MPEG-4 AVC stream watermarking

Marwen Hasnaoui

THESE DE DOCTORAT CONJOINT TELECOM SUDPARIS
et L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité : Informatique et Télécommunications

Ecole doctorale : Informatique, Télécommunications et Electronique de Paris

Présentée par

Marwen Hasnaoui

Pour obtenir le grade de

DOCTEUR DE TELECOM SUDPARIS

Tatouage du flux compressé MPEG-4 AVC

Soutenue le 28 mars 2014

devant le jury composé de :

Pr. Frédéric Truchetet (Université de Bourgogne)	**Rapporteur**
HDR. Patrick Bas (Ecole Centrale Lille)	**Rapporteur**
Pr. Patrick Gallinari (Université de Pierre et Marie Curie)	**Examinateur**
Pr. Adriana Vlad (Université Politehnica de Bucarest)	**Examinateur**
Pr. Françoise Prêteux (Mines ParisTech)	**Examinateur**
Pr. Amel Benazza (Ecole supérieure des communications de Tunis)	**Examinateur**
M. Eric Munier (Cassidian)	**Examinateur**
HDR. Mihai MITREA (Télécom SudParis)	**Directeur de thèse**

Thèse n° 2014TELE0009

Remerciement

Mes premiers remerciements iront à mon directeur de thèse, HDR. Mihai Mitrea, pour m'avoir soutenu durant mes trois ans années de thèse. J'aimerais lui adresser mes plus vifs remerciements pour son dynamisme et ses compétences scientifiques qui m'ont permis de mener à bien cette étude. Ce travail n'aurait jamais pu aboutir sans lui, qui a toujours su me consacrer de son temps, me conseiller, me guider et me témoigner de son soutien et sa confiance. Je souhaite lui transmettre l'expression de ma reconnaissance et de ma plus profonde gratitude.

Je remercie tout particulièrement les membres de mon jury de thèse, qui ont accepté d'évaluer ce travail. Je suis reconnaissant envers Pr. Frédéric Truchetet et HDR. Patrick Bas pour avoir rapporté ma thèse ainsi que pour leurs recommandations qui ont permis d'améliorer ce travail. Je tiens également à remercier Pr. Adriana Vlad, Pr. Françoise Preteux, Pr. Amel Benazza, Pr. Patrick Gallinari et M. Eric Munier pour avoir accepter de faire partie de mon jury de thèse et d'enrichir mon travail avec leurs remarques.

Je remercie vivement Pr. Adriana Vlad et Pr. Amel Benazza pour avoir accepté de participer au jury malgré le long voyage.

Je tiens également à remercier l'équipe SPY et particulièrement M. Eric Munier, M. Arthur Lallet et M. Boris Batteux de SPY-CASSIDIAN pour la collaboration et les plusieurs échanges qui ont contribués énormément à la conception et à l'implantation du système de vérification d'intégrité SPYART.

Mes plus vifs remerciements à Maher Belhaj qui m'a aidé à décortiquer les parties complexes du logiciel de référence JM86. Mes remerciements vont aussi à Afef Chammem, Adriana Gorban, Bojan Joveski et Rama Rao Ganji pour leurs aides ainsi que tous les bons moments qu'on a pu partager.

Je souhaite aussi remercier Mme Evlyne Taroni qui m'a largement aidé à dépasser les entraves administratives liés à la prolongation du contrat doctoral ainsi qu'à la préparation de la soutenance.

Un merci du fond du cœur ira à tous les membres du département ARTEMIS qui m'ont offert un excellent cadre de travail.

Je ne pourrais clôturer sans remercier ma chère mère et mon cher père à qui je dois tant, ainsi que toute ma famille qui m'a apporté bien plus que leur support, mais aussi tant d'amours, de motivations et de joies.

Je tiens enfin à dédier ce travail à ma femme qui par bon et mauvais temps m'a toujours assuré son soutien.

Table of Contents

List of Figures

List of Tables

Abstract

Context

During the last two decades, digital multimedia piracy has been considered a topic of concern given its impact on the industrial world: video piracy has cost more than one billion US Dollars to the cinema and television industry and because of it, sales and rentals of DVDs have declined by about 10 percent between 2002 and 2012 (source: Motion Picture Association of America, MPAA Filing on World's Most Notorious Piracy Markets, 2013).

In this context, the need for embedding digital information (watermarks) into digital video has attracted a great deal of interest for a large area of video applications, such as ownership protection, content integrity verification, piracy tracking or broadcast monitoring.

In practice, video sequences are stored and distributed in compressed bit stream formats. Consequently, watermarking the compressed video would require to decode this sequence, then to insert the watermark and, finally, to re-encode it. Such an approach would result into a large computing time, mainly because of the encoding and decoding; these operations are intrinsically avoided if the watermark is inserted directly in the compressed video domain.

Several research studies are conducted in this respect, with a special attention paid to the MPEG-4 Part 10 (a.k.a. MPEG-4 AVC or, alternatively H.264) compressed stream. In order to achieve higher compression efficiency, MPEG-4 AVC deploys particular compression features such as variable block-size motion estimation, directional spatial prediction and context-adaptive video coding. Consequently, the existing watermarking techniques, be they devoted to the uncompressed or to earlier compressed domains (MPEG-2 or MPEG-4 Part 2), are likely to fail in reaching the same performances in the MPEG-4 AVC compressed domain and specific methods should be devised in this respect.

The present thesis addresses the MPEG-4 AVC stream watermarking and considers two theoretical and applicative challenges, namely ownership protection and content integrity verification. While ownership protection is the preponderant application of watermarking techniques aiming to prevent or deter unauthorized copying of digital media, content integrity verification aims to check that the video has not been altered by modifying its semantic content.

Constraints

In order to be effective, watermarking should jointly reach constraints related to data payload, robustness/fragility, transparency, and computational complexity. The data payload is the amount of the embedded information (*i.e* the size of the watermark). The robustness is the ability of the watermark to survive mundane and/or malicious attacks; conversely, the fragility is the mark vulnerability against attacks. The transparency refers to the human imperceptibility of the artifacts induced by the mark in the host signal. In order for the watermarking techniques to be easily integrated into practical applications, the watermark insertion/detection should be achieved at a low (non prohibitive) computational cost.

The practical trade-off among these four properties is *a priori* set according to the targeted application context and purposes, as follow.

Concerning the data payload, the main deadlock is related to the conceptual contradiction between compressions and watermarking. In fact, the compression paradigm is based on eliminating the visual redundancy in order to achieve high rate of compression. In contrast, watermarking tends to take advantage of the visual redundancy so as to hide the mark. Therefore, compressed video stream leaves very little room to hide the watermark (*e.g.* serial number identifying user for ownership application, authentication signature for integrity verification application, *etc.*). Beside the mark size, some watermarking applications also impose constraints concerning the mark semantic. For instance, for ownership protections, the embedded mark may have no meaning and is randomly generated to serve just as an owner ID. In contrast, for video integrity verifications, the embedded mark should reflect the semantic for the video content so as to distinguish between content changing and content preserving alterations.

Concerning the robustness, the main deadlock is to recover the mark after any malicious and mundane attack that a pirate may apply: noise addition, compression, frame dropping, resizing, letter-boxing, removal, random geometric transformations induced by in-theater camera recording, *etc*. While the previous sentence holds for ownership protection, the content integrity verification comes across with an additional requirement: the mark should be robust against content preserving attacks (*e.g.* noise addition, transcoding, …) while being fragile against content alteration attacks (*e.g.* object deletion, spatio-temporal cropping, …).

The transparency also sets related yet different requirements for ownership protection and content integrity verification. The former application assumes ideal transparency, *i.e.* the artifacts induced during the mark insertion are humanly imperceptible. The latter application accepts a weaker transparency: human disturbing artifacts are tolerated assuming they have no impact in object/persons/events identification. Regardless of the application, note that directly watermarking the compressed stream would result in drift artifacts: due to the stream syntax, the modification of one block would result in the modification of its neighbors, thus increasing the transparency constraints.

The low computational complexity imposes the constraint that the insertion/detection should be lighter than the rest of video processing operations involved in the application (*e.g* encoding/decoding) and compatible with the real time.

Challenges

The present thesis deals with theoretical and methodological issues related to the MPEG-4 AVC watermarking for ownership protection and video integrity verification. These challenges have been identified under the framework of two collaborative R&D projects: the French MEDIEVALS (waterMarking et Embrouillage pour la DIffusion et les Echanges Vidéos et Audios Legalisés) project funded by ANR and the European SPY (Surveillance imProved sYstem) project funded by ITEA2.

From the theoretical point of view, the thesis main challenge is to develop a unitary watermarking framework (insertion/detection) able to serve the two above mentioned applications. From the

methodological point of view, the challenge is to instantiate this theoretical framework for serving the targeted applications

The ownership protection is considered under a VoD (Video on Demand) framework (*cf.* MEDIEVALS). The objective is the increase of the data payload for pre-established robustness and transparency levels. The robustness should be evaluated against noise addition, transcoding, and Stirmark random bending attacks. The transparency should correspond to the humanly imperceptible artifacts.

The video integrity verification is considered under a mobile video surveillance system (*cf.* SPY). The challenge is to extract from the video stream an authentication signature which is subsequently inserted so as to ensure the video integrity. This signature should be robust to content preserving attacks and fragile to content alteration attacks. The accuracy of the content altered regions should be evaluated both spatially and temporally.

For the two applications, the mark insertion/detection should be lighter than the MPEG-4 AVC encoding/decoding.

Contributions

The present thesis tackles the above mentioned challenges by the following theoretical and methodological contributions.

Multi symbol quantization index modulation watermarking (m-QIM)

The thesis first main contribution consists in building the theoretical framework for the multi-symbol watermarking based on quantization index modulation (*m*-QIM). The insertion rule is analytically designed by extending the binary QIM rule. The detection rule is optimized so as to ensure minimal probability of error under additive white Gaussian noise distributed attacks. It is thus demonstrated that the data payload can be increased by a factor of $\log_2 m$, for prescribed transparency and additive Gaussian noise power.

The *m*-QIM framework is first deployed for ensuring the VoD (Video on Demand) ownership protection. The main benefit is the increase of data payload by a factor of $\log_2 m$ for a prescribed robustness of 0.1 of BER (variations lower than 3% of the bit error rate after additive noise, transcoding and Stirmark random bending attacks) and transparency (set to average PSNR = 45dB and 65dB for SD and HD encoded content, respectively). Actually, the experiments considered 4 values of m, namely $m = 2$, $m = 3$, $m = 5$ and $m = 7$; just for illustration, for $m = 5$, a data payload of 150 bits per minute, *i.e.* about 20 times larger than the limit imposed by the DCI (Digital Cinema Initiatives) standard, is obtained. The processed corpus sums up to 1 h of video content granted by the MEDIEVALS industrial partners.

The second *m*-QIM application consists in designing a semi-fragile watermarking method for video integrity verification in compressed stream. In this respect, the MPEG-4 AVC syntax elements which can optimally (in the information theory sense) serve as authentication signature are identified. This authentication signature is further inserted by combining the *m*-QIM principle to an alteration detection strategy. The experiments results show fragility to content replacement (with an 1/81 frame and 3 seconds spatial and temporal accuracy, respectively) and robustness against noise addition and

transcoding (compression by a factor of 2). The *m*-QIM framework main advantage is this time a relative gain factor of 0.11 of PSNR for fixed robustness (against noise addition and transcoding), fragility (to content alteration) and the data payload. The processed corpus sums up 1h 20 minutes of heterogeneous video contents granted by the SPY industrial partners.

The computational time required by each operation included in the watermarking chain is evaluated on the following PC configuration: a Core4 CPU at 2.8 GHz and with 12 GB of RAM and a 500 GB HDD. The signature generation, insertion and detection are much faster than the entropic decoding/encoding and the video stream read/write from the hard disk. When considering the same example as above (protecting 1s of video), the entropic decoding/encoding and the read/write operations are about 10 and 8 times slower than the mark selection/insertion/detection, respectively.

Drift-free watermarking

The thesis second main theoretical contribution consists in specifying a preprocessing MPEG-4 AVC shaping operation which can eliminate the intra-frame (spatial) drift effect. The drift represents the distortion spread in the compressed stream related to the MPEG encoding paradigm. In this respect, the drift distortion propagation problem in MPEG-4 AVC is algebraically expressed and the corresponding equations system is solved under drift-free constraints.

The experiments consider the same *m*-QIM semi-fragile watermarking method and the same corpus. For prescribed data payload (100 bits/s), robustness (BER < 0.1 against transcoding at 50% in stream size), fragility (frame modification detection with accuracies of 1/81 from the frame size and 3 seconds) and complexity constraints, the drift-free shaping results in gains in transparency of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ.

MPEG-4 AVC compressed domain watermarking: properties, constraints, and thesis contributions.

Properties	Constraints	Thesis contributions
Data payload	***Compressed domain watermarking:*** ▪ Compressed video stream leaves very little room space to hide data.	***m-QIM insertion rule*** ▪ ***Theory:*** Increasing the data payload by a factor of *log2m.* ▪ ***Practice:*** Inserting a data payload of 150 bits per minute for prescribed robustness and transparency, for VoD ownership protection.
Robustness	***Ownership protection:*** ▪ The embedded mark should be recovered after additive noise, transcoding, and Stirmark random bending attacks.	***m-QIM optimal decision rule*** ▪ ***Theory:*** Minimize probability of error under additive white Gaussian noise distributed attacks. ▪ ***Practice:*** Robustness of 0.1 of BER (variations lower than 3% of the bit error rate after additive noise, transcoding and Stirmark random bending attacks).
Fragility	***Mark semantic:*** ▪ The embedded mark should reflect the semantic for the video content so as to distinguish between content changing and content preserving alterations. ***Integrity verification:*** ▪ The mark should be robust against content preserving attacks while being fragile against content alteration attacks.	***MPEG-4 AVC Syntax element based signature*** ▪ ***Theory:*** Identifying the syntax elements which can optimally (in the information theory sense) serve as authentication signature ▪ ***Practice:*** Robustness against transcoding and Fragility to content replacement (with an 1/81 frame and 3 seconds spatial and temporal accuracy, respectively).
Transparency	***Compressed domain watermarking:*** ▪ Compressed stream processing leads to drift distortion propagation.	***Drift-free shaping for MPEG-4 AVC compressed stream watermarking.*** ▪ ***Theory:*** Analytically expressing the drift distortion problem and resolving it under drift-free constraint. ▪ ***Practice:*** Gains in transparency of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ. for prescribed data payload (100 bit/s), robustness (BER < 0.1 against transcoding at 50% in stream size).
Complexity	***Compressed domain watermarking:*** ▪ Mark embedding should not increase the computational cost of the watermarking application.	***MPEG-4 AVC compressed domain watermarking*** ▪ ***Practice:*** The mark generation, selection, insertion and detection for one second of video are 10 times and 8 times faster than the MPEG-4 AVC entropic encoding/decoding and the stream read/write operations, respectively.

Part I: Introduction

Abstract

This Introduction is structured into three chapters. First, the various watermarking fundamentals (main properties, applications, theoretical model) are browsed. Secondly, the states of the art related to robust and semi-fragile watermarking are detailed and their limitations for ownership protection and integrity verification in MPEG-4 AVC compressed domain are identified. Finally, the thesis objectives, contributions and structure are presented.

I.1. Watermarking context

Digital watermarking can be defined as the process of embedding a pattern of information into a cover digital content (image, audio, video, *etc.*) [COX02] [MIT07], see Figure I-1. The insertion of the mark is always controlled by some secret information referred to as a key. While the key should be kept secret (*i.e.* known only by the owner), the embedded information and even the embedding method can be public. Once watermarked, the host data can be transmitted and/or stored in a hostile environment, *i.e.* in an environment where changes attempting to remove the watermark are likely to occur.

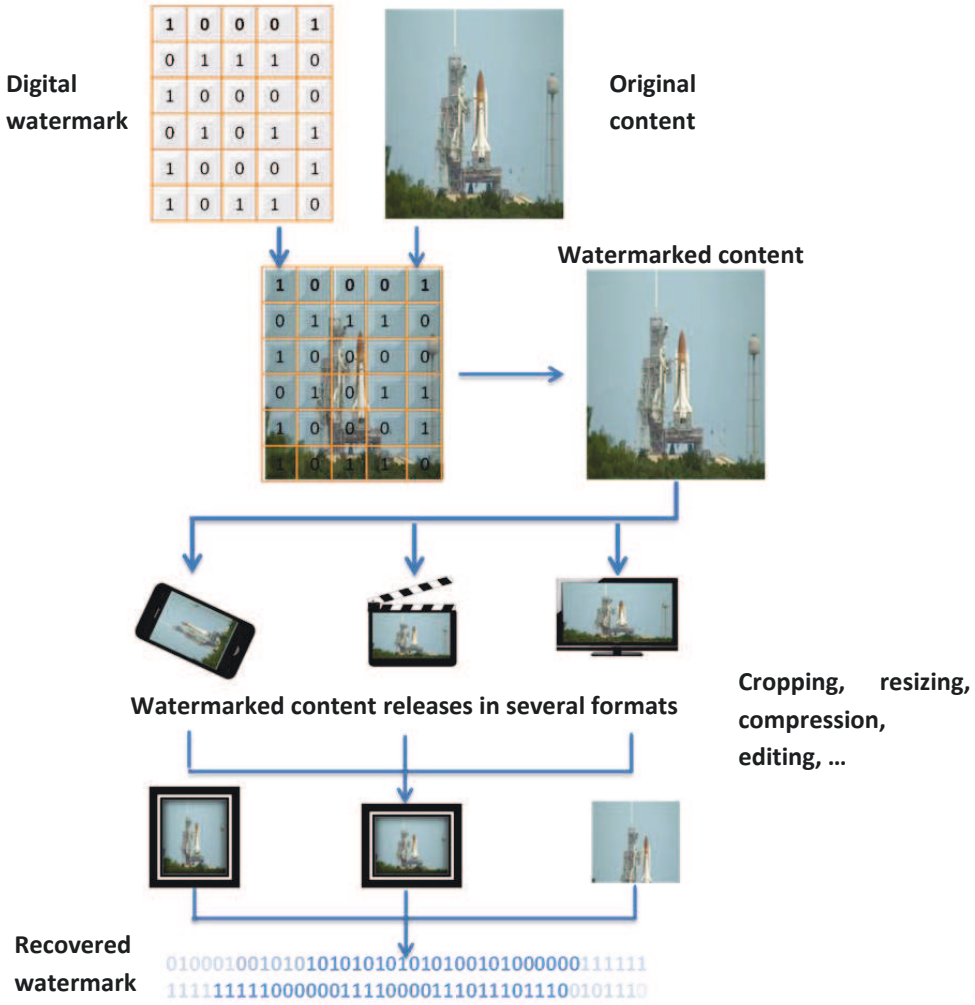


Figure I-1: Watermarking flowchart.

From the structural point of view, any watermarking procedure features three components: the watermark generation (*i.e.* the way in which the message to be inserted is encrypted with a secret key so as to obtain a watermark), the watermark embedding (*i.e.* the way in which the watermark is inserted in the host document) and the watermark detection (*i.e.* the way in which the watermark is recovered).

From the information theory point of view, the watermarking process can be considered as a communication system with side-information at the encoder, see Figure I-2. Using a secrete key *k,* the watermark message *m* is embedded into the host signal *x*. The watermarked signal *s* is then transmitted over the channel which can introduce a noise *n* resulting from the attacks. The decoder receives the signal *r* and, using the same key *k*, extracts the watermark message *m'*.

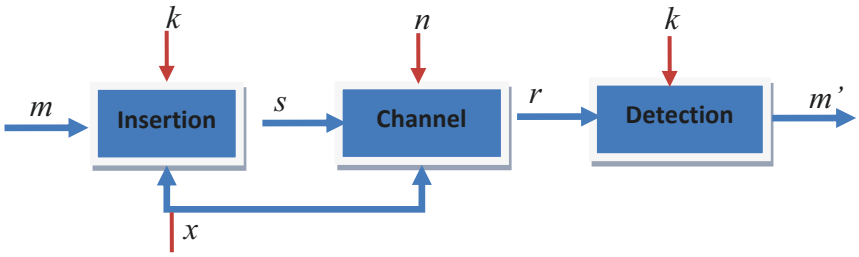


Figure I-2: Watermarking synopsis diagram.

The watermarking schemas are commonly divided into two main classes, namely spread spectrum (SS) and side information (SI).

The SS system have been already deployed in telecommunication applications (*e.g.* CDMA), by providing a preferment solution very low power signal transmission over noisy channel [COX95]. Consequently, an SS based watermarking method spreads the mark across the host signal by creating redundancy, requiring a much larger bandwidth than strictly necessary. In practice this approach remains robust against attacks, while offering limited data payload [MIT07].

The SI principle [SHA58], [COS83], [EGG03], [CHE98] stipulates that a given noise channel known at the transmitter and unknown at the receiver would not decrease the channel capacity (the maximum amount of information which can be theoretically transmitted). Thus, the original document should no longer be considered as a hindrance to the watermark detection. Consequently, the side information watermarking is *a priori* optimal from the data payload point of view (under fixed transparency and robustness constraints). However, in practice, the methods following this approach feature very weak robustness in spite of a very high quantity of embedded information.

Generally, a noisy channel is described by expressing the probabilistic dependencies between the input and the output of the information sources and by evaluating the average amount of the transmitted information.

A watermarking system can be model by a discrete channel, where the input m represents the inserted mark alphabet, the output m' represents the detected mark alphabet, and the noise n represents the attacks. An error means to receive a symbol that does not correspond to the inserted one. In this case the channel is modeled by a noise matrix expressing the conditional probabilities $p(m'/m)$.

I.1.1. Applicative panorama

The worldwide spread of watermarking solutions is mainly and historically boosted by the impact of the copyright in the industry. The current estimations (in USD) of the value added for the core and non-core copyright U.S. industries in 2009, 2010, 2011, and 2012, are resumed in Figure I-3 (source: Motion Picture Association of America report [MPAA13]). Figure I-3 shows an increase from $884.81 billion in 2009 to more than $1 trillion, in 2012. The estimated value added for the other copyright industries increase from $656.82 billion in 2009 to 794.53 billion in 2012 [MPAA13].

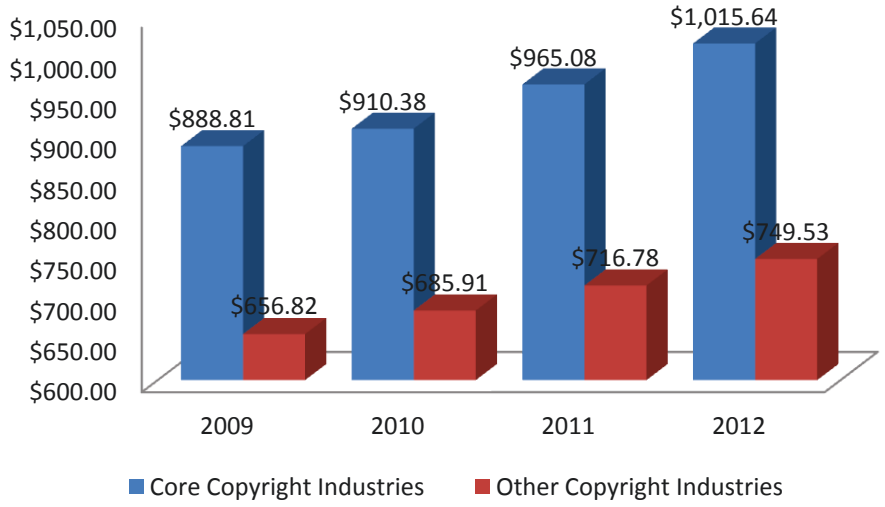


Figure I-3: Copyright industries value added (in billion USD).

Beyond copyright solutions, watermarking systems can serve a large variety of applications, from property and/or integrity proof to augmented reality.

Originally designed to track pictures and audio files piracy, watermarking now includes several applications [COX01] and handles many media content types ranging from still image to compressed video and stereoscopic content. This considerable expansion opens the door to many new applications, as illustrated below[1].

Digital Right Management

Copyright protection is the main application of watermarking techniques [MEM98], [CRA98], [BAS01]: it aims at preventing or deterring unauthorized copying of digital media. Digital watermarks contain a set of copy control instructions, telling the copy devices if copies are allowed, or not.

[1] In this thesis we are interested in two applications of the compressed domain watermarking: ownership protection for video on demand and content integrity verification for video surveillance.

For instance, under the VoD (Video on Demand) framework, consumers can have access to HD movies included in the VoD server. This represents a considerable revenue opportunity for PayTv industries. However, content owners must be ensured that video content piracy attacks can be deterred. Digital watermarking provides ownership protection solution by identifying the source of illegal distributed copies, see Figure I-4.

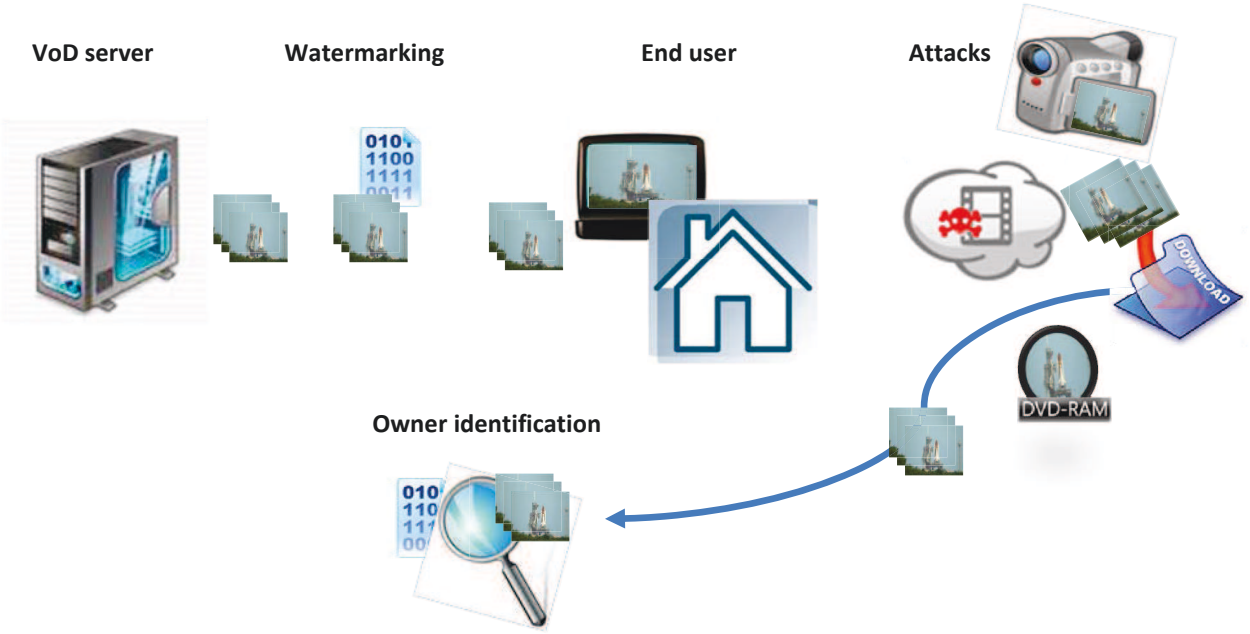


Figure I-4: VoD ownership protection.

Forensics and piracy tracking

Forensic watermarking applications enhance the content owner's ability to detect and respond to the misuse of his/her assets. Digital watermarking is used not only to gather evidence for criminal acts, but also to enforce contractual usage agreements between a content owner and the customers. It provides positive, irrefutable evidence of misuse for leaked content assets [DIG03].

Authentication and integrity

Digital watermarks are imperceptibly embedded into all forms of media content, be they individually or aggregately distributed/stored. The watermark can uniquely identify each specific item or instance of content and carry information about its consumption chain and intended destinations. Watermarks can be encrypted and secured so that only authorized reading devices can detect and access the data. Altering a watermark is virtually impossible and the carried data can immediately indicate if the content is genuine or a counterfeit [SAM09], [CHA00].

For instance, under video surveillance integrity verification framework, recorded videos are watermarked and stored. Further, when these videos are solicited to help in elucidating some crime acts, the embedded mark is detected to verify their integrity, see Figure I-5.

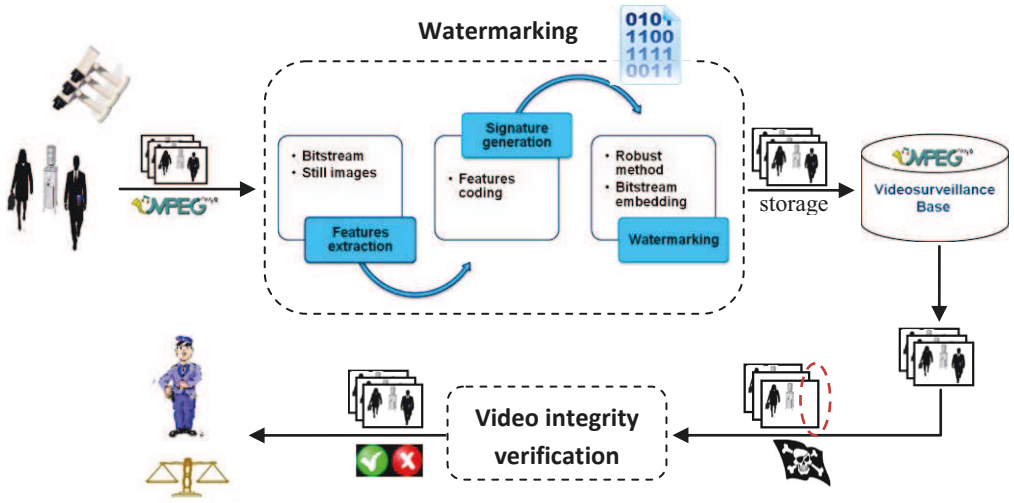


Figure I-5: Integrity verification synopsis.

Broadcast and Internet monitoring

Over the last few years, the amount of content flowing through television/radio channels continues to grow exponentially. Watermarking techniques offer the best solution to automate the monitoring of digital content. For such an application, the mark conveys a unique identifier (owner, distributor, data/time information) allowing the content owners and distributors to track their media [KAL99], [SAM09].

Asset and content management

Watermarking techniques enable effective content identification by giving a unique digital identity to any type of media content and by embedding this identity as additional hidden information [SAM09]. The watermarks must be imperceptible and have minimal or no impact on the visual quality of the original content. Hence they can be used as a persistent tag, acting as keys into a digital asset management system (DAM). Tagged content can lead back to the original content stored in the DAM system; it can also be linked to metadata in the DAM, such as keywords, rights and permissions.

Filtering/classification

Digital watermarks offer new opportunities for content owners, advertisers and more generally marketers searching for new ways to engage consumers in richer media experiences. In fact, the embedded information enables the identification, classification and filtering of multimedia content. The watermarking systems are able to selectively filter potential inappropriate content (*e.g.* parental control).

Navigation between print and online

Digital watermarks play here the role of some in-band enrichment information allowing the readers of printed documents to directly access hyperlinks. The readers need only to point their smart phone or

tablet at a digitally watermarked image, graphic element or text to be directly connected to related online experiences, see Figure I-6. This solution was advanced by Digimarc Discover platform in 2011 to compete QR code. According to the Digimarc traction report [DIG12], the number of digital watermarks placed in magazine and advertising content registers a growth of 486% from 2011 to 2012.

Figure I-6: Navigation between print and online.

Second-screen synchronization

For such an application, the watermark is inserted in the TV content prior to its broadcasting. The user, watching a TV set, can take a screen snapshot with a device (smartphone, tablet) of the distributed video and send it to the server which extracts the embedded mark and returns the content ID. The main advantage of the technique is that you don't have to be the owner of the video content to analyze it and it is a perfect way for building synchronized services [FIL13], see Figure I-7.

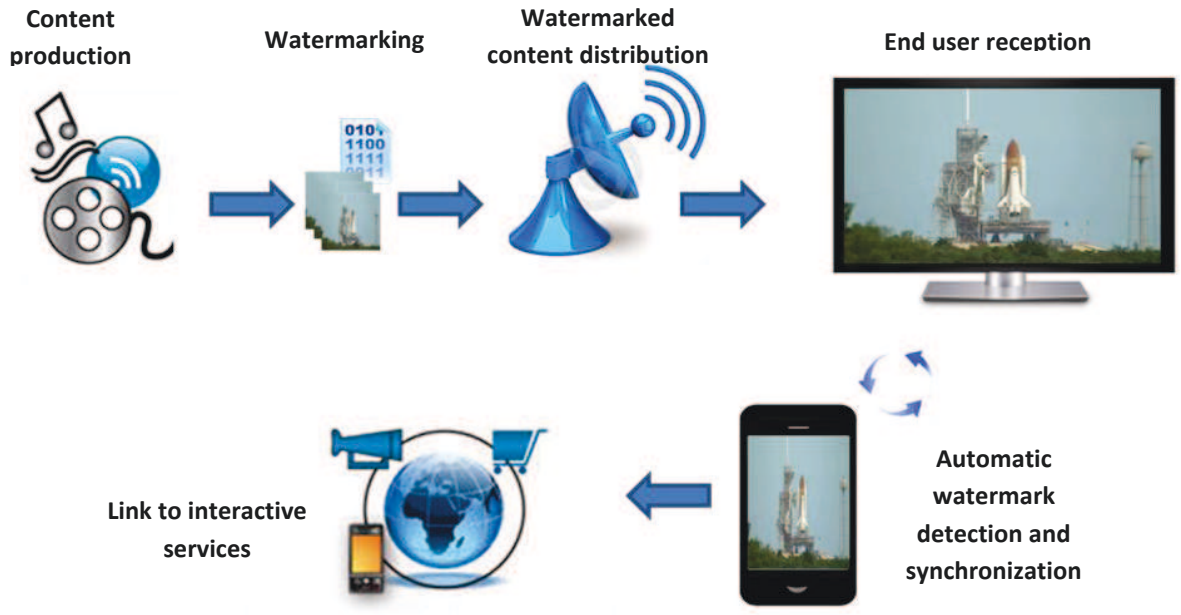


Figure I-7: Second screen synchronization.

I.1.2. Properties

The applicative panorama presented in the previous section demonstrates that there are no universal requirements to be satisfied by all watermarking applications. Nevertheless, some general directions can be given for most of the applications.

In order to be effective, the watermark should be perceptually invisible for a human observer (*i.e.* the transparency property – Section I.1.2.3) and its detection should be successful even when the watermarked content is attacked (*i.e.* the robustness property – Section I.1.2.2). Moreover, it should allow the insertion of the amount of information (referred to as data payload – Section I.1.2.1) required by the targeted application (*e.g.* a serial number identifying a user, a time stamp, *etc.*). In order for the watermarking techniques to be easily integrated into practical applications, the watermark insertion/detection should be achieved at a low computational cost (Section I.1.2.4).

Note that robustness comes across with a related yet different property, namely the security. According to [MAT13] and [KAL01], the security is "*the inability by unauthorized users to have access to the raw watermarking channel*" [KAL01]. From this point of view, the security relates to the robustness attacks, explicitly applied by a malicious hacker; hence, this concept is covered in the thesis by the robustness - Section I.2.2.2). However, as mentioned in [MAT13], "*following a cryptographic model, security in watermarking is generally based on Kerckhoffs principle [KER83] and relates to the use of a secret for the embedding and the decoding of the sequences"* [MAT13]. This latter definition of the security property is discussed in Section I.1.2.5. However, as from this point of view, the security of the watermarking solutions advanced by the present thesis was investigated by CASSIDIAN CyberSecurity, no experimental result will be reported in Part II.

I.1.2.1. Data payload

This is the total amount of information (in bits) inserted into original content. According to the targeted applications, the specifications on this factor may be very different, from 64 bits per sequence for the identification of ownership up to hundred of kilobits per frame for application of hyper-video. Different applications require different data payloads. For instance, for e-commerce applications, the additional data (the watermark) could bring information about the document buyer, vendor, date and time of purchase. In a right management context, the embedded watermark has to identify the content and specify the usage rules as well as the billing information. For authentication and integrity applications, the watermark identifies (spatially and temporally) the content's main modifications.

I.1.2.2. Robustness

Robustness is the ability of the mark to survive changes undergone by the host media. These changes (be they intentional or unintentional) define the set of attacks. The various possible attacks against watermarked video can be structured into four classes [PET98], according to the way they act: removal attacks, geometric attacks, cryptographic attacks, and protocol attacks, Figure I-8.

The removal attacks try to make the watermark unreadable. This class includes attacks by noise addition, denoising, transcoding quantization, …

The geometric attacks aim to destroy the synchronization of the watermark. After such an attack, the watermark is still present in the video, but its location is unknown at the decoder. Rotations, curvatures, jitter of pixels individually considered or combined into the Stirmark attacks, fall into this category [PET00].

Protocol attacks aim to make watermark unusable by creating some ambiguities concerning the mark usage. Attacks by inversion and copy belong to this class. The former creates a false key so that by applying the detection procedure, the watermark indicates a different owner for the video.

The cryptographic attacks try to manage the watermark (detect/copy/insert a new one) without knowledge of the secret key. One example is represented by the brute-force search. Another example, known as the oracle attack, consists in creating an unmarked version of the signal by exploiting the response of a detector (assuming it is available). In any case, this type of attack is very restrictive in practice because of its complexity.

A watermark system is fragile to an attack when the watermark cannot be detected after slightest modifications generated by this attack.

A watermarking system is semi-fragile when both particular robustness and fragility properties are imposed to the system. Once the classes of allowed and non-allowed attacks have been defined based on the targeted application, the watermark must survive all manipulation belonging in the former class (the robustness), but it should be destroyed by the manipulations belonging to the latter (the fragility).

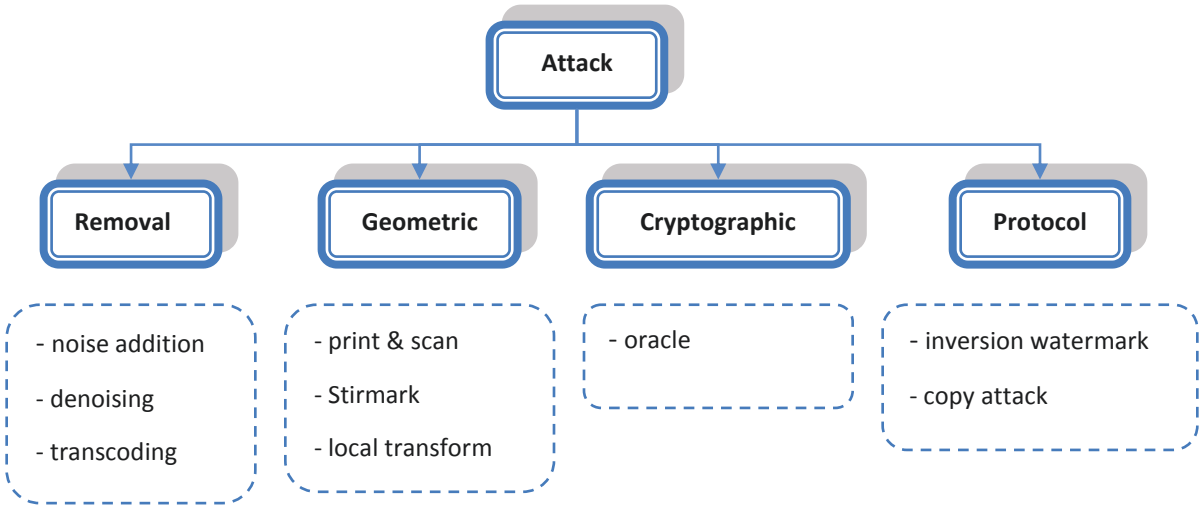


Figure I-8: Watermark attack classification [COX08].

In the applicative context of this thesis the most intensively considered attacks are the noise addition, the transcoding and the Stirmark. These attacks will be further detailed.

Additive Gaussian noise:

This type of attack has been mainly involved in communication and signal processing. It is invoked to model the overall behavior of different noise sources according to the Central Limit Theorem. The same assumption is adopted in watermarking, where several studies consider the Gaussian noise as a general attack model. Although it can be very effective model for some real life image/video filtering operations (*e.g.* linear filtering, JPEG compression, …) in-depth studies proved its inaccuracy for other types of attacks (*e.g.* rotations, random geometric attack, …) [MIT07]. Note that this inaccuracy involves the Gaussian behavior and not the additive hypothesis, which was practically each and every time validated by the experiments.

This attack can be achieved by altering the host data by introducing an additive noise following a Gaussian distribution.

Transcoding (lossy compression):

Lossy compression remains the most common attack that a multimedia content is likely to undergo. In fact, a marked multimedia content is susceptible to undergo this attack both during its creation and transmission/storage.

The transcoding attack may take place when the original encoded video according to a given encoding parameters/format is first decoded and further re-encoding according to some prescribed encoding parameters/format. For instance, the lossy compression can be achieved by performing a transcoding at a lower bit rate.

Note that that there is a deep-seated conflict between watermarking and lossy compression. In order to be efficient, a compressor tries to get rid of all the visual redundancy existing in the original video. Thus, it leaves fewer room for the watermarking which exploits that redundancy in order to hide the mark.

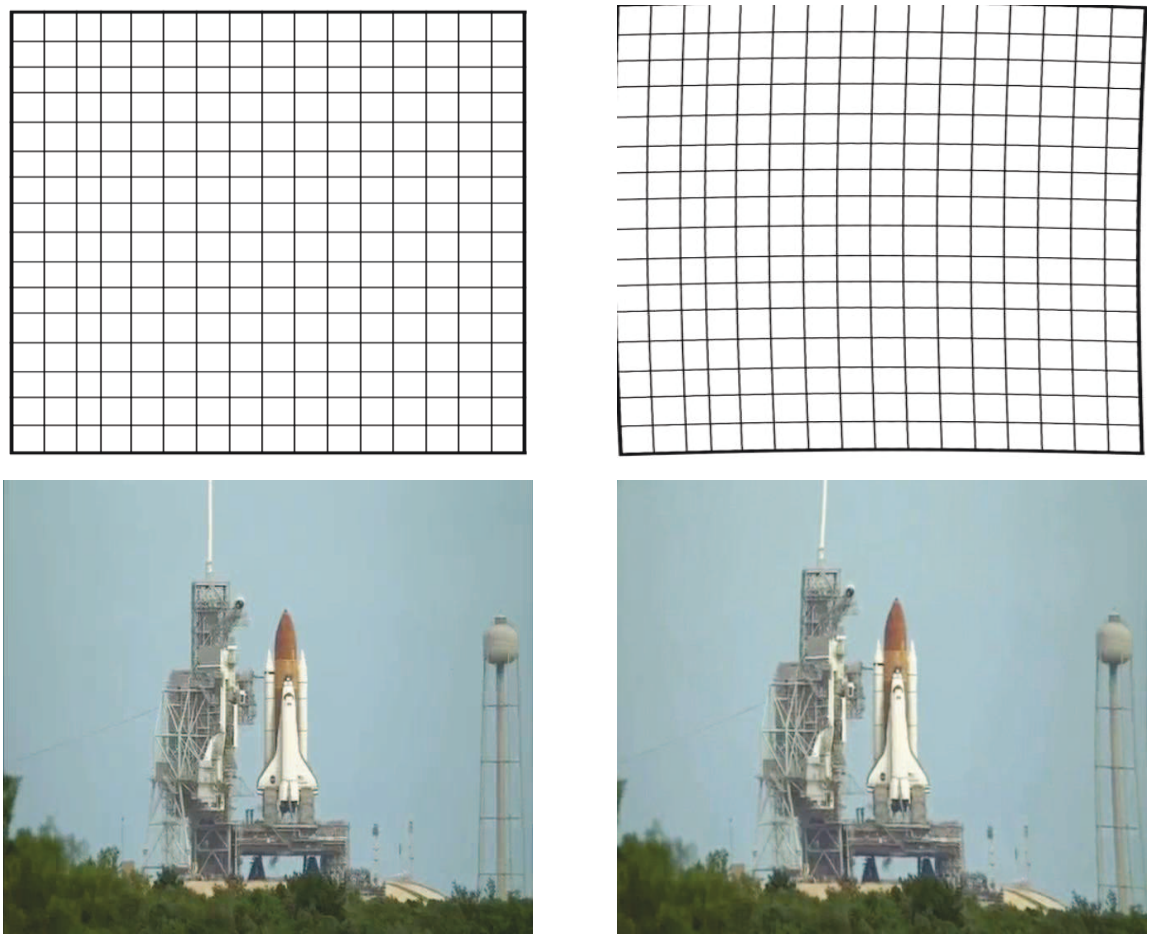
Stirmark:

Stirmark is a generic tool developed for assessing the robustness of image watermarking algorithms [PET00].

The Stirmark tool offers a several class of attacks such as filtering and geometric based attacks. Among the field of the proposed attacks, it also includes Stirmark random bending attack which is inspired from the jitter attack. It simulates the impact of a high quality printing followed by a low quality scanning. The attacked image receives minor local geometric modification as well as small amplitude additive Gaussian noise distortions. The obtained image retains a very high visual quality. However, the induced geometric distortions are often sufficient to bother the mark detection.

Figure I-9 shows the impact of the Stirmark random bending, on both a test and a natural image. While the affects are directly noticeable for the test image (see Figure I-9 up left *vs*. right), they are imperceptible for the natural image (see Figure I-9 down left *vs.* right).

In the present thesis, by Stirmark attack we shall denote the Stirmark geometric random bending attack. Moreover, this attack is applied to a video content at the frame level. Consequently, the video is first split into frames, then the random bending attack is applied, and the attacked frames are further re-encoded to generate the attacked video.



Original image | Stirmark random bending attack affects

Figure I-9: Stirmark random bending attack impact.

I.1.2.3. Transparency

The notion of transparency is related to the perception (visual, auditory …) of artifacts resulted from the insertion process. Watermarking should be imperceptible and invisible to a human observer (*i.e.* the embedded watermark should not affect the quality of the host data).

The visual quality assessment of the watermarked data remains an important criterion for validating the watermarking algorithm. However, it is a subjective concept that depends on various criteria: human visual system, age, experience, artistic sense, observation condition, *etc*. Thus, it is complicated to evaluate whether a watermarking method is transparent or not. Such an evaluation requires significant testing involving a wide observer's panel and many visual assessments [MAN04]. Alternatively, some objective transparency metrics can be used.

An objective measure is a function that takes as input some video information, calculates the distance to some reference information extracted from reference video and outputs a value somewhat associated to that differences. Based on the way they act, objective measures can be classified into three classes [AVC01]:

- ***Pixels difference measures*** are based on differences between the original and the modified image. The peak signal to noise ratio (*PSNR*), the maximum mean square error (*PMSE*), the image fidelity (*IF*) and the average absolute difference (*AAD*) are the most common, due to their easily use and implementation.
- ***Correlation measures*** reflect the similarity between two images even in the presence of a low noise compared to the pixel strength; the normalized cross correlation (*NCC*) and the structural content (*SC*) belong to this class.
- ***Psychovisual measures*** consider the human visual system as a spatio-temporal filter. For instance, the digital video quality (*DVQ*) models the human visual system according to luminance intensity, frequency contents and structural content.

These measures will be considered in the sequel according to their definition presented in [ESK95] and [WAN04]. Be there S and $\hat{S}$ two video to be compared, each of them having N_f frames of $W \times H$ pixels. Be $S_{i,j,k}$ the pixel at column i and row j of frame k. $PSNR$, AAD, SC and NCC are expressed as follow:

$$PSNR(S,\hat{S}) = \frac{1}{N_f}\sum_{k=1}^{N_f} 10\,log\left(\frac{W.H.\max\left({S_{i,j,k}}^2\right)}{\sum_{i=1}^{W}\sum_{j=1}^{H}\left(S_{i,j,k}-\hat{S}_{i,j,k}\right)^2}\right)$$

$$MSE(S,\hat{S}) = \frac{1}{WHN_f}\sum_{i=1}^{w}\sum_{j=1}^{h}\left(S_{i,j}-\hat{S}_{i,j}\right)^2$$

$$IF(S,\hat{S}) = 1 - \frac{\sum_{i=1}^{w}\sum_{j=1}^{h}\left(S_{i,j}-\hat{S}_{i,j}\right)^2}{\sum_{i=1}^{w}\sum_{j=1}^{h}{S^2}_{i,j}},$$

$$AAD(S,\hat{S}) = \frac{1}{N_f}\sum_{k=1}^{N_f}\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}\left|S_{i,j,k}-\hat{S}_{i,j,k}\right|}{W.H}$$

$$SC(S,\hat{S}) = \frac{1}{N_f}\sum_{k=1}^{N_f}\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}\left({S_{i,j,k}}^2\right)}{\sum_{i=1}^{W}\sum_{j=1}^{H}\left({\hat{S}_{i,j,k}}^2\right)}$$

$$NCC(S,\hat{S}) = \frac{1}{N_f}\sum_{k=1}^{N_f}\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}\left(S_{i,j,k}\hat{S}_{i,j,k}\right)}{\sum_{i=1}^{W}\sum_{j=1}^{H}\left({S_{i,j,k}}^2\right)}$$

Concerning the *DVQ* analytic computing, we can refer to [WAN04], [BEL11].

I.1.2.4. Computational complexity

The technical cost of the algorithm is also a significant feature of any watermarking method. From this point of view, the complexity of the algorithm is the main criterion of practical acceptance. The complexity of the watermarking algorithm can be evaluated by estimating the number of operations required during the watermarking process. Assuming a well defined applicative framework and a prior established processing set-up, information in this respect can also be obtained by estimating the processing time.

Commonly, watermarking a compressed video file would involve five main operations: stream reading, MPEG decoding, mark insertion, MPEG coding and stream writing. Just for illustration, the study in [BEL11] reports the following distribution of the time spent of each of these operations for HD video, watermarked by a binary QIM (Quantization Index Modulation) method, see Figure I-10.

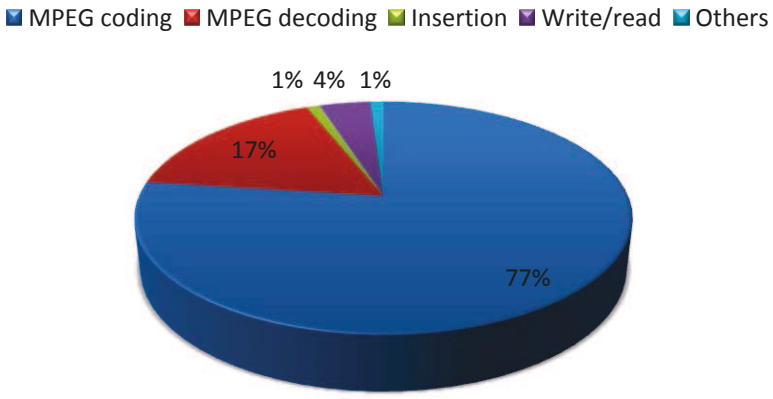


Figure I-10: Time consumption average for each video watermarking task [BEL11].

As shown in Figure I-10, about 95% of total processing time is consumed during the encoding/decoding process. Such an illustration brings to light that compressed domain watermarking is a promising solution to meet the low complexity level required by some real time applications.

I.1.2.5. Security

In any watermarking scheme, a secret key is used to insert and detect the mark; a security attack is a malicious attack allowing the hacker to estimate this key [MAT13], [KAL01] and [KER83]. Depending on the level of estimation, the hacker can modify, delete or copy the inserted message. The basic types of security attacks identified by [COX00] are:

Active attacks:

The hacker tries to remove the watermark or make it undetectable. This type of attack is critical for ownership protection and copy control. However, it is not serious problem for authentication.

Passive attacks:

In this case the hacker is not trying to remove the watermark, but he/she is simply trying to determine whether the mark is present.

Collusion attacks:

These are a special case of active attacks, in which the hacker uses several copies of one piece of original media, each with different mark, to construct a copy with no watermark [COX97].

Forgery attacks:

Here the hacker tries to embed a valid watermark rather than to remove one. This type of attack is a serious concern in proof of ownership [CRA98].

The present thesis does not consider the security evaluation of the watermarking solutions advanced in Part II. However, such an investigation was independently performed by Cassidian CyberSecurity experts, as a part of the SPY European project.

I.1.3. Constraints

The previous section introduced the watermarking main properties, namely data payload, robustness, transparency and computational complexity. The practical trade-off among watermarking properties is *a priori* set according to the targeted application context and purposes (see Figure I-11), as follow.

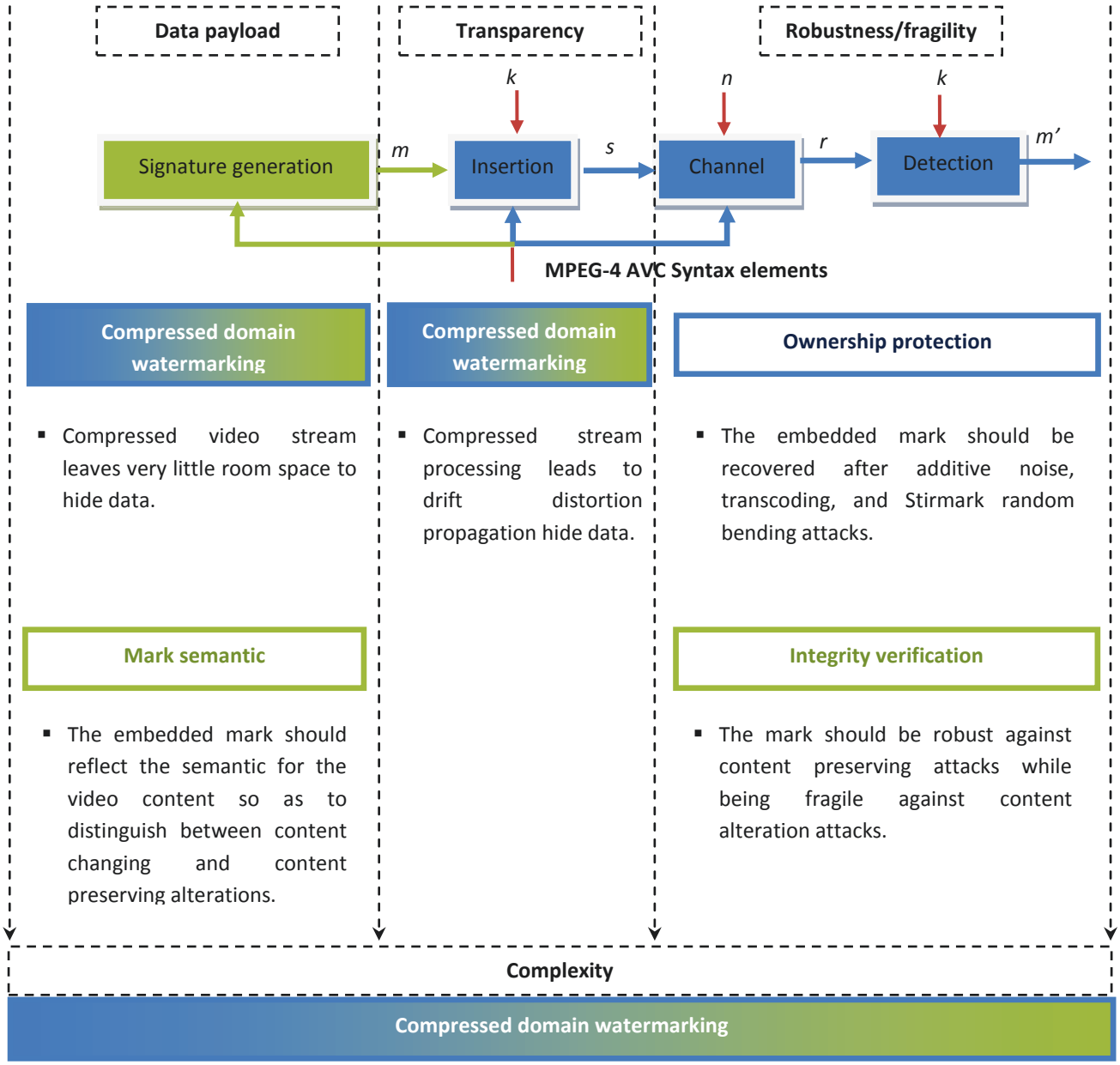
Concerning the data payload, the main deadlock is related to the conceptual contradiction between compressions and watermarking. In fact, the compression paradigm is based on eliminating the visual redundancy in order to achieve high rate of compression. In contrast, watermarking tends to take advantage of the visual redundancy so as to hide the mark. Therefore, compressed video stream leaves very little room to hide the watermark (*e.g.* serial number identifying user for ownership application, authentication signature for integrity verification application, *etc.*). Beside the mark size, some watermarking applications also impose constraints concerning the mark semantic. For instance, for ownership protection, the embedded mark may have no meaning and is randomly generated to serve just as an owner ID. In contrast, for video integrity verifications, the embedded mark should reflect the semantic for the video content so as to distinguish between content changing and content preserving alterations.

Concerning the robustness, the main deadlock is to recover the mark after any malicious and mundane attack that a pirate may apply: noise addition, compression, frame dropping, resizing, letter-boxing, removal, random geometric transformations induced by in-theater camera recording, *etc*. While the previous sentence holds for ownership protection, the content integrity verification comes across with an additional requirement: the mark should be robust against content preserving attacks (*e.g.* noise addition, transcoding, …) while being fragile against content alteration attacks (*e.g.* object deletion, spatio-temporal cropping, …).

The transparency also sets related yet different requirements for ownership protection and content integrity verification. The former application assumes ideal transparency, *i.e.* the artifacts induced during the mark insertion are humanly imperceptible. The latter application accepts a weaker transparency: human disturbing artifacts are tolerated assuming they have no impact in object/persons/events identification. Regardless of the application, note that directly watermarking the compressed stream would result in drift artifacts: due to the stream syntax, the modification of one block would result in the modification of its neighbors, thus increasing the transparency constraints.

The low computational complexity imposes the constraint that the insertion/detection should be lighter than the rest of video processing operations involved in the application (*e.g* encoding/decoding) and compatible with the real time.

The next chapter (State of the art) will investigate how these constraints are addressed by various research studies, devoted to both robust watermarking for ownership application and semi fragile watermarking for video content integrity verification.



- Mark generation/embedding/detection should not increase the computational cost of the watermarking application

Figure I-11: Constraints synopsis.

I.2. State of the art

This chapter will be structured into two parts devoted to the state of the art of the robust watermarking for ownership protection and the state of the art of semi-fragile watermarking for video integrity verification, respectively.

I.2.1. Robust watermarking for ownership protection

In practice, video sequence are stored and distributed in compressed format. Traditionally, watermarking a video requires to decode the sequence, then to insert the watermark and, finally, to recode the video. Such an approach requires a large computing time (*e.g.* about 95% from the total time) to be allocated to encoding/decoding. These operations can be intrinsically avoided if the watermark embedding acts directly in the compressed domain.

The first works in compressed domain were devoted to the watermarking of the MPEG-2 compressed video stream. Langelaar *et al.* [LAN 98] presented a real-time watermarking method. This method embedded the watermark directly by substituting the last-significant bits (LSB) of the selected suitable variable length codes (VLCs) by the watermark bit. In [HAA98], Haan and Beller advanced a more robust method consisted in inserting the watermark into selected high-frequency DCT coefficients of the stream. Despite their low complexity, these methods remain fragile against transcoding and geometric attacks.

A special attention in watermarking is paid to the MPEG-4 Part 10 (a.k.a. MPEG-4 AVC or, alternatively H.264) [RIC03] compressed stream. As its ancestors, this standard contains four basic functions: prediction, transformation, quantization and entropic encoding, denoted in Figure I-12 by P, T, Q and E, respectively. In order to achieve higher compression efficiency, MPEG-4 AVC deploys particular compression features such as variable block-size motion estimation, directional spatial prediction, DCT (Discrete Cosine Transform) approximation and context-adaptive entropic encoding. Consequently, the watermarking techniques devoted to uncompressed or to earlier compressed domains (MPEG-2 or MPEG-4 Part 2) are likely to fail in reaching the same performances in the MPEG-4 AVC domain and specific methods should be devised in this respect, [ZOU08], [GOL07], [NOO05], [BEL10] see Table I-1. Figure I-12 shows the actual position, along the watermarking chain, where the insertion takes place.

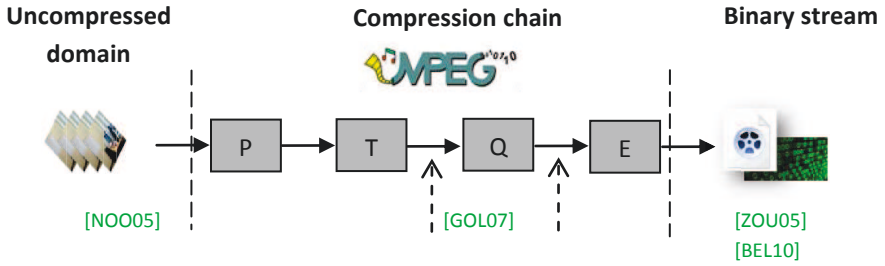


Figure I-12: Encoding chain and related robust watermarking studies, classified according to their insertion domain.

Table I-1: State of the art synopsis of compressed-domain watermarking.

Method	Fragility	Robustness	Transparency	Data payload	Complexity
QIM [GOL07]	Geometric	Transcoding Filtering	PSNR = 32.4 dB	**NA**	Entropic decoding Dequantization
Additive mark embedding Perceptual shaping [NOO05]	Geometric	Filtering	NA	800 bits per 5 minutes	Entropic decoding Dequantization
Stream substitution [ZOU08]	Geometric	Transcoding Filtering	PSNR = 32 dB	**NA**	Binary stream parsing
QIM Perceptual shaping [BEL10]		Transcoding Filtering Geometric	PSNR = 42 dB NCC = 0.99	279 bits per 5 minutes	Entropic decoding

Zou *et al.* [ZOU08] introduce a substitution watermarking method for MPEG-4 AVC stream. This study proves that the stream can be changed while ensuring a very fast technique and respecting the video format structure. Experimental evaluations show that watermarking directly the stream is possible within transparency constraint, but is very limited in terms of robustness and data-payload.

A. Golikeri, P. Nasiopoulos and Z. J. Wang [GOL07] propose an ST-QIM (Spread Transform-QIM) watermarking method. Although improving the performances of the traditional ST-DM, this method features a quite small data payload (one bit per macro-block) and has no robustness against the geometric attacks.

M. Noorkami advances a correlation-watermarking method [NOO05] based on perceptual masking principles: the mark is a simple bipolar message inserted according to a psycho-visual cost. This method provides a very good transparency with a high data payload (3 times better than its predecessors). Its main weakness remains the robustness: even the mundane transcoding attack is able to destroy the mark.

M. Belhaj *et al.* [BEL10] introduce a binary spread transform based QIM for MPEG-4 AVC stream watermarking. By combining QIM principles, spread transform, a perceptual shaping mechanism, and an information-theory driven selection criterion, they achieved a good transparency and robustness against transcoding and geometric attacks.

Table I-1 presents a synoptic comparison among these four methods and brings to light that no algorithm can find an optimal trade-off amongst transparency robustness and data-payload inserted in the MPEG-4 AVC domain. However, note that M. Belhaj *et al.* [BEL10] achieved a good transparency – robustness equilibrium by considering a binary spread transform based QIM technique.

In order to strength this qualitative discussion by some quantitative illustrations, we have implemented the methods in [BEL10] and [GOL07]. The functional evaluations are carried out on the MEDIVALS corpus

(*cf.* Appendix B) and are conducted at three levels so as to evaluate the three-folded data payload-robustness-transparency.

First, the data payload is estimated while keeping a fixed value of transparency (average PSNR of 45 dB and 65 dB, for SD and HD respectively) and robustness (maximal BER of 0.1± 0.03 against bipolar additive noise, transcoding and Stirmark geometric random bending attacks). The results show that:

- The method in [BEL10] features a data payload of 60 bits per minute and 90 bits per minute for SD and HD corpus, respectively.
- The method in [GOL07] cannot allow even 1 bit per minute to be inserted under the prescribed robustness and transparency.

Secondly, the robustness against additive noise, transcoding and Stirmark geometric random bending attack is investigated at fixed data payload (150 bits per minutes) and transparency (PSNR of 45 dB and of 65 dB, for SD and HD respectively). The obtained results are illustrated in Figure I-13:

- The method in [BEL10] ensures a robustness expressed by a BER equal to 0, 0.03 and 0.08 for SD corpus and equal to 0, 0.05 and 0.07 for HD corpus against additive noise, transcoding up to 50% from the bit stream size and Stirmark random bending, respectively.
- The method in [GOL07] shows a BER of 0, 0.17 and 0.45 for SD corpus and of 0, 0.15 and 0.43 for HD corpus against additive noise, transcoding up to 50% from the bit stream size and Stirmark random bending, respectively.

Finally, the transparency is assessed at fixed data payload (150 bits per minutes) and robustness (maximal BER of 0.1± 0.03 against bipolar additive noise, transcoding and Stirmark random bending attacks).

- The method in [BEL10] features a transparency expressed PSNR of 47 dB and 65 dB for SD and HD corpus, respectively.
- The method in [GOL07] cannot provide the prescribed data payload and robustness constraint. Thus, the transparency experiments cannot be performed.

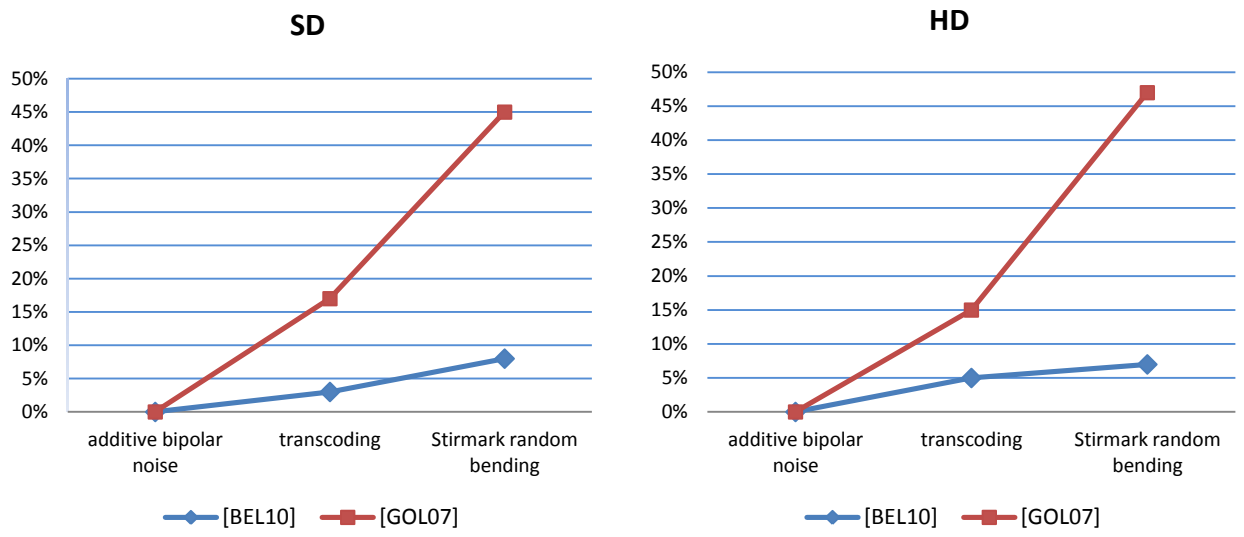


Figure I-13: Robustness evaluations for [BEL10] and [GOL07]

I.2.2. Semi fragile watermarking for video integrity verification

Figure I-5 and Figure I-11 hint to the way in which video integrity verification systems and their underlying block diagrams can be represented. First, the captured video is encoded. Then, an authentication signature is generated based on video content features. This signature is embedded in the video data by the means of some watermarking technique.

In order to verify the integrity of the video, the watermark is extracted and compared to the content based generated signature. This chapter is structured into two subsections. First (Chapter I.2.2.1), the state of the art related to the signature extraction is presented. Second (Chapter I.2.2.1), the video integrity verification watermarking methods are analyzed.

I.2.2.1. Signature extraction

Watermarking based integrity verification was already the subject of several studies [TIT99], [QUE98], [CHE08], [THI05], [THI06], and [ATI10], see Table I-2. Several insertion domains are considered: still images [TIT99], MPEG-1/2 [QUE98], [THI05], [CHE08] and MPEG-4 AVC [THI06], [ATI10].

Titman [TIT99] and Queue [QUE98] use image edges and corners to generate the authentication signature. The signature is embedded according to additional modification rules of overlaying 8x8 blocks. They show that these features are sensitive to content changing alteration. These advantages should be traded-off for the method drawbacks: fragility (against compression and scaling), large size for the signature, and complex generation operations (thus imposing particular constraints on the watermarking insertion method).

Chen and Leung generate the signature based on a chaotic system [CHE08]. Temporal authentication information (frame index and GOP index) is used to compute the signature for each *I* (Intra) frame. The experiments carried out on a video sequence of 795 frames proved that this method detects temporal changes, but the properties of spatial detection of alterations have not been evaluated. The robustness to compression (up 30%) was also shown. Likewise, the signature generation requires an additional level of complexity.

S. Thiemert *et al.* [THI05] calculate the authentication signature in the uncompressed domain, from the points of interest obtained through the Moravec operator [MOR77]. A binary mask is generated for each frame *I*, then embedded into the high-frequency DCT coefficients belonging to adjacent frames. The method detects content changing alterations (object removing/inserting) while being robust to content preserving manipulation (compression up to 50%, scaling). However the authentication signature calculation increases the method complexity. The study in [THI06] resumes and extends these principles. The difference relies on the use of the entropy of gray level in groups of blocks to generate the binary signature. The advanced method is robust against compression down to 50% and detects content changing alterations (object removing). Nevertheless, the signature increases the complexity of the video integrity verification system.

K. Ait Saadi *et al.* [AIT10] consider a signature generated from low frequency quantized DCT coefficients. For each *I* frame, the low frequency DCT coefficients are collected in a buffer to be further hashed using an MD5 function. This results a 128 bit binary signature. The obtained signature is embedded in the *P* and *B* motion vectors. Experiments show that the resulting system remains fragile to all manipulations. Moreover the signature generation requires an MPEG-4 AVC entropic decoding.

To conclude with, Table I-2 brings to light that the trade-off between fragility, robustness, and complexity is not yet reached in the compressed domain. Moreover, the signature is heuristically generated, without any theoretical support.

Table I-2: State of the art studies related to signature extraction.

Method/domain	Signature	Robustness	Fragility	Complexity
J. Titman *et al.* [TIT99]	Edges based	Fragile against transcoding and scaling	Sensitive to spatial alterations	Edge detection Binary edge pattern VLC encoding
P. M. Queue [QUE98] **MPEG-2**	Edges based	Robust against MPEG-2 compression	Detect spatial alteration	Edge detection MPEG decoding
Chen and Leung [CHE08] **MPEG-4 AVC**	Temporal information (GOP index & frame index)	Robust against compression (up to 30%)	Sensitive to temporal alterations	Chaotic modulation
Thiemert *et al.* [THI05] **MPEG-1/2**	Points of interest	Transcoding	Sensitive to Spatio-temporal alterations	Moravec operator MPEG-4 AVC decoding
Thiemert *et al.* [THI06] **MPEG-1/2**	Entropy value pixels	Transcoding	Sensitive to Spatio-temporal alterations	Entropic decoding Entropy computation
Saadi *et al.* [AIT10] **MPEG-4 AVC**	MD5 of DCT residual coefficients		Sensitive to all alterations	MD5 hash function Entropic decoding

I.2.2.2. Integrity verification watermarking techniques

Video integrity verification by means of watermarking techniques was already the object of several research studies, see Table I-3. Such studies consider different signature insertion techniques and different insertion domains, see Figure I-14.

D. Xu and R. Wang [XU11] perform the watermarking at the MPEG-4 AVC entropic encoding level. In this respect, the Exp-Golomb code elements which are eligible to be watermarked without destroying the stream synchronization are first detected. Then, a mapping rule between these elements and the watermark bits is established. The detection is performed by directly parsing the Exp-Golomb code words of the watermarked stream. The performance evaluations show perfect transparency (no quality degradation being induced) but a total fragility to transcoding.

Table I-3: State of the art synopsis for video integrity verification.

Method	Fragility	Robustness	Transparency	Data payload	Complexity
Exponential-Golomb code word mapping [XU11]	Sensitive to all manipulations		**NA**	**NA**	Binary stream parsing
Motion vector and macroblock mode LSB embedding [KIM12]	Sensitive to all manipulations		PSNR = 40 dB	**NA**	Entropic decoding
Changing the parity of the last nonzero coefficient [WAN10]	Sensitive to all manipulations		**NA**	**NA**	Entropic decoding
Modifying the nonzero quantized coefficients [ZAN06]	Sensitive to all manipulations		PSNR > 35 dB	**NA**	Entropic decoding
Imposing local intensity relations into a group of adjacent blocks [CHE08]	Temporal alterations	Frame-level JPEG(*QF=30*) Median filetring	PSNR = 40 dB	**NA**	Uncompressed domain
Enforcing DCT coefficients relations [THI06]	Sensitive to all manipulations	Frame-level JPEG(*QF=50*)	**NA**	**NA**	Entropic decoding
Reactivating skipped macroblocks [PRO05]	Sensitive to all manipulations		PSNR > 50 dB	**NA**	Entropic decoding
Modifying the number of quantized nonzero AC coefficients [WAN08]	Sensitive to all manipulations	**NA**	**NA**	**NA**	Entropic decoding

T. Kim *et al.* [KIM12] embed the watermark bits in the motion vectors of the inter blocks or in the intra mode number of the intra blocks. The advanced method features a high data payload with small image quality degradation (a PSNR = 40 dB is reported). Nevertheless, a large sensitivity to transcoding is featured.

C. C. Wang and Y. C. Hsu [WAN10] present a fragile watermarking algorithm to authenticate MPEG-4 AVC stream. The mark is computed as the MD5 (message digest algorithm) hash function of a random generated binary sequence and embedded by changing the parity of the high-frequency quantized DCT coefficients of *I* frames. While such a technique provides the ideal case of fragility and features low complexity (only the MPEG-4 AVC entropic decoding being required), it is conceptually unable to make any distinction between mundane and malicious attacks.

J. Zang and A. T. S. Ho [ZAN06] adopt the same principles and insert the mark in the *P* frames. The overall results show good transparency (PSNR > 35 dB), a very good sensitivity to spatio-temporal alterations, low complexity but no robustness to content preserving attacks (*e.g.* transcoding).

S. Chen and H. Leung present a semi-fragile watermarking scheme based on chaotic systems for the authentication of individual frames in the MPEG-4 AVC stream [CHE08]. The authentication information is represented by both the GOP index and the frame index in that GOP. This information is modulated in a chaotic signal and inserted in the DCT transformed blocks of each frame by imposing local intensity relationships into a group of adjacent blocks. The insertion requires the entropic decoding, the de-quantizing and the reverse of the prediction operations, thus becoming computationally complex. Experiments carried out on a 795 frames video sequence proved a transparency expressed by a PSNR = 40 dB and robustness against JPEG compression (quality factor QF of 30) and median filtering. This method also detects the temporal modifications (with one frame accuracy) but the spatial modification properties were not assessed.

S. Thiemert *et al.* [THI06] advance a semi-fragile watermarking system devoted to the MPEG-1/2 video sequences. The mark computation is based on the properties of the entropy computed at the 8x8 block levels. The mark is embedded by enforcing relationship between the DCT coefficients of some blocks. The experiments are run on one sequence (whose length is not précised) encoded at 1125 kbps. The method proved both robustness (against JPEG compression with QF=50) and fragility against temporal (with 2 frame accuracy) and spatial (with a non-assessed accuracy) content changing. However, the main drawback of this method remains its inner computation complexity: beyond the complete MPEG decoding/encoding, it also requires sophisticated entropy estimation at frame levels.

Proforck *et al*. [PRO05] propose an integrity authentication schema of MPEG-4 AVC. The authentication information which consists of the encrypted hash value and a certificate with public key is embedded by reactivating some skipped macroblocks. The advantage of the method is the possibility of erasing the watermark, but the considered hash algorithm increases the computation cost of the scheme. The transparency was evaluated at a PSNR > 50 dB.

Chen, Chen and Wang [WAN08] compute the authentication data as the block sub-index. Then, the obtained signature is embedded by modifying the number of nonzero DCT AC coefficients of *I* frames. The experimental results show that the proposed system can detect the illegally altered area. However, neither the transparency nor the robustness against non malicious alteration has been evaluated.

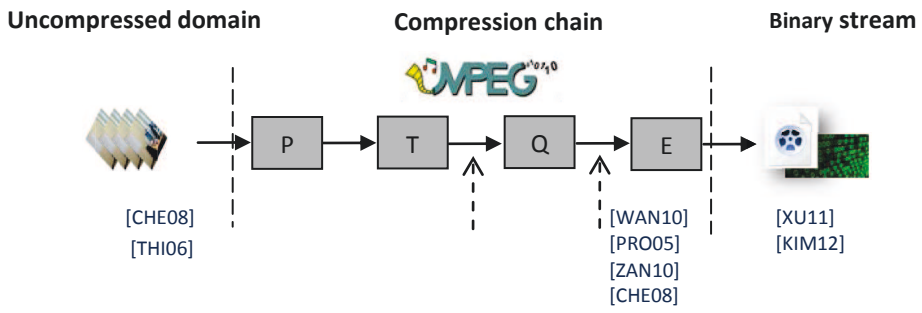


Figure I-14: Encoding/decoding chain and related semi-fragile watermarking studies.

As it can be seen, Table I-3, the trade-off among fragility, robustness and complexity is not yet achieved. Moreover, the studies related to semi-fragile and robust watermarking are divergent in their concept and approach.

I.2.3. Conclusion

Tables I-1, I-2 and I-3 show that the large variety of state of the art methods allow each particular constraint (transparency, robustness, data payload, computational cost) to be individually reached for particular applications. However, none of these studies is able to jointly reach these four requirements.

For instance, for robust watermarking, the method in [BEL10], based on binary *QIM*, seems to provide the best robustness, transparency and computational trade off but leaves room for data payload improvement.

For semi-fragile watermarking, the trade-off between fragility, robustness, and complexity is not yet reached in the compressed domain. Even though some studies seem able to reach a certain functional balance between the fragility and robustness such as [TEI06] and [CHE08], the authentication signature generation/insertion is still performed in the uncompressed domain and subsequently increase the watermarking computational cost. Moreover, the signature is heuristically generated, without any theoretical support.

Furthermore, to the best of our knowledge, no watermarking method is able today to serve both the purposes of robust and semi-fragile watermarking.

I.3. Thesis overview

This chapter will be structured into three parts devoted to the challenges, the main contributions and the thesis structure, respectively.

I.3.1. Challenges

The present thesis deals with theoretical and methodological issues related to the MPEG-4 AVC watermarking for ownership protection and video integrity verification.

From the theoretical point of view, the thesis main challenge is to develop a unitary watermarking framework (insertion/detection) able to serve the two above mentioned applications. From the methodological point of view, the challenge is to instantiate this theoretical framework for serving the targeted applications.

The ownership protection is considered under a VoD (Video on Demand) framework, *cf*. the MEDIEVALS project. The objective is the increase of the data payload for pre-established robustness and transparency levels. The robustness should be evaluated against noise addition, transcoding, and Stirmark random bending attacks. The transparency should correspond to the humanly imperceptible artifacts. The advanced watermarking method should not require any additional decoding/encoding operation.

The video integrity verification is considered under a mobile video surveillance system, *cf*. the SPY project. The challenge is to extract from the video stream an authentication signature which is subsequently inserted so as to ensure the video integrity. This signature should be robust to content preserving attacks and fragile to content alteration attacks. The accuracy of the content altered regions should be evaluated both spatially and temporally.

I.3.2. Contributions

The present thesis tackles the above mentioned challenges by the following theoretical and methodological contributions, see Figure I-15.

Multi symbol quantization index modulation watermarking (m-QIM)

The thesis first contribution consists in building the theoretical framework for the multi-symbol watermarking based on quantization index modulation (*m*-QIM). The insertion rule is analytically designed by extending the binary QIM insertion rule. The detection rule is optimized so as to ensure minimal probability of error under additive white Gaussian noise distributed attacks. It is thus demonstrated that the data payload can be increased by a factor of $\log_2 m$, for prescribed transparency and additive Gaussian noise power.

The *m*-QIM framework is first deployed for ensuring the VoD (Video on Demand) ownership protection. The main benefit is the increase of data payload by a factor of $\log_2 m$ for a prescribed robustness of 0.1 of BER (variations lower than 3% of the bit error rate after additive noise, transcoding and Stirmark random

bending attacks) and transparency (set to average PSNR = 45dB and 65dB for SD and HD encoded content, respectively). Actually, the experiments considered 4 values of m, namely $m = 2$, $m = 3$, $m = 5$ and $m = 7$; just for illustration, for $m = 5$, a data payload of 150 bits per minute, *i.e.* about 20 times larger than the limit imposed by the DCI (Digital Cinema Initiatives) standard, is obtained. The processed corpus sums up to 1 h of video content granted by the MEDIEVALS industrial partners.

The second *m*-QIM application consists in designing a semi-fragile watermarking method for video integrity verification in compressed stream. In this respect, the MPEG-4 AVC syntax elements which can optimally (in the information theory sense) serve as authentication signature are identified. This authentication signature is further inserted by combining the *m*-QIM principle to an alteration detection strategy. The experiments results show fragility to content replacement (with an 1/81 frame and 3 seconds spatial and temporal accuracy, respectively) and robustness against noise addition and transcoding (compression by a factor of 2). The *m*-QIM framework main advantage is this time a relative gain factor of 0.11 of PSNR for fixed robustness (against noise addition and transcoding), fragility (to content alteration) and the data payload. The processed corpus sums up 1h 20 minutes of heterogeneous video contents granted by the SPY industrial partners.

The computational time required by each operation included in the watermarking chain is evaluated on the following PC configuration: a Core4 CPU at 2.8 GHz and with 12 GB of RAM and a 500 GB HDD. The signature generation and insertion consume 0.004 s, and 0.008 s respectively, for one second of video; this represents 0.28% and 0.55% respectively from the total embedding processing time. When compared to MPEG-4 AVC reference software (JM86), the mark generation and insertion is about 100 times faster than entropic encoding/decoding and stream read/write.

Drift-free watermarking

The thesis second theoretical contribution consists in specifying a preprocessing MPEG-4 AVC shaping operation which can eliminate the intra-frame (spatial) drift effect. The drift represents the distortion spread in the compressed stream related to the MPEG encoding paradigm. In this respect, the drift distortion propagation problem in MPEG-4 AVC is algebraically expressed and the corresponding equations system is resolved under drift-free constraints.

The experiments consider the same *m*-QIM semi-fragile watermarking method and the same SPY corpus. For prescribed data payload (100 bits/s), robustness (BER < 0.1 against transcoding at 50% in stream size), fragility (frame modification detection with accuracies of 1/81 from the frame size and 3 seconds) and complexity constraints, the drift-free shaping results in gains in transparency of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ.

Properties	Constraints	Thesis contributions
Data payload	***Compressed domain watermarking:*** ▪ Compressed video stream leaves very little room space to hide data.	***m-QIM insertion rule*** ▪ ***Theory:*** Increasing the data payload by a factor of *log2m.* ▪ ***Practice:*** Inserting a data payload of 150 bits per minute for prescribed robustness and transparency, for VoD ownership protection.
Robustness	***Ownership protection:*** ▪ The embedded mark should be recovered after additive noise, transcoding, and Stirmark random bending attacks.	***m-QIM optimal decision rule*** ▪ ***Theory:*** Minimize probability of error under additive white Gaussian noise distributed attacks. ▪ ***Practice:*** Robustness of 0.1 of BER (variations lower than 3% of the bit error rate after additive noise, transcoding and Stirmark random bending attacks).
Fragility	***Mark semantic:*** ▪ The embedded mark should reflect the semantic for the video content so as to distinguish between content changing and content preserving alterations. ***Integrity verification:*** ▪ The mark should be robust against content preserving attacks while being fragile against content alteration attacks.	***MPEG-4 AVC Syntax element based signature*** ▪ ***Theory:*** Identifying the syntax elements which can optimally (in the information theory sense) serve as authentication signature ▪ ***Practice:*** Robustness against transcoding and Fragility to content replacement (with an 1/81 frame and 3 seconds spatial and temporal accuracy, respectively).
Transparency	***Compressed domain watermarking:*** ▪ Compressed stream processing leads to drift distortion propagation.	***Drift-free shaping for MPEG-4 AVC compressed stream watermarking.*** ▪ ***Theory:*** Analytically expressing the drift distortion problem and resolving it under drift-free constraint. ▪ ***Practice:*** Gains in transparency of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ. for prescribed data payload (100 bit/s), robustness (BER < 0.1 against transcoding at 50% in stream size).
Complexity	***Compressed domain watermarking:*** ▪ Mark embedding should not increase the computational cost of the watermarking application.	***MPEG-4 AVC compressed domain watermarking*** ▪ ***Practice:*** The mark generation, selection, insertion and detection for one second of video are 10 times and 8 times faster than the MPEG-4 AVC entropic encoding/decoding and the stream read/write operations, respectively.

Figure I-15: Thesis contributions and results

I.3.3. Structure

In order to reach the above detailed objectives, the thesis manuscript is structured into three Parts preceded by this Introduction and followed by three appendixes.

Part II is structured into three chapters. Chapter II.1 is devoted to specifying the multi symbol *m*-QIM watermarking framework by generalizing the binary rule and optimizing the underlying detection rule with respect to the minimization of the average error probability, under the hypothesis of white, additive Gaussian behavior for the attacks. Chapter II.2 advances the robust *m*-QIM watermarking method for the ownership in the MPEG-4 AVC compressed domain and evaluates its performances under the MEDIVALS

corpus. The third chapter also exploits the *m*-QIM principles, this time in conjunction with authentication signature: an integrity verification system for MPEG-4 AVC video surveillance is thus designed and demonstrated under the SPY corpus.

Part III tackles the transparency issue under drift constraints in the MPEG-4 AVC compressed stream. First, by considering the analytic expressions of the MPEG-4 AVC encoding operations, it algebraically models the drift distortion spread as an optimization problem. Second, it solves this problem under drift-free constraints. Finally, the advanced solution is adapted so as to take into account the watermarking restrictions

Conclusions are drawn and perspectives are opened in Part IV.

The thesis has three Appendixes. Appendix A is devoted to MPEG-4 AVC encoding/decoding features. Appendix B presents the corpora processed in the experiments. Appendix C details a key issue related to the *m*-QIM framework, namely the probability of error at the detection as function of the Gaussian noise standard deviation σ.

Part II: Multi symbol QIM watermarking

Abstract

Part II contains three chapters. Chapter II-1 advances the *m*-QIM theoretical framework by generalizing the binary insertion rule and optimizing the underlying detection rule with respect to the minimization of the average error probability, under the hypothesis of white additive Gaussian behavior for the attacks. Chapter II-2 deploys the robust *m*-QIM watermarking method for the ownership protection in the MPEG-4 AVC compressed domain and evaluates its performances under the MEDIVALS corpus. Chapter II-3 conducts a theoretical investigation on the syntax elements based authentication signature and advances a semi –fragile watermarking method for MPEG-4 AVC compressed domain integrity verification and evaluates its performances on the SPY corpus.

II.1. Theoretical contribution: *m*-QIM watermarking framework

The efficiency of any noisy channel coding (modulation) technique is ultimately evaluated by its bit-rate *i.e.* by the quantity of information which is transmitted through that channel. In this respect, while the binary modulation solutions are the most popular, *m*-ary solution are very appealing for increasing the bit rate [PRO01].

From the conceptual point of view, the increase of the alphabet size from 2 to m would result in an increase of the bit rate by a factor of $log_2 m$. However, the larger the size of the alphabet, the closer the amplitude level of the symbol carriers. Hence, in the presence of noise, it will be *a priori* difficult at the receiver to discriminate between two symbols [PRO01].

Consequently, the practical impact of m-ary modulation should be evaluated according to the targeted application. For instance in [FIT00], a comparison is conducted among 7 types of modulation (BPSK, BFSK, 4QAM, 4PSK, 8PSK, 16QAM, 16PSK). Table II-1 illustrates the obtained bandwidth efficiency ratio (Capacity/Bandwidth) and the error free ratio (Binary error/Noise). By analyzing these results, the following discussions are raised:

- Quadratic amplitude modulation (4QAM) features a gain factor of 2 in bandwidth efficiency with respect to the binary frequency shift keying (BFSK) at the same error free ratio (around 10dB); hence, the hypothesis of a gain of $log_2 m$ holds in this case.
- Phase shift keying multi modulation (8PSK) offers a gain factor of 3 in bandwidth efficiency with respect to the binary frequency shift keying (BFSK) modulation at the expense of increasing the error free ratio by 1.5 dB; hence the gain of $log_2 m$ is here an approximation.

Similar types of result are reported in [JAC67] for BPSK, 4PSK and 8PSK, in [YAN02] for BPSK, 4PSK and 8PSK and in [DES12] for 4PSK, 16 QAM and 64QAM.

Table II- 1: m-ary modulation performances comparison.

	Bandwidth efficiency (Capacity/Bandwidth)	**Error free (Binary error/Noise)**
16PSK (16 Phase Shift Keying)	4	18dB
16QAM (16 Quadratic Amplitude Modulation)	4	15dB
8PSK (8 Phase Shift Keying)	3	14.5dB
4PSK (4 Phase Shift Keying)	2	10.1dB
4QAM (4 Quadratic Amplitude modulation)	2	10.1dB
BFSK (Binary Frequency Shift Keying)	1	13dB
BPSK (Binary Phase Shift Keying)	1	10.5dB

Our study follows the same approach: first, it theoretically investigates the *m*-QIM multi symbol quantization index modulation then its effectiveness for watermarking applications.

As the state of the art of watermarking in compressed domain brought to light that the requirements of transparency and robustness could be reached by spread transform and quantization index modulation based methods [BEL10], the present chapter takes the challenge to increase the data payload performances while keeping the transparency and robustness features. To this aim, an extension to multi symbol insertion has been targeted.

The main contribution consist in advancing the theoretical framework allowing for the binary Quantization Index Modulation (QIM) embedding techniques to be extended towards multiple-symbol QIM (m-QIM, where m stands for the number of symbols on which the mark is encoded prior to its embedding). The underlying detection method is optimized with respect to the minimization of the average error probability, under the hypothesis of white, additive Gaussian behavior for the attacks. This way, for prescribed transparency and robustness constraints, the data payload is increased by a factor of $log_2 m$.

II.1.1. *m*-QIM insertion rule

II.1.1.1. Binary QIM

Consider the case in which some binary information b is to be inserted in some (floating point) original data x by means of binary QIM methods [GOL07], [BEL10]. To do so, x is quantized using multiple quantizers whose indexes are chosen based on the message to be embedded [CHE98].

To implement such quantizers, dither modulation (DM) can be used [CHE98], thus obtaining the watermarked signal y:

$$\begin{cases} q = Q_\Delta\left(x - \Delta(b/2 + k)\right) - \left(x - \Delta(b/2 + k)\right) \\ y = x + \alpha q \end{cases} \quad \text{(II-1)}$$

where Δ is a fixed quantization step size, k a random key (sampled from a white, uniform noise, $0 < k \leq 1$) and α a fixed parameter, $0 < \alpha \leq 1$.

The quantizer Q_Δ is defined as follows:

$$Q_\Delta(x) = \Delta Round(x/\Delta) \quad \text{(II-2)}$$

where $Round()$ is the approximation to the closest integer.

The practical balance between the transparency and the robustness can be reached by adjusting the α and Δ parameters: the lower the α and Δ values, the lower the difference between x and y and, consequently, the greater the transparency but the worse the robustness.

At the decoder, the embedded message bits are recovered by a scalar quantization of the received signal sample *r* which represents the y signal after its corruption by attacks.

The $Y(b)$ decision variable is computed as follows [CHE98], [BEL10]:

$$Y(b) = Q_{\Delta}(r - k\Delta) - r + k\Delta \tag{II-3}$$

The aim is to decide whether the inserted bit was $b = 0$ or $b = 1$. The optimal decision rule, assuming the attacks are modelled by additive white Gaussian noise is (see Figure II-1) [CHE98]:

$$\begin{cases} |Y(b)| < (1-\alpha)\,\Delta/2 \;\rightarrow\; \hat{b} = 0 \\ |Y(b)| \geq (1-\alpha)\,\Delta/2 \;\rightarrow\; \hat{b} = 1 \end{cases} \tag{II-4}$$

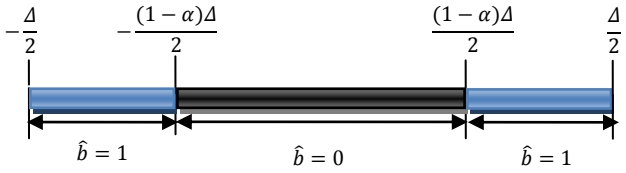


Figure II-1: Decision regions for binary QIM.

II.1.1.2. Spread transform dither modulation

ST-DM is a particular form of QIM. The watermark is not directly embedded into the original signal x but into the projection x' of x onto a randomly generated vector u. This technique is called Spread Transform as proposed by Chen and Wornell in [CHE98]. The resulting scalar value is then quantized before being added in the component of the signal:

$$\begin{cases} q = Q_{\Delta}\left(x^T u - \Delta\left({}^{b}/_{2} + k\right)\right) - \left(x^T u - \Delta\left({}^{b}/_{2} + k\right)\right) \\ y = x + (\alpha q)u \end{cases} \tag{II-5}$$

The corresponding detection is given by equation (II-4) where the detection function *Y* (*b*) is:

$$Y(b) = Q_{\Delta}(r^T u - k\Delta) - r^T u + k\Delta \tag{II-6}$$

ST-DM is more robust to re-quantization (*e.g.* JPEG compression) than regular DM [CHE98]. However, ST-DM introduces relatively higher perceptual distortions. To reduce perceptual distortion, perceptual models can be considered [WAT01], [NOO05], [BEL10].

II.1.1.3. *m*-QIM insertion

In order to generalize the insertion technique, be there the same binary message b to be inserted in the same original data x.

Prior to its insertion, the message b is encoded into a message d belonging to an m-ary alphabet D; assuming m is an odd value:

$$D = \{-\tfrac{m-1}{2}, -\tfrac{m-2}{2}, \ldots\ 0, \ldots \tfrac{m-2}{2}, \tfrac{m-1}{2}\}.$$

On average, each d symbol corresponds to $log_2 m$ bits from the message b [PRO01].

By following the principles above, the *m*-QIM insertion can be expressed as:

$$\begin{cases} q = Q_{\Delta}\left(x - \Delta\left({}^{d}/_{m} + k\right)\right) - \left(x - \Delta\left({}^{d}/_{m} + k\right)\right) \\ y = x + \alpha q \end{cases} \quad \text{(II-7)}$$

As in the binary case, the lower the α and the Δ, the lower the difference between x and y and the greater the transparency but the worse the robustness. However, this time, each x sample bears one d *m*-ary symbol, thus increasing the data payload by a factor of $log_2 m$, with respect to the binary case.

II.1.2. *m*-QIM detection rule

While keeping practically the same insertion rule, see (II-1) *vs*. (II-7), the multi-symbol generalisation requires the modification of the decision rule. In the present paper, a decision rule minimising the probability error in mark detection under the white additive Gaussian noise hypothesis is derived.

Let us consider first the case in which no attack occurs. The decision is based on the value of the $Y(d)$ variable:

$$Y(d) = Q_{\Delta}(x + \alpha q - k\Delta) - (x + \alpha q - k\Delta) \quad \text{(II-8)}$$

$Y(d)$ is a quantization error belonging to the $\left[-\frac{\Delta}{2};\frac{\Delta}{2}\right]$ interval. Hence, specifying a decision rule means to divide the decision region $\left[-\frac{\Delta}{2};\frac{\Delta}{2}\right]$ into *m* non-overlapping intervals. These intervals are computed based on a three-steps development presented below. First, the $Y(d)$ expression is reformulated so as to no longer depend on the original data x. Secondly, the decision intervals are computed by expressing the quantization error as a function of α, d, m and Δ. Finally, the value of the α parameter ensuring non-overlapping intervals is computed.

Step 1: $Y(d)$ reformulation:

We denote: $A = x - \Delta\left({}^{d}/_{m} + k\right)$ and $B = x + \alpha q - k\Delta$.

The expression of A as a function of B is obtained as follows:

$$\begin{aligned} B &= x + \alpha\left(Q_{\Delta}\left(x - \Delta\left({}^{d}/_{m} + k\right)\right) - \left(x - \Delta\left({}^{d}/_{m} + k\right)\right)\right) - k\Delta \\ &= x + \alpha Q_{\Delta}(A) - \alpha\left(x - \Delta\left({}^{d}/_{m} + k\right)\right) - k\Delta \\ &= (1-\alpha)\left(A - Q_{\Delta}(A)\right) + Q_{\Delta}(A) + \Delta\,{}^{d}/_{m} \\ &= (\alpha - 1)q + Q_{\Delta}(A) + \Delta\,{}^{d}/_{m} \end{aligned} \quad \text{(II-9)}$$

Then, $Y(d)$ can be written as follows:

$$Y(d) = Q_{\Delta}(B(d)) - B(d)$$

$$= Q_\Delta\left((\alpha-1)q + Q_\Delta(A) + \Delta\, d/m\right) - (\alpha-1)q - Q_\Delta(A) - \Delta\, d/m$$

As Q_Δ is by its definition a multiple of Δ, $Y(d)$ can be simplified as follows:

$$Y(d) = Q_\Delta\left((\alpha-1)q + \Delta\, d/m\right) - (\alpha-1)q - \Delta\, d/m$$

Let $C(d) = (\alpha-1)q + \Delta\, d/m$ then:

$$Y(d) = Q_\Delta(C(d)) - C(d) \tag{II-10}$$

Note that while (II-8) and (II-10) are equivalent from the mathematical point of view, the right-hand term of (II-10) does no longer depend on the original (unmarked) data x.

Step 2: Decision intervals as a function of α, d, m and Δ:

We have $-\Delta/2 < q < \Delta/2$ and $0 < \alpha \leq 1$; this implies:

$$\Delta((\alpha-1)m + 2d)/2m < C(d) < \Delta((1-\alpha)m + 2d)/2m \tag{II-11}$$

Be there:

$$\begin{cases} I_{sup,\alpha}(d) = \dfrac{\Delta((1-\alpha)m + 2d)}{2m} \\ I_{inf,\alpha}(d) = \dfrac{\Delta((\alpha-1)m + 2d)}{2m} \end{cases} \tag{II-12}$$

From equations (II-10) and (II-11) we obtain:

$$-I_{sup,\alpha}(d) + Q_\Delta(C(d)) < Y(d) < -I_{inf,\alpha}(d) + Q_\Delta(C(d)) \tag{II-13}$$

$Q_\Delta(C(d))$ is the quantized value of $C(d)$ with quantization step Δ; hence:

$$\begin{cases} Q_\Delta(C(d)) = l\Delta, l \in Z \\ \left(l - \dfrac{1}{2}\right)\Delta \leq C(d) < \left(l + \dfrac{1}{2}\right)\Delta \end{cases} \tag{II-14}$$

Equations (II-11) and (II-14) imply:

$$\begin{cases} \dfrac{((1-\alpha)m + 2d)}{2m} < l + 1/2 \\ \dfrac{((\alpha-1)m + 2d)}{2m} > l - 1/2 \end{cases} \tag{II-15}$$

As d is a symbol from the *m*-ary alphabet *D*, we have:

$$|d| \leq (\mathrm{m} - 1)/2$$

Hence, equations (II-15) gives:

$$|l| < \frac{\left|\frac{(m-1)}{m} - \alpha\right|}{2} \quad \text{(II-16)}$$

Then: $l = 0$ and consequently $Q_\Delta(C(d)) = 0$. According to (II-10) we get:

$$-I_{sup,\alpha}(d) \leq Y(d) \leq -I_{inf,\alpha}(d) \quad \text{(II-17)}$$

Eq. (II-17) demonstrates that the insertion of the d symbol results in $Y(d)$ values belonging to the $I_\alpha(d) = [-I_{sup,\alpha}(d) \quad -I_{inf,\alpha}(d)]$ interval. $I_\alpha(d)$ depends on Δ, α, and m: while Δ and m are fixed for an application, α is a parameter which can be chosen so as to ensure non-overleaping decision intervals.

Step 3: Computing the optimal α value:

For a fixed value of α parameter, $I_{sup,\alpha}$ and $I_{inf,\alpha}$ defined in equation (II-12) are positive slope affine functions, *i.e.* increasing functions of d. Hence, if each two successive symbols (*d*, *d*+1) have nonoverlapping decision intervals, then we will have m nonoverlapping decision intervals:

$$I_{sup,\alpha}(d) - I_{inf,\alpha}(d+1) \leq 0$$

This yields to:

$$\frac{\Delta((\alpha-1)m + 2d)}{2m} - \frac{\Delta((1-\alpha)m + 2(d+1))}{2m} \leq 0 \quad \text{(II-18)}$$

Equation (II-18) implies that the condition for obtaining non-overleaping $I_\alpha(d)$ intervals is $\alpha \geq \frac{m-1}{m}$; be $\alpha^* = \frac{m-1}{m}$. The influence of α on the $I_\alpha(d)$ intervals is illustrated in Figure II-2 for $m = 5$ (hence, $\alpha^* = 0.8$) and $\Delta = 70$. Three particular cases are considered, namely $\alpha = 0.76 < \alpha^*$ (see Fig II-2 up-left), $\alpha = 0.8 = \alpha^*$ (see Figure II-2 up-right) and $\alpha = 0.84 > \alpha^*$ (see Figure II-2 bottom). In these three plots, the abscissa corresponds to the value of d, while the ordinate stands for the $I_{inf,\alpha}$ (in blue diamond) and $I_{sup,\alpha}$ (in red square).

The α^* can be considered as the optimal α value for computing the decision intervals: it ensures non overlapping intervals (hence ideal robustness, assuming no attack occurs) and the best transparency (the minimal differences between the host and the marked signals). Consequently, the optimal decision rule associates to each inserted d symbol a detection interval $I_{\alpha^*}(d) = [-I_{sup,\alpha^*}(d) \quad -I_{inf,\alpha^*}(d)]$. In other words, when a particular $Y(d)$ value is computed at the watermarking detection side, we decide that the inserted symbol was $\hat{d}$, where:

$$if\ Y(d) \in I_{\alpha^*}(\hat{d}) => d = \hat{d} \quad \text{(II-19)}$$

Assume now the case in which the attacks are present: the insertion of a d symbol can result now in a value $Y(d)$ outside the I_{α^*} interval. Intuitively, in order to decrease the errors induced by such a situation, a value $\alpha > \alpha^*$ should be considered at insertion/detection, *i.e.* a lower transparency should be accepted in order to grant some additional robustness against attacks, see Figure II-2. The I_{α^*} and I_α obtained for *m* = 5 (hence $\alpha^* = 0.84$), $\alpha = 0.84$ and $\Delta = 70$ are illustrated in Figure II-3.

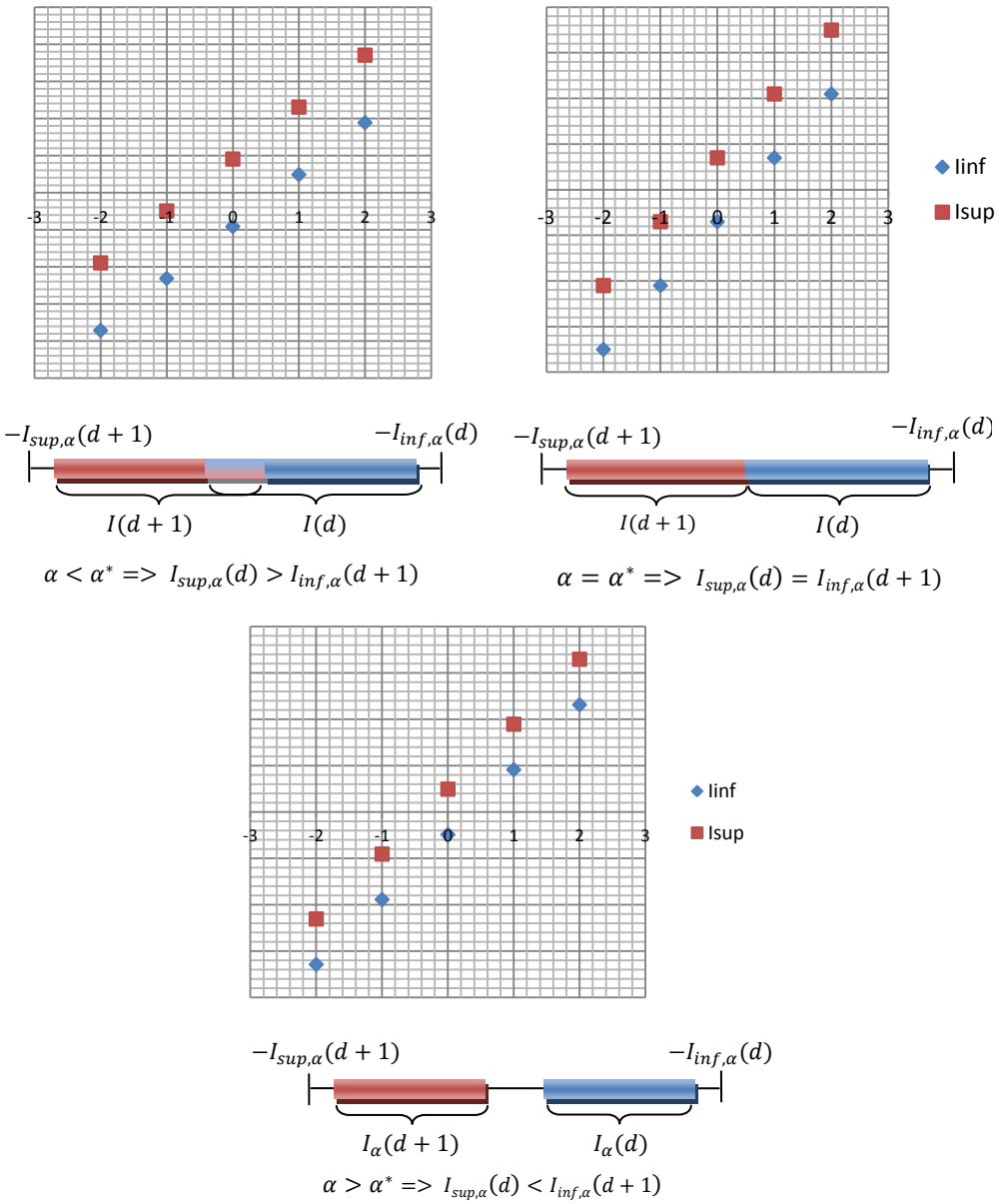


Figure II-2: $\boldsymbol{I_{sup,\alpha}}$ and $\boldsymbol{I_{inf,\alpha}}$ as a function of $\boldsymbol{d}$, illustrated for $\boldsymbol{m = 5}$, $\boldsymbol{\Delta = 70}$ and three values of $\boldsymbol{\alpha}$: $\alpha = 0.76 < \alpha^*$ (up-left), $\alpha = 0.8 = \alpha^*$ (up-right) and $\alpha = 0.84 > \alpha^*$ (bottom).

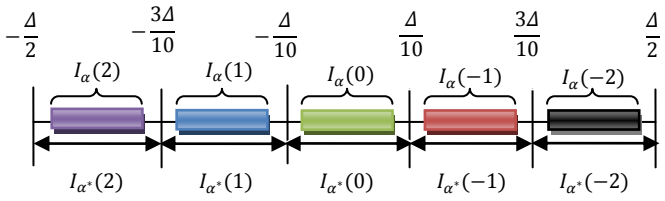


Figure II-3: Decision regions for m=5, $\Delta = \boldsymbol{70}$ and $\boldsymbol{\alpha = 0.84 > \alpha^* = 0.8}$.

As usual in the watermarking studies [EGG03], we shall assume that the attacks are independent with respect to the inserted symbol and that they can be modelled by a white, Gaussian noise which is added to the $Y(d)$ value computed for a fixed $\alpha > \alpha^*$ parameter; the $\tilde{Y}(d)$ variable becomes:

$$\tilde{Y}(d) = Y(d) + n$$

$\tilde{Y}(d)$ can fall between two $I_\alpha(d)$ decision regions or even outside the $\left[-\frac{\Delta}{2};\frac{\Delta}{2}\right]$ interval. Consequently, the optimal decision rule defined in equations (8) and (19) should be extended so as to cope with such a situation, as follows. First, although the insertion is performed for an $\alpha > \alpha^*$ value, the decision regions will be computed on the I_{α^*} basis: this way, all the values inside the $\left[-\frac{\Delta}{2};\frac{\Delta}{2}\right]$ are considered. Moreover, in order to avoid the cases in which $\tilde{Y}(d)$ would fall outside the $\left[-\frac{\Delta}{2};\frac{\Delta}{2}\right]$ interval, a modulo operator is applied prior to the detection. The corresponding decision variable $\tilde{Y}_\Delta(d)$ and the underlying decision rule are:

$$\begin{aligned} &\tilde{Y}_\Delta(d) = \tilde{Y}(d) modulo\ \Delta - \frac{\Delta}{2} \\ &if\ \tilde{Y}_\Delta(d) \in I_{\alpha^*}(\hat{d}) => d = \hat{d} \end{aligned} \qquad \text{(II-20)}$$

The error probability associated to the decision rule in equation (II-20) and its optimality are discussed in the following chapter.

II.1.3. Optimal decision rule

II.1.3.1. Computing the probability density functions for decision variables

In the sequel, we shall incrementally express the probability density function for the $Y(d)$, $\tilde{Y}(d)$ and $\tilde{Y}_\Delta(d)$ decision variables.

The $Y(d)$ variable is computed as the result of a quantization error and belongs to the $I_\alpha(d) = [-I_{sup,\alpha}(d) \quad -I_{inf,\alpha}(d)]$ intervals, *cf.* (II-8) and (II-19). Consequently, the probability density function of the $Y(.)$ variable conditioned on the insertion of the d symbol is denoted by $p_Y(u/d)$ and can be modelled by a uniform law in that interval [PRO01]:

$$p_Y(u/d) = \begin{cases} \frac{1}{(1-\alpha)\Delta}, & if\ u \in I_\alpha(d) \\ 0, & if\ not \end{cases}$$

The noise probability density function, denoted by $p_n(n)$, is assumed to follow a normal (Gaussian) law of $\mu = 0$ mean and σ standard deviation:

$$p_n(n) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-n^2}{2\sigma^2}}$$

As $\tilde{Y}(d) = Y(d) + n$, its probability density function conditioned on the insertion of the d symbol can be expressed as the convolution between the $p_Y(y/d)$ and $p_n(n)$:

$$p_{\tilde{Y}}(y/d) = \big(p_Y(u/d) \otimes p_n(n)\big)(y) = \frac{1}{(1-\alpha)\Delta}\int_{-I_{sup,\alpha}(d)}^{-I_{inf,\alpha}(d)} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(y-t)^2}{2\sigma^2}} dt$$

Thus, $p_{\tilde{Y}}(y/d)$ can be expressed by using the $erfc$ function:

$$p_{\tilde{Y}}(y/d) = \frac{1}{2(1-\alpha)\Delta}\left(erfc\left(-\frac{I_{sup,\alpha}(d)+y}{\sqrt{2}\sigma}\right) - erfc\left(-\frac{I_{inf,\alpha}(d)+y}{\sqrt{2}\sigma}\right)\right) \quad \text{(II-21)}$$

where $erfc(.)$ is the complementary error function function defined by:

$$erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2} dt$$

As $\tilde{Y}_\Delta(d) = \tilde{Y}(d) modulo\ \Delta - \frac{\Delta}{2}$, its conditional probability on the insertion of the d symbol was computed in our study from the $p_{\tilde{Y}}(y/d)$, by following basic principles in non-linear random variable filtering:

$$p_{\tilde{Y}_\Delta}(y/d) = \sum\nolimits_i p_{\tilde{Y}}\left((\frac{i\Delta}{2}+y)/d\right), i = 2j+1, j \in Z \quad \text{(II-22)}$$

Equations (II-21) and (II-22) show that irrespective to the d symbol, $p_{\tilde{Y}_\Delta}(y/d)$ has a maximal value corresponding the centre of the $I_\alpha(d)$ and symmetrically decreases from that point, as illustrated in Figure II-4 for $m = 5$, $\Delta = 70$ and $\alpha = 0.84$.

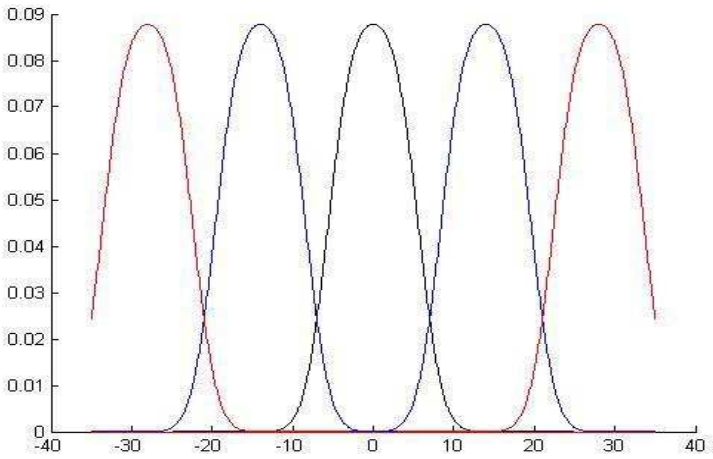


Figure II-4: Received signal distribution: illustration for $m = 5$, $\Delta = 70$ and $\alpha = 0.84$.

II.1.3.2. The optimal decision rule

In its widest acceptation, specifying a decision rule means to define a partition of the interval in which the detection variable takes values [EGG03]. In the *m*-QIM case, this means to define a partition Δ_i, $i \in \{0,1,\dots,m-1\}$ of the $[-\Delta/2, \Delta/2]$ interval:

$$\cup_i \Delta_i = [-\Delta/2, \Delta/2] \text{ and } \cap_i \Delta_i = \emptyset \quad \text{(II-23)}$$

Consider now the case in which a symbol d_i is inserted. A correct decision is made when the $\tilde{Y}_\Delta(d_i)$ variable belongs to the Δ_i interval. On the contrarily, an error occurs when the attacks act in such a way that the $\tilde{Y}_\Delta(d_i)$ variable belongs to a Δ_j, with $i \neq j$; $i, j \in \{0, 1, \dots, m-1\}$.

Table II-2 illustrates the correct/erred decisions for $m = 5$: the rows correspond to the inserted symbols, while the columns to the decision; C stands for a correct decision while E stands for an erred decision.

Table II-2: Detection matrix.

inserted symbol \ detected symbol	-2	-1	0	1	2
-2	C	E	E	E	E
-1	E	C	E	E	E
0	E	E	C	E	E
1	E	E	E	C	E
2	E	E	E	E	C

The probability of the correct decision when inserting the d_i symbol is denoted by $P_c(d_i)$ and can be computed as follows:

$$P_c(d_i) = \int_{\Delta_i} p_{\tilde{Y}_\Delta}(y/d_i)\, dy \quad \text{(II-24)}$$

The average probability of a correct decision is denoted by P_c and can be computed by averaging the $P_c(d_i)$ values:

$$P_c = \sum_{d_i} P(d_i)\, P_c(d_i) \quad \text{(II-25)}$$

The optimal detection rule should ensure the maximal P_c value over all possible Δ_i partitions of the $[-\Delta/2, \Delta/2]$ interval.

For watermarking applications, the inserted symbols are equally likely: $P(d_i) = P(d_j)$, $i, j \in \{0,1,\dots,m-1\}$. Hence, at the detection side, the probabilities of correct detection of the inserted symbols should also be equal:

$$P_c(d_i) = P_c(d_j),\ i, j \in \{0,1,\dots,m-1\} \quad \text{(II-26)}$$

Equations (II-24) and (II-26) yield to:

$$\int_{\Delta_i} p_{\tilde{Y}_\Delta}(y/d_i)\,dy = \int_{\Delta_j} p_{\tilde{Y}_\Delta}(y/d_j)\,dy \tag{II-27}$$

where $i, j \in \{0,1,\dots,m-1\}$.

Equation (II-27) demonstrates that the optimal decision rule should ensure at the same time maximal and equal $\int_{\Delta_i} p_{\tilde{Y}_\Delta}(y/d_i)\,dy$ values over all possible Δ_i partitions.

When coming now back to the expression of $p_{\tilde{Y}_\Delta}(y/d)$, see equation (II-22), it can be stated that:

1) equal values for the left and right side expressions in (II-27) can be obtained when the the Δ_i intervals have the same length (*i.e.* a fifth of the Δ value);
2) for a given length of a decision interval Δ_i, $i \in \{0,1,\dots,m-1\}$, a maximal value for $\int_{\Delta_i} p_{\tilde{Y}_\Delta}(y/d_i)\,dy$ is obtained when $p_{\tilde{Y}_\Delta}(y/d_i)$ is centred within the Δ_i interval.

The two observations above demonstrate that the decision rule in (II-20) is an optimal decision rule, in the sense of maximizing the probability of correct decision, hence of minimizing the probability of error.

II.1.3.3. Computing the probability of error

Assume now the case in which a d_i symbol is inserted and a wrong decision d_j ($d_j \neq d_i$) is made. The probability of such an error, denoted by $P_e(d_i, d_j)$, can be computed as:

$$P_e(d_i, d_j) = P(d_i) \int_{I_{\alpha^*}(d_j)} p_{\tilde{Y}_\Delta}(y/d_i)\,dy$$

The average error probability, when considering all the possible symbols to be inserted and all the possible errors in detection can be computed as the sum of the individual error probabilities:

$$P_e = \sum_{\substack{d_i, d_j \\ d_i \neq d_j}} P_e(d_i, d_j), i, j \in \{0,1,\dots,m-1\} \tag{II-28}$$

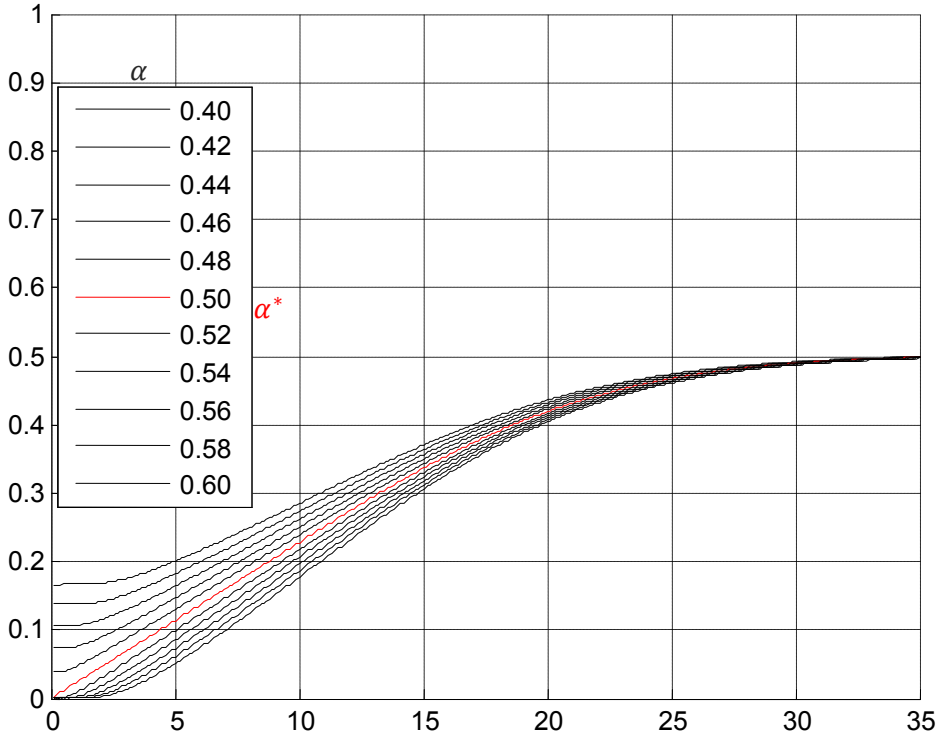
The error probability expressed by equation (II-28) corresponds to the error in detecting a symbol d from an m-ary alphabet. Should we be interested in the error of detecting a symbol from the initial binary message b, the general conversion formula can be applied [SPA87]:

$$P_{e_b} = \frac{1}{2}\frac{m}{m-1} P_e \tag{II-29}$$

Figures II-5 to II-8 illustrate the average probability error expressed by equation (II-28) as a function of the Gaussian noise standard deviation σ (presented on the abscissa), for $\Delta = 70$. Four alphabet sizes have been considered, namely $m = 2$, $m = 3$, $m = 5$ and $m = 7$. In each case, 11 values for the α parameter are illustrated; these values are evenly distributed with a step 0.02 and are centered on the corresponding α^* value; the case of α^* is plotted in red.

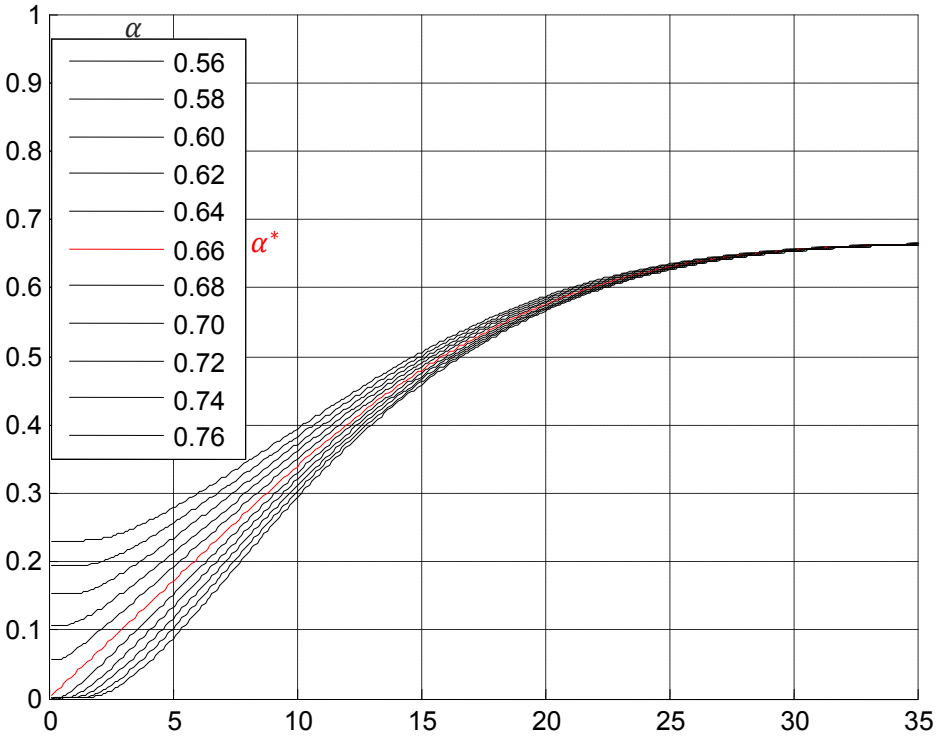
By analyzing these plots, the following conclusions are brought to light:

- when $\alpha < \alpha^*$, $P_e > 0$ even in the absence of attacks (*i.e.* even when $\sigma = 0$); this is a consequence of the overlapping between the decision intervals, see Figures II-5 to II-8;
- when $\alpha = \alpha^*$ and in the absence of attacks (*i.e.* $\sigma = 0$), $P_e = 0$; this result derives from the fact that in the absence of attacks, the decision rule in (II-19) is deterministic and $\alpha = \alpha^*$ ensures the best robustness–transparency trade-off;
- if $\alpha = \alpha^*$ and $\sigma > 0$, then $P_e > 0$; consequently, the α^* value ensuring the best transparency cannot be exploited in practical applications: in order to grant some robustness to the attacks, an $\alpha > \alpha^*$ should be considered, thus impacting in the transparency;
- if the practical application requires a value $P_e < 0.1$ for $\sigma < \Delta/(4m)$ (*i.e.* for a Gaussian noise covering at 95% a particular decision interval $I_{\alpha^*}(d)$, see Figure II-9), then values $\alpha \geq \alpha^* + 0.04$ should be considered. This lower limit $\alpha = \alpha^* + 0.04$ will be further considered for the theoretical investigation of the P_e variation as a function of Δ (see Figure II-10 to II-13) and for the experimental validations in Chapter II.2 and II.3;
- all the P_e plots converge towards $(m-1)/m$, when $\sigma \to \Delta/2$ (*i.e.* in the case of a very strong Gaussian noise, covering practically all the $\left[-\frac{\Delta}{2}; \frac{\Delta}{2}\right]$ interval), see Figure II-9.



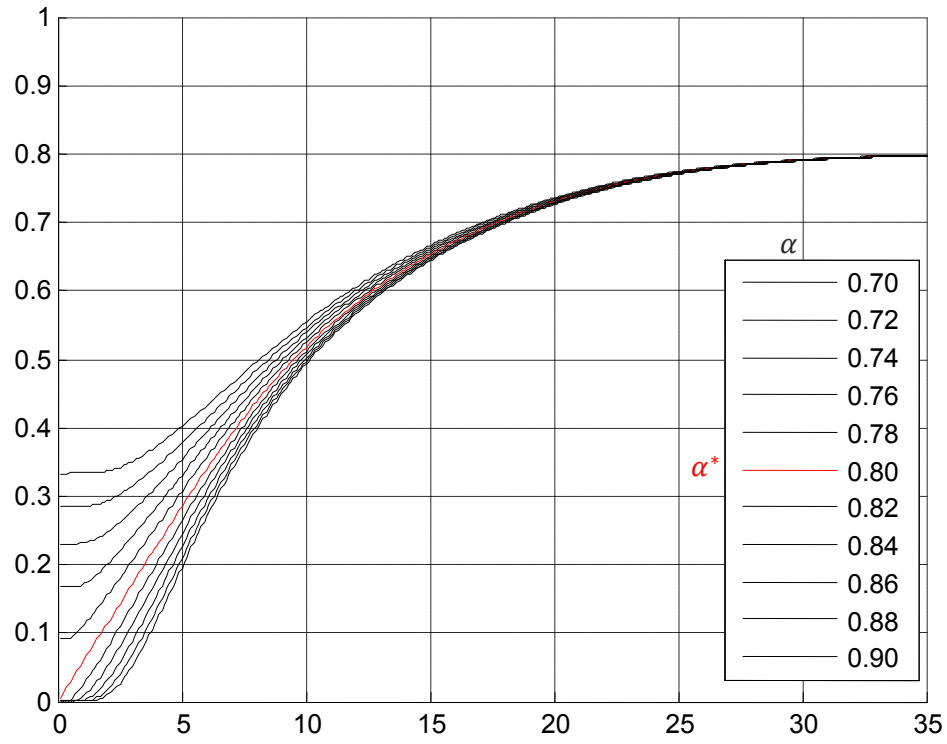
$m = 2$

Figure II-5: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 2}$ and for a fixed value $\Delta = \mathbf{70}$.



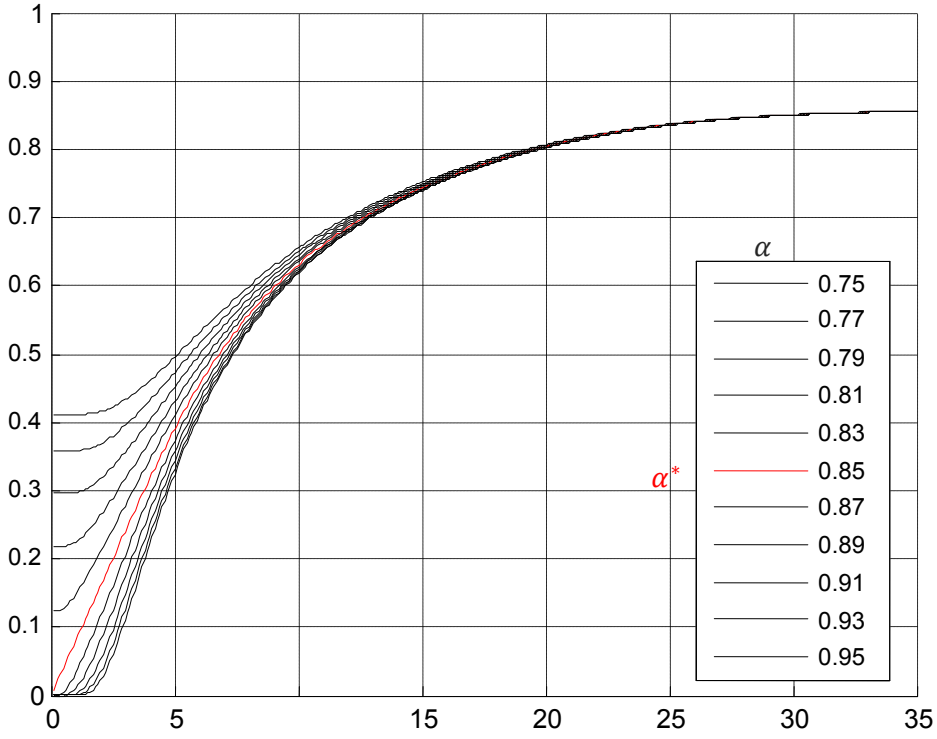
$m = 3$

Figure II-6: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 3}$ and for a fixed value $\Delta = \mathbf{70}$.



$m = 5$

Figure II- 7: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 5}$ and for a fixed value $\Delta = \mathbf{70}$.



$m = 7$

Figure II-8: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 7}$ and for a fixed value $\Delta = \mathbf{70}$.

The experiments reported in Figure II-5 to II-8 are resumed in Figure C-1 to C-4 and Figure C-5 to C-8 (*cf.* Appendix C) for for two different Δ values, namely $\Delta = 50$ and $\Delta = 90$, respectively.

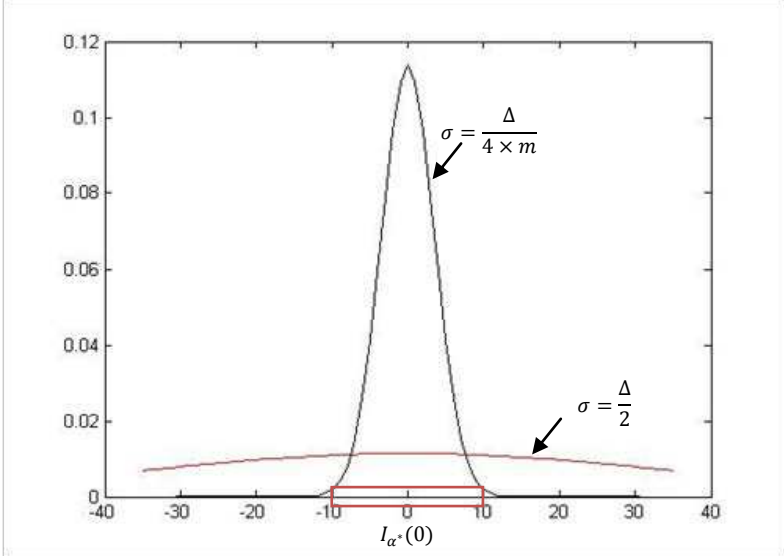
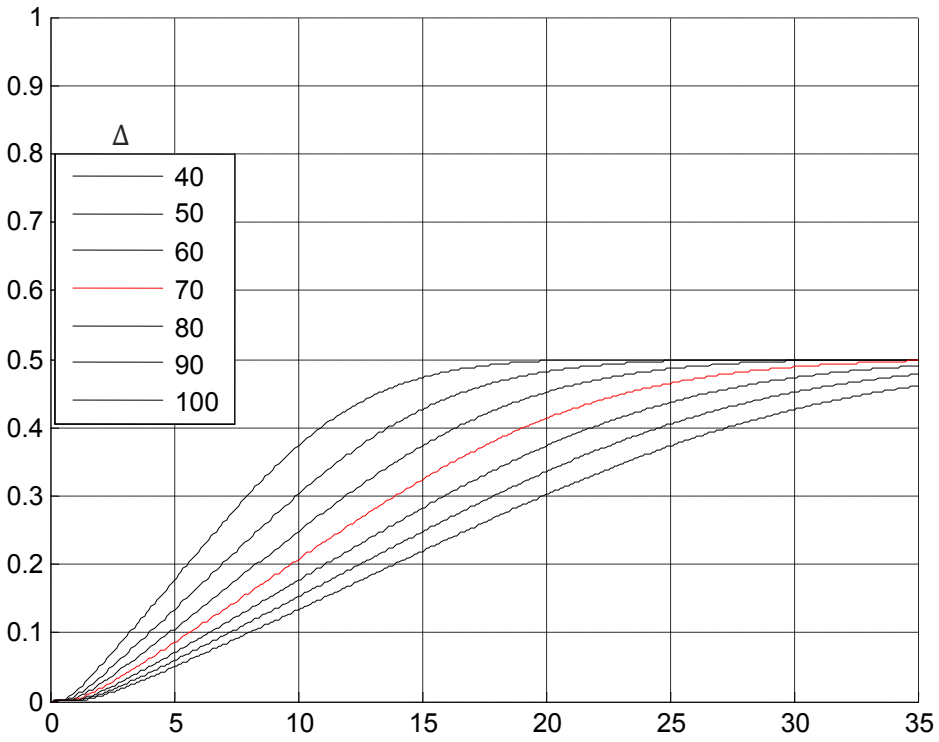


Figure II-9: Gaussian noise for $\boldsymbol{\sigma = \frac{\Delta}{4 \times m}}$ *(i.e. a noise covering only one decision interval) and for* $\boldsymbol{\sigma = \frac{\Delta}{2}}$ *(i.e. a very strong noise, covering all the* $\left[-\frac{\Delta}{2}; \frac{\Delta}{2}\right]$ *interval).*

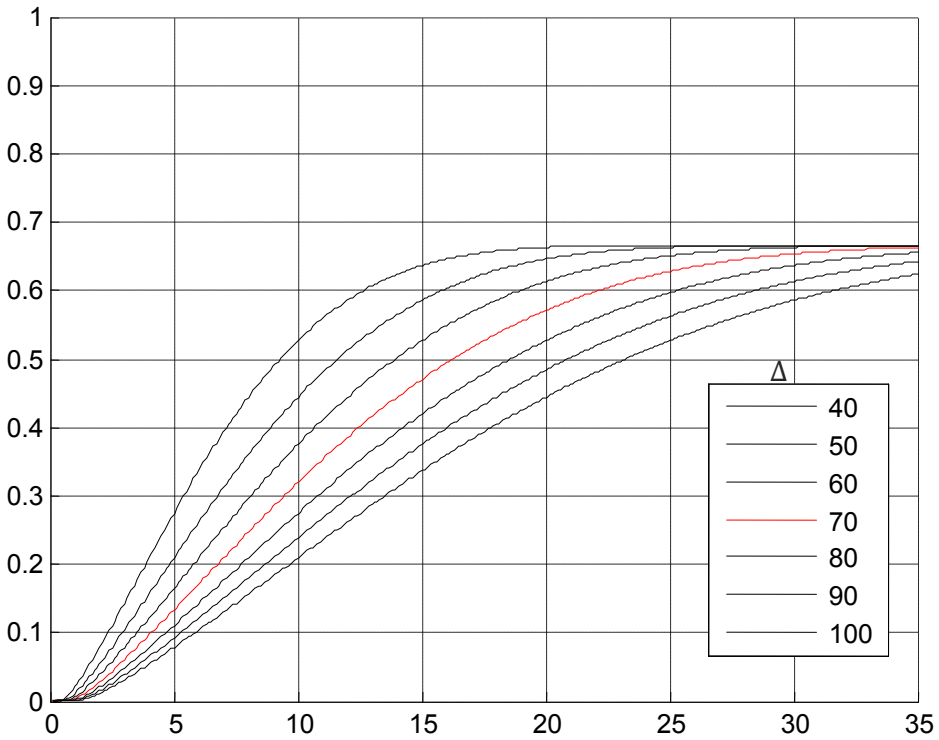
Figures II-10 to II-13 illustrates the average probability error expressed by equation (II-28) as a function of the Gaussian noise standard deviation σ (represented on the abscissa), for $\alpha = \alpha^* + 0.04$; the same four values for m have been investigated: $m = 2$, $m = 3$, $m = 5$ and $m = 7$. In each case, seven Δ values are illustrated, namely $\Delta = 40$, $\Delta = 50, \Delta = 60, \Delta = 70, \Delta = 80$, $\Delta = 90$, and $\Delta = 100$. By analyzing these plots, we can see that P_e is a decreasing function of Δ, at a fixed value of σ. Hence, the increase of Δ means *a priori* a stronger robustness, obtained at the expense of a weaker transparency. In the sequel, we shall consider Δ values between 50 and 90 and we shall discuss their practical relevance.

The experiments reported in Figure II-9 to II-12 are resumed in Figure C-9 to C-12 and Figure C-13 to C-16 (*cf.* Appendix C) for two different α values, namely $\alpha = \alpha^* - 0.04$ and $\alpha = \alpha^*$, respectively.



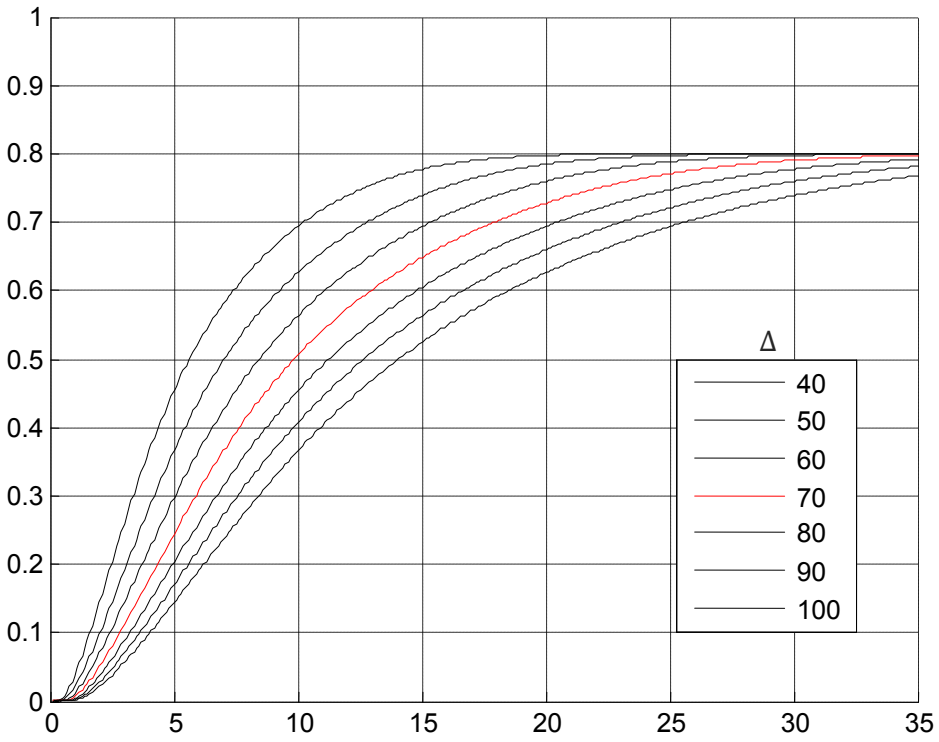
$m = 2, \alpha = \alpha^* + 0.04 = 0.54$

Figure II-10: P_e as a function of σ, for $\Delta \in \{40, 50, 60, 70, 80, 90, 100\}$, $m = 2$ and $\alpha = \alpha^* + 0.04$.



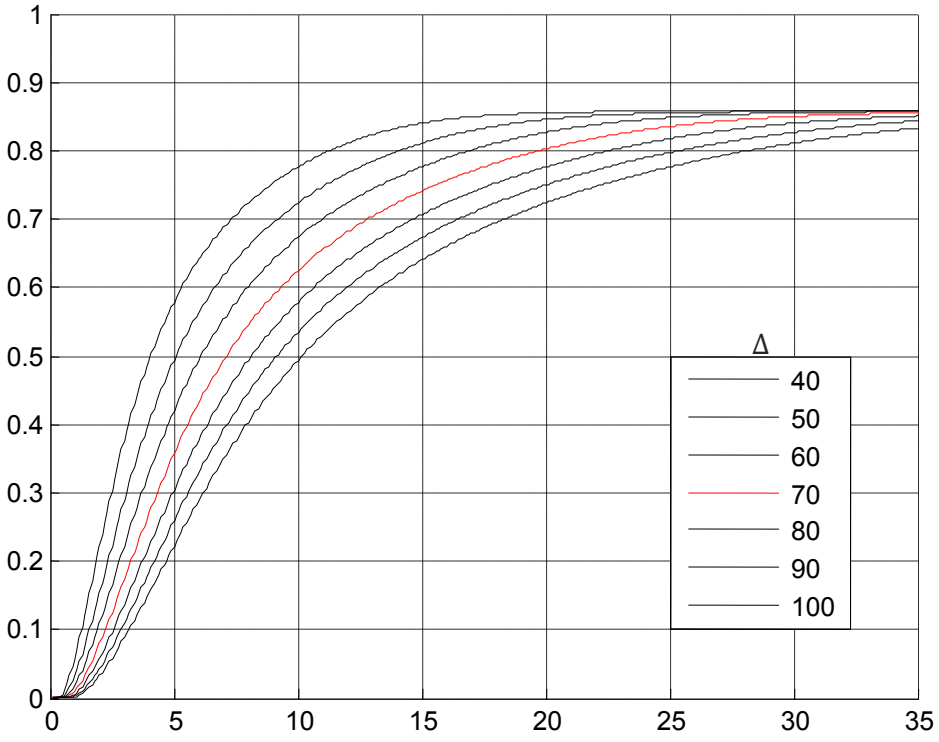
$m = 3, \alpha = \alpha^* + 0.04 = 0.7$

Figure II-11: P_e as a function of σ, for $\Delta \in \{40, 50, 60, 70, 80, 90, 100\}$, $m = 3$ and $\alpha = \alpha^* + 0.04$.



$m = 5, \alpha = \alpha^* + 0.04 = 0.84$

Figure II-12: P_e as a function of σ, for $\Delta \in \{40, 50, 60, 70, 80, 90, 100\}$, $m = 5$ and $\alpha = \alpha^* + 0.04$.



$m = 7, \alpha = \alpha^* + 0.04 = 0.9$

Figure II-13: P_e as a function of σ, for $\Delta \in \{40, 50, 60, 70, 80, 90, 100\}$, $m = 7$ and $\alpha = \alpha^* + 0.04$.

II.1.3.4. Computing the probability of error at constant distortion

While searching for the optimal parameters configuration and/or detection rule under the m-QIM framework, the previous sections considered as optimization criteria (1) the tradeoff between transparency and robustness and (2) the minimum probability of error.

Such an approach considers that the Δ parameter is fixed, that a single x MPEG-4 AVC stream element is watermarked and the α^* value was computed. Moreover, it was implicitly assumed that the original MPEG-4 AVC stream stands for an infinite insertion space.

The present section considers an application-oriented approach: the condition of fixed distortion is imposed as constraint, the α parameter is fixed and the relationship between Δ, m and the number of the stream syntax element which can be watermarked is investigated.

Relationship between Δ and m

Be there a binary message b of size N_2. This message is embedded by a binary modulation into a vector $X = [x_1, x_2, \dots, x_{N_2}]$ composed by N_2 elements is constructed from the MPEG-4 AVC stream. The obtained watermarked vector is denoted by $Y = [y_1, y_2, \dots, y_{N_2}]$.

The same message b can alternatively encoded into an m-airy massage S, prior to its insertion. The multi-symbol message size N_m is obtained by (II-30).

$$N_m = \frac{N_2}{log_2(m)} \tag{II-30}$$

Hence, the m-airy message S will be inserted into a vector $X = [x_1, x_2, \dots, x_{N_m}]$. The watermarked signal vector $Y = [y_1, y_2, \dots, y_{N_m}]$ is obtained according to the following equation (II-7):

$$Y = X + \alpha Q$$

where $Q = [q_1, q_2, \dots, q_{N_m}]$ is quantization error vector by Δ quantization step.

In the sequel, the watermarking distortion between will be expressed by the difference of the quadratic average errors of the Y and X vectors (*i.e.* by the energy of the watermarked artifacts):

$$D_m = \overline{Y^2} - \overline{X^2}$$

From the statistical point of view, the X, Q and Y components can be considered as following the $i.i.d$ (independent and identically distributed) model (*i.e.* inside each X, Q and Y vector, the component are independent and identically distributed). Be there $\boldsymbol{y}$, $\boldsymbol{x}$ and $\boldsymbol{q}$ the random variables modeling the values taken by x_i, q_i and y_i components (note that a bold typeset was used for random variables)

Hence,

$$D_m = \sum_{i=1}^{N_m}\left(\overline{y_i^2} - \overline{x_i^2}\right) \quad = N_m(\overline{\boldsymbol{y}^2} - \overline{\boldsymbol{x}^2}) \tag{II-31}$$

According to (II-7)

$$\overline{\boldsymbol{y}^2} = \overline{\boldsymbol{x}^2} + 2\alpha\overline{\boldsymbol{x}\boldsymbol{q}} + \alpha^2\overline{\boldsymbol{q}^2} \tag{II-32}$$

On the one hand, as $\boldsymbol{x}$ represents the syntax element values, it is considered as being a zero-mean *pdf* (probability distribution function). On the other hand, as $\boldsymbol{q}$ represents a quantization error by step Δ, its *pdf* is uniform of zero-mean and $\frac{\Delta^2}{12}$ quadratic average value.

Consequently,

$$\overline{\boldsymbol{y}^2} = \overline{\boldsymbol{x}^2} + \alpha^2\frac{\Delta^2}{12} \tag{II-33}$$

Hence, the watermarking distortion (II-31) can be obtained according to (II-34):

$$D_m = N_m\alpha^2\frac{\Delta^2}{12} = \frac{N_2}{log_2(m)}\alpha^2\frac{\Delta^2}{12} \tag{II-34}$$

For a fixed α value, in order to obtain the same distortion in the m-ary and binary case (*i.e.* $D_m = D_2$), the Δ and m relationship should jointly fulfill the following equation (II-35):

$$D_m = D_2 \Rightarrow \Delta_m = \Delta_2\sqrt{log_2 m} \tag{II-35}$$

Probability of error for constrained m and Δ_m

Assume we are interested in benchmarking the performances of m-QIM system for different m values, under the constraint of equal distortion. In this respect, the probability of error Pe_b expressed by (II-29) should computed by considering m and Δ_m values connected through (II-35); be Peb_m the binary probability error considering the adapted value Δ_m:

$$Peb_m = Pe_b(\Delta_m) \tag{II-36}$$

In order to investigate the Peb_m behavior, a Gaussian noise modeling the attacks of standard deviation σ is considered; the same four values m have been investigated $m = 2$, $m = 3$, $m = 5$ and $m = 7$; the α value is set to $\alpha = 0.9$, which corresponds to $\alpha = \alpha^* + 0.04$ for $m = 7$ (hence, in all cases, the non overlapping condition of decision intervals is satisfied); the Δ_2 value is set to $\Delta_2 = 30$; hence, $\Delta_3 = 47$, $\Delta_5 = 69$ and $\Delta_7 = 84$.

Figure II-14 illustrates the average binary probability error Peb_m expressed by equation (II-36) as a function of the Gaussian noise standard deviation σ (represented on the abscissa),

By analyzing this plot, the following conclusions can be drawn:

- if $\sigma < \frac{\Delta_2}{4}$ the probability of error obtained in the binary case is lower than the probability error value obtained in all m-ary case ($m > 2$).
- if $\sigma \geq \frac{\Delta_2}{4}$ multi symbol modulation can show lower average binary error probability than the binary case.

These results confirm the state of the art, see Table II-1 and the discussion in Section II-1: the *m*-QIM practical impact depends on the actual applicative conditions.

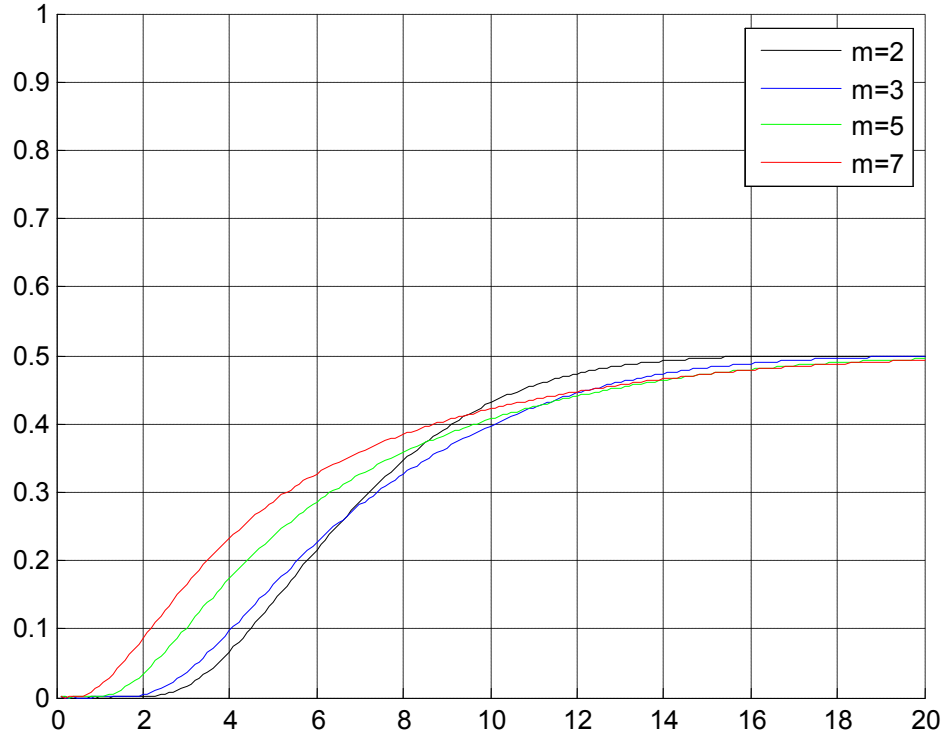


Figure II- 14: Peb_m as a function of σ, for 4 values of m, for $\Delta_2 = 30$ and fixed $\alpha = 0.9$.

Probability of error for constrained m and Δ_m and for limited insertion space

The present thesis deals with the practical case of compressed domain watermarking, where the insertion space is hence *a priori* limited: from the conceptual point of view, the better the compression efficiency, the smaller the insertion space. In other words, in practice it is expected that the maximum size of the x and y vectors would be limited at a value N_w representing the maximum number of syntax elements which can be watermarked.

While the case $N_w \geq N_m$ is already discussed above (Figure II-14), let's consider now the case in which $N_m \leq N_w$ (that is, the case in which there is no enough room in the host signal for inserting the mark for a particular m value).

In such a case, when detecting the mark, N_w symbols are recovered according to the *m*-QIM principles; hence, their binary probability of error can be computed according to (II-36). The rest of $N_m - N_w$ are randomly detected; hence, their binary probability of error is 0.5. Consequently, when $N_m \leq N_w$, the binary probability of error can be computed as a weighted average between Peb_m in (II.36) and 0.5; be this probability of error under joint constraints of fixed distortion and limited insertion space denoted by Peb_{ml}:

$$Peb_{ml} = \frac{N_w}{N_m} Peb_m + \frac{N_m - N_w}{N_m} \frac{1}{2} \qquad \text{(II-37)}$$

Then:

$$Peb_{ml} = log_2(m)\frac{N_w}{N_2}Peb_m + \frac{1}{2}\left(1 - log_2(m)\frac{N_w}{N_2}\right) \quad \text{(II-38)}$$

Let $\beta = \frac{N_w}{N_2}$ (*i.e.* β quantifies the available insertion space with respect to the insertion space required for a binary insertion).

Then:

$$Peb_{ml} = log_2(m)\beta Peb_m + (1 - log_2(m)\beta)\frac{1}{2} \quad \text{(II-39)}$$

We shall investigate Peb_f as function of β, see Figure II-15 to (II-19).

The same experiment conditions as in Figure II-14 are kept: the same four values m namely $m = 2$, $m = 3$, $m = 5$ and $m = 7$; the α value is set to $\alpha = 0.9$; $\Delta_2 = 30$.

Figures II-15 to II-19 illustrate the average binary probability error Peb_{ml} expressed by equation (II-39) as a function of the Gaussian noise standard deviation σ (represented on the abscissa). Four β values have been considered, namely $\beta = 0.9$, $\beta = 0.8$, $\beta = 0.7$ and $\beta = 0.6$.

By analyzing these plots, we can notice that for $\beta < 1$, multi symbol methods feature lower average binary error probability than the binary case and consequently outperforms it in term of robustness.

This behaviour can be explained by the fact that the embedding space offered by the watermarking channel does not fit the binary information size. Consequently, reducing the required insertion space by considering higher alphabet size may feature lower probability error, at least for such a parameter configuration.

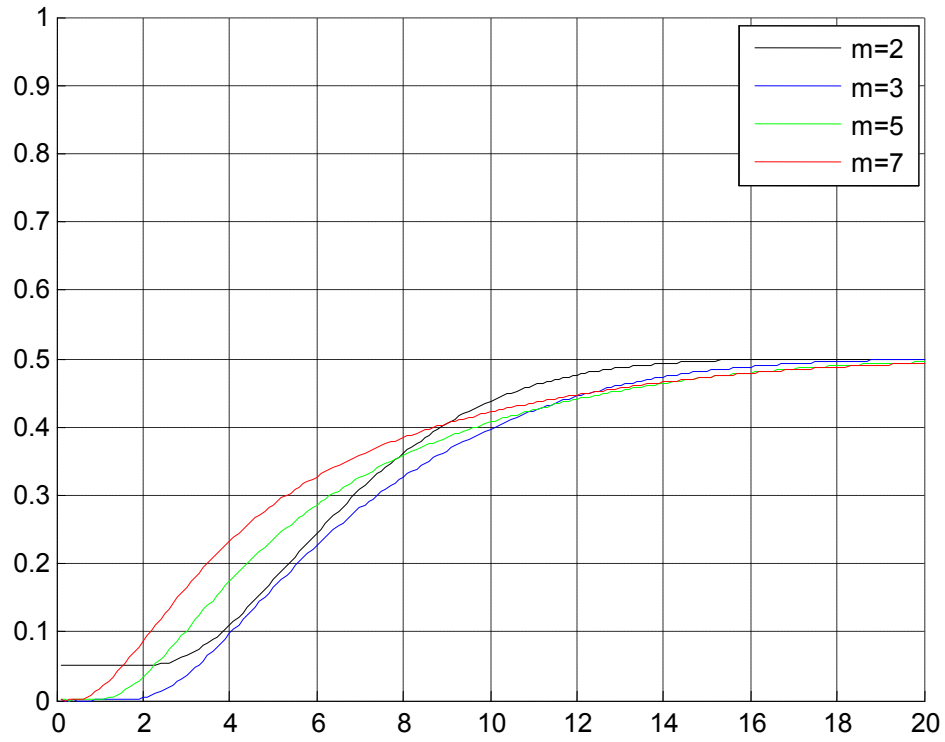


Figure II- 15: Peb_{ml} as a function of σ, for 4 values of m, for $\Delta_2 = 30$, $\alpha = 0.9$ and $\beta = 0.9$.

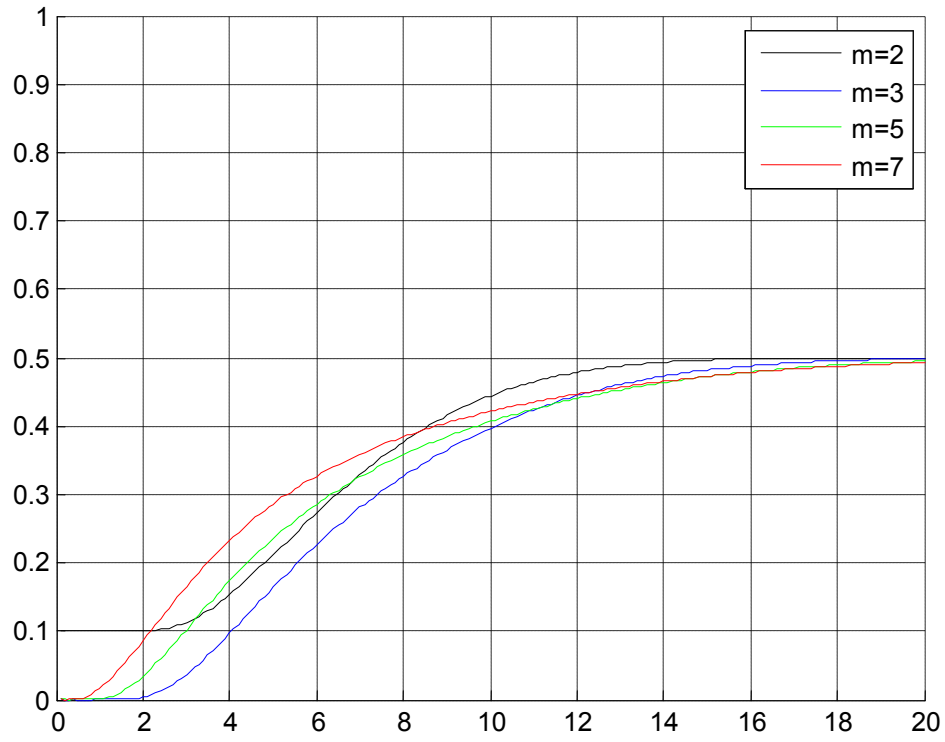


Figure II- 16: Peb_{ml} as a function of σ, for 4 values of m, for $\Delta_2 = 30$, $\alpha = 0.9$ and $\beta = 0.8$.

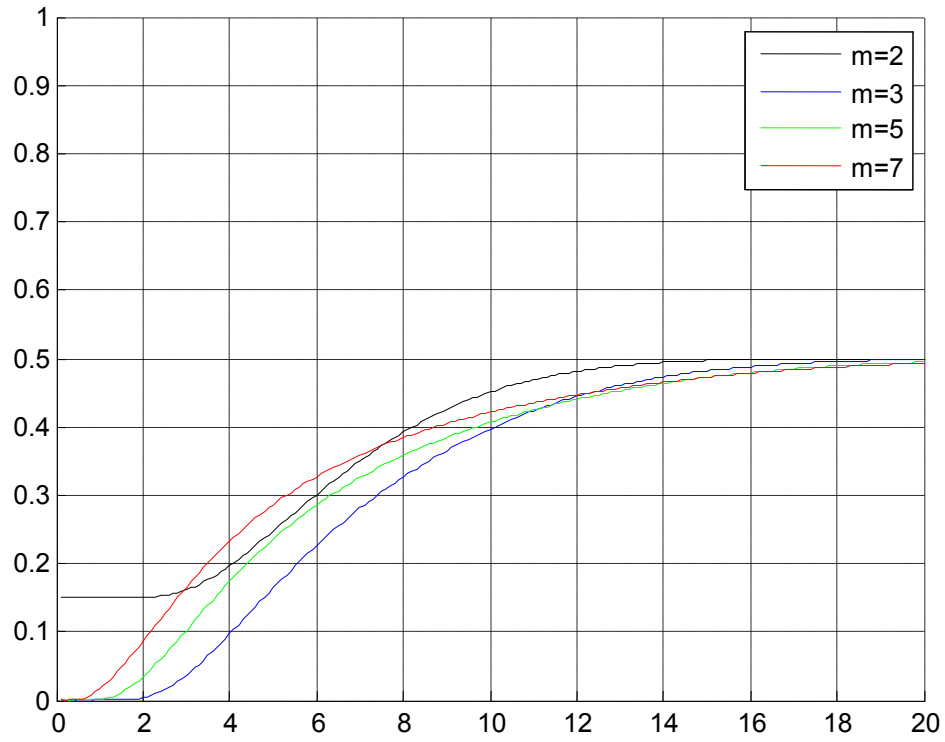


Figure II- 17 : Peb_{ml} as a function of σ, for 4 values of m, for $\Delta_2 = 30$, $\alpha = 0.9$ and $\beta = 0.7$.

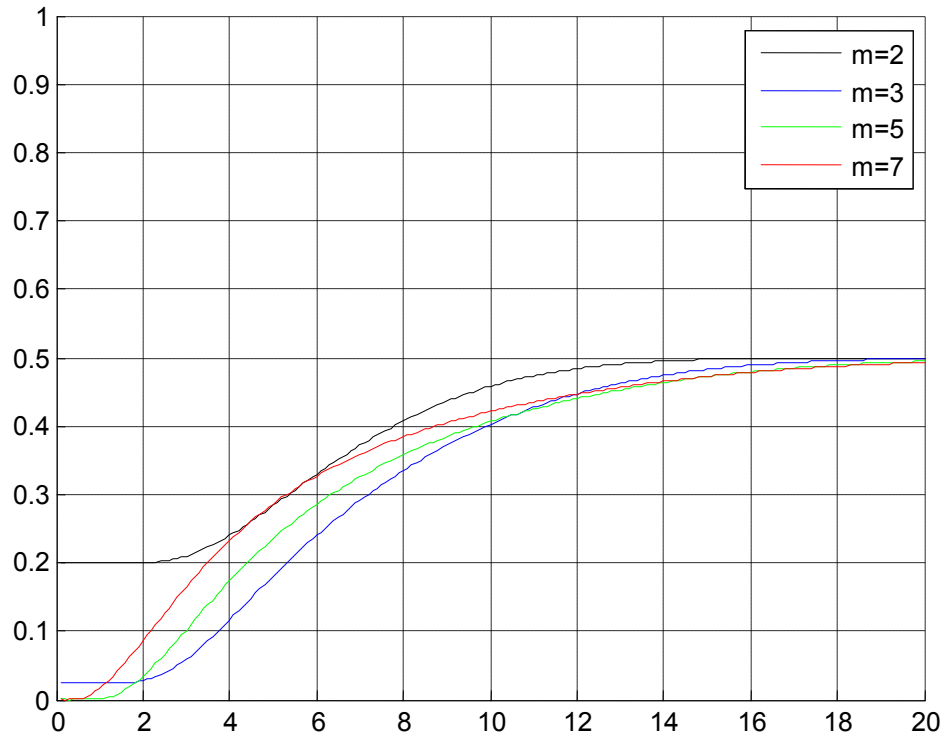


Figure II- 18 : Peb_{ml} as a function of σ, for 4 values of m, for $\Delta_2 = 30$, $\alpha = 0.9$ and $\beta = 0.6$.

II.1.4. Conclusion

This chapter presents the theoretical framework for the m-QIM watermarking. First, it generalizes the insertion rule, from the binary to m-ary case, see equation (II-7). Secondly, it derives the optimal decision rule in the sense of minimizing the error probability of detection under an additive white Gaussian noise (II-20) and computes the underlying probability error (II-28). These results also identify optimal (in the error probability sense) parameters for the method and/or hinted to more suitable configurations to be considered in the practical validation. The way in which these theoretical issues are reflected in the practical data payload-robustness-transparency tradeoff is investigated in next chapters devoted to robust and semi-fragile applications.

II.2. Case study 1: MPEG-4 AVC robust watermarking for ownership protection

This chapter demonstrates the m-QIM performances when integrated into an MPEG-4 AVC robust watermarking method.

II.2.1. Advanced method

The robust watermarking embedding process starts by following the principles detailed in Chapter II.1. The mark is inserted in the quantized AC luma residual coefficients; this way, the MPEG-4 AVC decoding/reencoding overhead is limited to the binary decoding/reencoding.

The host vector x is obtained by zig-zag scanning the 15 components of a 4×4 AC luma residual coefficients.

The watermark is not directly embedded into the original 4×4 block x but into the projection x' of x onto a perceptual mask, denoted by the v_{mask} vector.

The insertion/detection follows the equations (II-7) and (II-20).

II.2.1.1. Block selection

The mark is inserted after a block selection based on a criterion expressed by (II-40). Only in the blocks whose computed detection variables $Y(d)$ after the insertion of d verify the condition in (II-30) will be selected for the final mark embedding:

$$\left| Y(d) + d\frac{\Delta}{m} \right| < \frac{\Delta}{4m} \qquad \text{(II-40)}$$

Where $Y(d)$ represents the detection variable computed according to (II-8), d is the embedded symbol, m is the alphabet size and Δ is the quantization step.

The watermarked block positions are recorded to be used at the insertion and the detection procedure.

PSEUDOCODE FOR THE SELECTION
Begin video
If the current frame is *I* ***then***
Begin frame
1- Select randomly *n* nonzero blocks that are disjoints
If the current block is selected ***then***
- Mark the block according to (II-7).
- Compute the detection variable according to (II-8).
- Compute the condition (II-30)
- ***If*** the condition (II-30) is verified ***then*** the bloc is selected and its position is recorded.
Else go to the next block
End frame
Else go to the next frame
End video

II.2.1.2. Embedding process

The embedding process combines *m*-QIM and perceptual shaping. It is structured into three main modules as shown in Figure II-14: *perceptual shaping*, *mark generation* and *mark embedding.*

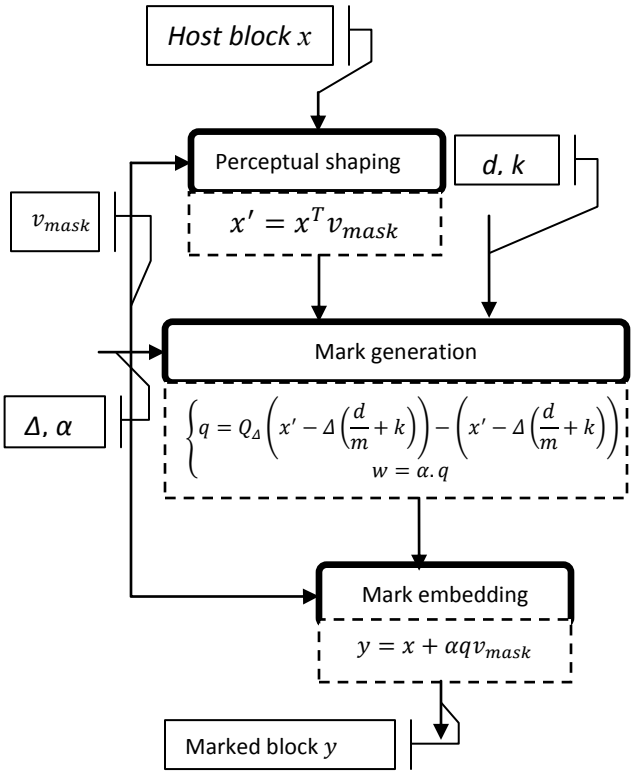


Figure II-19: The embedding synopsis: three inputs (the message d, the host $\boldsymbol{x}$ and the key $\boldsymbol{k}$) and three parameters (the perceptual mask $\boldsymbol{v_{mask}}$, the quantization step $\boldsymbol{\Delta}$ and the scaling factor $\boldsymbol{\alpha}$) are considered to compute the marked data y. $\boldsymbol{m}$ is the number of symbols in the message alphabet.

Perceptual shaping:

In order to match the watermarking method to the human visual system, a perceptual shaping mechanism is also considered. This means that the watermark is not directly embedded into the original vector x but into its projection onto a perceptual mask v_{mask}. This mask is obtained by adapting and extending [BEL10] for MPEG-4 AVC the popular Watson model [WAT93] developed for still images.

Mark generation:

The mark to be inserted into the host x vector depends of the m-ary message, *D* = {-(*m*-1)/2, - (*m*-2)/2,… 0, … (*m*-2)/2, (*m*-1)/2}, and of the original vector itself:

$$\begin{cases} q = Q_\Delta\left(x^T v_{mask} - \Delta\left(\frac{d}{m} + k\right)\right) - \left(x^T v_{mask} - \Delta\left(\frac{d}{m} + k\right)\right) \\ w = \alpha q \end{cases}$$

Embedding mark:

This module generates the watermarked 4×4 MPEG-4 AVC block. It has as input the mark w, the original content and the perceptual mask v_{mask}. In the present paper the insertion follows a simple additive rule as follows:

$$y = x + w\, v_{mask}$$

PSEUDOCODE FOR THE INSERTION
Begin video
If the current frame is *I* ***then***
Begin frame
If the current block is selected ***then***
for i=1 to 15 **do**
$\boldsymbol{x' = x' + CurrBck\_coeff[i]}$ // compute the projection
Compute the mark ***w*** according (II-7)
for i=1 to 15 **do**
$\boldsymbol{CurrBck\_coeff[i] = CurrBck\_coeff[i] + w\, v_{mask}[i]}$
Else go to the next block
End frame
Else go to the next frame
End video

II.2.1.3. Detection

For each supposed marked block, the detector starts by projecting the vector of 15 AC coefficients of the 4×4 sub-macroblocks in the *I* frames onto the perceptual mask v_{mask}. Then the detection is achieved by following the rule described in Chapter II.1.

PSEUDOCODE FOR THE DETECTION
Begin video
If the current frame is *I* ***then***
Begin frame
If the current block is selected ***then***
for i=1 to 15 **do**
$\boldsymbol{y}' = \boldsymbol{y}' + \boldsymbol{CurrBck\_coeff}[\boldsymbol{i}]$ // compute the projection
- Compute the decision value $Y(d)$ of the block according to (II-8).
- Detect the mark using the detection rule (II-20)
Else go to the next block
End frame
Else go to the next frame
End video

II.2.2. Functional evaluation

The *m*-QIM parameters are chosen so as to serve the practical purpose of the robust watermarking application and are guided by the theoretical demonstration in Chapter II.1. Four m values and three quantization steps Δ are compared all through the study, namely $m \in \{2, 3, 5, 7\}$ and $\Delta \in \{50, 70, 90\}$. Regardless the particular m and Δ values, we set $\alpha = \alpha^* + 0.04$ (see the discussion in Chapter II.1.3.3).

Experimental evaluations are carried under MEDIEVALS corpus (*cf.* Appendix B) consisting of 4 video sequences of 15 minutes each. They were encoded in MPEG-4 AVC Baseline Profile (no B frames, CAVLC entropy encoder) at 512 kb/s (SD) and 2Mbps (HD). The GOP size is set to 10.

The experiments are conducted at three successive levels, so as to investigate the three-folded data payload-robustness-transparency watermarking challenge.

First (Chapter II.2.2.1), the data payload is estimated for different *m* values while keeping fixed values for transparency (average PSNR of 45 dB and of 65 dB, for SD and HD respectively) and robustness (maximal BER of 0.1± 0.03 against bipolar additive noise, transcoding and geometric random bending attacks).

Secondly (Chapter II.2.2.2), the robustness against the same three types of attacks is investigated at fixed data payload (150 bits per minute) and transparency (PSNR of 45 dB and of 65 dB, for SD and HD respectively) constraints.

Finally (Chapter II.2.2.3), the transparency (expressed by five subjective metrics) is assessed at fixed data payload (150 bits per minute) and robustness (maximal BER of 0.1 ± 0.03 against additive noise, transcoding and geometric random bending attacks).

II.2.2.1. Data payload

As the *m*-QIM is intrinsically a side-information method [CHE98], [EGG03] the data payload depends on the particular video sequence and cannot be *a priori* predicted.

Consequently, our experiments consider for each value m an incremental approach: we start from the value reported in [BEL10] for $m = 2$ and SD (namely 56 bits/minute), and we gradually increase it up to the limit allowing prescribed average values for transparency and robustness. On the one hand, the average transparency is set at a PSNR = 45dB in the SD case and at a PSNR = 65 dB in the HD case (with minimal/maximal limits of 43/47 dB and 63/67 dB, respectively). On the other hand, for both the SD and the HD corpora, the robustness is set at an average BER of 0.1 (with minimal/maximal limits of 0.07/0.13) after noise addition, transcoding and geometric random bending attacks[2]. A $\Delta = 70$ has been considered. The values thus obtained for the data payload are plotted in Figure II-15 in blue diamonds.

However, with respect to the binary QIM, the *m*-QIM insertion is theoretically expected to increase the data payload by a factor of $log_2 m$ [PRO01]. Consequently, we consider the data payload value obtained for $m = 2$ and we also plot the corresponding theoretical logarithmic function, see the red squares in Figure II-15.

Figure II-15 shows a good concordance between the theoretical and the experimental plot, the relative average differences being lower than 0.02 in the SD case and lower than 0.06 in the HD case. These average differences are computed by averaging the relative differences obtained on the corresponding corpora, for all the considered m values.

[2] All through the manuscript the geometric attack is ensured by the Stirmark random bending [PET98]; the terms random geometric and Stirmark attacks will be alternatively employed.

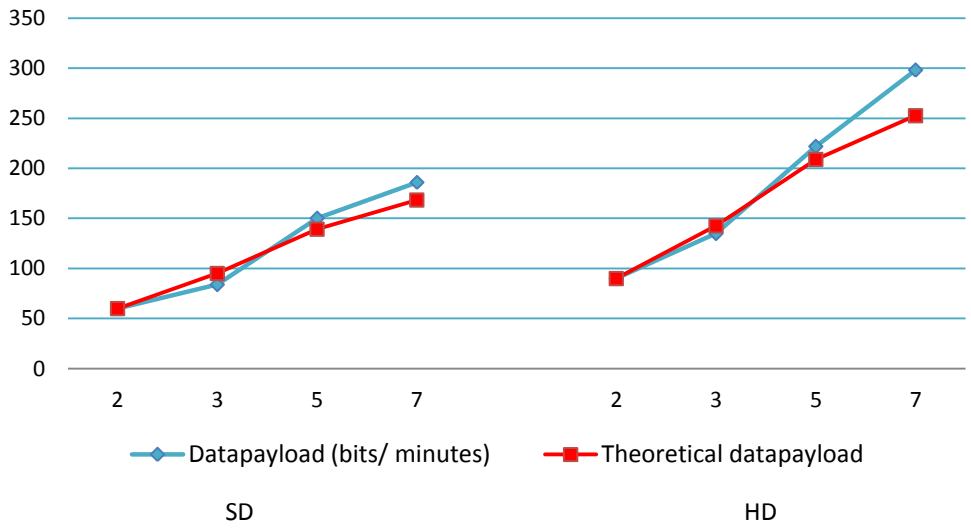


Figure II-20: Data payload as a function of m*, for* $\Delta = \mathbf{70}$ *and for SD (left) and HD (right) content. Fixed transparency and robustness constraints are kept.*

The experiments reported in Figure II-15 are resumed, in Table II-3 and Figure II-16, for $\alpha = \alpha^* + 0.04$, for $m = 5$ but for two different Δ values, namely $\Delta = 50$ and $\Delta = 70$. With respect to the reference case $\Delta = 70$, when keeping the same transparency/robustness constraints, it can be noticed that:

- $\Delta = 50$ ensures relative increases of the data payload by factor of 0.1 and 0.07 in the SD and HD cases, respectively.
- $\Delta = 90$ ensures relative decreases of the data payload by factor of 0.13 and 0.11 in the SD and HD cases, respectively.

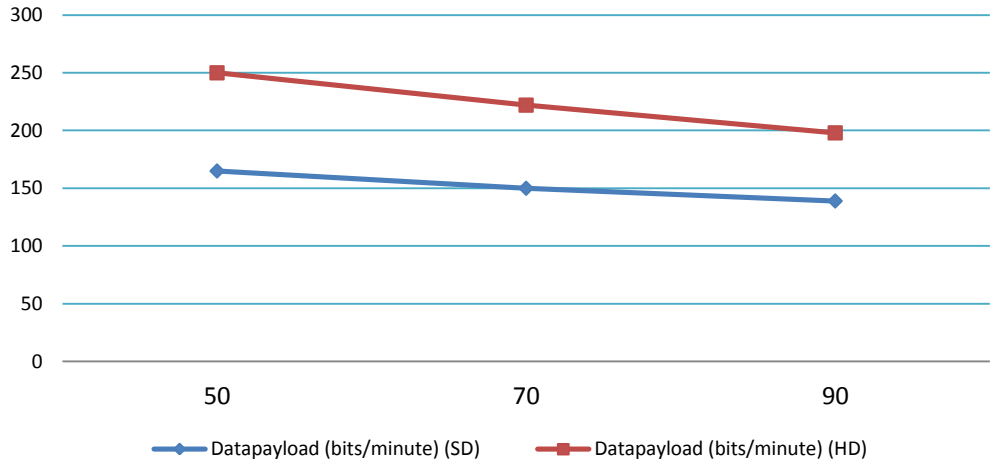


Figure II-21: Data payload as function of Δ*,* $\boldsymbol{m = 5}$.

Table II-3: Data payload behavior as function of Δ, the relative gain are computed according to (II-41).

	SD			HD		
Δ	50	**70**	90	50	**70**	90
Data payload (per minute)	165	**150**	139	225	**222**	198
Relative gain ρ	0.10	**0**	-0.13	0.07	**0**	-0.11

The relative gain ρ represented in the paragraph above is computed according to the formula:

$$\rho = \frac{value(\text{tested } \Delta) - value(\Delta = 70)}{value(\Delta = 70)} \quad \text{(II-41)}$$

II.2.2.2. Robustness

While the theoretical demonstration presented in Chapter II.1 considered the case of the attacks modelled by additive white Gaussian noise, the present section goes outside this hypothesis, by experimentally investigating attacks which cannot be modelled by the Gaussian laws [MIT07]: bipolar -1/1 white noise added in the 4×4 MPEG-4 AVC coefficient domain, MPEG-4 AVC transcoding (down to 25% from the original stream bit rate) and Stirmark geometric random bending attacks.

The data payload is set at 150 bits per minute while the transparency is set at 45dB (with $\pm 2\text{dB}$ maximal variation) in the SD case and at 65dB (with $\pm 2\text{dB}$ maximal variation) in the HD case. The Δ parameter is set at $\Delta = 70$. The same four values for m are investigated: $m \in \{2, 3, 5, 7\}$.

The robustness is investigated by computing the BER, see Figure II-17: the case of bipolar additive noise is represented in blue diamonds, the case of transcoding in red squares while the random geometric attacks in green triangles. The values reported in Figure II-17 are obtained by averaging the BER obtained at the frame level. The following conclusions can be drawn:

- the -1/1 bipolar noise addition results in perturbations described by a unitary standard deviation; as this value is very small compared to the Δ (see discussion in Chapter II.1.3), the choice of $\alpha = \alpha^* + 0.04$ ensured a BER = 0, although the law of such perturbation is not Gaussian;
- the MPEG-4 AVC transcoding resulted in average[3] BER of 0.05 and 0.06, for the SD and HD content, respectively; it can be noticed that BER is quite constant with respect to m, the corresponding variances (over the four investigated m values) being 2.2×10^{-4} and 1.7×10^{-4}, respectively;

[3] Averaged over the four considered m values.

- the standard deviation of the noise induced by the transcoding attacks is estimated in our experiments at 3.2 and 2.8, for SD and HD, respectively; the theoretical error obtained by equation (II-29) for these sigma values are 0.07 and 0.06; while a slight difference between the practice and the theory is identified in the SD case (0.05 vs. 0.07), the two values are identical in the HD case;
- the geometric random bending attacks also result in quite constant averaged BER as a function of m: average values of 0.105 and 0.090 and variances of 5.7×10^{-4} and 4.4×10^{-4} are obtained for the SD and HD content, respectively;
- the standard deviation of the noise corresponding to these attacks is estimated at 5.3 and 4.5, respectively; the theoretical average error computed with (II-29) are 0.12 and 0.11; we can see a lower concordance between the theoretical and experimental results than in the case of transcoding; this is the consequence of the fact that the geometrical attacks do not follow a Gaussian law;
- the numerical values reported in this section are statistical relevant, in the sense that the 95% confidence intervals [WAL02] computed for the BER after the transcoding and the random bending attacks resulted in relative errors lower than 5×10^{-3} (the BER in the bipolar noise addition was constantly equal to 0; hence, no confidence interval has been computed).

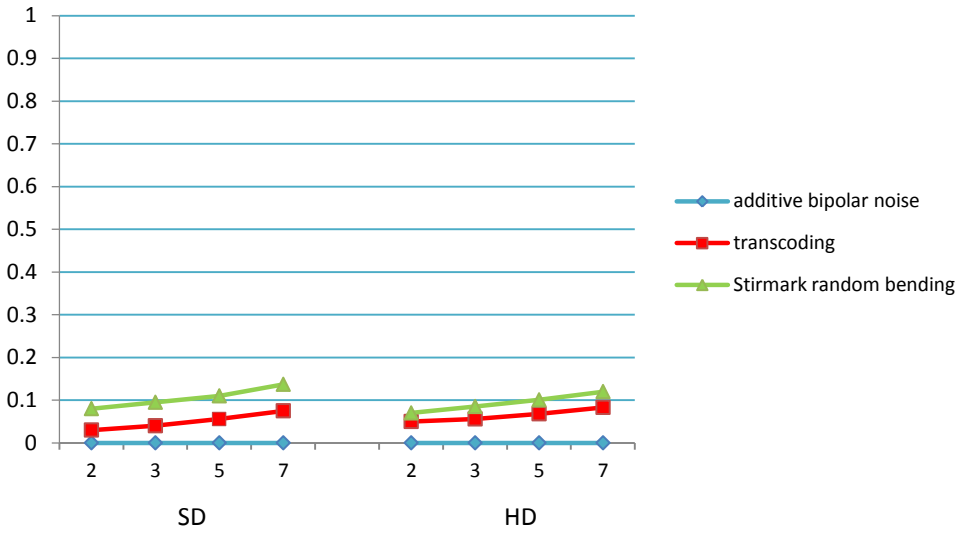


Figure II-22: BER as a function of m*, for SD (left) and HD (right) content. Fixed data payload and transparency performances are kept,* $\Delta = \mathbf{70}$.

The overall BER results prove the robustness of the *m*-QIM watermarking method against the investigated attacks. In order to illustrate the practical impact of these attacks for SD content, Figure II-18 presents the case in which the ARTEMIS logo is considered as the mark. The experiments in Figure II-18 considered the same context as above: a data payload of 150 bit/minute, $\alpha = \alpha^* + 0.04$, Δ =70.

	Original	Additive noise	Transcoding	Stirmark
$m = 2$				
$m = 3$				
$m = 5$				
$m = 7$				

Figure II-23: The original ARTEMIS (left) and reconstructed ARTEMIS after additive noise, transcoding and geometric random bending attacks, respectively.

The experiments reported in Figure II-17 are resumed in Table II-4 and Figure II-19 for the same $\alpha = \alpha^* + 0.04$, for $m = 5$ but for two different Δ values, namely $\Delta = 50$ and $\Delta = 90$. With respect to the case $\Delta = 70$, when keeping the same data payload/transparency constraints, it can be noticed that:

- $\Delta = 50$ ensures relative increases of the BER by a factor of 0.5 and 0.27 for transcoding and Stirmark random bending, respectively in the SD case and by a factor of 0.28 and 0.2 for transcoding and Stirmark random bending, respectively in the HD case.
- $\Delta = 90$ results in relative decreases of the BER by a factor of 0.33 and 0.09 for transcoding and Stirmark random bending, respectively in the SD case and by a factor of 0.28 and 0.1 for transcoding and Stirmark random bending, respectively in the HD case.

Note: the relative increases/decreases in the BER are computed according to (II-41).

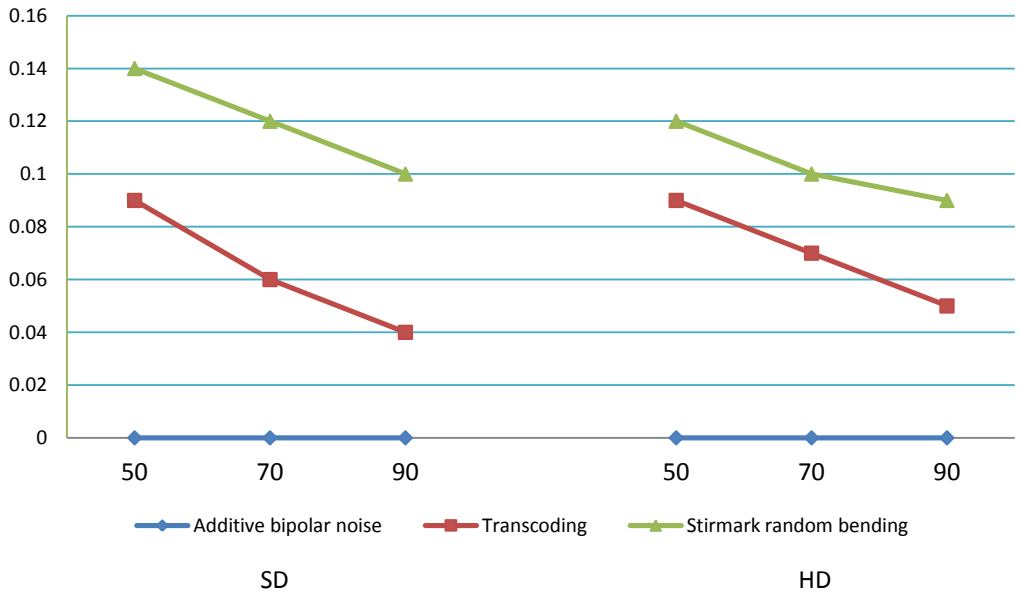


Figure II-24: Robustness as function of Δ., $\boldsymbol{m = 5}$.

Table II-4: Robustness behavior as function of Δ, $\boldsymbol{m = 5}$.

	SD			HD		
Δ	50	**70**	90	50	**70**	90
Transcoding (BER)	0.09	**0.06**	0.04	0.09	**0.07**	0.05
Relative gain $(\boldsymbol{\rho})$	0.50	**0**	-0.33	0.28	**0**	-0.28
Stirmark (BER)	0.14	**0.12**	0.10	0.12	**0.10**	0.09
Relative gain $(\boldsymbol{\rho})$	0.27	**0**	-0.09	0.2	**0**	-0.1

II.2.2.3. Transparency

In order to evaluate the transparency of the watermarking method, three types of metrics have been considered: pixel difference-based measures (peak signal to noise ratio – PSNR and absolute average difference – AAD), correlation based measures (structural continent – SC and normalized cross correlation - NCC), and psycho-visual measures (digital video quality - DVQ). These metrics are computed at the frame level according to their definitions presented in Chapter I.1.1, then averaged over the video corpus.

A fixed data payload of 150 bits per minute and robustness (expressed by average maximal BER of 0.1 ± 0.03 against the three above-mentioned attacks) are considered.

Figures II-20, II-21 and II-22 display the average values of these metrics alongside with their 95% confidence intervals computed according to [WAL02].

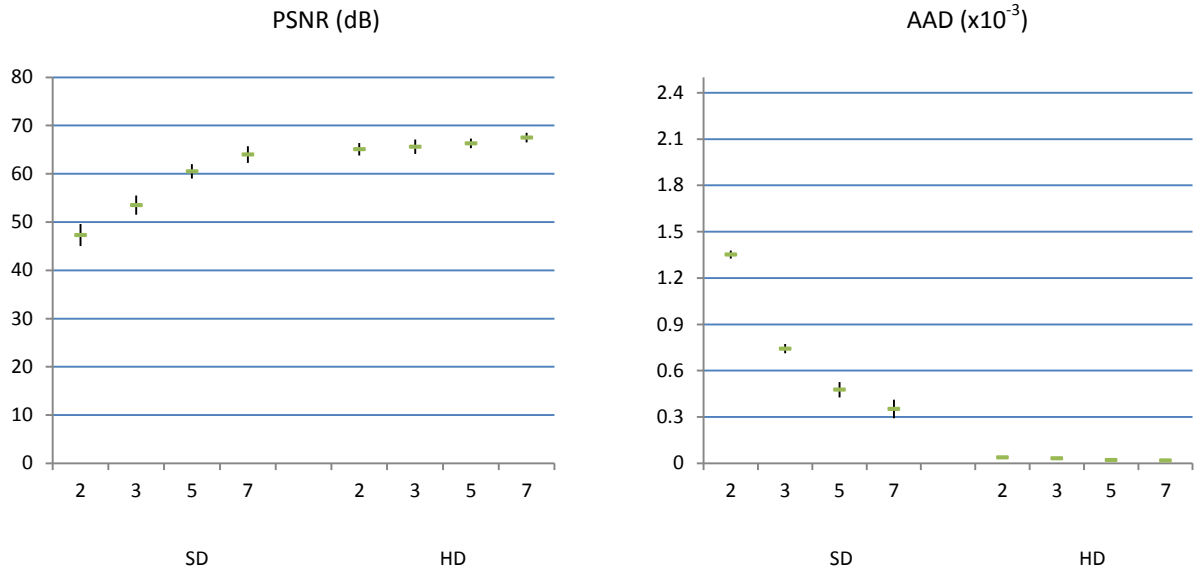


Figure II-25: PSNR and AAD as a function of $\boldsymbol{m}$ *(* $\Delta = \mathbf{70}$*): average values and 95% confidence limits.*

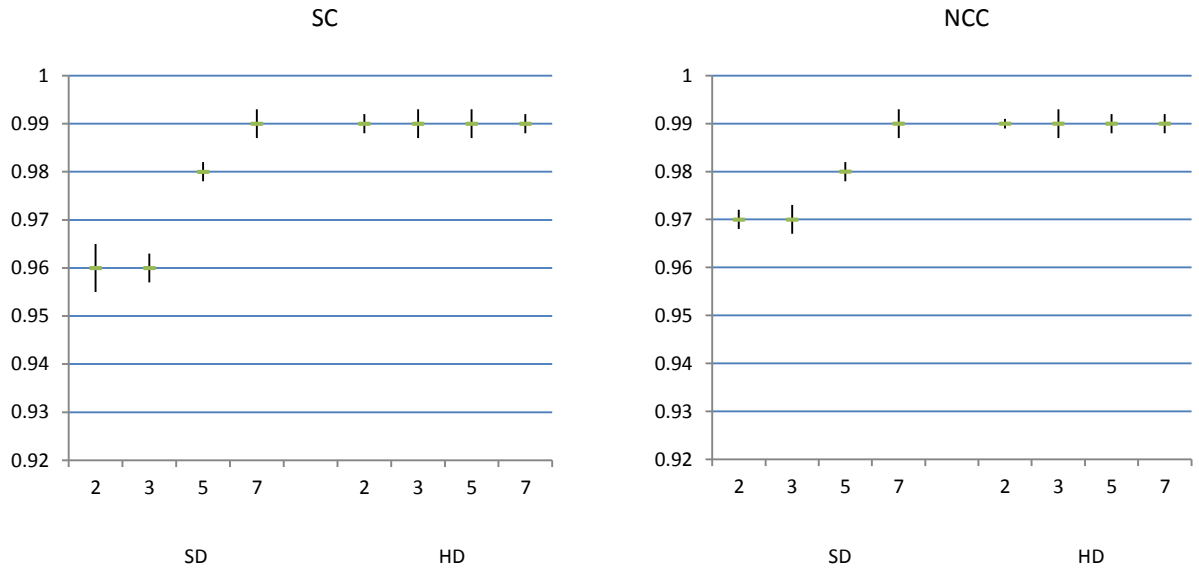


Figure II-26: SC and NCC as a function of $\boldsymbol{m}$ ($\Delta = \mathbf{70}$): average values and 95% confidence limits.

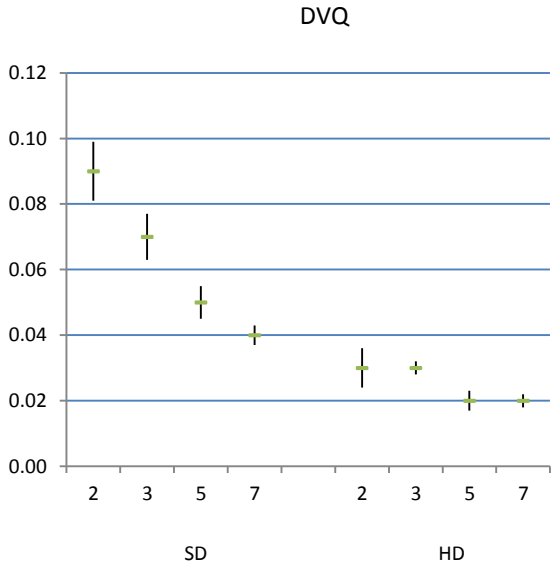


Figure II-27: DVQ as a function of $\boldsymbol{m}$ ($\Delta = \mathbf{70}$): average values and 95% confidence limits.

As an overall trend, we can see that the transparency improves with increasing the alphabet size m. This can be explained by the fact that, for a fixed data payload, increasing m decreases the number of MPEG-4 AVC blocks needed for embedding the mark and thus reduces the drift artefacts inner to compressed domain watermarking. However, an in-depth analysis of the different variations of the quality metrics, with respect to m, brings to light that the variation of the quality metrics is stronger in the case of SD content than in the case of HD content. In order to objectively asses this behavior, we consider the average relative variation between two successive m values, denoted by η:

$$\eta = \frac{1}{3}\sum_{i=1}^{3}\frac{metric(m_{i+1}) - metric(m_i)}{metric(m_i)} \quad \text{(II-42)}$$

where $metric \in$ {PSNR, AAD, SC, NCC and DVQ} and i is the index of the deployed m value belonging to the set $\{2, 3, 5, 7\}$. The η values are presented in Table II-5.

Table II-5: Variation of the quality metric with respect to $\boldsymbol{m}$.

	SD					HD				
	PSNR	**AAD**	**SC**	**NCC**	**DVQ**	**PSNR**	**AAD**	**SC**	**NCC**	**DVQ**
$\boldsymbol{\eta}$	0.1	-0.5	0.01	0.06	-0.23	0.012	-0.21	0	0	-0.16

The experiments reported in Figure II-20 to Figure II-22 are resumed in Table II-6 and Figure II-23 to Figure II-25 for the same $\alpha = \alpha^* + 0.04$, for $m = 5$ and for two different Δ values, namely $\Delta = 50$ and $\Delta = 70$. With respect to the case Δ =70, when keeping the same data payload/robustness constraints, we note that $\Delta = 50$ ensures:

- Relative increases by a factor of 0.06 and 0.04 for PSNR in the SD case and HD case, respectively.
- Relative decreases by a factor 0.2 and 0.33 for DVQ, in the SD case and in the HD case, respectively.

Similarly, with respect to the case $\Delta = 70$, when keeping the same data payload/robustness constraints, we note that $\Delta = 90$ ensures:

- Relative decreases by a factor of 0.07 and 0.06 for PSNR, in the SD case and in the HD case, respectively.
- Relative increases by a factor of 0.4 and 0.33 for DVQ in the SD case and in the HD case, respectively.

The relative gains are computed according to (II-41).

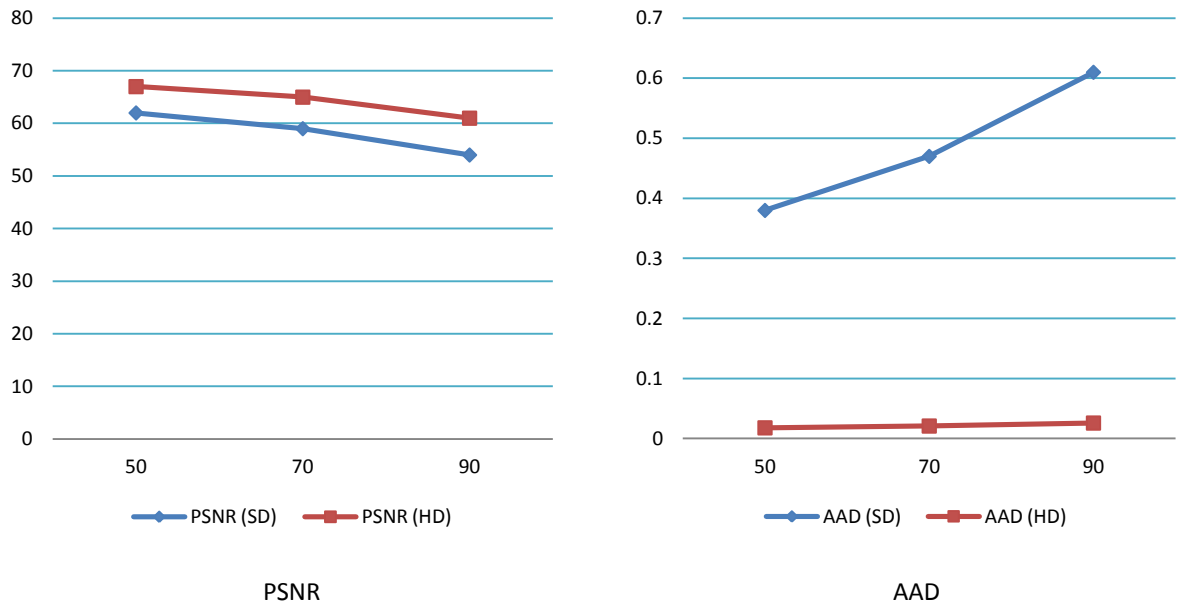


Figure II-28: PSNR and AAD as function of Δ, $\boldsymbol{m} = \mathbf{5}$.

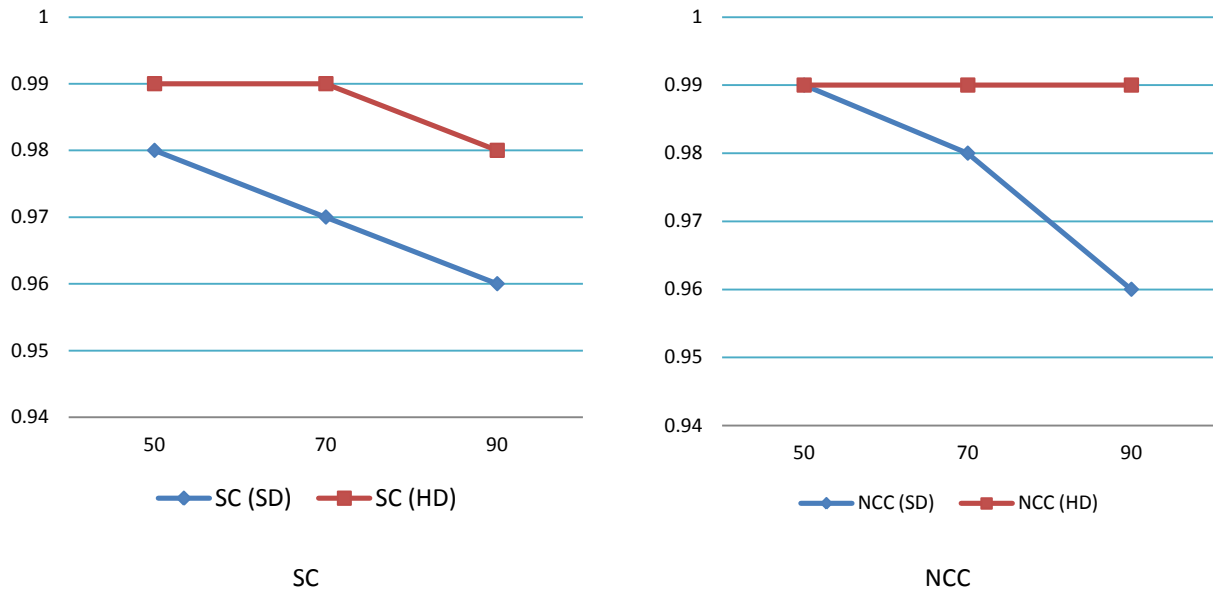


Figure II-29: SC and NCC as function of Δ, $\boldsymbol{m} = \mathbf{5}$.

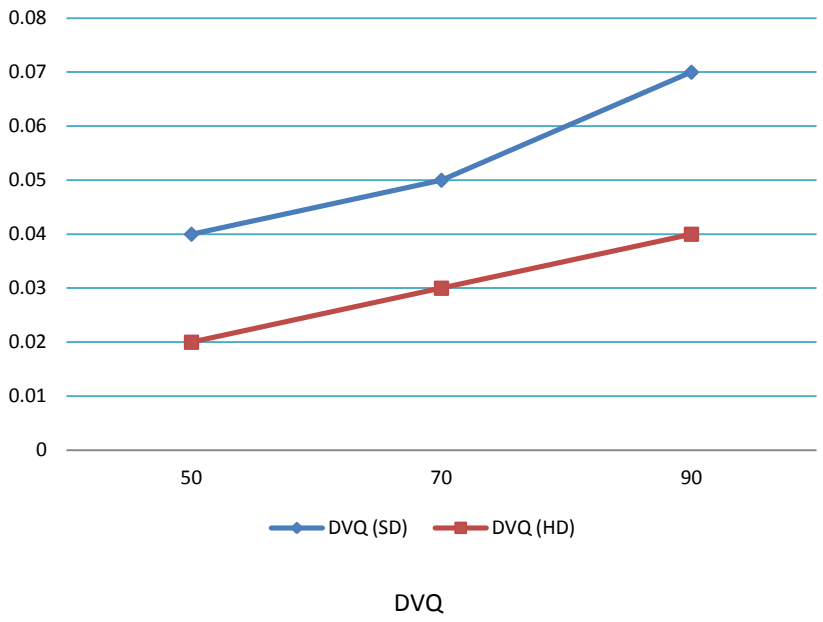


Figure II-30: DVQ as function of Δ, $\boldsymbol{m = 5}$.

Table II-6: Transparency behavior as function of Δ, $\boldsymbol{m = 5}$.

	SD			HD		
Δ	50	**70**	90	50	**70**	90
PSNR	62	**59**	54	67	**65**	61
Relative gain (ρ)	0.06	**0**	-0.07	0.04	**0**	-0.06
AAD	0.38	**0.47**	0.61	0.018	**0.021**	0.026
Relative gain (ρ)	-0.23	**0**	0.28	-0.18	**0**	0.23
SC	0.98	**0.97**	0.96	0.99	**0.99**	0.98
Relative gain (ρ)	0.01	**0**	-0.02	0.01	**0**	-0.01
NCC	0.99	**0.98**	0.96	0.99	**0.99**	0.99
Relative gain (ρ)	0.01	**0**	-0.02	0.00	**0**	-0.00
DVQ	0.04	**0.05**	0.07	0.02	**0.03**	0.04
Relative gain (ρ)	-0.20	**0**	0.4	-0.33	**0**	0.33

II.2.2.4. Computational cost

The execution time required by each operation included in the watermarking chain is evaluated on the following configuration: a PC with the following configuration: a Core4 CPU at 2.8 GHz and with 12 GB of RAM and a 500 GB HDD.

The values reported in this chapter are obtained on a corpus 60 min of video (MEDIVALS SD corpus Appendix B).

Figures II-26 and II-27 illustrate the allocation time and the time consuming during the embedding process for one second of video. It is noticed that the inner watermarking operations (selection and insertion) consume 0.045 s and 0.008 s respectively, *i.e.* 3.32% and 0.22% respectively of the total embedding processing time.

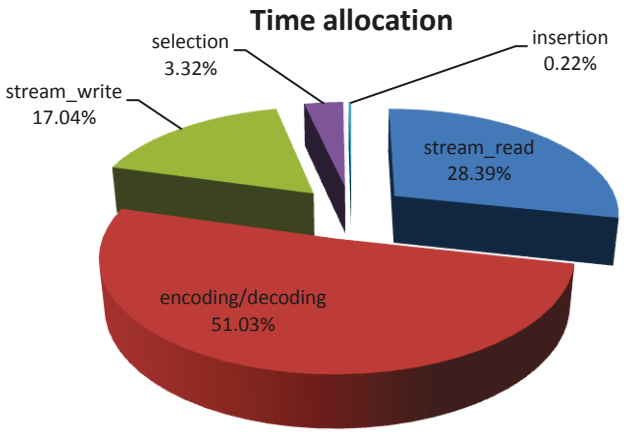


Figure II-31: Sharing time for the embedding process for one second of video.

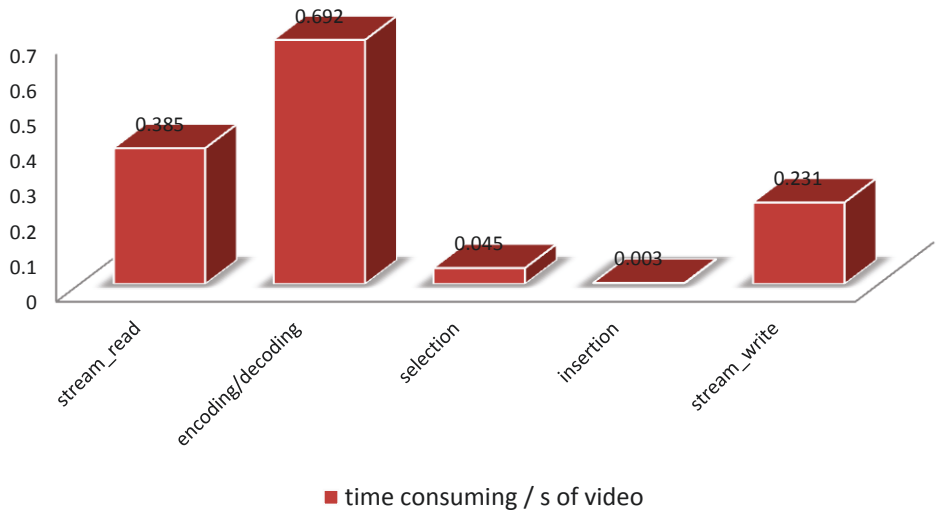


Figure II-32: Time consuming (in seconds) during the mark embedding for one second of video.

Figures II-28 and II-29 illustrate the allocation time and the time consuming during the mark detection process for one second of video. It is noticed that the detection consume 0.005 s for one second of video which presents 0.88% of the total embedding processing time.

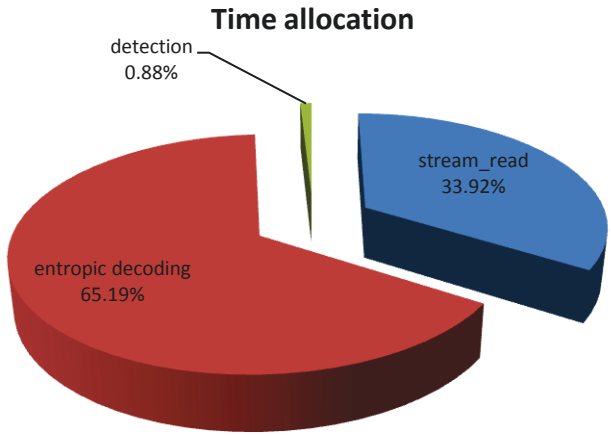


Figure II-33: Sharing time for the detection process for one second of video.

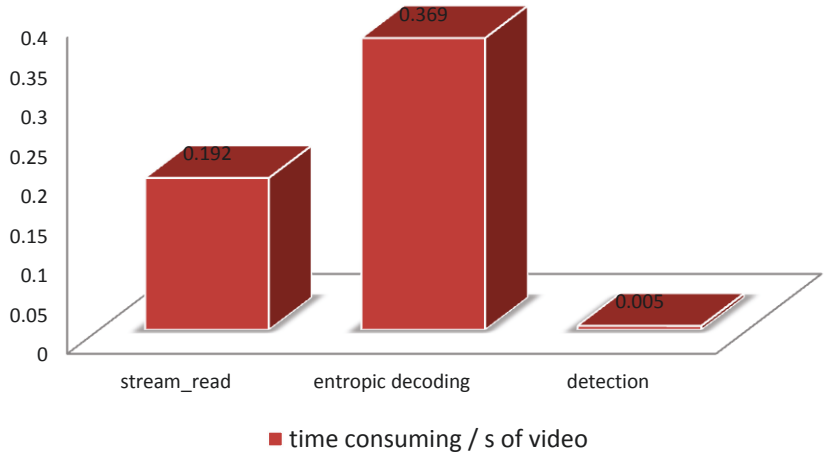


Figure II-34: Time consuming (in seconds) during the mark detection for one second of video.

The complexity analysis of the method shows that it is compatible with the real time (although its current implementation it is not):

- The operations intrinsically related to the mark selection, insertion and detection are less complex (and, implicitly, faster) than the MPEG-4 AVC entropic encoding/decoding and the stream read/write. For instance, in the current implementation, for protecting 1s of video, the

selection, insertion and detection sums up to 0.08s while the entropic decoding/encoding and the video stream read/ write operations reach 1.061s and 0.808s, respectively;

- The operations intrinsically related to the mark selection, insertion and detection are also much faster than the entropic decoding/encoding and the video stream read/write from the hard disk. When considering the same example as above (protecting 1s of video), the entropic decoding/encoding and the read/write operations are about 13 and 10 times slower than the mark selection/insertion/detection, respectively.

II.2.3. Conclusion

This chapter stands for the applicative validation of the *m*-QIM insertion/detection under the framework of the robust watermarking. The experimental investigation was conducted at three levels and considered $\alpha = \alpha^* + 0.4$ (according the *a priori* hints discussed in Chapter II.1.3).

First, for fixed transparency and robustness, an overall $log_2 m$ gain in data payload has been obtained.

Secondly, for fixed data payload and transparency constraints, no practical loss in robustness has been encountered, the variance of the BER with respect to m value being lower than 6×10^{-4}; such a result holds for both SD and HD corpora.

Finally, an average relative increase by a factor of 0.1 and 0.01 of PSNR and an average relative decrease by a factor of 0.23 and 0.16 of DVQ are obtained for fixed data payload and robustness in the SD case and HD case, respectively; the correlation based measures are quite constant with respect to m. This slight increase in transparency is the consequence of the decrease of the number of MPEG-4 AVC blocks needed for embedding when the alphabet size is increased and of the subsequent drift effect attenuation.

The experiments also investigated the impact of the Δ parameter in the overall results (the larger the Δ, the better the robustness but the worse the transparency). They also brought to light that a quite large interval of Δ values (from 50 to 90) can be considered for robust watermarking purposes.

Concerning the complexity cost, the evaluation shows that the selection/insertion/detection of the method are much faster than the MPEG-4 AVC entropic decoding/encoding and the video stream read/write from the hard disk which are 13 times and 10 times slower for one second of video.

In order to allow an overall comparison between our *m*-QIM framework and state-of-the-art binary QIM approaches, we also implemented the method in [GOL07]. We resumed the previous experiments while keeping the data payload – robustness – transparency constraints the same.

The data payload experiments (*cf.* Chapter II.2.2.1) demonstrated that the method in [GOL07] cannot allow even 1 bit per minute to be inserted.

The robustness experiments (*cf.* Chapter II.2.2.2) brought to light that the QIM method in [GOL07] resulted in:

- BER=0, after additive noise (for both SD and HD original content);
- BER = 0.17 and 0.15 after transcoding attacks, for SD and HD content, respectively; the underlying 95% relative errors were 0.02 and 0.019, respectively; these BER values are about 3 times larger than the average values corresponding to the *m*-QIM;
- BER = 0.45 and 0.43 after the Stirmark random bending attack, for SD and HD content, respectively; the underlying 95% relative errors were 0.025; these BER values are about 4 times larger than the average values corresponding to the *m*-QIM.

A synoptic view on these robustness results is provided by Figure II-30 which is analogous to Figure II-18. It illustrates the ARTEMIS logo recovered by using the method in [GOL07], after bipolar additive noise,

transcoding and Stirmark random bending attacks. It can be noticed that the Stirmark random bending attacks completely destroy the meaning of the recovered logo.

As the method in [GOL07] cannot provide the expected data payload and robustness constraints, the transparency experiments in Chapter II.2.2.3 were not resumed.

Figure II-35: The original ARTEMIS logo (left) and reconstructed ARTEMIS logo after additive noise, transcoding and geometric random bending attacks, respectively. These experiments correspond to the method in [GOL07].

II.3. Case study 2: MPEG-4 AVC semi-fragile watermarking for video surveillance application

In this chapter, we focus on the MPEG-4 AVC (Advanced Video Coding) [RIC03] integrity verification. First, the MPEG-4 AVC syntax elements that jointly reflect the semantic content and ensure the semi-fragility propriety are identified. The theoretical supported is granted by information theory tools. Secondly, the *m*-QIM (multiple symbols Quantization Index Modulation) insertion technique (whose optimality in proved in Chapter II.1) is considered in order to insert the mark. The experiments are carried out on 1 hour of video surveillance content and show that the proposed video watermarking based video integrity verification system is able to distinguish between content preserving and video content changing alterations.

II.3.1. Problem statement

Today, video-surveillance systems have become ubiquitous. The need to feel safe despite the high crime rate, the low cost compared to a human surveillance and the social acceptance enacted the intensive use of video-surveillance despite its intrinsic privacy intrusive character. Video surveillance can be found today not only in particularly sensitive areas such as banks, airports and government buildings but also in public places (stadiums, parks and residential areas). For instance, more than 9400 cameras are deployed in the London public transportation [DUR05].

This expansion of video-surveillance usage comes across with new challenges for the underlying video processing systems: ensuring methodological support for video surveillance content authenticity is the main deadlock for its consideration as evidence in justice courts.

Two approaches can be considered in order to ensure video authenticity, namely data and content based. The former considers the data (binary) representation of the video and extracts an authentication signature (*e.g.* a hash function) which meets the uniqueness and sensitivity (fragility) requirements. This signature is further stored as metadata. The latter no longer targets the binary representation of the video but its visual/semantic content. In order to be effective, such a signature should be robust to content preserving alterations and sensitive to content changing alterations. Consider the example in Figure II-31. Figure II-31.a represents an original content while Figure II-31.b shows its JPEG compressed version at quality factor Q = 70; Figure II-31.c gives a content-modified version of the image in Figure II-31.b in which a person has been added. From the binary representation point of view, the three images in Figure II-31.a, Figure II-31.b and Figure II-31.c are completely different; hence their data-based signatures should be different. However, from the semantic content point of view, Figure II-31.a and Figure II-31.b are identical while differing from Figure II-31.c. Consequently, the content-based signatures corresponding to Figure II-31.a and Figure II-31.b should be identical while differing from the signature extracted from Figure II-31.c.

(a) (b) (c) (d)

Figure II-36: The original frame (a) suffers a content preserving attack (a compression) (b) then a content alteration attack (the insertion of a person) (c). Signature based integrity verification should discriminate between the legal/fake areas (d).

Content-based authentication methods can be classified as passive or active [UPA11].

Passive authentication, also called forensic analysis [GER99] attempts to verify whether the content has been altered or not by using a direct statistical analysis. This way, no a priori processing/storing operation is required for the original content. While very appealing by its functional principle, passive authentication is not always possible and doubts about its reliability and security are raised for some applications [GER99].

This is not the case of the active approach, where the integrity of digital content is ensured by the embedding of an authentication signal (or signature) in the content itself, before its sharing/distribution. The active authentication is also called watermarking based authentication. Consider again the example in Figure II-31 and assume that Figure II-31.a carries (in a visual imperceptible way) a content authentication signature. On the one hand, this signature should be robust to compression; hence it should be recovered from the unaltered content areas in Figure II-31.c. On the other hand, it should be fragile against content-modifications; hence, it should no longer be recovered from the areas where the person was inserted. This way, the legal/faked areas can be discriminated into a same video frame (see Figure II-31.d).

From the applicative point of view, the main semi-fragile watermarking difficulties are the choice of the signature and of the embedding technique. The signature must ensure the semi-fragility property by being both sensitive to content changing alterations (frame cropping, object removing, …) and robust to content preserving alterations (transcoding, resizing, ...). As video sequences are preponderantly stored and distributed in compressed format, the signature should be both generated and inserted directly from/in the compressed domain, thus avoiding the computational overhead required by the decompression/compression.

II.3.2. Theoretical investigation on the authentication signature

In order to minimize the overhead induced by the decoding/encoding operations, the present study generates the signature directly from the MPEG-4 AVC syntax elements. The optimal syntax elements are identified by a study carried out on the basis of information theory, in order to build a syntax element based signature. The targeted signature must meet the trade-off between the robustness against content preserving and the sensitivity to content changing alterations at a low level of complexity.

II.3.2.1. Syntax elements identification

MPEG-4 AVC video sequences are structured into group of pictures (GOP) (*cf.* Appendix A.1). A GOP is constructed by fixed number of successive images of three main types (*I*, *P*, and *B*).

An *I* frame describes a full image coded independently, containing only references to itself. Secondly, the unidirectional predicted frames *P* use one or more previously encoded/decoded frames as reference for picture encoding/decoding. Finally, bidirectional predicted frames *B* consider in their computation both forward and backward reference frames. According to the coding principles, *I* frames preserve more information and require more bits for encoding than the other two types.

Our study focuses on the *I* frames. The *I* frames contain the salient visual/semantic information which is also exploited by the P and B frames in that GOP. Consequently, extracting the signature from the *I* frames has two main a priori advantages: the signature can be related to the video semantic and represent the whole GOP. Figure II-32 details the *I* frame encoding block diagram. MPEG-4 AVC transforms the uncompressed data in a classical compression chain: prediction ***P***, transformation ***T***, quantization ***Q*** and entropic coding ***E***.

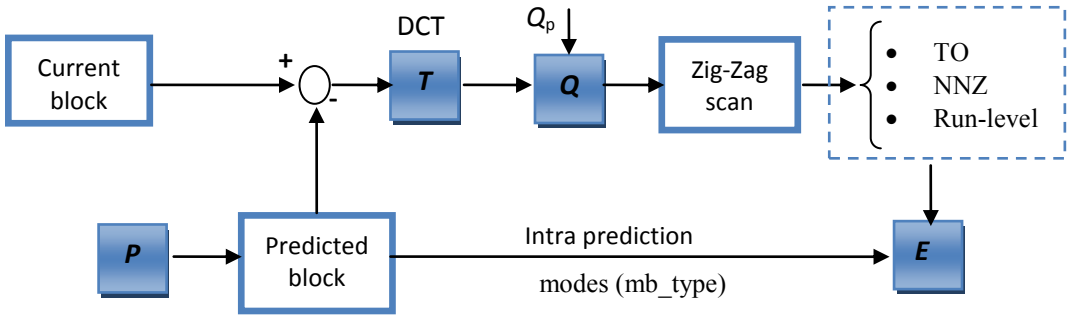


Figure II-37: Intra frame coding diagram.

I frames are encoded according to Intra prediction modes which exploit the spatial redundancy to enhance the compression efficiency. The MPEG-4 AVC standard offers 13 directional intra prediction modes (*cf.* Appendix A.2). For each current block, a predicted block is constructed from the boundary pixels of the neighboring blocks which are previously encoded. MPEG-4 AVC offers two intra prediction modes classes: *I*16×16 (with four prediction modes) for smoothed regions and *I*4x4 (with 9 prediction

modes) for the textured regions. For each block, the prediction mode minimizing the rate-distortion cost is selected.

The residual block computing the difference between the current block and the predicted block is transformed by using a DCT and a quantizer. Each quantized transformed residual block is further mapped into a 15 coefficients vector. In the baseline profile, the resulting vector is encoded by the CAVLC (Context Adaptive Variable Coding Length) encoder (*cf.* Appendix A. 2):

- CAVLC uses the run-level entropic encoding to represent compact zero sequences.
- The non-zero values taken by coefficients are often expressed in ± 1 sequences. CAVLC encodes the ±1 number of the large compact succession of ±1 (Trailing-Ones).
- The number of non-zero coefficients (NNZ) is encoded using a lock-up table. The encoding table is determined based on the NNZ of neighboring blocks.
- Levels values of nonzero coefficients in low frequencies are larger than those in high frequencies. CAVLC takes advantage of this behavior by adapting the choice of the encoding table (VLC look-up table). The standard provides four encoding tables, depending on the NNZ value, see Table II-7.

The *I* frames contain four main syntax elements per intra block: mb_type, TO (Trailing Ones), NNZ and run-level. The extraction of these elements is performed through a syntax parser. This study will not consider the run-level. In fact, to be effective, the signature should have a limited alphabet size; this is not the case of run-level element. Table II-7 illustrates the syntax elements that will be investigated in this study.

Table II-7: MPEG-4AVC syntax elements.

Syntax elements	Description
mb_type	Intra prediction mode class type: (*I*4x4 or *I*16×16).
TO (Trailing Ones)	It takes a value from 0 to 3. If there is a succession of more than three ±1, only the last three are considered.
NNZ (Number of Non Zero coefficients)	NNZ encoded according to 4 tables are considered based on its value: Table 1 (0, 1), Table 2 (2, 3), Table 3 (4, 5, 6, 7) and Table 4 (8 or more).

II.3.2.2. Syntax elements behavior to content preserving attacks

II.3.2.2.1. Experimental protocol

This section investigates the possibility of building an authentication signature from the 3 syntax elements identified above (mb_type, TO, NNZ). To do this, the behavior of each of these elements to content preserving attacks (transcoding, Gaussian filtering, sharpening and scaling) is first investigated. Then, the considered attacks are applied according to two scenarios: (S1) the MPEG-4 AVC encoder is

allowed to choose the quantization parameter *Qp*, and (S2) the quantization parameter *Qp* is set to 31. Table II-8 shows the steps involved in both scenarios.

Table II-8: Test scenarios.

S1	S2
(1) Encode the video at variable *Qp*	(1) Encode the video at *Qp* = 31
(2) Extract the syntax elements and store them	(2) Extract the syntax elements and store them
(3) Attack the video and re-encode it according to (1)	(3) Attack the video and re-encode it according to (1)
(4) Extract the syntax elements and compare them to those extracted at (2)	(4) Extract the syntax elements and compare them to those extracted at (2)

First, the syntax elements (mb_type, TO, NNZ) are extracted using a syntax parser and stored as signatures. Secondly, the video sequence is attacked (transcoded, filtered, scaled, …) and further re-encoded according to the same initial configuration. Finally, the syntax elements are extracted from the attacked video and compared to those extracted previously from the original video. The syntax elements extracted from each 4x4 block intra are coded as follows:

$$\text{mb\_type} = \begin{cases} 0 & \text{if } I4\text{x}4 \\ 1 & \text{if } I16 \times 16 \end{cases} \qquad \text{TO} = \text{TO} \in \{0, 1, 2, 3\}$$

$$\text{NNZ} = \begin{cases} 0 & \text{if } \text{NNZ} \leq 1 \\ 1 & \text{if } 1 < NNZ \leq 3 \\ 2 & \text{if } 3 < NNZ \leq 7 \\ 3 & \text{if } \text{NNZ} > 7 \end{cases}$$

II.3.2.2.2. Corpus

The experiments were carried out on a video surveillance corpus composed of 8 sequences of 10 minutes each, downloaded from internet [WEB01] or recorded under the framework of the SPY project (*cf.* Appendix B). This corpus is encoded in MPEG-4 AVC in Baseline Profile at 512 kbps, 640x480 pixel frames; the GOP size is set to 8.

II.3.2.2.3. Robustness against content preserving attacks

The initial value of a syntax element is likely to change after an attack on a random basis, given by both the attack and the content itself; hence we can investigate these modifications by modeling the attack with noise matrices. In such a matrix, the lines correspond to the values of original elements, while the columns to the values after attacks. An element in a matrix is the corresponding conditional probability, estimated on the corpus. These noise matrices are estimated by successively applying four content preserving attacks (transcoding, sharpening, Gaussian filtering and scaling) according to the two scenarios presented above. The size of the corpus was large enough so as to ensure the statistical relevance of the results: for each syntax element, 95% confidence limits are computed with relative error $\hat{\varepsilon}_r < 5 \times 10^{-3}$.

By further computing the probability of correct detection (P_c) and the mutual information (I) the related decision can be made on the optimal syntax element.

$$P_c = \sum_{i=1}^{N} P_{ii} * P_i \,, \qquad I = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} * P_i * log\,(P_{ij}/P_j) \tag{II-43}$$

where P_{ij} presents the noise matrix element of coordinates *i* and *j*, P_i is the average probability that the syntax element takes the value i in the original video and P_j is the average probability that the syntax element takes the value j in the attacked video. N is the size of the corresponding alphabet (*i.e.* $N = 2$ for mb_type and $N = 4$ for NNZ and TO).

Table II-9, Table II-11 and Table II-13 illustrate the distribution probability for mb_type, NNZ and TO respectively.

Table II-10, Table II-12 and Table II-14 illustrate the computed noise matrix for mb_type, NNZ and TO respectively.

The obtained probability of correct detection and mutual information values for each syntax element and for each tested attack according to the two scenarios are plotted in Figure II-33 and Figure II-34, respectively. The analysis of these results brings to light two main conclusions:

- The probability of correct detection and the mutual information values show that the syntax element (mb_type) remains more robust than the two other syntax elements (TO and NNZ) against all the investigated attacks. The obtained averaged *Pc* over the 4 attacks and the 2 scenarios are 0.92, 0.76, and 0.47 for mb_type, NNZ and TO, respectively. I values feature an average of 0.55, 0.44, and 0.15 for mb_type, NNZ and TO, respectively.
- A better robustness is achieved for (S2). A fixed quantization step increases the mb_type correct detection probability by 0.06, 0.02, 0.04 and 0.03 and the mb_type mutual information by 0.21, 0.13, 0.01 and 0.02 for transcoding, sharpening, Gaussian filtering and scaling, respectively

Table II-9:mb_type distribution probability.

	S1		***S2***	
	0	***1***	***0***	***1***
P	0.62	0.38	0.54	0.46

Table II-10: mb_type transition matrix.

	Transcoding				**Sharpening**			
	S1		*S2*		*S1*		*S2*	
	0	***1***	***0***	***1***	***0***	***1***	***0***	***1***
0	0.95	0.04	0.98	0.02	0.78	0.22	0.82	0.18
1	0.11	0.88	0.03	0.96	0.06	0.94	0.03	0.96
	Gaussian filtering				**Scaling**			
	S1		*S2*		*S1*		*S2*	
	0	***1***	***0***	***1***	***0***	***1***	***0***	***1***
0	0.94	0.06	0.95	0.05	0.92	0.08	0.93	0.07
1	0.15	0.85	0.05	0.94	0.12	0.88	0.07	0.93

Table II-11: NNZ distribution probability.

	S1				**S2**			
	0	***1***	***2***	***3***	***0***	***1***	***2***	***3***
P	0.68	0.21	0.09	0.02	0.74	0.17	0.08	0.01

Table II-12: NNZ transaction matrix.

	Transcoding								**Sharpening**							
	S1				*S2*				*S1*				*S2*			
	0	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***
0	0.97	0.02	0.01	0.00	0.97	0.02	0.00	0.01	0.68	0.22	0.09	0.01	0.69	0.20	0.09	0.01
1	0.45	0.52	0.02	0.01	0.33	0.64	0.02	0.01	0.16	0.24	0.50	0.10	0.08	0.28	0.51	0.13
2	0.05	0.38	0.50	0.07	0.02	0.34	0.63	0.01	0.02	0.03	0.38	0.55	0.02	0.02	0.35	0.61
3	0.06	0.02	0.61	0.31	0.01	0.01	0.62	0.36	0.01	0.01	0.02	0.96	0.01	0.01	0.01	0.97
	Gaussian filtering								**Scaling**							
	S1				*S2*				*S1*				*S2*			
	0	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***
0	0.90	0.09	0.01	0.00	0.95	0.04	0.00	0.01	0.94	0.06	0.00	0.00	0.95	0.03	0.01	0.01
1	0.45	0.49	0.05	0.01	0.47	0.50	0.02	0.01	0.43	0.51	0.05	0.00	0.48	0.48	0.03	0.01
2	0.10	0.46	0.43	0.01	0.11	0.53	0.34	0.02	0.08	0.50	0.41	0.01	0.11	0.55	0.34	0.00
3	0.02	0.09	0.83	0.06	0.02	0.13	0.83	0.02	0.02	0.11	0.82	0.05	0.01	0.13	0.83	0.03

Table II-13: TO distribution probability.

	S1				**S2**			
	0	***1***	***2***	***3***	***0***	***1***	***2***	***3***
P	0.41	0.3	0.16	0.13	0.49	0.27	0.13	0.1

Table II-14: TO transaction matrix.

	Transcoding								**Sharpening**							
	S1				*S2*				*S1*				*S2*			
	0	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***
0	0.84	0.11	0.03	0.02	0.82	0.13	0.02	0.01	0.47	0.22	0.17	0.14	0.50	0.25	0.12	0.13
1	0.68	0.22	0.06	0.04	0.39	0.46	0.09	0.04	0.40	0.24	0.16	0.20	0.32	0.30	0.18	0.20
2	0.60	0.18	0.14	0.08	0.14	0.33	0.39	0.14	0.40	0.21	0.16	0.23	0.28	0.27	0.20	0.25
3	0.52	0.14	0.13	0.21	0.07	0.15	0.26	0.52	0.34	0.24	0.18	0.24	0.29	0.26	0.20	0.25
	Gaussian filtering								**Scaling**							
	S1				*S2*				*S1*				*S2*			
	0	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***	***0***	***1***	***2***	***3***
0	0.57	0.26	0.10	0.07	0.65	0.23	0.07	0.05	0.65	0.21	0.09	0.05	0.72	0.18	0.06	0.04
1	0.42	0.33	0.15	0.10	0.38	0.40	0.14	0.08	0.45	0.33	0.15	0.07	0.42	0.38	0.14	0.06
2	0.36	0.29	0.21	0.14	0.26	0.35	0.26	0.13	0.33	0.31	0.24	0.12	0.27	0.36	0.25	0.12
3	0.25	0.27	0.24	0.24	0.21	0.28	0.26	0.25	0.27	0.28	0.23	0.20	0.20	0.29	0.26	0.25

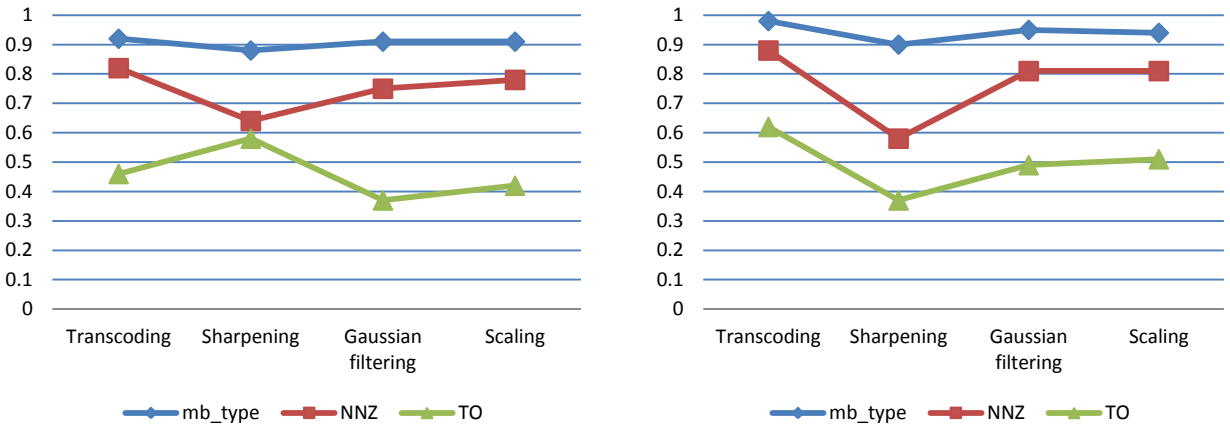


Figure II-38: Probability of correct detection, for (S1) – left and (S2) - right.

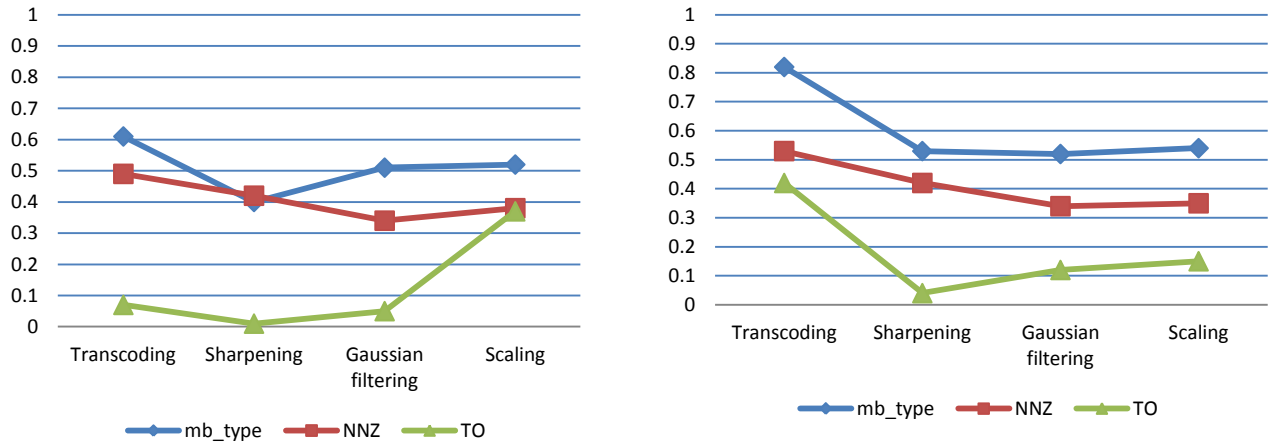


Figure II-39: Mutual information, for (S1) – left and (S2) - right.

These results prove that the mb_type is the single element able to satisfy the constraint of robustness over the three investigated syntax elements and that the Second scenario (the quantization parameter is fixed at 31) features better robustness performances. Thus, the rest of the study will concern only the mb_type elements and Second scenario (S2) conditions are kept.

II.3.2.2.4. Sensitivity to content changing alterations

This section investigates the sensitivity of mb_type syntax element to content changing alterations. In this study, the content changing alteration attack is performed by removing a person as illustrated in Figure II-35-a and Figure II-35-b. Just for illustration, such an attack can be performed by any non-professional user on a home PC, with software available on Internet; it takes about 10 minutes to process 1 second of video.

First *I* frame of original video (a) First *I* frame of attacked video (b) Content alterations detection (c)

Figure II-40: Content changing alterations.

In order to spatially localize the content manipulations, an alteration detection image is generated. When the mb_type is detected as changed, the pixel value corresponding to it is set to red. The resulting detection image is illustrated in Figure II-36. To spatially locate alterations, positions reported as manipulated by the detection matrix are further projected onto the inspected image, see Figure II-36-c. It can be noticed the presence of false alarms and that the red block are denser in the altered area than other areas. To enhance the alteration detection and avoid the false alarms, the following filter is applied to the detection image:

$$M(i,j) = \begin{cases} 1 & if \sum_{i_0}^{a} \sum_{j_0}^{b} M(i+i_0, j+j_0) > s \\ 0 & otherwise \end{cases} \quad (44)$$

where M is the alteration detection image, a × b presents neighborhood window filter size and s is the decision threshold. i_0 and j_0 present the line/column indexes in the filter window. Figure II-36 illustrates the evolution of the detection image as function of the decision threshold and Figure II-37 reports the obtained false alarm probability as a function of s. We note that from a given threshold (s = 3 in our case) false alarms disappear and the altered area is surrounded. We also notice that the optimization of the parameter s may be mandatory to improve the spatial accuracy of the alteration detection according to the targeted application.

M for s = 0 (a) M for s = 1 (b) M for s = 2 (b) M for s = 3 (c)

Figure II-41: Alteration detection matrix.

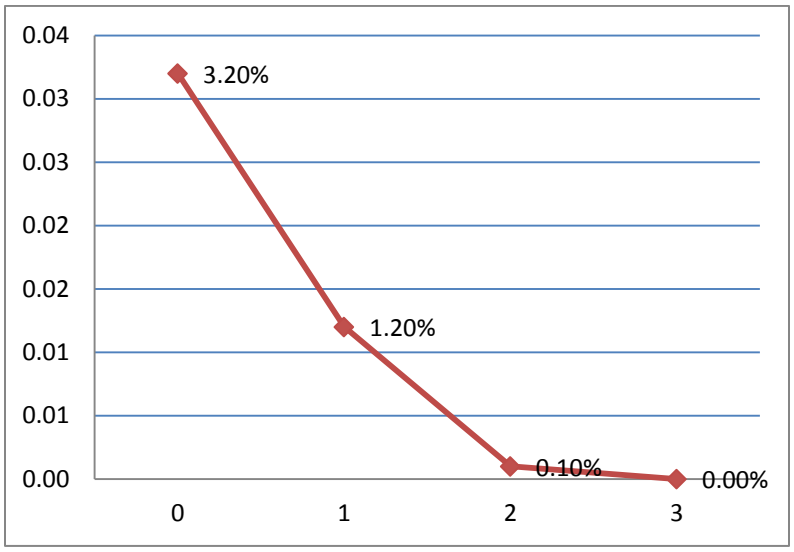


Figure II-42: False alarm as function of $\boldsymbol{s}$.

II.3.2.3. Conclusion

This section investigates the MPEG-4 AVC syntax elements with respect to their potential usage as authentication signature robust to content preserving attacks and sensitive to content changing. After an *a prior* study, 3 syntax elements (mb_type, NNZ, and TO) are identified and tested by using information theory based entities. First, noise matrices, $\boldsymbol{P_c}$ and $\boldsymbol{I}$ demonstrate that mb_type is the optimal choice for content preserving attacks. Secondly, mb_type proved to be able to result into probability of false alarms equal to zero under content changing attacks. The next section will present a video integrity verification system whose inserted signature is based on mb_type syntax element.

II.3.3. Video integrity verification method

This section details SPYART, the advanced video integrity verification system, further referred to as SPYART. The mb_type based signature is deployed jointly with the *m*-QIM watermarking technique (*cf.* Chapter II.1) to build the video integrity verification system.

While the same insertion/detection rule as in the case of robust watermarking is basically kept the semi-fragile watermarking application comes with particular constraints on (1) the mark semantics and (2) the data payload.

First, from the semantic point of view, the mark no longer relates to the intellectual property rights information but to an authentication signature of the content itself. The low complexity requirement is met when such a signature is extracted and inserted directly from/in the MPEG-4 AVC syntax elements, with minimal decoding/reencoding operations.

Secondly, from the data payload point of view, such an authentication signature results in about 4.5 kbits per minute (assuming SD corpus with frame size of 640x480). As such a data payload is 30 times larger than the one investigated in Chapter II.2 for robust watermarking purposes, the robustness and transparency investigation should be resumed accordingly.

To meet this requirement, SPAYRT considers individual groups of *i* successive *I* frames (further referred to as *I-Group*) sampled from an MPEG-4 AVC video sequence, see Figure II-38. The mb_type based signature is computed from the first frame I_0 in such an *I-Group* and further embedded into the rest of *i*-1 frames by means of the *m*-QIM watermarking technique. The shorter the *I-Group*, the more accurate the temporal localization of the altered content.

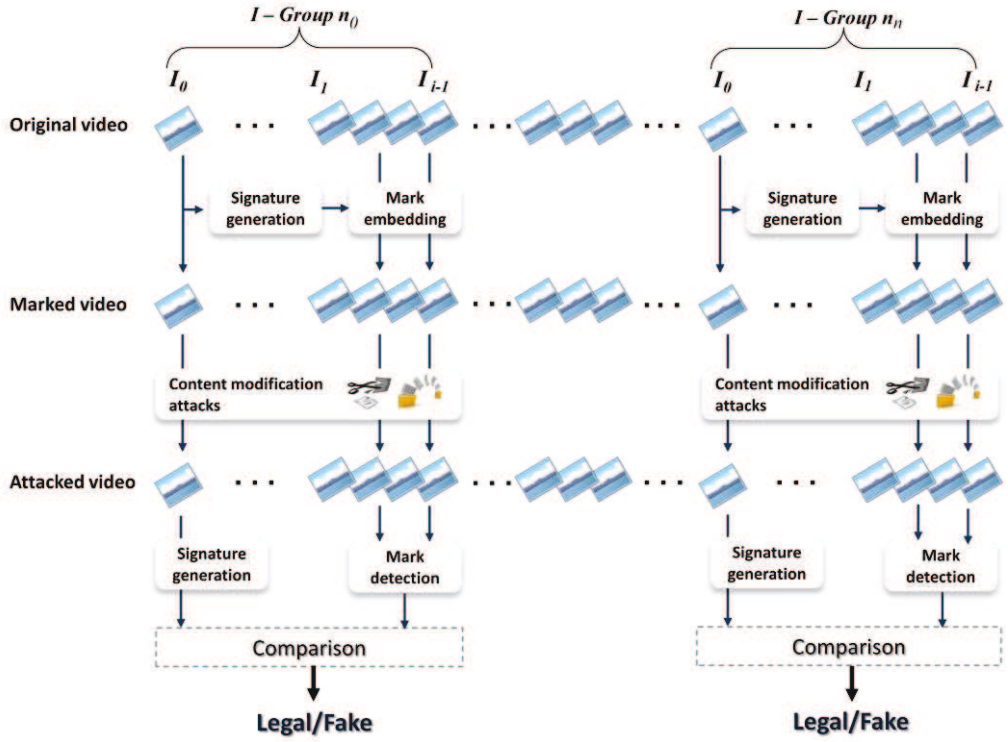


Figure II-43: Semi-fragile watermarking system.

II.3.3.1. Signature generation

Signature generation is structured into three modules as shown in Figure II-39: *feature extraction*, *binary mask generation* and *signature encoding*.

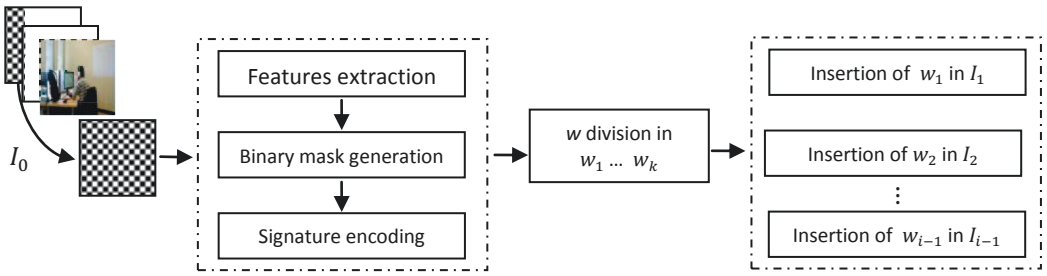


Figure II-44: Mark generation $\boldsymbol{W} = \{\boldsymbol{w_1}, \boldsymbol{w_2}, \dots, \boldsymbol{w_k}\}$ based on syntax element.

PSEUDOCODE FOR THE SIGNTATURE GENERATION
Begin video ***If*** the current frame is *I* ***then*** ***Begin frame*** ***Begin Macroblock*** - *Decode_One_Macroblock ()* - *Read_On_Macroblock ()* - *Read_ipread_modes()* - *Assign one bit of signature according to the prediction mode.* ***End Macroblock*** ***End frame*** ***Else*** go to the next frame ***End video***

II.3.3.1.1. Feature extraction

The mb_type syntax element has proven its relevance to generate an authentication signature conveying information about the content and allowing its authentication (*cf.* Chapter II.3.2.2).

When encoding the *I* frames, the MPEG-4 AVC standard can consider two types of blocks, referred to us as mb_type: 16×16 pixel blocks for smoothed regions (corresponding to the 4 ways of achieving the *I*16*MB* prediction modes) and 4x4 pixel blocks for textured areas (corresponding to the 9 ways of achieving the *I*4*MB* prediction modes).

In this study the mb_type syntax element will be considered as the feature generating the authentication signature. On the one hand, according to the MPEG-4 AVC coding principles, any alteration that changes the texture of the content will *a priori* change the prediction modes, thus allowing content integrity verification (related to the fragility property). On the other hand, there is *a priori* hint about the robustness of this signature against content preserving attacks. Hence, a statistical investigation on the behavior of the mb_type sythaxe element under different attacks was conducted and described in Chapter II.3.2.

For each *I-Group*, the mb_type syntax element of the first *I* frame I_0 will be extracted according to an MPEG-4 AVC syntax element parser (*cf.* Appendix A.4).

II.3.3.1.2. Binary mask generation

For the first frame from an *I-Group* (the I_0 frame in Figure II-39), a binary mask is generated by assigning one bit for each macroblock $B(x, y)$ based on its extracted feature:

$$BM(x,y) = \begin{cases} 1, & if\ I4MB \\ 0, & if\ I16MB \end{cases},$$

where x and y represent positions of the B macroblock within the frame I_0 and BM is the generated binary mask.

II.3.3.1.3. Signature encoding

In order to cope with the *m*-QIM (multi symbol Quantizing Index Modulation) principles, the BM binary mask should be encoded into an *m*-arry alphabet; as in the SPYART case $m = 5$, the encoding alphabet is $\{-2, -1, 0, 1, 2\}$. For real life watermarking applications, this encoding procedure should also ensure limited error propagation: hence, a fixed-length encoding is preferred instead of an optimal (minimal average length) one. Note that according to the Shannon's first theorem, the optimal encoding average length between a binary and an *m*-ary alphabet is $log_2 m$; in our case, $log_2 5 = 2.31$. Hence, we considered an encoding scheme based on 3 bit overlapping blocks (with the overlap of 1 bit). The binary value of such a block gives information about the sign and the parity of the 5ary alphabet, Table II-15 and Figure II-40.

Table II-15: Encoding table.

000	001	010	011	110	101	100	111
0	-1	-1	-2	1	1	2	2

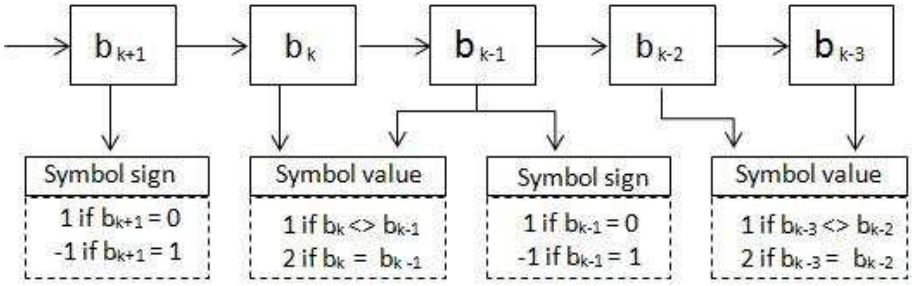


Figure II-45: Signature encoding.

In order to illustrate this principle, consider the encoding of the 101001110 bit string. This bit string results into 4 overlapping groups of 3 bits: 101, 100, 011 and 110. Hence, it will be encoded as -1, -2, 2, -1; notice that both 101 and 110 are encoded as -1. The decoding procedure is applied at two levels: first, the signs of the 5ary symbol alphabet are converted into the corresponding bits and then the parity information is considered to decode the remaining sequences. In order to illustrate the decoding, reconsider the example above; we have to decode the string -1, -2, 2, -1. First, we decode the bits placed at the odd position in the bit string, by considering the signs of the 5ary symbols: 1x1x0x1x (by x we denoted the bits unknown at this stage). Further on, the encoding dictionary in Table II-15 gives the following sequence: 110100111x. In order to decode the last bit, a new symbol should be received (this is achieved by always padding the useful information with two fixed bits).

The encoding performance has been evaluated with respect to the binary encoding in terms of efficiency, according to the Shannon definition [SHA48]:

$$\eta = \frac{L_{min}}{\bar{L}}$$

where L_{min} is the minimum possible code length (according to the Shannon's first theorem) and $\bar{L}$ is the average code length.

In the binary case, the distribution probabilities of the symbols {0, 1} are {0.54, 0.46}, resulting an efficiency $\eta_2 = 0.995$. Concerning the 5-ary encoding, the distribution probability of the symbols {-2,-1, 0, 1, 2} are {0.19, 0.21, 0.23, 0.19, 0.18}, resulting an efficiency $\eta_5 = 0.995$. Consequently, we notice that the binary and 5-ary encoding have quite the same efficiency (differences of 0.002 in favor of the 5-ary code).

II.3.3.2. Mark embedding

For each *I-Group*, the signature generated from the first frame (I_0 in Figure II-41) is shuffled (according to a private key) and divided into $i-1$ sub-marks w_i. Each sub-mark is inserted into one of the I_i frames of that *I-Group*, Figure II-41. The insertion is performed within the 15 AC coefficients of 4x4 blocks of *I* frame by *m*-QIM techniques (*cf.* Chapter II.1 and Chapter II.2).

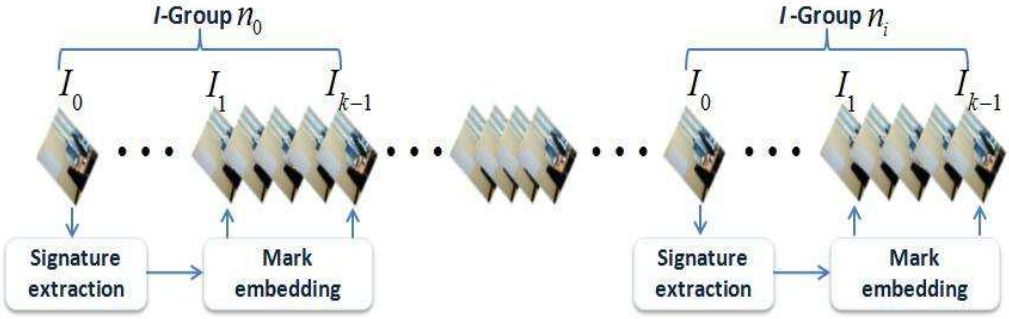


Figure II-46: Mark embedding.

II.3.3.3. Mark detection and integrity verification

Consider now the watermarked and potentially corrupted video sequence, see Figure II-42.

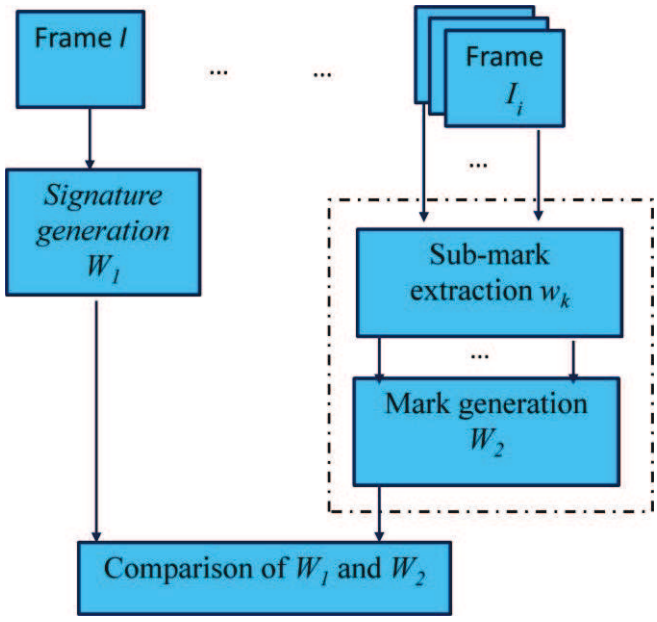


Figure II-47: Integrity verification.

This sequence is first re-encoding with the original encoder parameters and then divided into *I-Groups*. The $\widehat{w}_k$ sub-marks are individually extracted according to the *m*-QIM principles from each I_k , $k = 1, \dots, i-1$ frames; be there $\widehat{w}$ the vector obtained by concatenating these extracted sub-marks.

In parallel, the signature corresponding to the first I_0 frame is extracted and the corresponding would-be mark w_a is computing.

As $\widehat{w}$ conveys information about the original *I-Group* features and w_a about its attacked replica, by comparing these two watermarks, a decision concerning the integrity of the video content can be made. SPYART considers that an area in a frame was modified when at least 50% of the $\widehat{w}$ elements extracted from that area do not match the corresponding $\widehat{w}_a$ elements, see Figure II-43. Consequently, SPYART has a temporal precision given by the duration of the *I-Group* and a spatial accuracy given by the size of the area on which the alteration is investigated.

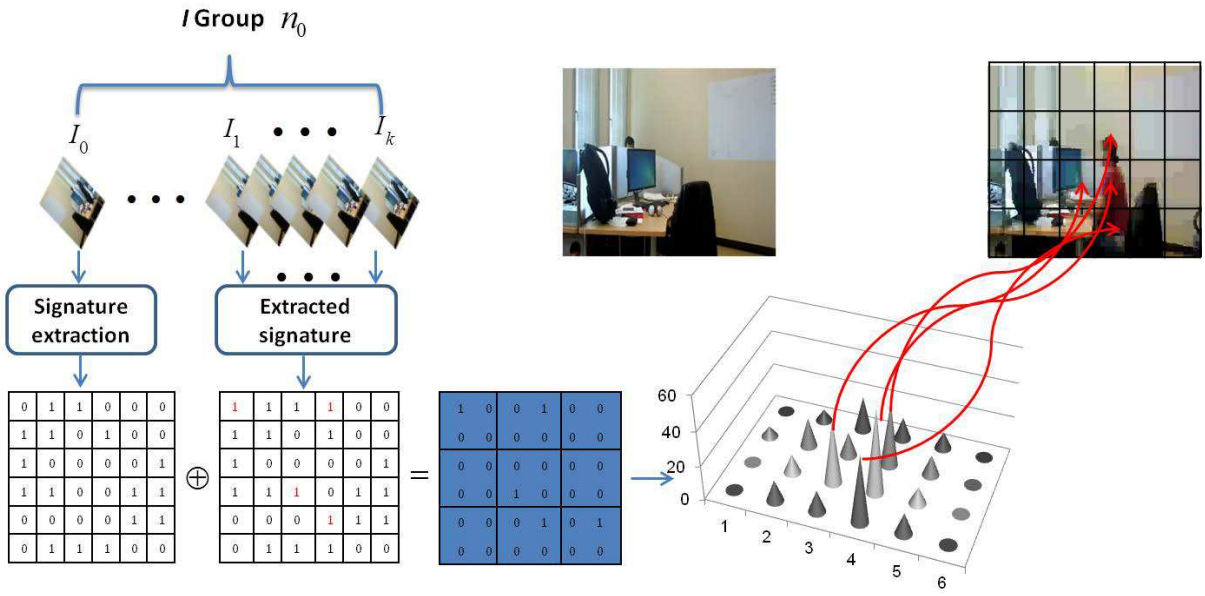


Figure II-48: Spatial alterations detection.

From the result interpretation point of view, feed-back provided by professional under the framework of the SPY project brought to light that when an *I-Group* is maliciously altered, it is very likely to have other successive blocks altered. Consequently, the decision according to equation (II-20) is followed by a post-processing rule of the type: one *I-Group* is considered as altered if at least two *I-Groups* that succeed it or precede it are detected as altered.

II.3.4. Functional evaluation

The experiments investigate the robustness, fragility and transparency properties, for a prescribed data payload value, set by the size of the authentication signature (*i.e.* 4.5 kbits per min in the case of the tested SPY corpus).

First, under the semi-fragile watermarking framework, transcoding and additive noise are the most harmless authorized attacks. Consequently, Chapter II.3.4.1 investigates the BER after these attacks, for prescribed transparency (average PSNR = 32 dB).

Secondly, the fragility is investigated in Chapter II.3.4.2 by computing the *Precision* and *Recall* rates when identifying the content altered blocks at fixed robustness (maximal BER of 0.1± 0.03 after transcoding). The same transparency constraint is kept (average PSNR = 32 dB).

Finally, the transparency is assessed in Chapter II.3.4.3 for prescribed robustness/fragility limits (maximal BER of 0.1± 0.03 and minimal *Precision* and *Recall* of 0.9).

Note that current day video surveillance cameras are mainly IP-based and generate SD content; hence our study considers only an SD corpus. The experiments were carried out on a video surveillance corpus composed of 8 sequences of 10 minutes each, downloaded from internet [WEB01] or recorded under the framework of the SPY project. Their content is heterogeneous, combining city streets, highways,

industrial objectives, shopping centers, … This corpus is encoded in MPEG-4 AVC in Baseline Profile (no B frames, CAVLC entropy encoder) at 512 kbps, 640x480 pixel frames; the GOP size is set to 8.

Four m values and three quantization steps Δ are compared all through the study, namely $m \in \{2,3,5,7\}$ and $\Delta \in \{50,70,90\}$. Each and every time, $\alpha = \alpha^* + 0.4$, see the discussion in Chapter II.1.3.3.

II.3.4.1. Robustness

This section investigates the robustness of the semi-fragile watermarking method based on m-QIM against transcoding (down to 25% from the original stream bit rate) and bipolar noise addition, for a data payload of 4.5 kbits per minutes. The average transparency is set to PSNR = 32 dB. Figure II-44 illustrates the BER as a function of m, when $m \in \{2,3,5,7\}$, $\Delta = 70$ and $\alpha = \alpha^* + 0.04$.

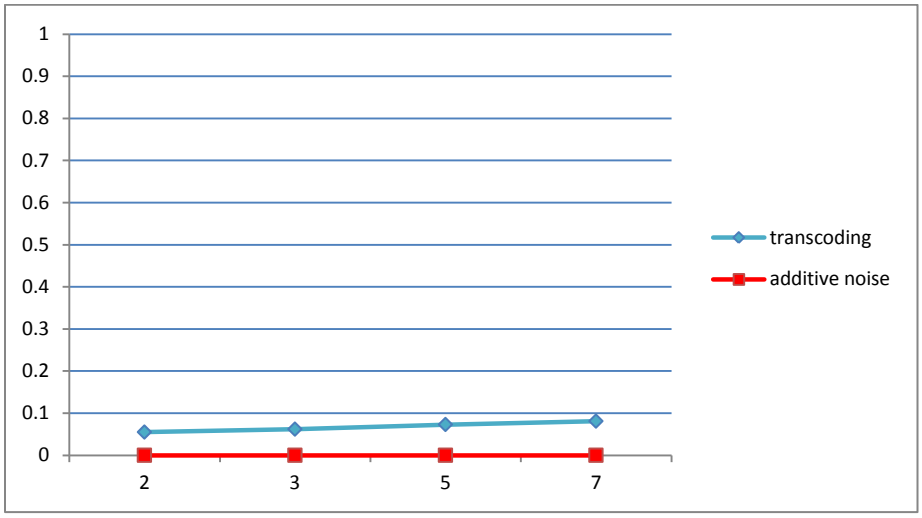


Figure II-49: BER as a function of $\boldsymbol{m}$ *for* $\Delta = \mathbf{70}$.

The BER values reported in Figure II-44 prove the robustness of the semi-fragile method against the investigated attacks: average BER of 0.08 and 0 against transcoding and additive noise are obtained, respectively. It can also be noticed that the BER is quite constant with respect to m, its variance being lower than 4×10^{-4}. These numerical values are statistical relevant, in the sense that the 95% confidence intervals [WAL02] computed for the BER after the transcoding resulted in relative errors lower than 5×10^{-3} (the BER in the bipolar noise addition was constantly equal to 0; hence, no confidence interval has been computed).

The experiments are resumed so as to investigate the robustness behavior as a function of Δ. This time, $m = 5$ and $\Delta \in \{50,90\}$, see Table II-16 and Figure II-45.

When keeping the same transparency constraints, $\Delta = 90$ results in relative decreases of the BER by a factor of 0.14, while $\Delta = 50$ results in a relative increase of the BER by a factor of 1.28; these relative gains are computed by (41). Table II-16 demonstrates that, at least from the robustness point of view, a value $\Delta = 50$ is no longer of practical relevance. Note that these results confirm the theoretical ground

as well as the behaviour which was experimentally identified in Chapter II.2.2.2: the larger the Δ value, the better the robustness.

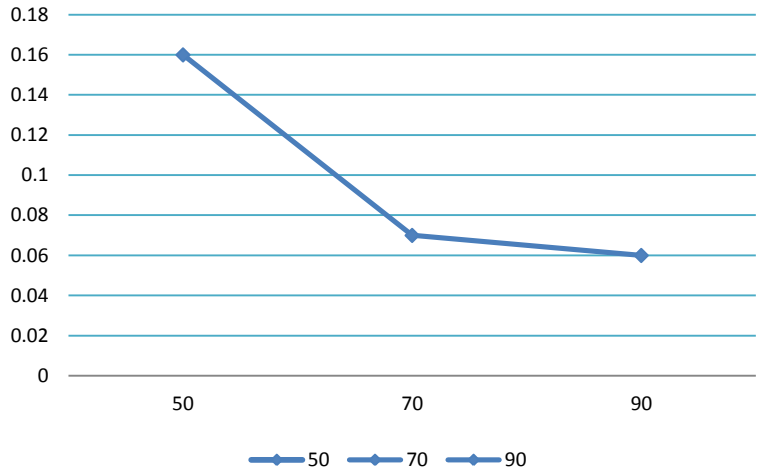


Figure II-50: Robustness as function of Δ, $\boldsymbol{m = 5}$.

Table II-16: Robustness behavior as function of Δ, $\boldsymbol{m = 5}$.

Δ	50	**70**	90
Transcoding (BER)	0.16	**0.07**	0.06
Relative gain ($\boldsymbol{\rho}$)	1.28	**0**	-0.14

II.3.4.2. Fragility evaluation

The video content is considered as being altered when one object is removed, inserted or substituted. To simulate this attack, we wrote a piece of code that tampers the marked video by arbitrarily changing 1/81 areas of the frame.

From the fragility point of view, an ideal watermarking method should fail in detecting the mark from each and every area which was subject to content alterations, thus detecting the altered area. While such a behavior can be also expressed in terms of probability of missed detection and false alarm, the literature brings to light two more detailed measures, namely the *Precision* and the *Recall* ratios, defined as follows [BUC94].

$$Precision = tp/(tp + fp), Recall = tp/(tp + fn)$$

where t_p is the number of true positive (*i.e.* the number of content modified areas which do not allow the mark to be recovered), fp is the false positive (*i.e.* the number of content preserved areas which do not allow the mark to be recovered) and fn is the false negative number (*i.e.* the number of content modified areas which allow the mark to be detected).

Figure II-46 illustrates the obtained *Precision* and *Recall* values as a function of *m*, for $\Delta = 70$.

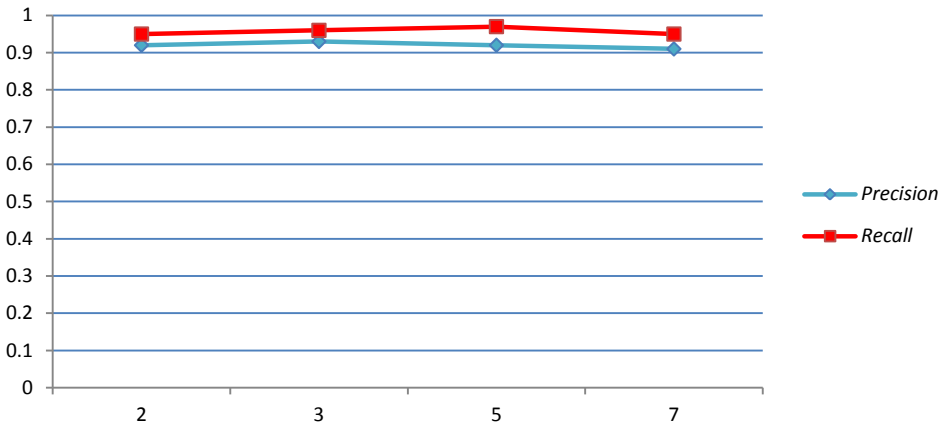


Figure II-51: Precision and Recall as a function of $\boldsymbol{m}$ *(*$\Delta = \mathbf{70}$*).*

Figure II-46 shows average values for *Precision* and *Recall* of 0.92 and 0.95, respectively. The *Precision* and *Recall* value are quite constant with respect to m, the relative variances being of of 6×10^{-5} and 9×10^{-5}, respectively.

The above fragility experiments are resumed for $\Delta = 50$ and $\Delta = 90$; this time $m = 5$, see Table II-17 and Figure II-47. With respect to the case $\Delta = 70$, $\Delta = 90$ results in a quite constant *Precision* and *Recall* values. However, $\Delta = 50$ results in important relative decrease by a factor of 0.4 and 0.29, respectively. The relative gains are computed according to (41). Hence, the fragility property also refutes the $\Delta = 50$ value (see Chapter II.3.4.1).

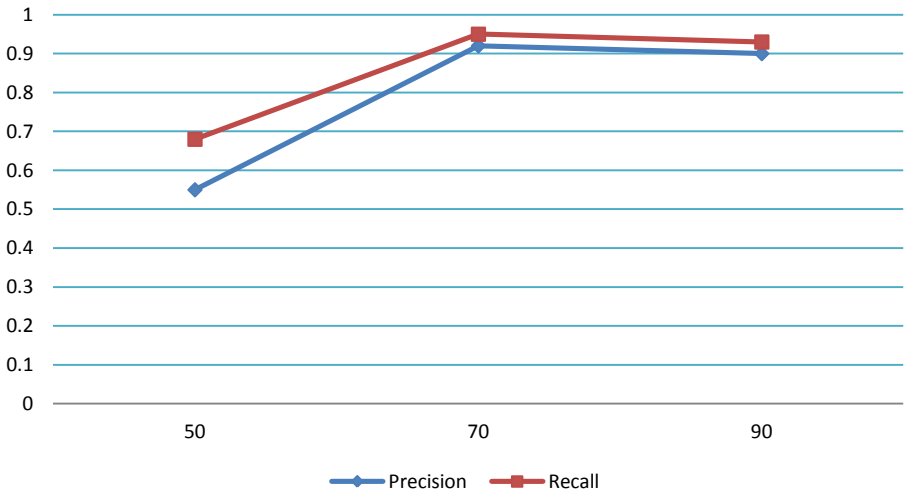


Figure II-52: Fragility as function of Δ*,* $\boldsymbol{m} = \mathbf{5}$*.*

Table II-17: Fragility behavior as function of Δ, $\boldsymbol{m = 5}$.

	Precision			Recall		
Δ	50	**70**	90	50	**70**	90
Value	0.55	**0.92**	0.9	0.68	**0.97**	0.95
Relative gain (ρ)	-0.4	**0**	-0.02	-0.29	**0**	-0.02

II.3.4.3. Transparency

The transparency of the semi-fragile watermarking method is assessed considering the same three types of metrics as in Chapter II.2.2.3.

Here again, the data-payload is set at 4.5 kbits per min. The robustness expressed by a BER = 0.1±0.03 against transcoding while the fragility is kept at *Precision* and *Recall* rates larger than 0.9. The same four m value are investigated, namely $m \in \{2, 3, 5, 7\}$.

The experimental results are reported in Figures II-48 to Figure II-50[4]. These numeric values meet the *a priori* expectancies: the transparency increases with m. This behavior is assessed by computing the η coefficient defined by (II-42), see Table II-18.

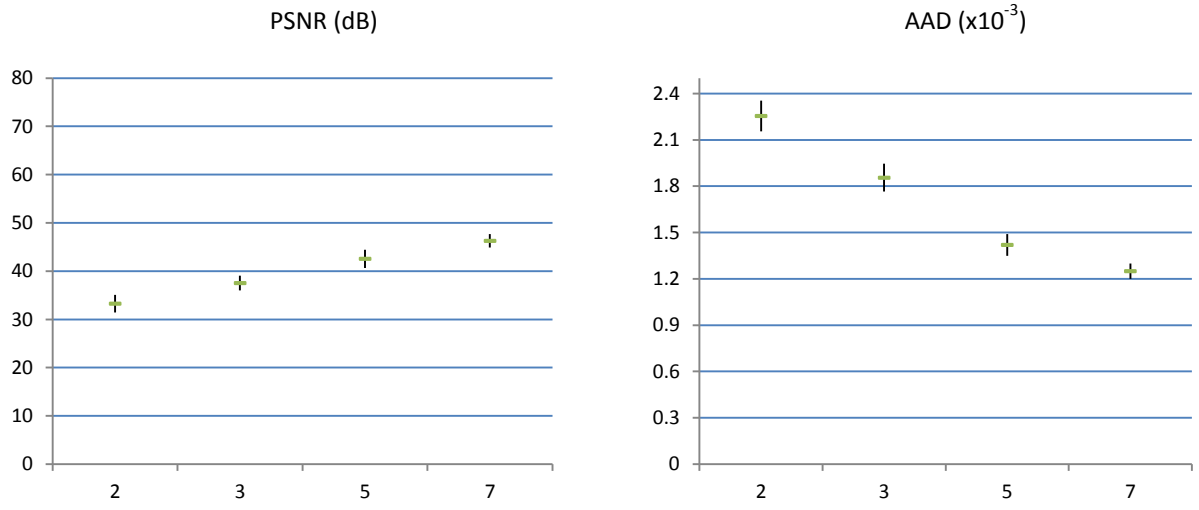


Figure II-53: PSNR and AAD as a function of $\boldsymbol{m}$ *(*$\Delta = \mathbf{70}$*): average values and 95% confidence limits.*

[4] In order to allow a synoptic comparison with robust watermarking, the same axe ranges as in Figures II-31 to II-33 are kept.

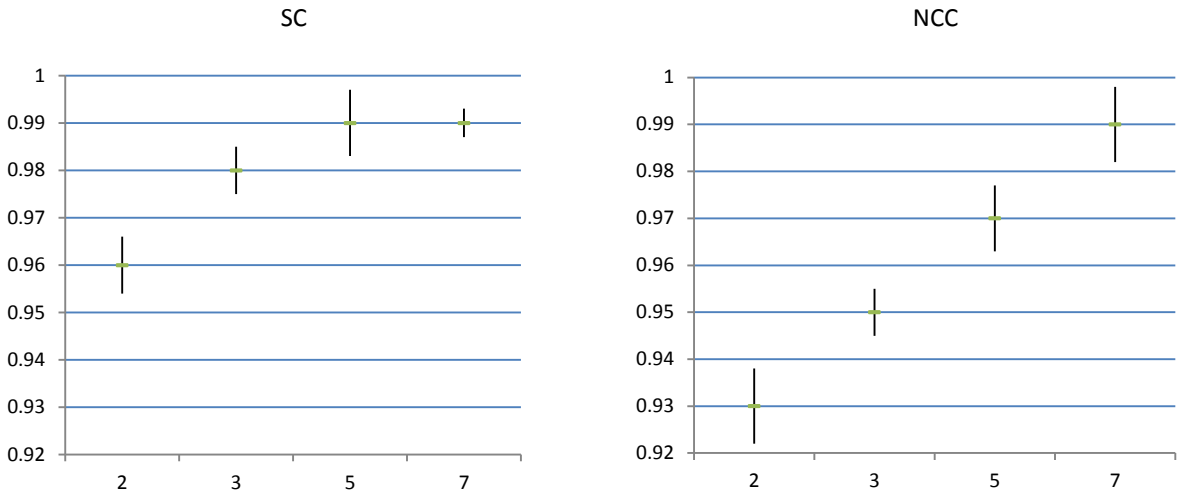


Figure II-54: SC and NCC as a function of $\boldsymbol{m}$ ($\Delta = \mathbf{70}$): average values and 95% confidence limits.

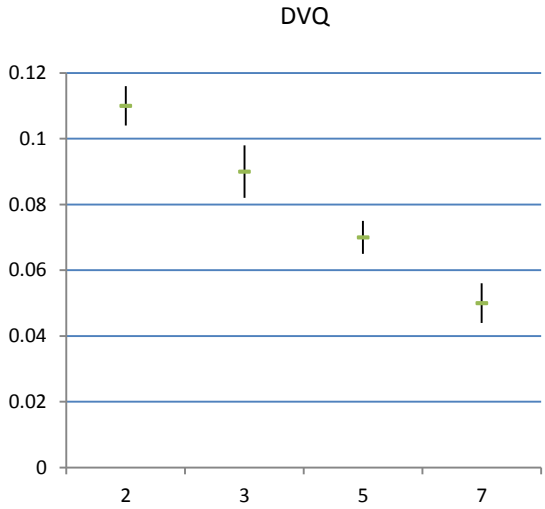


Figure II-55: DVQ as a function of $\boldsymbol{m}$ ($\Delta = \mathbf{70}$): average values and 95% confidence limits.

Table II-18: Variation of the quality metric with respect to $\boldsymbol{m}$, $\Delta = \mathbf{70}$.

	PSNR	AAD	SC	NCC	DVQ
$\boldsymbol{\eta}$	0.1	-0.5	0.01	0.06	-0.23

The transparency investigation is resumed in Table II-19 for $\Delta = 90$ and $m = 5$. Note that Chapter II.3.4.1 and Chapter II.3.4.2 demonstrate that $\Delta = 50$ is unable to meet the robustness/fragility requirements. With respect to the case $\Delta = 70$, $\Delta = 90$ results in a significantly depreciated transparency see Table II-19 where the relative gains, computed according to (II-41), are reported. Consequently, $\Delta = 90$ is no longer of practical relevance.

Table II-19: Quality metric behavior as function of Δ, $\boldsymbol{m = 5}$.

	PSNR	AAD	SC	NCC	DVQ
$\boldsymbol{\Delta = 70}$	42	1.42	0.98	0.97	0.07
$\boldsymbol{\Delta = 90}$	35	2.41	0.96	0.95	0.10
Relative gain ($\boldsymbol{\rho}$)	-0.15	0.7	-0.02	-0.02	-0.42

II.3.4.4. Computational cost

The execution time required by each operation included in the watermarking chain is evaluated on the following configuration: a PC with the following configuration: a Core4 CPU at 2.8 GHz and with 12 GB of RAM and a 500 GB HDD.

The values reported in this section are obtained on a corpus 80 min of video (SPY corpus Appendix B).

Figures II-51 and II-52 illustrate the allocation time and the time consuming during the embedding process for one second of video. It is noticed that the inner watermarking operations (selection, signature generation, and insertion) consume 0.078 s, 0.004 s and 0.009 s respectively for one second of video which present 5.58%, 0.29% and 0.64% respectively of the total embedding processing time.

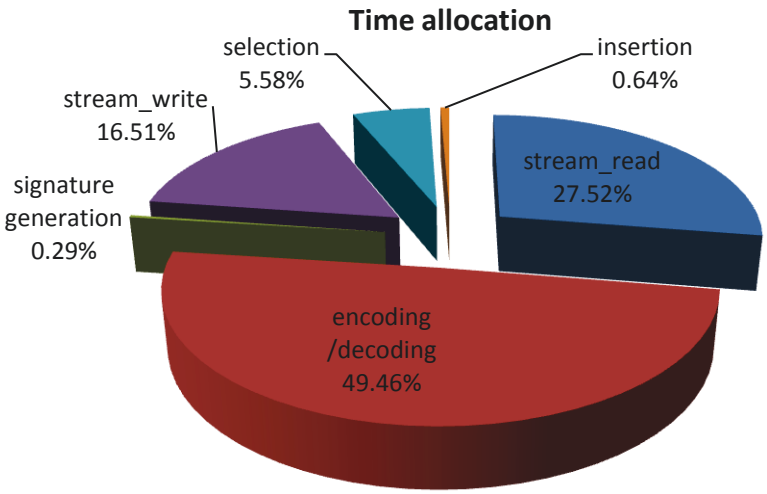


Figure II-56: Sharing time for the embedding process for one second of video.

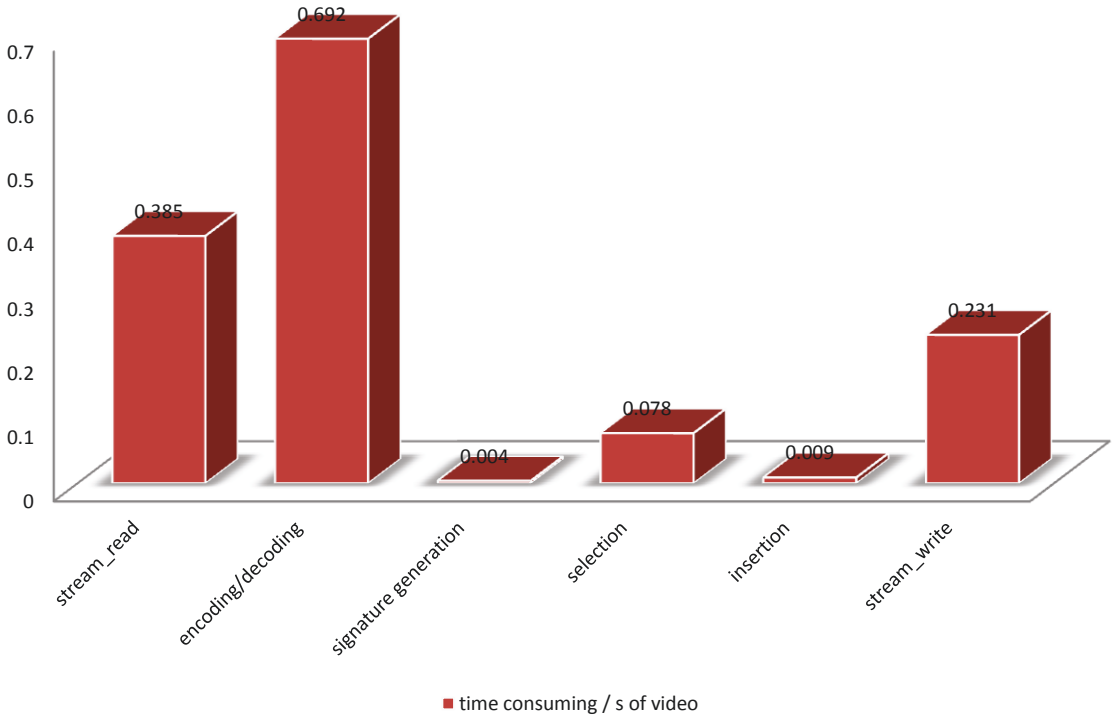


Figure II-57: Time consuming (in seconds) during the mark embedding for one second of video.

Figures II-53 and II-54 illustrate the allocation time and the time consuming during the mark detection process for one second of video. It is noticed that the signature generation and the detection consume

0.004s and 0.01s respectively for one second of video which presents 0.70 % and 1.74% respectively of the total embedding processing time.

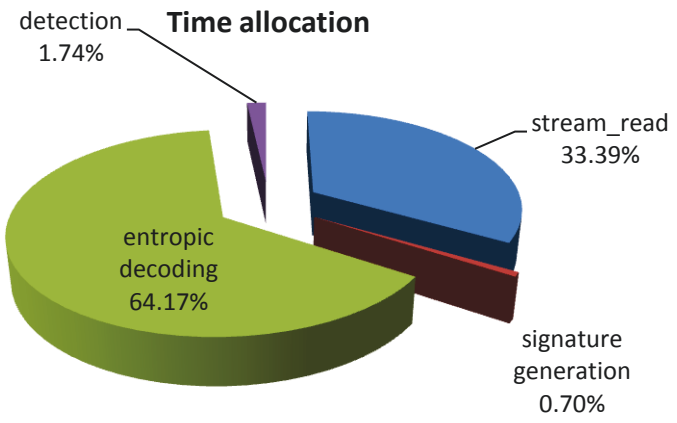


Figure II-58: Sharing time for the detection process for one second of video.

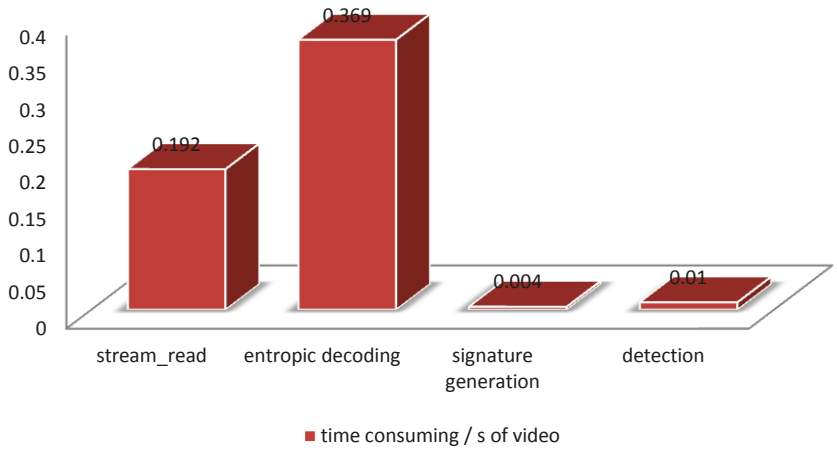


Figure II-59: Time consuming (in seconds) during the mark detection for one second of video.

The complexity analysis of the method shows that it is compatible with the real time (although its current implementation it is not):

- The operations intrinsically related to the mark generation, selection, insertion and detection are less complex (and, implicitly, faster) than the stream read/write and the MPEG-4 AVC entropic encoding/decoding; for instance, in the current implementation, for protecting 1s of video, the signature generation, selection, insertion and detection sums up to 0.105 s while the entropic decoding/encoding and the stream read/write operations reach 1.061s and 0.808s, respectively.
- The operations intrinsically related to the mark generation, selection, insertion and detection are also much faster than the entropic decoding/encoding and the video stream read/write from the hard disk. When considering the same example as above (protecting 1s of video), the entropic

decoding/encoding and the read/write operations are about 10 and 8 times slower than the mark selection/insertion/detection, respectively.

- The signature generation does not increase the complexity of the watermarking algorithm, for one second of video; it consumes 0.004 s and presents 0.29% and 0.70% of the time embedding and detection processing, respectively.

II.3.5. Conclusion

This chapter advances SPAYART, a novel video integrity verification system for MPEG-4 AVC. The advanced system makes an accurate usage of Intra modes which are extracted directly from the compressed stream to generate the authentication signature. This signature is further embedded by an *m*-QIM technique.

The new method was evaluated under the framework of a videosurveillance application; the results exhibit fragility to content replacement (with an 1/81 frame and 3s spatial and temporal accuracy, respectively) and robustness against transcoding (MPEG-4 AVC compression by a factor of 4). As both the signature extraction and mark embedding take place at the MPEG-4 AVC syntax element level, the method also features low complexity.

In order to also allow an overall comparison between our *m*-QIM framework and state of the art binary QIM approaches, we resumed the experiment for the method in [GOL07]. The robustness experiments (*cf.* Chapter II.3.4.1) brought to light average BER values of 0 and 0.25, after the additive noise and transcoding attacks, respectively. As the latter BER value, computed with a 95% relative error of 0.011 is that high, we can state that the method in [GOL07] does not meet our targeted semi-fragile watermarking requirements. Consequently, the fragility (*cf.* Chapter II.3.4.2) and transparency (*cf.* Chapter II.3.4.3) were not performed for the method in [GOL07].

To conclude with, the usefulness of the *m*-QIM technique for semi-fragile MPEG-4 AVC video watermarking was demosntrated. Such an application requires a data payload 30 times larger than in the robust watermarking case. It was experimentally demonstrated that the robustness and fragility property be properly achieved while having as main benefit an increase of the transparency by a factor of 0.1 in PSNR, 0.06 in NCC and 0.23 in DVQ. However, this application is more restrictive than the robust watermarking with respect to interval to which the Δ value belongs.

II.4. Conclusion

The study presented in Part II generalizes the QIM watermarking techniques from binary to the *m*-ary case. From the theoretical point of view, the insertion/detection rules are paired derived so as to minimize the probability of error under white Gaussian noise attacks.

While the *m*-QIM can be potentially applied to any insertion domain, our experimental study considers the quantized residual coefficients of the MPEG-4 AVC stream, and investigates the possibility of designing both robust and semi-fragile applications.

First, the robust watermarking investigation is carried out on 1 hour of video content encoded at two resolutions (SD and HD). It is thus demonstrated that:

- while keeping fixed robustness and transparency, the data payload is increased by a factor of $\log_2 m$;
- while imposing fixed transparency and data payload, the robustness is quite constant with respect to m;
- while setting a prescribed data robustness and data payload, an average relative increase by a factor of 0.1 and 0.01 of PSNR and an average relative decrease by a factor of 0.23 and 0.16 of DVQ are achieved in the SD case and HD case.

Such results are brought to light for $m \in \{2, 3, 5, 7\}$ and for a quite broad interval of Δ values, ranging from 50 to 90.

Secondly, a syntax element based authentication signature is identified by making an accurate usage of Intra prediction modes. The obtained signature and the *m*-QIM framework are jointly used to design the SPYART integrity verification system.

SPYART was evaluated under the framework of a videosurveillance application; the results exhibit fragility to content replacement (with an 1/81 frame and 3s spatial and temporal accuracy, respectively) and robustness against transcoding (MPEG-4 AVC compression by a factor of 2). As both the signature extraction and mark embedding take place at the MPEG-4 AVC syntax element level, the method also features low complexity.

According to the integrity verification context, the experiments consider a strong data payload constraint (4.5 kbits per min) imposed by the signature size. It is demonstrated that:

- while imposing a prescribed transparency, the robustness and the fragility are quite constant with respect to m;
- while keeping fixed robustness/fragility constraints, the transparency is increased by 0.1 in PSNR, 0.06 in NCC and 0.23 in DVQ.

These results are obtained at the expense of restricting the Δ parameter to a quite narrow interval centered on 70.

Note that the MPEG-4 AVC watermarking by means of *m*-QIM technique avoids complex operations by requiring just binary MPEG-4 AVC encoding/decoding and *m*-QIM quantization operations. The mark

generation, selection, insertion and detection for one second of video are 10 times and 8 times faster than the MPEG-4 AVC entropic encoding/decoding and the stream read/write operations, respectively.

Note that, although not discussed before, the impact of the inserted signature on the file size has also been evaluated as a function of the alphabet size and the quantization step.

Table II-20 illustrates the obtained average relative differences between the watermarked video and the original video sizes corresponding to the SPY corpus. Four alphabet sizes namely $m = 2$, $m = 3$, $m = 5$ and $m = 7$ and three quantization steps namely $\Delta = 50$, $\Delta = 70$ and $\Delta = 90$ have been considered; $\alpha = \alpha^* + 0.04$. The results reported in Table II-20 are computed according to:

$$\frac{Watermaked\ video\ size - Original\ video\ size}{Original\ video\ size}$$

The analysis demonstrates that:

- for a fixed quantization step, the larger the m value, the less file size impact; hence, in this respect, $m = 2$ is the worst case as it induces the largest increase in the file size;
- for a fixed value m, the larger the Δ value, the larger the impact on the file size.

However, in all investigated cases, this variation is insignificant: its relative values are of 10^{-4}.

Table II- 20 : Impact of the inserted signature on the size of the original video.

	2	**3**	**5**	**7**
$\mathbf{\Delta = 50}$	$6.2\ 10^{-4}$	$4.8\ 10^{-4}$	$4.1\ 10^{-4}$	$3.5\ 10^{-4}$
$\mathbf{\Delta = 70}$	$7.6\ 10^{-4}$	$5.8\ 10^{-4}$	$5.5\ 10^{-4}$	$4.9\ 10^{-4}$
$\mathbf{\Delta = 90}$	$8.9\ 10^{-4}$	$8.1\ 10^{-4}$	$7.3\ 10^{-4}$	$6.5\ 10^{-4}$

These overall results demonstrate that the *m*-QIM watermarking technique outperforms the state-of-the art studies presented in Chapters I.2.1 and I.2.2. Hence, we can state that *m*-QIM technique is an unitary theoretical framework jointly reaching functional equilibrium for both robust and semi-fragile watermarking applications.

Part III: Drift-free compressed domain watermarking

Abstract

Our study aims at avoiding the intra-frame (spatial) drift effect for MPEG-4 AVC watermarking. First, by considering the analytic expressions of the MPEG-4 AVC encoding operations, it algebraically models the drift distortion spread as an optimization problem. Second, it solves this problem under drift-free constraints. Finally, the advanced solution is adapted to take into account the watermarking restrictions. The experiments consider an *m*-QIM semi-fragile watermarking method.

III.1. Problem statement

Intra-frame (spatial) drift is a major concern for all types of MPEG-4 AVC compressed-domain video processing applications: due to the MPEG-4 AVC intra prediction paradigm, the modification of one block generally results in the alteration of the neighboring blocks. Moreover, such an effect can be propagated several times (according to the visual content, encoding configuration, etc), in an uncontrolled way. This drift effect is of particular importance in watermarking applications, where it can depreciate the performances of all the three watermarking properties: data payload, transparency and robustness.

In this context, several research works have been proposed in order to avoid intra drift distortion in MPEG-4 AVC watermarking.

Ma *et al.* [MA09] suggested the use of DCT coefficients for embedding the mark and compensated the intra frame distortion by adjusting the DCT coefficients already altered by the drift mechanism. The experimental results show that the advanced scheme can ameliorate the visual quality of the watermarked video. However, the intra frame drift distortion compensation requires additional computational operations to be integrated to the mark embedding process.

In their subsequent study [MA10], Ma *et al.* employed the directions of intra prediction modes to define a block set able to avert the drift distortion. The method exploits several paired-coefficients of 4X4 DCT residual block to embed the mark. The experimental results show that the proposed method enhances the visual quality compared to methods that have not considered the intra-frame distortion drift. Nerveless, intra prediction mode direction based selection may reduce the number of the watermarking candidate blocks and consequently the watermarking data-payload.

A controllable error-drift compensation scheme is advanced by Huo, Zhu and Chen [HUO11]. In order to reduce the compensation computational complexity, the advanced scheme considers that the error propagation is caused by different DCT coefficients but only some of them need to be compensated. Achieved results demonstrated that the advanced scheme reduce the compensation cost while keeping the same performances in terms of data-payload and transparency compared to the method proposed by Gong and Lu [GON08], where only the DC coefficients are involved.

A compensation method proposed by Zhang *et al.* [ZHA10] eliminates the intra frame drift distortion by considering the difference between the original and watermarked reconstructed samples and further compensate it over all the DCT coefficients. The experimental results show that the proposed scheme achieves a PSNR (Peak Signal to Noise Ratio) average of 38 dB with little increase in the bit rate. However the compensation procedure increases the computational complexity of the watermarking algorithm.

To conclude with, in order to avoid the drift drawbacks, two classes of solutions are currently considered in the literature, see Figure III-1: compensation and selection-based. The former consists in inserting the mark in the compressed domain, estimating the sub-sequent drift distortions and in compensating them by decoding/re-encoding operations. This way, the side effects can be completely avoided at the expense of increasing the computational complexity and of modifying the compressed stream parameters (hence, of implicitly adding some involuntary attacks). The latter consists in restricting the insertion domain: only the blocks which are not involved in the prediction process can be considered for

the mark insertion. This way, no drift effects occur, the computational complexity is kept constant but the data payload is drastically reduced; moreover, the security of the system is also implicitly reduced, making it easier for a brute force attack to be performed.

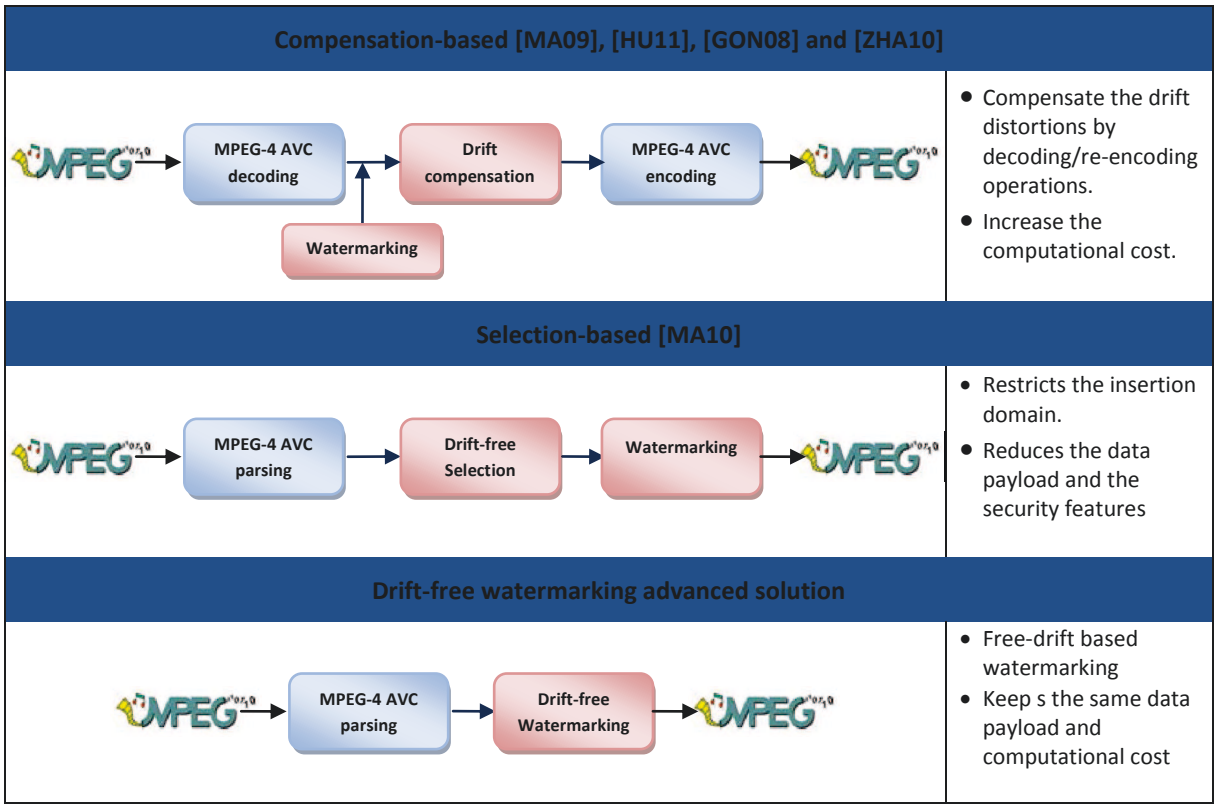


Figure III-1: Drift distortion avoiding solutions.

In order to achieve drift-free watermarking insertion, while keeping the same data payload, robustness, computational complexity, and security constraints, the solution presented in this chapter follows a different approach, see Figure III-1. It algebraically models the drift behavior and allows the insertion procedure to be customized so as to get rid of this undesired effect.

III.2. Theoretical contribution: Algebraic-based drift cancelation

When trying to avoid the drift distortions for MPEG-4 AVC watermarking, a two-folded difficulty is encountered. On the one hand, the mark is inserted in the compressed domain; in this respect, several studies demonstrated that the quantized DCT-transformed prediction residuals ensure an optimal trade-off among the watermarking functional properties. On the other hand, the prediction modes are established and the prediction itself is performed in the pixel domain. Hence, the dual compressed-uncompressed representations should be considered when eliminating the drift undesired effects.

In order to avoid any decoding – reencoding operation for each watermarked block and for all of its neighbors, our study starts by investigating the analytic expressions of the MPEG-4 AVC encoding operations. This will led to algebraically formulate the drift problem and subsequently solve it under the drift-free constraints.

III.2.1. Algebraic models for intra-frame prediction

I frames are encoded according to Intra prediction modes which exploit the spatial redundancy to enhance the compression efficiency. The MPEG-4 AVC standard features 13 prediction modes; despite their peculiarities, all of them act in the pixel domain and compute the predicted blocks based only on the most right column and bottom row of the reference blocks.

For each current block, the prediction mode minimizing the rate-distortion cost is selected and the predicted block is constructed from the boundary pixels of the neighboring blocks which are previously encoded.

Figure III-2 illustrates the intra prediction process for 4x4 block. For more details, we can refer to Appendix A.

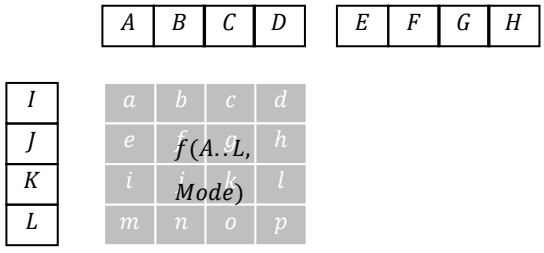


Figure III-2: Intra-frame prediction process.

As illustrated in Figure III-2, the predicted samples (from a to p) of the current block are computed from the adjacent samples (from A to L) of the neighboring blocks using a prediction formula f according to a selected prediction mode.

The residual block computing the difference between the current block and the predicted block is further transformed by using a DCT and a quantizer. Each quantized transformed residual block is further mapped into a 16 coefficients vector by a zig-zag scanning and encoded using an entropic encoding.

III.2.2. Intra frame drift elimination

Since the watermarking takes place in the MPEG-4 AVC compressed domain, the distortion is induced in the 4×4 residual block of quantized coefficients. We suppose that a distortion vector $\widetilde{W}$ is induced into a given quantized transformed residual block $\widetilde{R}$. As any distortion can be modeled by additive operation, the altered block $\widetilde{R}_w$ is obtained according to (III-1):

$$\widetilde{R}_w = \widetilde{R} + \widetilde{W} \tag{III-1}$$

At the decoder side, after applying the inverse DCT, the altered quantized residual block R_w is obtained according to (III-2):

$$R_w = IDCT(\widetilde{R} + \widetilde{W}) \tag{III-2}$$

where $IDCT$ is the MPEG-4 AVC inverse DCT transformation.

The linearity of the $IDCT$ function leads to:

$$R_w = R + W \tag{III-3}$$

where R is the residual block and W presents the pixel domain mark.

The predicted block P is computed according Chapter III.2.1 and added to the watermarked residual block to obtain the watermarked pixel domain block X_w:

$$\begin{aligned} X_w &= R + W + P \\ &= X + W \end{aligned} \tag{III-4}$$

X and W can be expressed in matrix from according to (III-5):

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}, W = \begin{bmatrix} W_{00} & W_{01} & W_{02} & W_{03} \\ W_{10} & W_{11} & W_{12} & W_{13} \\ W_{20} & W_{21} & W_{22} & W_{23} \\ W_{30} & W_{31} & W_{32} & W_{33} \end{bmatrix} \tag{III-5}$$

Figure III-3 illustrates the intra frame drift distortions principal. The drift distortion occurs when a given altered pixel block is further used to compute the prediction block for some neighboring blocks and subsequently propagates the distortion to these blocks.

For a given current block $B_{i,j}$, the set of the neighboring blocks that may be use this block in the prediction are defined as following (see Figure III-3):

- The *right-block* ($B_{i,j+1}$): the block on the right of the current block;

- The *under-block* ($B_{i+1,j}$): the block under the current block;
- The *under-left-block* ($B_{i+1,j-1}$): the block on the left of the *under-block*;
- The *under-right-block* ($B_{i+1,j+1}$): the block on the right of the *under-block*.

For instance, given an altered block $B_{i,j}$ (see Figure III-3), its most-right column samples that are marked in red in Figure III-3 may be used to predict the $B_{i,j+1}$ and $B_{i+1,j+1}$ block samples and its bottom line samples that are marked in green in Figure III-3 may be used to predict the $B_{i+1,j-1}$, $B_{i+1,j}$ and $B_{i+1,j+1}$ block samples, according to the MPEG-4 AVC intra-frame prediction principle (see Figure III-1). Like this, the distortion will spread to affect other neighboring blocks, see Figure III-4.

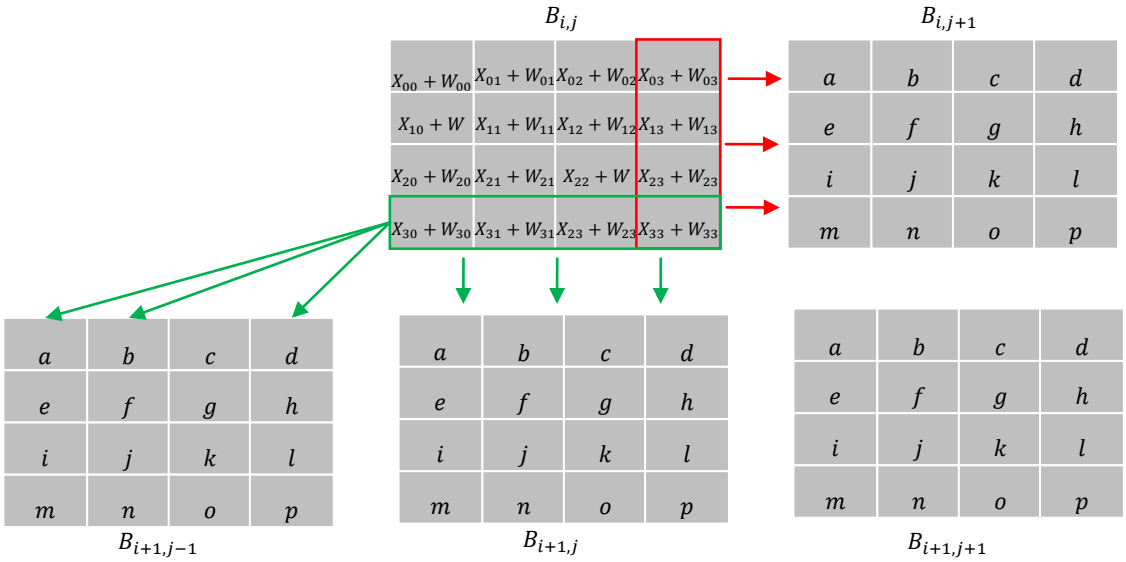


Figure III-3: Intra-frame drift principle.

In this sense, drift based distortion spread takes place once the pixel sample values at the bottom line and the most right column of the altered block are deviated from their original pixel values.

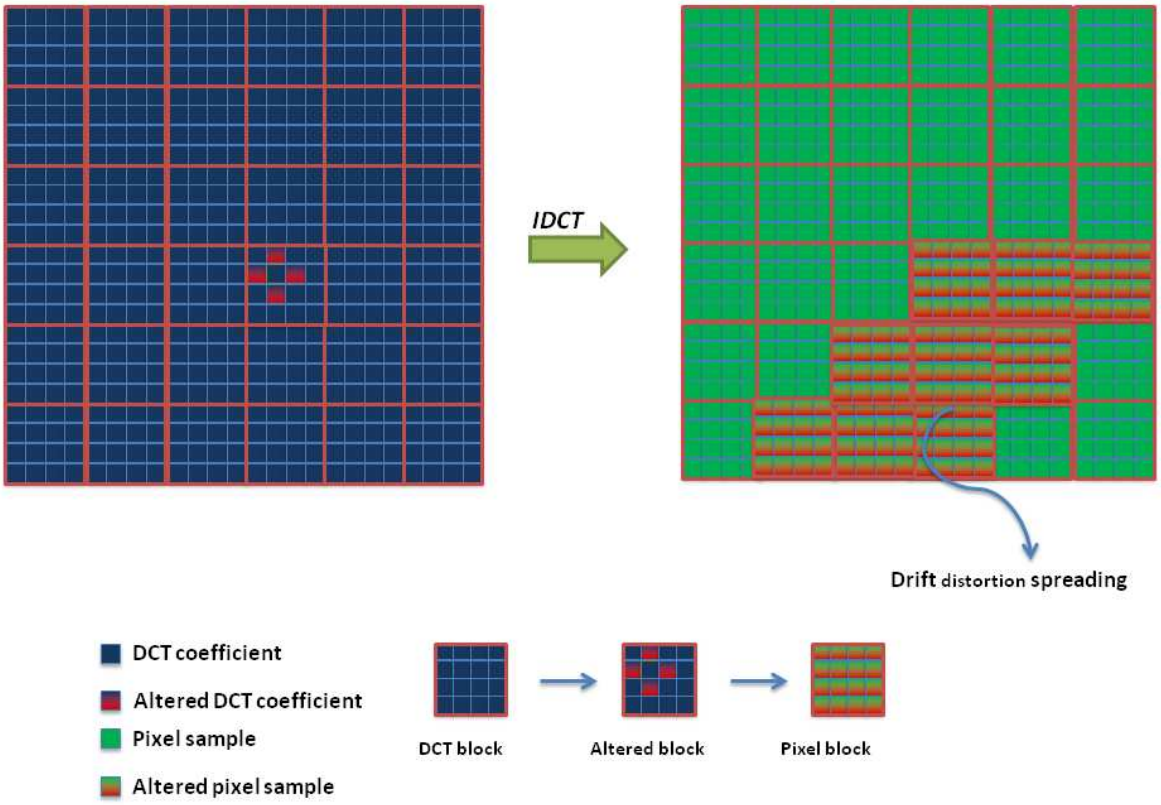


Figure III-4: Drift distorsion propagation.

MPEG-4 AVC has 13 predictions modes: 9 $I4 \times 4$ intra prediction modes devoted for textured areas and 4 $I16 \times 16$ prediction modes devoted for smoothed areas. Despite their peculiarities, all these predictions modes act only on the same specific samples of the pixel domain.

Consequently, to prevent drift distortion, the bottom line and most-right column pixel values of the altered block should not be modified. Hence, we shall achieve drift-free insertion by constraining the insertion procedure to not change the most right column and bottom row of the host block in the pixel domain.

Consequently, the block distortion system considering the drift-free condition in the compressed domain can be formulated according to (III-6):

$$\begin{cases} \tilde{R}_w = \tilde{R} + \tilde{W} \\ IDCT(\tilde{W}) = \begin{bmatrix} W_{00} & W_{01} & W_{02} & 0 \\ W_{10} & W_{11} & W_{12} & 0 \\ W_{20} & W_{21} & W_{22} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{cases} \quad \text{(III-6)}$$

For all W belonging to the pixel domain solutions space (denoted by S_p) of the system described by the equation (III-6), W may be expressed as matrix multiplication (III-7):

$$W \in S_p \quad Then \quad W = E * W_1 * E$$

$$where\ W_1 \in M_{4\times4}\ and\ E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{III-7}$$

As the distortion is performed in the compressed domain, in order to avoid any increase of the computational complexity, the compressed domain solution space (denote by further S_C) should be defined.

Equations (III-6) and (III-7) imply:

$$S_c = \{\widetilde{W} \quad / \quad \widetilde{W} = DCT(E * W_1 * E)\} \tag{III-8}$$

Knowing that DCT transformation is defined as:

$$DCT(X) = A^T * X * A$$
$$where\ A^T = A^{-1}$$

$\widetilde{W}$ can be expressed:

$$\begin{aligned} \widetilde{W} &= A^T * E * W_1 * E * A \\ &= A^T * E * A * A^T * W_1 * A * A^T * E * A \\ &= DCT(W_1) * DCT(W_1) * DCT(E) \\ &= \tilde{E} * \widetilde{W}_1 * \tilde{E} \end{aligned}$$

Therefore, the compressed domain solution space S_c can be defined according to (III-9):

$$S_c = \left\{ \widetilde{W} \quad / \quad \widetilde{W} = \tilde{E} * \widetilde{W}_1 * \tilde{E} \qquad where\ E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} and\ \widetilde{W}_1 \in M_{4\times4} \right\} \tag{III-9}$$

In other words, in order to prevent the drift problem, the distortion induced in the compressed domain at the 4×4 block should be before and after multiplied by the Drift-free shaped mask $M_d = \tilde{E}$ before being induced to the host quantized transformed residual block.

The MPEG-4 AVC standard way of computing the integer DCT transformation is given by (III-10):

$$Y = C_f * X * {C_f}^T \quad E_f \tag{III-10}$$

$$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, E_f = \begin{bmatrix} 0.25 & 0.158 & 0.25 & 0.158 \\ 0.158 & 0.1 & 0.158 & 0.1 \\ 0.25 & 0.158 & 0.25 & 0.158 \\ 0.158 & 0.1 & 0.158 & 0.1 \end{bmatrix}$$

Where:

- X is an original 4x4 matrix, while Y is the transform matrix,
- " $*$ " is the matrix product,
- " " is the element-wise product (Hadamard product).

Thus, the numerical expression of the M_d can be obtained:

$$M_d = C_f * E * {C_f}^T \quad E_f = \begin{bmatrix} 0.75 & 0.3162 & -0.25 & 0.1581 \\ 0.3162 & 0.60 & 0.3162 & -0.20 \\ -0.25 & 0.3162 & 0.75 & 0.1581 \\ 0.1581 & -0.20 & 0.158 & 0.90 \end{bmatrix} \quad \text{(III-11)}$$

By combining (III-9) and (III-11), the compressed domain space solutions S_c can be expressed, (III-12):

$$S_c = \left\{ \widetilde{W} / \ \widetilde{W} = M_d * \widetilde{W}_1 * M_d, \ M_d = \begin{bmatrix} 0.75 & 0.3162 & -0.25 & 0.1581 \\ 0.3162 & 0.60 & 0.3162 & -0.20 \\ -0.25 & 0.3162 & 0.75 & 0.1581 \\ 0.1581 & -0.20 & 0.158 & 0.90 \end{bmatrix}, \ \widetilde{W}_1 \in M_{4\times4} \right\} \quad \text{(III-12)}$$

This generic solution to avoid the intra-frame drift distortion will be considered in the following section and adapted to take into account the restrictions required by watermarking embedding methods for MPEG-4 AVC compressed domain.

III.2.3. Drift-free for watermarking

Commonly, in order to enhance the transparency feature of the watermarked MPEG-4 AVC videos, mark embedding procedure takes into account tow main restrictions. The first restriction is related to the human visual perception and is considered by the deployment of a perceptual mask. The second restriction is set by the DCT transformation properties and is considered by not involving the DC coefficient in the watermarking.

III.2.3.1. Perceptual masking

Watermarking techniques deploy a perceptual mask based on the human visual perception model to enhance the transparency of the watermarked MPEG-4 AVC videos. Beside the human visual perception model, the exploited mask does not take into account the drift distortion problem. Our purpose is to adapt the advanced Drift-free shaped mask (*cf.* Chapter III.2.2) to consider the human perceptual model.

The perceptual mask M_p (III-13) was obtained by first sub-sampling the Noorkami [NOO05] matrix and further adapted to take into consideration the amendments introduced in MPEG-4 AVC integer DCT transformation.

$$M_p = \begin{bmatrix} 0.7 & 0.410 & 1.20 & 2.429 \\ 0.410 & 0.827 & 1.180 & 1.847 \\ 1.20 & 1.180 & 3.07 & 4.447 \\ 2.429 & 1.847 & 4.447 & 7.539 \end{bmatrix} \quad \text{(III-13)}$$

A value in the M_p matrix represents the visibility threshold, *i.e.* the maximal value of a distortion added on a pixel (classical) DCT coefficient which is still transparent (imperceptible) for a human observer.

The embedded mark $\widetilde{W}$ is computed according to (III-14):

$$\widetilde{W} = w \frac{M_p}{\|M_p\|} \quad \text{(III-14)}$$

where w is a scalar value and $\|M_p\|$ is the norm of M_p.

As $\frac{w}{\|M_p\|}$ is a scalar and M_p is belonging to $M_{4\times4}$, the multiplication of W by M_d belongs to S_c.

Like this, the new perceptual $\widetilde{M}_{pd}$ mask avoiding the drift distortion can be obtained (III-15):

$$\widetilde{M}_{pd} = M_d * M_p * M_d = \begin{bmatrix} 0.823 & 0.458 & 1.294 & 2.432 \\ 0.458 & 0.506 & 1.266 & 1.713 \\ 1.294 & 1.180 & 3.134 & 4.469 \\ 2.432 & 1.713 & 4.469 & 7.485 \end{bmatrix} \quad \text{(III-15)}$$

III.2.3.2. DC coefficient

In the compressed domain watermarking embedding case, another constraint should be taken into account. In fact, only the 15 AC coefficients are commonly involved in the mark embedding. Consequently, the induced mark may be modeled as follow:

$$\widetilde{W} = \frac{w}{\|\widetilde{M}_{pd}\|} \begin{bmatrix} 0 & 0.458 & 1.294 & 2.432 \\ 0.458 & 0.506 & 1.266 & 1.713 \\ 1.294 & 1.180 & 3.134 & 4.469 \\ 2.432 & 1.713 & 4.469 & 7.485 \end{bmatrix}$$

While applying the $IDCT$ to the inserted mark W, we obtain:

$$W = \frac{w}{\|\widetilde{M}_{pd}\|} * \begin{bmatrix} 6.340 & -5.180 & 2.458 & -0.206 \\ -5.180 & 3.528 & -2.223 & -0.206 \\ 2.458 & -2.223 & 1.463 & -0.206 \\ -0.206 & -0.206 & -0.206 & -0.206 \end{bmatrix}$$

Here, we note that the obtained mark no longer belongs to the pixel domain solutions space S_p, as its resulting pixel samples of the most right column and the bottom row are not null.

In order to handle this additional constraint a new equations system is formulated (III-16):

$$\begin{cases} IDCT(\widetilde{M}_{pd}) \in S_p \\ \widetilde{M}_{pd_{00}} = 0 \end{cases} \quad \text{(III-16)}$$

The setting of $\widetilde{M}_{pd_{00}}$ to zero can be achieved if the sum of the M_{apd} pixel domain samples is equal to zero. This condition can be met by computing an adapted drift free perceptual mask M_{apd} according to (III-17):

$$M_{apd} = M_{pd} - \frac{\begin{pmatrix} M_{pd} & M_{pd} \end{pmatrix}}{\|M_{pd}\|} * \sum_{i=1}^{4}\sum_{j=1}^{4} M_{pd_{i,j}} \quad \text{(III-17)}$$

where:

$$M_{pd} = IDCT(\widetilde{M}_{pd}) = \begin{bmatrix} 6.546 & -4.974 & 2.664 & 0 \\ -4.974 & 3.734 & -2.017 & 0 \\ 2.664 & -2.017 & 1.669 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where $IDCT(X) = C_i^T(X \quad E_i)C_i$ and:

$$C_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0.5 & -0.5 & -1 \\ 1 & -1 & -1 & 1 \\ 0.5 & -1 & 1 & -0.5 \end{bmatrix} \text{ and } E_i = \begin{bmatrix} 0.25 & 0.3162 & 0.25 & 0.3162 \\ 0.3162 & 0.4 & 0.3162 & 0.4 \\ 0.25 & 0.3162 & 0.25 & 0.3162 \\ 0.3162 & 0.4 & 0.3162 & 0.4 \end{bmatrix}$$

We obtain:

$$M_{apd} = \begin{bmatrix} 5.473 & -5.595 & 2.486 & 0 \\ -5.595 & 3.385 & -2.119 & 0 \\ 2.486 & -2.119 & 1.599 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{(III-18)}$$

When applying the DCT to M_{apd}, we obtain:

$$\widetilde{M}_{apd} = \begin{bmatrix} 0 & -0.248 & 1.181 & 2.363 \\ -0.248 & -0.121 & 1.153 & 1.674 \\ 1.181 & 1.153 & 3.108 & 4.476 \\ 2.364 & 1.674 & 4.476 & 7.468 \end{bmatrix} \quad \text{(III-19)}$$

Equations (III-18) and (III-19) show that the adapted mask $\widetilde{M}_{apd}$can meet the condition imposed by the system (III-16). Like this, $\widetilde{M}_{apd}$ adapts the drift-free shaping mask to consider the human visual perception model and the non-involvement of the DC coefficient during the mark embedding.

III.3. Case study: Drift-free *m*-QIM semi-fragile watermarking

In this chapter, the *m*-QIM (multiple-symbols Quantizing Index Modulation) insertion method (*cf.* chapter II.2) is adapted by considering advanced perceptual drift-free solution. The experiments are first devoted to the evaluation of the impact of the new masking model in the *m*-QIM watermarking transparency. Then, the performances of the advanced drift-free solution will be compared to the selection based method proposed in [MA10] which features the same complexity level as our advanced solution compared to the other state of the art method.

III.3.1. Advanced method

The *m*-QIM method inserts the mark in the AC quantized residual coefficients of intra frame 4×4 luma blocks, hence the host vector stands for 15 component vector obtained by the zig-zag scanning such a block. The embedding process combines multi symbol quantization indexing and perceptual drift-free shaping:

The watermark is not directly embedded into the original signal X but into the projection x' of X onto the perceptual drift-free mask M_{apd}.

Figure III-5 illustrates the flowchart of the embedding watermarking method.

For each supposed marked block Z, the detector starts by projecting the vector obtained by zig-zag scanning the 15 AC coefficients of 4×4 *I* frame blocks onto the perceptual drift-free mask M_{apd}. Then, the resulting scalar is used to compute a detection variable D.

The corresponding detection is given by locating the decision interval in which belongs the detection variable D, as expressed in (III-20):

$$\begin{cases} D = Q_{\Delta}\left(Z^{T}\dfrac{M_{apd}}{\|M_{apd}\|} - k\Delta\right) - Z^{T}\dfrac{M_{apd}}{\|M_{apd}\|} + k\Delta \\ \dfrac{\Delta((\alpha-1)m-2d)}{2m} < D \leq \dfrac{\Delta((1-\alpha)m-2d)}{2m} \rightarrow \hat{d} = d \end{cases} \quad \text{(III-20)}$$

where Z is the received block, m presents the alphabet size and $\hat{d}$ is the detected symbol.

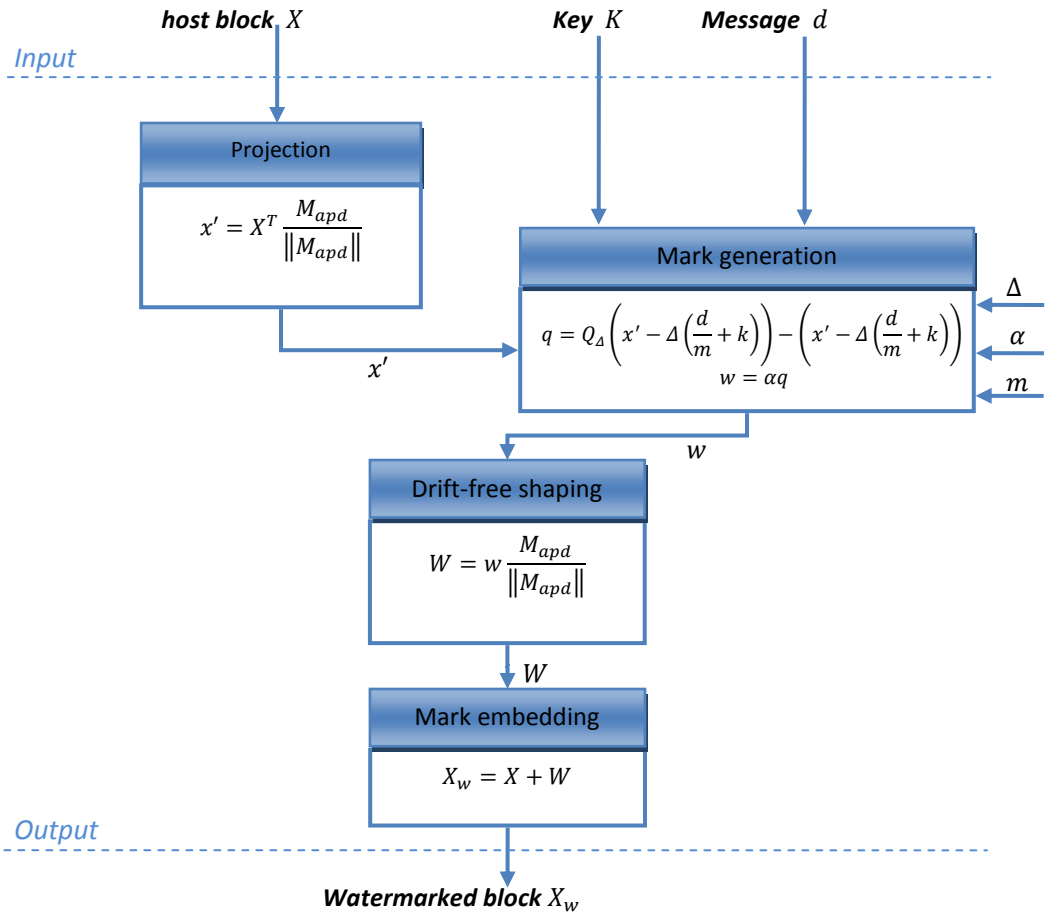


Figure III-5: The embedding synopsis: three inputs (the message $\mathbf{d}$, the host block $\mathbf{X}$ and the key $\mathbf{k}$) and four parameters (the perceptual drift-free mask $\mathbf{M_{apd}}$, the quantization step Δ and the alphabet size $\mathbf{m}$) are considered

III.3.2. Experimental evaluations

III.3.2.1. Corpus

The experiments were carried out on a video corpus composed of 8 sequences of about 10 minutes each, downloaded from internet [WEB01] or recorded under the framework of the SPY project, *cf Appendix B*. Their content is heterogeneous, combining city streets, highways, industrial objectives, shopping centers, *ect*.

This corpus is encoded in MPEG-4 AVC in Baseline Profile (no B frames, CAVLC entropy encoder) at 512 kbps, 640x480 pixel frames; the GOP size is set to 8.

III.3.2.2. Performances evaluation

The watermarking method presented in Chapter III.3.1 was applied by considering two perceptual masks: (1) the perceptual mask M_p expressed by (III-13) and (2) the perceptual mask considering the intra-frame drift M_{apd} expressed by (III-20). The experiment was devoted to the evaluation of the impact of the new masking model in the watermarking transparency. In this respect, a fixed data payload (100 bit/s), robustness (BER < 0.1 against transcoding at 50% in stream size) and fragility (frame modification detection with accuracies of 1/81 from the frame size and 3s) are imposed.

The corresponding transparency was evaluated according to three types of metrics (*cf.* Chapter I.1)

Figure III-6 to Figure III-12 reports the results for PSNR, AAD, IF, SC, NCC, and DVQ, respectively for the two masks: (1) the perceptual mask and (2) the perceptual drift-free mask. The analysis of these plots shows a gain in the transparency feature when the perceptual drift-free mask is deployed. This gain is expressed by average gains of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ.

We note that the *m*-QIM parameters are fixed at $m = 5$, $\alpha = 0.84$ and $\Delta = 70$.

Figure III-6: PSNR results: (1) perceptual mask is used and (2) the perceptual drift-free mask is used.

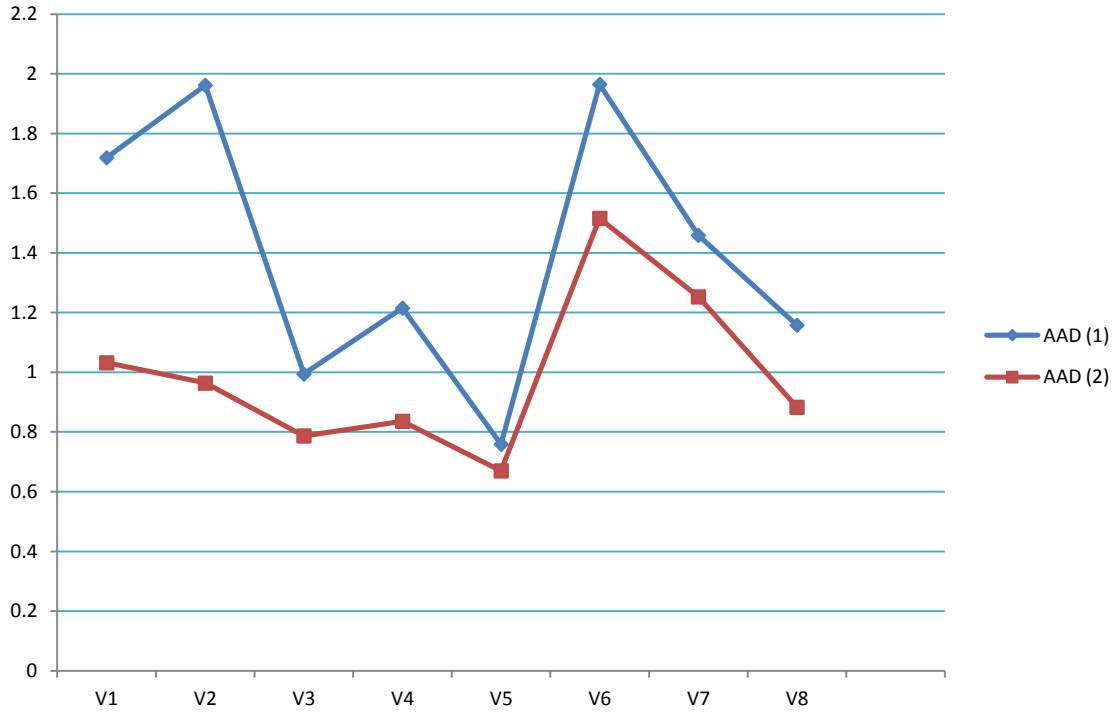


Figure III-7: AAD results: (1) perceptual mask is used and (2) the perceptual drift-free mask is used.

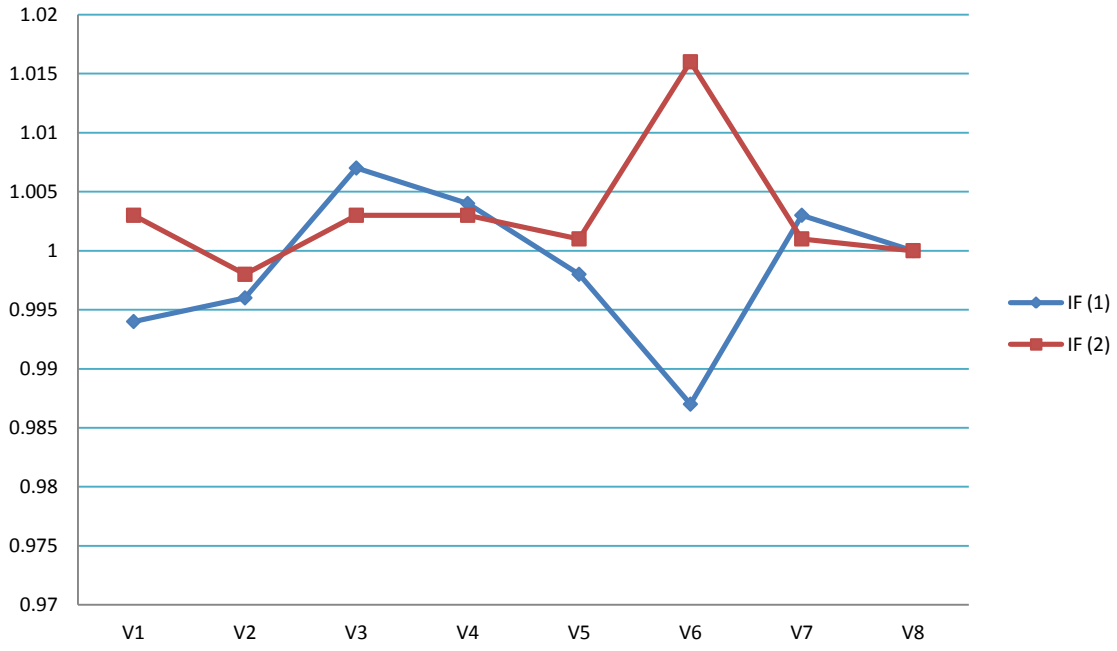


Figure III-8: IF results: (1) perceptual mask is used and (2) the perceptual drift-free mask is used.

Figure III-9: SC results: (1) perceptual mask is used and (2) the perceptual drift-free mask is used.

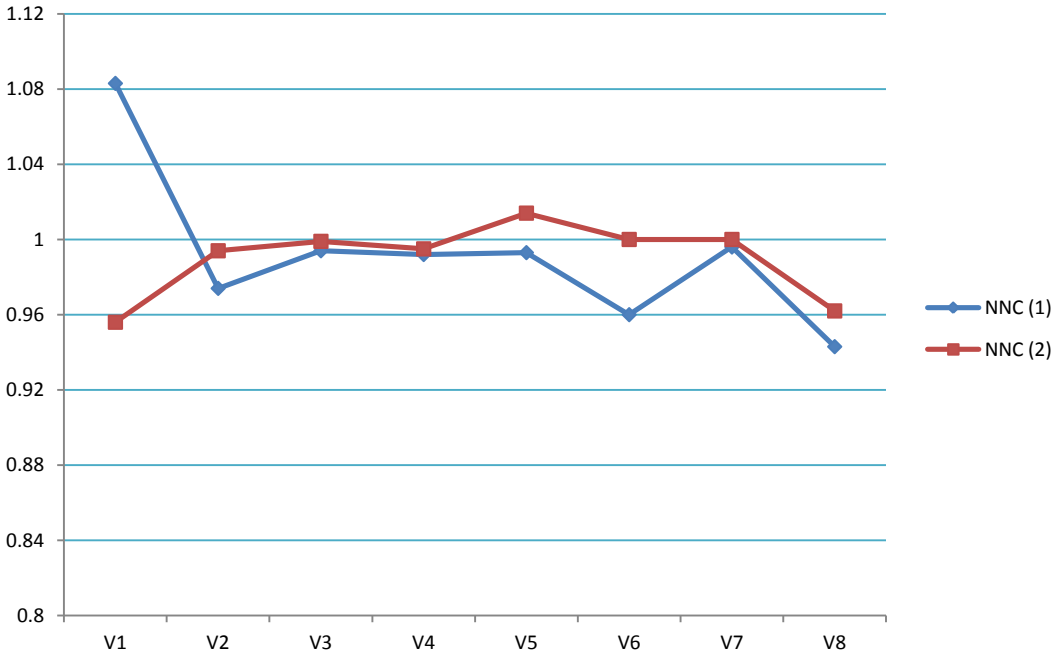


Figure III-10: NCC results: (1) perceptual mask is used and (2) the perceptual drift-free mask is used.

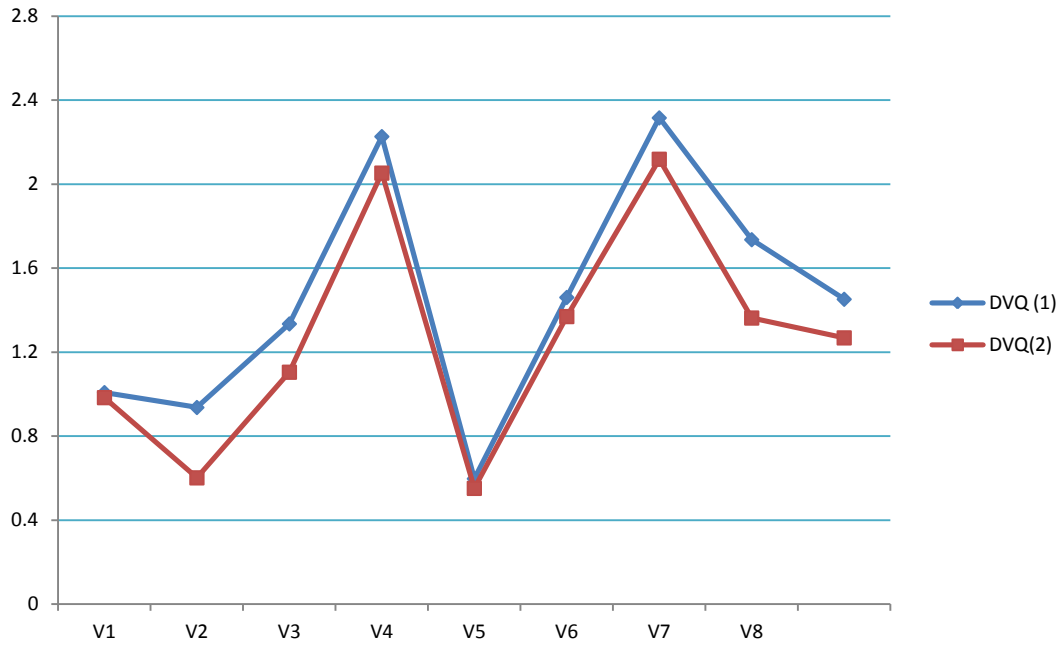


Figure III-11: DVQ results: (1) perceptual mask is used and (2) the perceptual drift-free mask is used.

In order to evaluate the performances of the advanced drift-free shaping solution, we also implemented a selection-based state of the art method, featuring the same level of complexity as our method, namely [MA10]. According to [MA10], only blocks with specific characteristics are selected to embed the mark. Therefore, the current selected block for watermarking in [MA10] must meet three conditions as illustrated in Figure III-12:

1) The *right-block* prediction mode belongs to $\{0,3,7\}_{I4\times4}$ or to $\{0\}_{I16\times16}$ (*cf.* Appendix A).
2) The *under-left-block* prediction mode belongs to $\{0,1,2,4,5,6,8\}_{I4\times4}$ or to $\{0,1,2,3\}_{I16\times16}$ and the *under-block* prediction mode belongs to $\{1,8\}_{I4\times4}$ or to $\{1\}_{I16\times16}$.
3) The under-right-block prediction mode belongs to $\{0,1,2,3,7,8\}_{I4\times4}$ or to $\{0,1,2,3\}_{I16\times16}$.

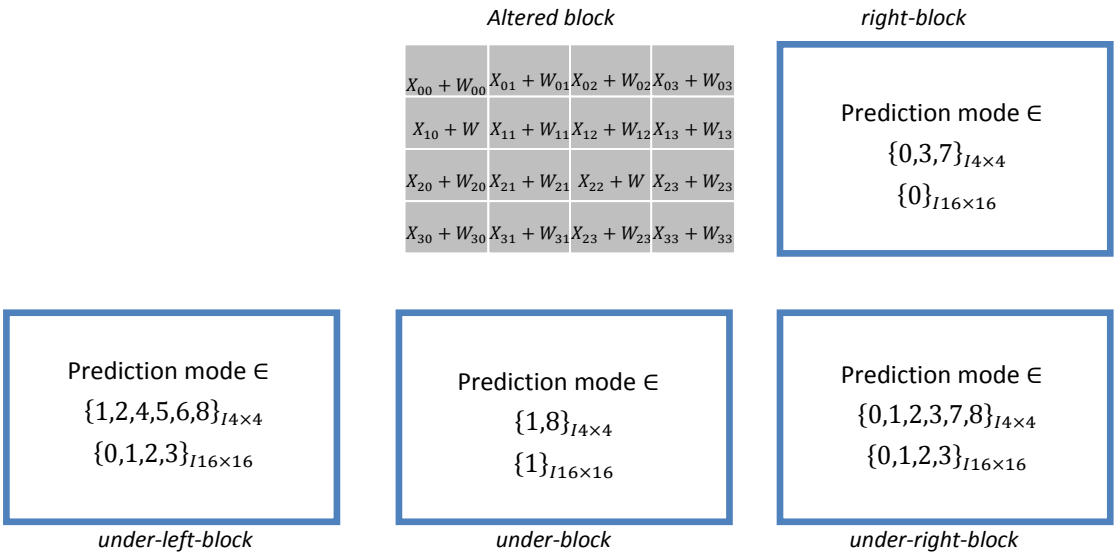


Figure III-12: Drift prevention conditions in [MA10].

Table III-1 compares the performances of the proposed perceptual drift-free shaping solution in terms of transparency and robustness/fragility with those of the selective based method in [MA10] and with those obtained in the case where the drift-free is not considered.

Table III-1: Performances evaluation.

	Transparency			Robustness		Fragility	
Method	**PSNR**	**DVQ**	**SC**	**Transcoding (50%)**	**Additive noise**	**Recall**	**Precision**
Perceptual shaping	41.7	1.44	0.97	7%	0%	0.91	0.92
Drift-free shaping	44	1.26	0.99	8%	0%	0.89	0.91
Selective [MA10]	46	1.05	0.99	40%	0%	0.07	0.05

The analysis of the results reported in Table III-1 shows that the proposed drift-free shaping based method features better performances in terms of reaching the functional balance between transparency, robustness and fragility. Also, it is noticed that the selective based method features better transparency results while remaining fragile against transcoding attack and unable to detect content changing alterations. This weakness in terms of robustness/fragility performances of the selective based method [MA10] can be explained by the reducing of the insertion space. In fact, the drift avoiding selection step leaves few blocks potentially suitable for conveying the mark.

III.5. Conclusion

The present study deals with drift-free semi-fragile watermarking. First, by considering the analytic expressions of the MPEG-4 AVC encoding operations, it algebraically models the drift distortion problem. Second, it solves this problem under drift-free constraints. This way, the M_{apd}*= [0 -0.24 1.18 2.36; -0.24 -0.12 1.15 1.67; 1.18 1.15 3.10 4.47; 2.36 1.67 4.47 7.46]* matrix which should multiply the mark prior to its insertion is computed.

The experiments consider an *m*-QIM semi-fragile watermarking method and a video surveillance corpus of 120 minutes. For prescribed data payload (100 bit/s), robustness (BER < 0.1 against transcoding at 50% in stream size), fragility (frame modification detection with accuracies of 1/81 from the frame size and 3s) and complexity constraints, the modified insertion results in gains in transparency of 2 dB in PSNR, of 0.01 in IF and NCC and 22 in DVQ.

Note that the M_{apd} matrix is independent with respect to the video content and/or encoding; hence, its use does not represent an additional attack and does not increase the computational complexity.

Part IV: Conclusion and future work

IV.1. Conclusion

In this thesis, the issues related to watermark-based ownership protection and watermark-based video integrity verification in the field of MPEG-4 AVC compressed stream have been investigated.

Despite preliminary results published in the literature inserting some extra information into a compressed stream remains a challenging research topic, mainly because of its underlying conceptual contradiction. On the one hand, the compression goal is to eliminate the redundancy from the visual content. On the other hand, the watermarking exploits the visual redundancy to hide the mark.

Our study aims at (1) enhancing the performances of compressed domain robust watermarking for MPEG-4 AVC video stream and (2) designing a compressed stream watermark-based video authentication for video surveillance application.

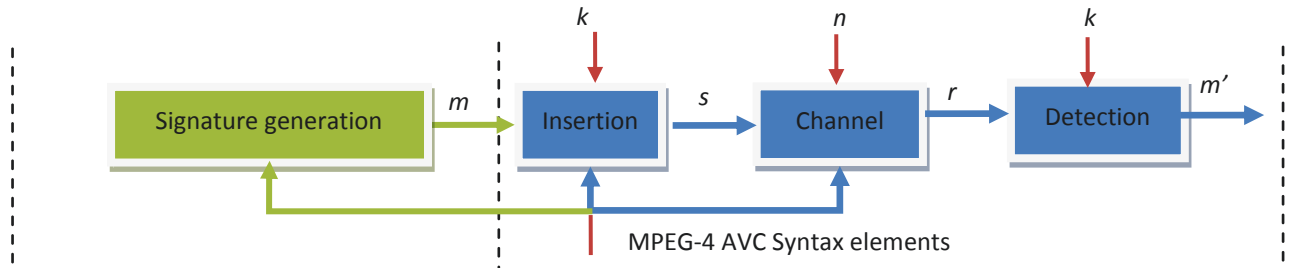
The present thesis tackles this challenges by providing the following main results (see Figure IV-1):

- By advancing the *m*-QIM theoretical framework, this thesis extends the QIM watermark principle beyond the binary case. In this respect, the research was structured at two levels: (1) extending the insertion rule from the binary to *m*-ary case and (2) computing the optimal detection rule, in sense of average probability error minimization under the condition of Gaussian noise constraints. Thus, the size of the inserted mark is increased by a factor $\log_2 m$.
- The theoretical framework *m*-QIM was deployed so as to advance a robust compressed domain watermarking method for MPEG-4 AVC ownership protection. While applied to the MEDIEVALS corpus (one hour of heterogeneous video content), the method demonstrated that the trade-off transparency-robustness-data payload can be reached. The main benefit is the increase of data payload by a factor of $\log_2 m$ while keeping fixed robustness (variations lower than 3% of the bit error rate after additive noise, transcoding and Stirmark random bending attacks) and transparency (set to average PSNR = 45dB and 65dB for SD and HD encoded content, respectively). The data payload averaged 150 bits per minute, *i.e.* about 20 times larger than the limit imposed by the DCI (Digital Cinema Initiatives) standard.
- A semi-fragile watermarking method for video integrity verification in compressed stream was designed. The research was structured at two levels. First, the possibility of identifying MPEG-4 AVC stream syntax elements describing the semantic content was investigated. Thus, we make an accurate usage of Intra prediction mode to generate the authentication signature. The signature generation block is first built by an empirical approach and subsequently validated by information theory tools. Secondly, the embedding strategy was specified. The designed system considers individual groups of *k* successive *I* frames (referred to as *I*-Group) sampled from an MPEG-4 AVC video sequence. Within such an *I*-Group, an authentication signature is extracted from the first *I* frame (thus ensuring fragility) and inserted into the rest of *k*-1 *I* frames by means of the *m*-QIM watermarking technique. The low complexity requirement can be met when the signature is extracted and inserted directly from/in the MPEG-4 AVC syntax elements, with minimal decoding/re-encoding operations. This method was evaluated for videosurveillance applications, under the framework of the SPY project. The results exhibit fragility to content replacement (with an 1/81 frame and 3s spatial and temporal accuracy, respectively) and

robustness against transcoding (compression by a factor of 2). The *m*-QIM framework main advantage is a relative gain factor of 0.11 of PSNR for fixed robustness (against transcoding), fragility (to content alteration) and the data payload. The experiments consider 1h 20min of video content.

- The intra-frame drift error propagation problem related to the compressed MPEG-4 AVC encoding features is solved by considering the analytic (algebraic) expressions of the MPEG-4 AVC encoding operations. The experiments consider an *m*-QIM semi-fragile watermarking method and a video surveillance corpus of 1 hour and 20 minutes. For prescribed data payload (100 bits per second), robustness (BER < 0.1 against transcoding at 50% in stream size), fragility (frame modification detection with accuracies of 1/81 from the frame size and 3s) and complexity constraints, the modified insertion results in gains in transparency of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ.

We note that the detailed contributions met the low level of complexity required by acting directly on the compressed domain, thus avoiding unnecessary additional encoding/decoding operation.



_m_-QIM insertion rule

- ***Theory:*** Increasing the data payload by a factor of *log2m.*
- ***Practice:*** Inserting a data payload of 150 bits per minute for prescribed robustness and transparency, for VoD ownership protection

_m_-QIM optimal decision rule

- ***Theory:*** Minimize probability of error under additive white Gaussian noise distributed attacks.
- ***Practice:*** Robustness of 0.1 of BER (variations lower than 3% of the bit error rate after additive noise, transcoding and Stirmark random bending attacks).

MPEG-4 AVC syntax element based signature

- ***Theory:*** Identifying the syntax elements which can optimally (in the information theory sense) serve as authentication signature
- ***Practice:*** Robustness against trascoding and Fragility to content replacement (with an 1/81 frame and 3 seconds spatial and temporal accuracy, respectively).

Drift-free shaping for MPEG-4 AVC compressed stream watermarking

- ***Theory:*** Analytically expressing the drift distortion problem and resolving it under drift-free constraint.
- ***Practice:*** Gains in transparency of 2 dB in PSNR, of 0.4 in AAD, of 0.002 in IF, of 0.03 in SC, of 0.017 NCC and 22 in DVQ. for prescribed data payload (100 bit/s), robustness (BER < 0.1 against transcoding at 50% in stream size).

MPEG-4 AVC compressed stream watermarking

- ***Practice:*** The mark generation, selection, insertion and detection for one second of video are 10 times and 8 times faster than the MPEG-4 AVC entropic encoding/decoding and the stream read/write operations, respectively.

Figure IV-1: Synopsis of the results.

IV.2. Future work

The perspectives of our future work are illustrated in Figure IV-2; they are structured on three directions, related to robust and semi-fragile watermarking as well as to drift effect cancelation.

Compressed domain robust watermarking

- From the theoretical point of view, work will be done in order to optimize the alphabet size *m* as a function of the quantization step and to estimate the channel capacity as a function of *m* and the quantization step.
- From the applicative point of view, we project to exploit the *m*-QIM for another type of content (*e.g.* 3DTV), to investigate the possibility to watermark other MPEG-4 AVC syntax elements and to evaluate the performances of the system against other attacks.
- Applying the *m*-QIM framework for protecting the MPEG-4 HEVC/H265 standard is also part of our future work.

Compressed domain semi-fragile watermarking

- From both theoretical and applicative points of view, a study on the usefulness of the other MPEG-4 AVC syntax elements for authentication applications will be carried out.
- The extensions towards MPEG-4 HEVC stream authentication will be investigated.

Drift free watermarking

- Work is already started in order to compensate the inter-frame (temporal) drift effect.
- The MPEG-4 HEVC standard imposes two challenges to the drift-free compensation, related to both perceptual shaping and drift *per-se*.

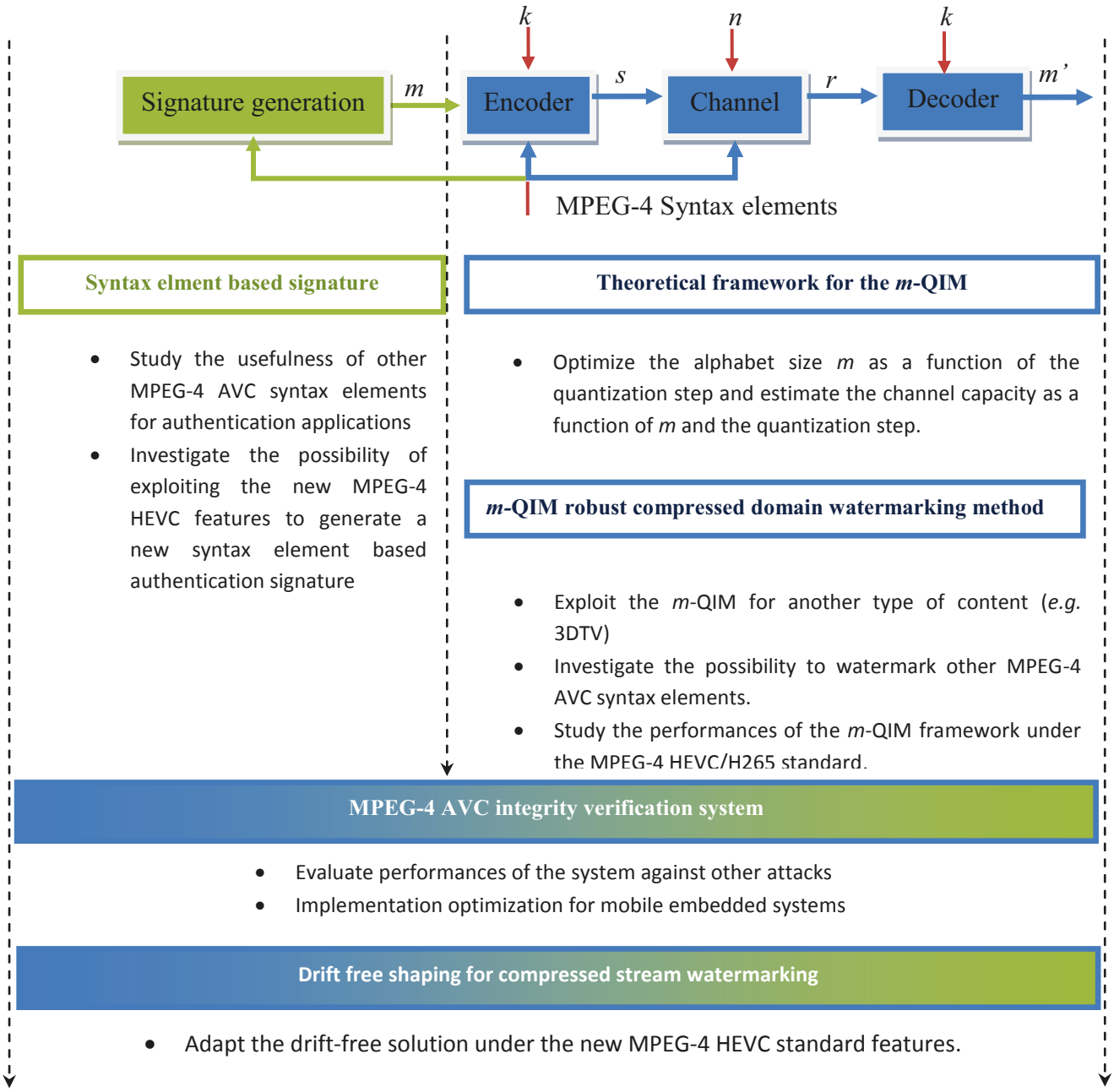


Figure IV-2: Perspectives.

Appendixes

A. MPEG-4 AVC overview

MPEG-4 AVC (Advanced Video Coding Standard) is a video coding standard, elaborated by the ITU-T Coding Video Expert Group (VEG) together with the ISO/IEC Moving Picture expert Group (MPEG) as the product of collective partnership effort known as the Joint Video Team (JVT). This standard is especially suitable for low rate video applications. It provides substantial better video at those same data rates compared to previous standard (MPEG-2, MPEG-4 Part 2, H.263) with only a moderate increase of complexity [RIC03]. Used in a wide range of applications, from mobile phones to High Definition TV, it helped to revolutionize the quality of the video image operating over several types of networks and systems.

The MPEG-4 AVC standard has a number of advantages that distinguish it from existing standards, while at the same time, sharing common features within other existing standards [RIC03].

The following are some of the key advantages of MPEG-4 AVC standard:

- Up to 50% in bit rate saving: compared to MPEG-2 or MPEG-4 Part 2, MPEG-4 AVC allows a reduction in bit rate by up to 50% for a similar degree of encoder optimization at most bit rates.
- High quality video: MPEG-4 AVC offers consistently good video quality at high and low bit rates.
- Error resilience: MPEG-4 AVC provides necessary tools to deal with packet loss in packet networks and bit errors in wireless networks.
- Network friendliness: MPEG-4 AVC bit stream can be easily transported over different networks through the Network Adaptation Layer.

The MPEG-4 AVC standard does not defines a new encoder. However, it defines new encoding syntax elements and refines the principal encoding functions.

The purpose of this Appendix is to outline the concept of the MPEG-4 AVC encoding standard and its advantages with respect to previous standards.

A.1. Structure

The MPEG-4 AVC architecture is designed based on two main layers: The Video Coding Layer (VLC) which is constructed to efficiently represent the video contents and the Network Abstraction Layer (NAL) which encapsulates the content represented by the VCL and provides header information in an appropriate way for conveyance by a variety of transport layer or storage media.

Figure A-1 shows the structure of the MPEG-4 AVC video encoder which will be explained later [RIC03].

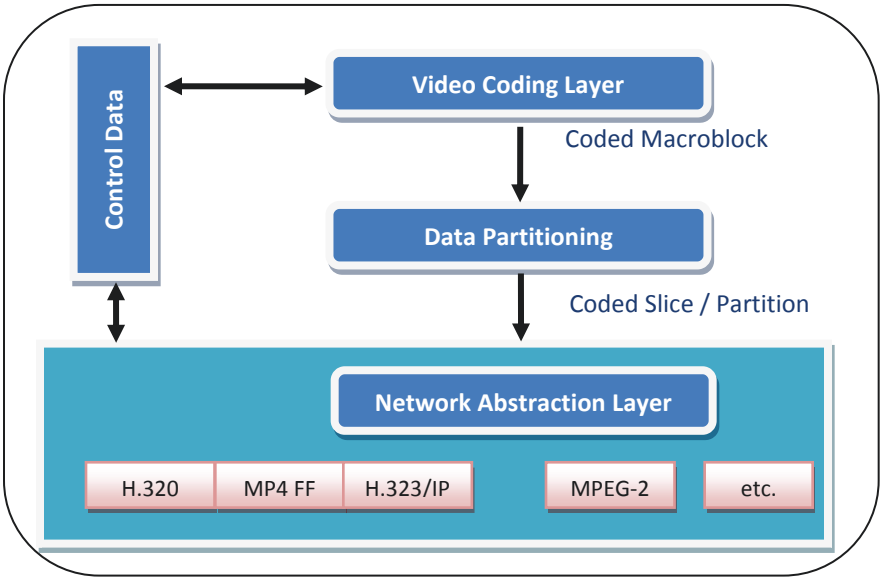


Figure A-1: MPEG-4 AVC architecture.

Network abstraction layer

NAL layer is specified to standardize a data encapsulation and to provide header information in an appropriate way for conveyance by the transport layers or storage media. All data are contained in NAL units, each of which contains effectively a packet with an integer number of bytes; the first byte belongs to the header. A NAL unit specifies a generic format to be used in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and bitstream is identical, except that each NAL unit can be preceded by a start code prefix to identify beginning of NAL unit in a bitstream-oriented transport layer.

The NAL facilitates the ability to map MPEG-4 AVC VCL data to transport layers such as:

- RTP/IP: Real-time internet services;
- File format: *e.g.*, ISO MP4 for storage and multimedia services;
- H.32X: wired and wireless conversational services;
- MPEG-2: broadcasting.

NAL units are classified into VCL and non-VCL units. VCL units contain the data that represents the values of samples in the video pictures and the non-VCL units contain any associated additional information such as parameter sets and supplemental enhancement information.

Video coding layer

The video coding layer of MPEG-4 AVC is similar in spirit to other standards. It consists on hybrid of temporal and spatial prediction, in conjunction with integer transform and entropic coding. The VCL working process consists of the four main stages:

- Dividing each video frame into blocks of pixels so that the video frame processing can be conducted at the block level.

- Exploiting the spatial redundancies that exist within the video frame by coding the original blocks through spatial prediction, transform, quantization and entropy coding (or variable-length coding).
- Exploiting the temporal redundancies that exist between blocks in successive frames, so that only changes between successive frames need to be encoded. This is achieved according to motion estimation and compensation. For a given block, a block search is performed in the previously coded frames (one or more) to obtain the motion vectors that will be used further by the encoder/decoder to predict the subject block.

Figure A-2 shows a block diagram of the MPEG-4 AVC encoding; this figure is kept unchanged from its reference [RIC03]

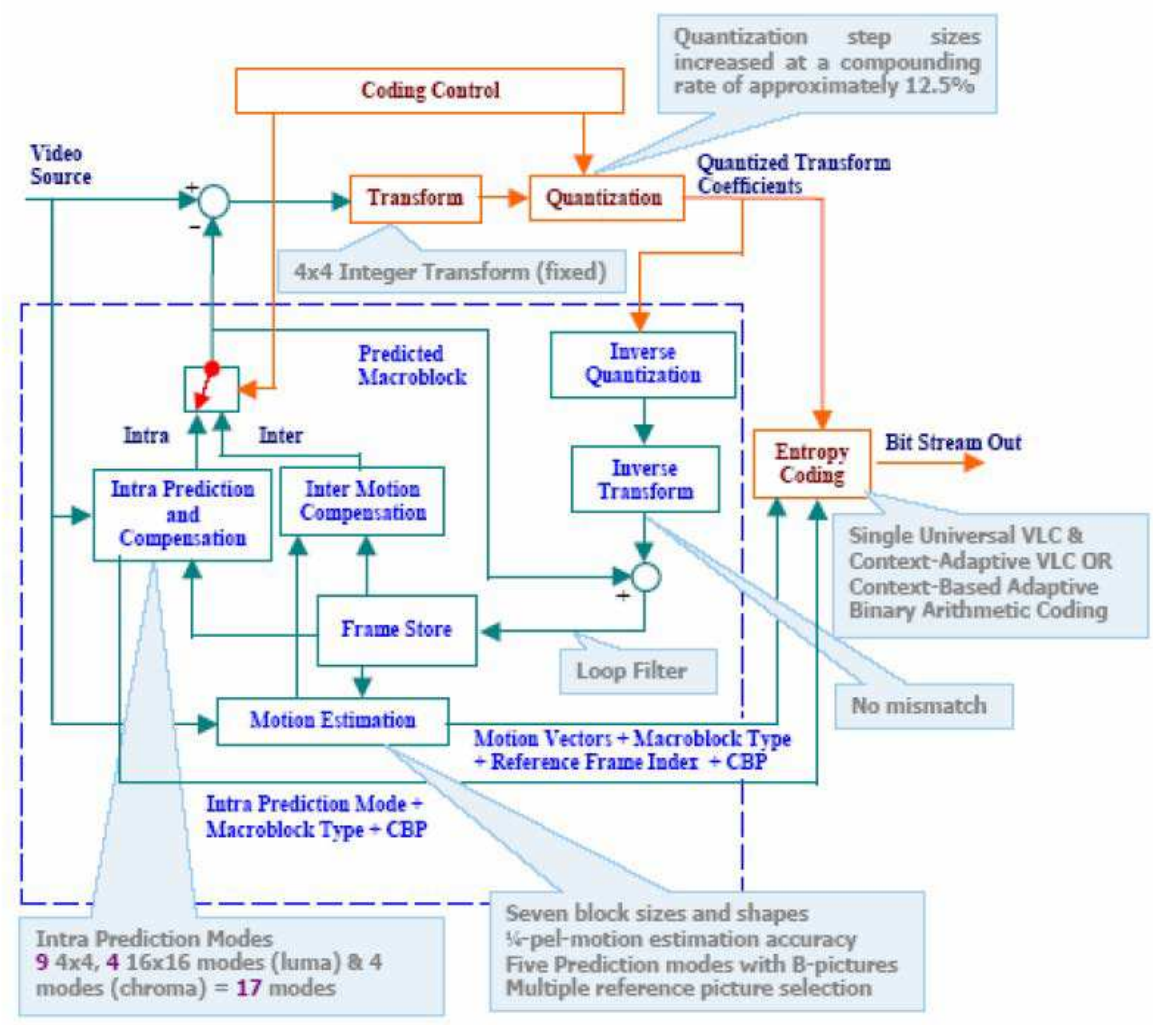


Figure A-2: Block diagram of the MPEG-4 AVC encoder [RIC03].

The VCL is structured into five layers: GOP (Group Of Picture), picture, slice, macroblock and block. Headers of each layer provide information on the encoding/decoding order for the lower layers.

A GOP consists of a number of images that can be 3 types, grouped according to predetermined decoding order:

- **The *I* frames** correspond to a coded image independently; note that only one field *I* can be at the beginning of a GOP, as it serves as a starting point for coding images of two other types;
- **The *P* frames** are associated with the a motion compensated image, predicted either from an *I* or from other frame;
- **The B frames** refer to any image being double (forward and backward) motion compensated.

Block dividing

Each video image is partitioned into 16×16 macroblocks. Each macroblock consists of 16×16 luminance samples Y of 8×8 samples for each of two chrominance components Cb and Cr. These blocks are encoded/decoded with respect to the order described in the Figure A-3.

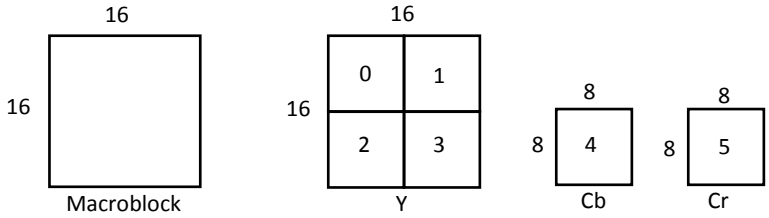


Figure A-3: Y, Cb and Cr encoding/decoding order.

A.2. Encoding

Prediction

Each frame of a video sequence is processed in units of macroblock (corresponding to 16×16 pixels). Each macroblock is encoded in intra or inter mode. In intra mode, the block B is constructed by samples in the current frame that have previously been encoded/decoded. In inter mode, the Block B is constructed by motion compensation from one or more reference frames.

Intra prediction

In MPEG-4 AVC, two different types of intra prediction: intra 4×4 and intra 16×16 can be considered. On the one hand, the intra 4×4 mode is based on predicting each 4×4 block separately and is well suited for encoding the textured frame area. On the other hand, the intra 16×16 mode performs the prediction of the whole 16×16 macroblocks and is more suited for encoding smoothed frame area.

In order to perform the intra prediction mentioned above, MPEG-4 AVC offers nine modes for the prediction of 4×4 luminance blocks [RIC03], including DC prediction (Mode 2) and eight directional modes, namely 0, 1, 3, 4, 5, 6, 7 and 8 , see Figures A-4 and A-5; these figures are kept unchanged from their reference [RIC03].

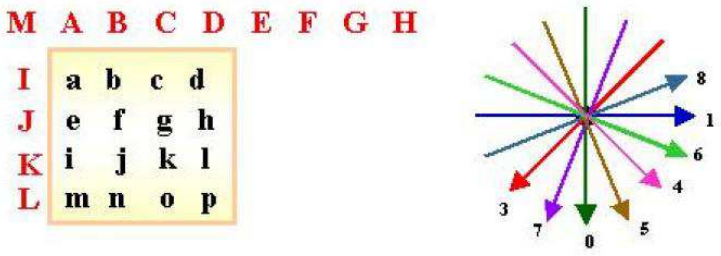


Figure A-4: Intra prediction.

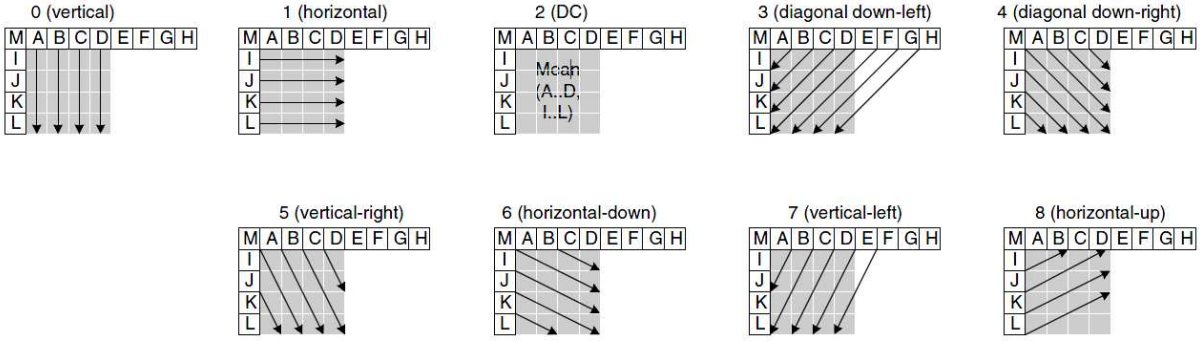


Figure A-5: Intra prediction modes for $\mathbf{4} \times \mathbf{4}$ *luminance blocks [RIC03].*

The predicted block is obtained by using the already encoded samples (from A to M) from neighboring blocks.

For instance, if the vertical prediction mode (mode 0) is selected in the prediction, then the samples of the predicted block (from a to p) are obtained as follows:

- a, e, I and m are set to A,
- b, f, j and n are set to B,
- c, g, k and o are set to C, and
- d, h, l and p are set to D.

For smoothed regions (with less spatial detail), MPEG-4 AVC consider 16×16 intra prediction mode. One of four prediction modes (DC, Vertical, Horizontal and Planar) may be chosen for computing the luminance predicted macroblock.

In addition, MPEG-4 AVC supports intra prediction for 8×8 chrominance blocks by also using four prediction modes (DC, Vertical, Horizontal and Planar). Finally, the prediction mode for each block is efficiently coded by assigning shorter symbol to more likely modes, where the probability of each mode is determined based on the modes used for coding the surrounding blocks.

Inter prediction

Inter prediction is based on using motion estimation and compensation to take advantage of the temporal redundancies that exist between successive frames, hence, providing very efficient coding. Motion estimation in MPEG-4 AVC supports most of the key features adopted in earlier video standards, but its efficiency is improved through added flexibility and functionality.

Compared to previous video coding standard which support only 16×16 and 8×8 block sizes for motion estimation, MPEG-4 AVC supports a block sizes ranging from 16×16 to 4×4. Motion compensation for each block 16×16 can be performed according to different block sizes and shapes.

Partitions with luma block size of 16×16, 16×8, 8×16, and 8×8 samples are initially supported by the syntax. In case partitions with 8×8 samples are chosen, one additional syntax element specifies whether the corresponding 8×8 partition is further partitioned into partitions of 4×8, 8×4 or 4×4 luma samples and corresponding chroma samples. Figure 6-A illustrates the partitioning principal.

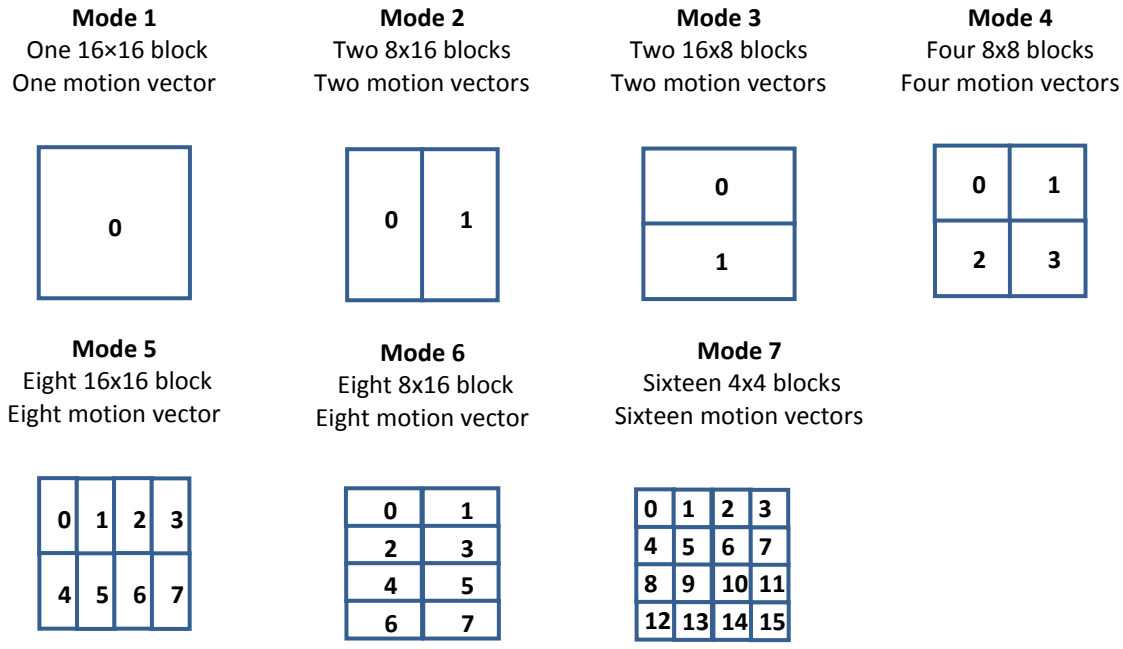


Figure A-6: Different modes of dividing a macroblock for motion estimation in MPEG-4 AVC.

The availability of smaller motion compensation blocks improves the prediction in general, and in particular, the small blocks improves the ability of the model to handle fine motion detail and result a better subjective viewing quality because they do not produce large blocking artifacts.

MPEG-4 AVC allows the motion vector to be determined at higher levels of spatial accuracy with respect to the existing standards. Quarter pixel accuracy is deployed in MPEG-4 AVC motion compensation, in contrast with prior standards based primarily on half-pixel accuracy. A more detailed investigation of fractional sample accuracy is presented in [WED03].

The MPEG-4 AVC standard offers the option of having multiple reference frames in inter picture coding, resulting in better subjective video quality and more efficient video coding. Moreover, from the implementation point of view, there would be additional processing delays and higher memory at both the encoder and the decoder.

Each partition or sub-partition in an inter macroblock a motion vector is estimated from an area of the same size in a reference picture. The motion compensated vector is obtained by computing the

displacement between the current area and the reference area. The difference between the current motion vector and the predicted motion vector and the prediction mode are encoded.

Compared to anterior standards, MPEG-4 AVC has brought a new innovation by introducing the kipped macroblock type. For macroblock encoded according to this type, neither prediction residual information, nor motion information or reference index is transmitted for the decoder side. The advantage of this induced type is that large area with no change such as background can be represented with very few bits.

Regardless the type of prediction, the pixel values are subtracted from the corresponding predicted values to obtain the residual macroblock. Each residual macroblock is transformed, quantized and binary encoded.

Transformation

Following the prediction, the transformation is applied with the view of representing the data as uncorrelated (separate components with a minimum interdependence) and compacted (the energy is concentrated in a small number of frequencies) [HAL02].

Compared to previous standards which use the 8×8 Discrete Cosine Transform (DCT) as the basic transformation, MPEG-4 AVC uses three transformations depending on the type of the data that is to be encoded:

- An integer DCT transformation which is applied to all 4×4 blocks of luminance and chrominance components in the residual data.
- A Hadamard transformation applied to 4×4 blocks constructed of luma dc coefficients in intra macroblocks predicted according to the 16×16 mode.
- A Hadamard transformation applied to 2×2 blocks constructed of chroma dc coefficients in any macroblock.

One of the main improvements of this standard is the using of smaller 4×4 block transformation. In fact, the smaller shape transformation has visual benefits resulting in reducing blocking and ringing (noise around edges) artifacts. Also, the smaller transformation requires less computations and subsequently less energy consuming.

Instead of a classical 4×4 discrete cosine transform, a separable integer transform with similar properties as a 4×4 DCT is used. The new advanced transform approaching the 4×4 DCT has several advantages:

- The core part of the transformation can be implemented using additions and shifts, resulting less level of computation complexity.
- The precise integer specification eliminates any mismatch issues between the encoder and decoder in the inverse transform (this has been a problem with earlier standards).

For instance, given a 4×4 residual block X, the transformed block Y is obtained according to the following matrix operations:

$$Y = AXA^T \odot E$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & -1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, E = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}; a = \frac{1}{2}; b = \sqrt{\frac{2}{5}}$$

where A^T is the transposed matrix of A and the operator $\odot$ denote the element per element multiplication between two matrices.

The 4×4 Hadamard transformation is performed according to the following equation:

$$H_{DC} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} Y_{DC} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

The 2×2 Hadamard transformation is performed according to the following equation:

$$H_{DC} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} Y_{DC} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Where Y_{DC} represents a DC block and H_{DC} denotes the Hadamard transformed block.

Figure A-7 illustrates the way in which the data is structured and transmitted within a macroblock. If the macroblock is coded in 16×16 intra mode, then the block containing the DC coefficient of each 4×4 luma block (labeled -1 in Figure A-7) is transmitted first. Secondly, the luma residual blocks ranging from 0 to 15 are transmitted in the order shown in Figure A-7 where the DC coefficients are set to zero. Blocks 16 and 17 contain a 2×2 array of chroma coefficients are transformed and sent. Finally, chroma residual blocks ranging from 18 to 25 (with DC coefficient set to 0) are sent.

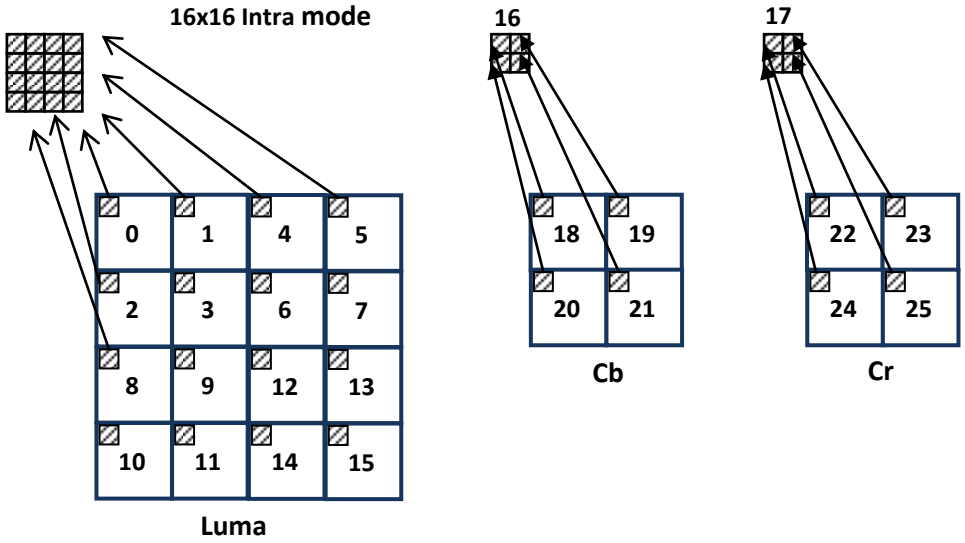


Figure A-7: Block construction for DCT and Hadamard transformations.

Quantization

The quantization step is where a significant portion of data compression takes place and is the phase where information is lost in the compression chain [HAL02]. In MPEG-4 AVC, the transformed coefficients are quantized using a scalar quantization. The basic forward quantization operation is performed as follows:

$$Z_{ij} = round(Y_{ij} / Q_{step})$$

where Y_{ij} is a coefficient of the transformed 4×4 block described above, $\mathrm{Q_{step}}$ is the quantization step size and Z_{ij} is the quantized coefficient.
The MPEG-4 AVC supports a total of 52 of quantization step value which are indexed by a quantization parameter Q_p as illustrated in Table A-1.

Table A-1: Quantization steps.

Q_p	0	1	2	3	4	5	6	7	8	9	10	11	12	…
Q_{step}	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	…
Q_p	…	18	…	24	…	30	…	36	…	42	…	48	…	51
Q_{step}		5		10		20		40		80		160		224

The quantizers are arranged so that an increase of 1 in quantization parameter means an increase of quantization step size by approximately 12% (an increase of 6 means an increase of quantization step size by exactly a factor of 2).

To circumvent the disadvantages of the entire division, the MPEG-4 AVC standard offers another form of quantization performing, this time is involving right shift:

$$Z_{ij} = sign\{Y_{ij}\}\left[|Y_{ij}|A(Q_p) + f2^L \gg L\right]$$

Where f and $A(Q_p)$ are association of the quantization parameter, L is the bit length parameter for the encoding process.

Entropy coding

The quantized transformed coefficients are generally scanned in a zig-zag manner and transmitted to be encoded. The zig-zag scan illustrated in Figure A-8 is used in all frame coding cases and it is identical to the conventional scan used in earlier video coding standards. The zig-zag scan arranges the coefficient in an ascending order of the corresponding frequencies.

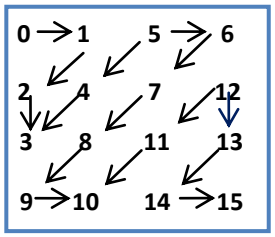


Figure A-8: Zig-zag scanning.

Entropy coding is the last step in video coding process. Its paradigm consists in assigning shorter codewords to symbol with higher occurrence probability and longer codewords to symbol with less occurrences probability. Two types of entropy coding have been adopted by the MPEG-4 AVC standard. The first category represents a combination of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC). The second method is represented by Context-Based Adaptive Binary Arithmetic Coding (CABAC).

UVLC/CAVLC

In anterior video coding standards, symbols the associated codewords are organized in a look-up tables, referred to as variable length coding (VLC) tables which are stored at both the encoder and the decoder. In MPEG-2, a number of VLC tables are used, depending on the type of encoded data (*e.g.*, transformed coefficient, motion vector, …).

In contrast with the previous standards, MPEG-4 AVC offers a single Universal VLC (UVLC) table that is used in entropy coding of all symbols in the encoder except the transform coefficients. Despite the fact that the use of a single table is simple, a major disadvantage is introduced, a single table may ignore the correlation between the encoder symbols.

In MPEG-4 AVC, the residual transformed coefficients are encoded using a Context Adaptive Variable Length Coding (CAVLC). CAVLC is designed to take advantage of several features of the zig-zag scanning of 4×4 blocks:

- Non-zero coefficients at the end of the zig-zag scanning are often equal to +/-1. CAVLC encodes the number of these coefficients (“trailing 1s”) in a compact way.

- CAVLC employs run-level coding to represent the string of zero in a quantized 4×4 block. Furthermore, the numbers of non-zero coefficients in neighboring blocks are usually correlated. Consequently, the number of non-zero coefficients is encoded according to a look-up table that depends on the number of non-zero coefficients in neighboring blocks.
- The level of non-zero coefficients tends to be higher at the start of the 4×4 zig-zag scanned block (near the DC coefficient) and lower towards the higher frequencies. CAVLC takes advantage of this by adapting the choice of the VLC look-up table for the level encoding depending on recently encoded level magnitude.

CAVLC is supported in all MPEG-4 AVC profiles [WEI03].

CABAC

CABAC (Context-Adaptive Binary Arithmetic Coding) is used in MPEG-4 AVC to encode quantized coefficients, based on arithmetic coding [MAR01]. The main difference between CABAC and CAVLC is that CABAC has the feature to adapt to the context: coding table change according to symbols already transmitted. CABAC is more efficient than CAVLC but has a higher computational complexity.

On the one hand, the usage of arithmetic coding allows the assignment of a non-integer number of bits to each symbol of an alphabet, which is extremely beneficial for symbol probabilities that are greater than 0.5. On the other hand, the usage of adaptive codes permits adaptation to non-stationary symbol statistics. Another important property of CABAC is its context modeling. The statistics of already coded syntax elements are used to estimate conditional probabilities. These conditional probabilities are used for switching several estimated probability models. In H.264/AVC, the arithmetic coding core engine and its associated probability estimation are specified as multiplication-free low-complexity methods using only shifts and table look-ups. Compared to CAVLC, CABAC typically provides a reduction in bit rate between 5-15%.The highest gains are typically obtained when coding interlaced TV signals. More details on CABAC can be found in [MAR03].

A.3. Profiles

A profile defines a set of coding tools and features that can be used in generating a conforming stream. MPEG-4 AVC defines three popular profiles which are namely the Baseline, Main and extended profile.

The Baseline profile allows the use of Arbitrary Slice Ordering (ASO) to ensure real time communication applications, as well as the use of Flexible Macroblock Ordering (FMO) to improve error resilience in the coded bit stream. The Baseline profile supports all features in the MPEG-4 AVC except the following two features sets [WEI 03]:

- Set 1: B slices, weighted prediction, CABAC.
- Set 2: redundant slice data partitioning.

The Baseline profile is mainly deployed for real time applications such as video conferencing and mobile applications.

The first set of the additional feature is introduced by the Main profile. However, the Main profile does not support the FMO, ASO and redundant slice data partitioning which are supported by the Baseline profile. The Main profile is deployed for Broadcast application.

The extended profile support all features of the Baseline profile, and both the sets of features detailed above except CABAC. The extended profile is mainly deployed for High Definition (HD) video encoding.

A.4. MPEG-4 AVC parser

In order to allow the modification of individual elements in an MPEG-4 AVC compressed stream while observing to the related syntax constraints, the following parser has been implemented, Figure A-9.

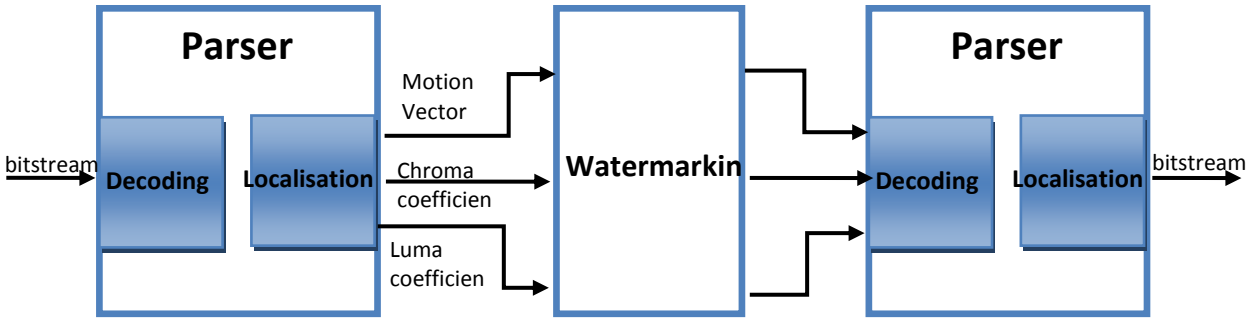


Figure A-9: Parser of the MPEG-4 AVC bit stream.

Our parser exploits the layer architecture of the MPEG-4 AVC Codec to separate the parsing and location process of MPEG-4 AVC syntax element. As shown in Figure A-10, two main layers exist: the video coding layer describes the chain compression while the network abstraction layer controls the package of the in NALU (Network Abstraction Layer Unit). After this second layer, the parser analyzes the stream and recovers the syntax elements according to the hierarchy shown in Figure A-10.

Our parser, Figure A-10, performs a partial decoding of the AVC streams read from a file, changes its syntax elements according to the watermarking procedure, records it into new stream and then writes into another file. These operations are also performed into two levels architecture.

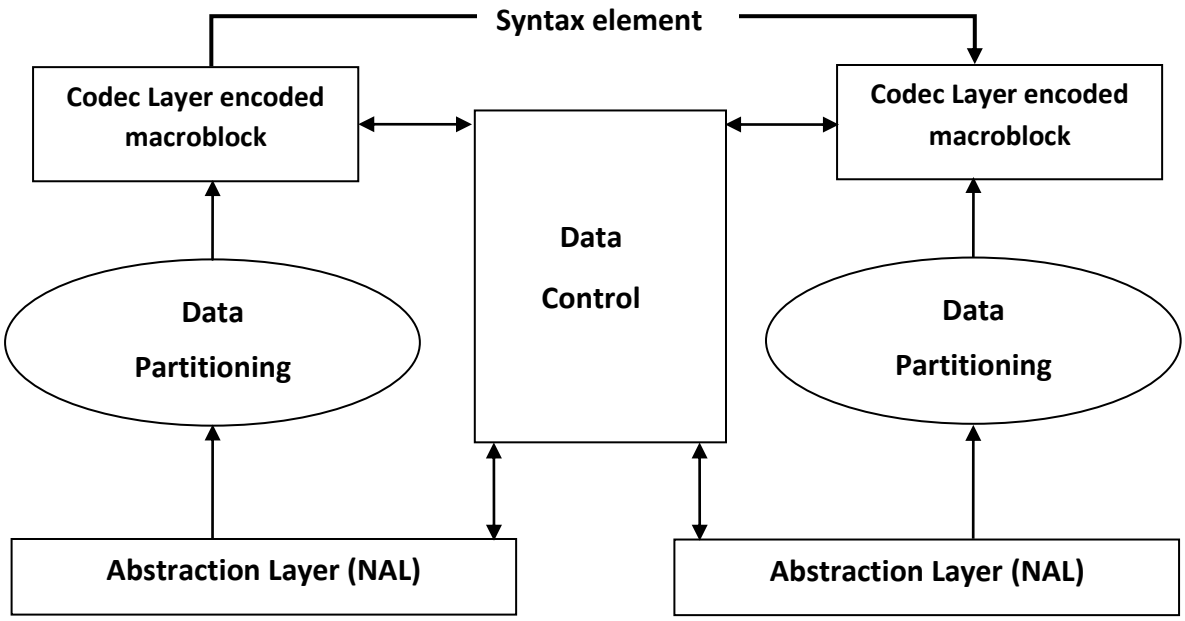


Figure A-10: Layer structure MPEG-4 AVC Decoder/Parser/Encoder.

B. Video corpus

B.1. MPEG-4 AVC encoding parameters

In generally the standard includes two different configuration of codec called profiles, each targeting a specific class of applications:

- Baseline Profile (BP): Primarily for low-cost applications that use fewer resources, this profile is widely used in mobile applications and videoconferencing.
- Main Profile: This profile is used for standard definition digital TV broadcasts that use the MPEG-4 format as defined in the DVB standard.

In the sequel, Table B-1 will provide details about the profile parameters considered in the corpus while Table B-2 provides the level parameter descriptions.

Table B-1: MPEG-4 AVC profiles parameters.

Compression Process	Configuration	Baseline	Main
Pre –processing	4:2:0 format	Yes	Yes
	4:0:0 format	No	No
	4:2:2 format	No	No
	4:4:4 format	No	No
	Deblocking filter	Yes	Yes
Prediction	Slice I and P	Yes	Yes
	Slice B	No	Yes
	Slice SI ans SP	No	No
	Multiple Reference	Yes	Yes
	Redundant Slices (RS)	Yes	No
Quantization	Quantization Matrix	No	No
Encoding	CAVLC	Yes	Yes
	CABAC	No	Yes
	8 bits per pixel	Yes	Yes

Table B-2: MPEG-4 AVC level parameters of our experimental corpus.

Level	Maximum bloc number per frame	Maximum rate for Base	Resolution & fps
1	99	64 kbit/ s	128x96/ 30.9
2.2	1620	256 kbit/ s	352x576/ 25.0
3	1620	5 Mbit/s	720x576/ 25.0
4	8192	10 Mbit/ s	1920x1080/ 30.1

B.2. Corpus

The experiments are carried out on large number of heterheneous video sequences, as illustraited bellow. Our expirements were conducted under two corpora the size and types of the various content are presendted in Table B-3 and Table B-4.

Table B-3: Experimental corpora (MEDIEVALS SD corpus).

MEDIEVALS SD CORPUS	
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 15 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 15 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 15 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 15 minutes

Table B-4: Experimental corpora (MEDIEVALS HD corpus).

MEDIEVALS HD CORPUS	
	• 2024 kbps • Baseline profile • 1920x1080/ 25 fps • 15 minutes
	• 2024 kbps • Baseline profile • 1920x1080/ 25 fps • 15 minutes
	• 2024 kbps • Baseline profile • 1920x1080/ 25 fps • 15 minutes
	• 2024 kbps • Baseline profile • 1920x1080/ 25 fps • 15 minutes

Table B-5: Experimental corpora (SPY corpus).

SPY CORPUS	
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes

	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes
	• 512 kbps • Baseline profile • 640x480/ 25 fps • 10 minutes

C. Additional results related to the *m*-QIM probability of error

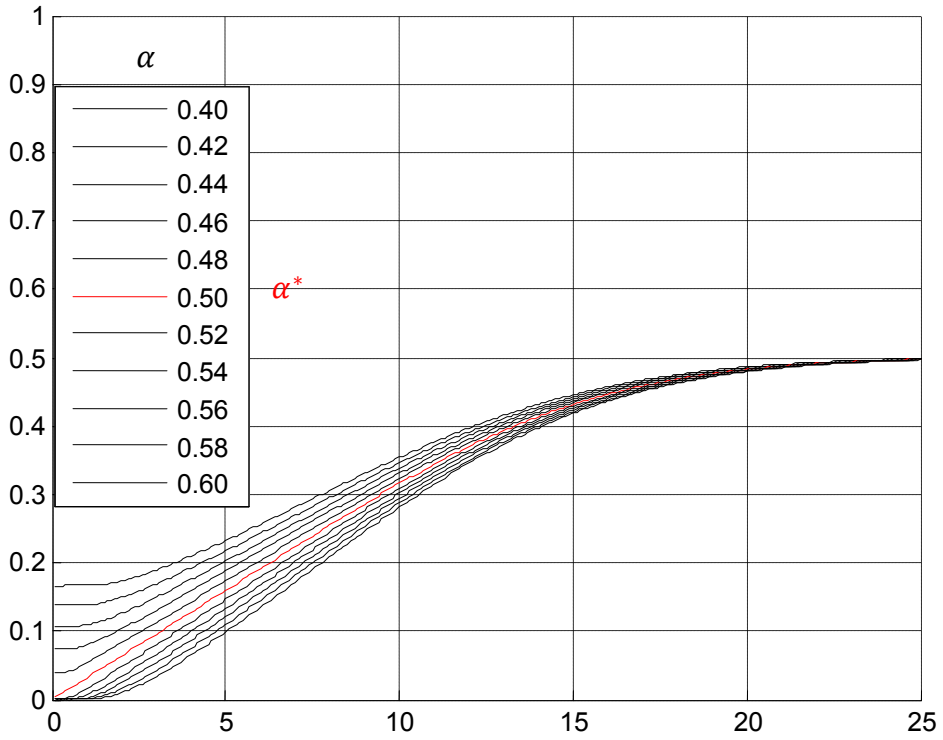
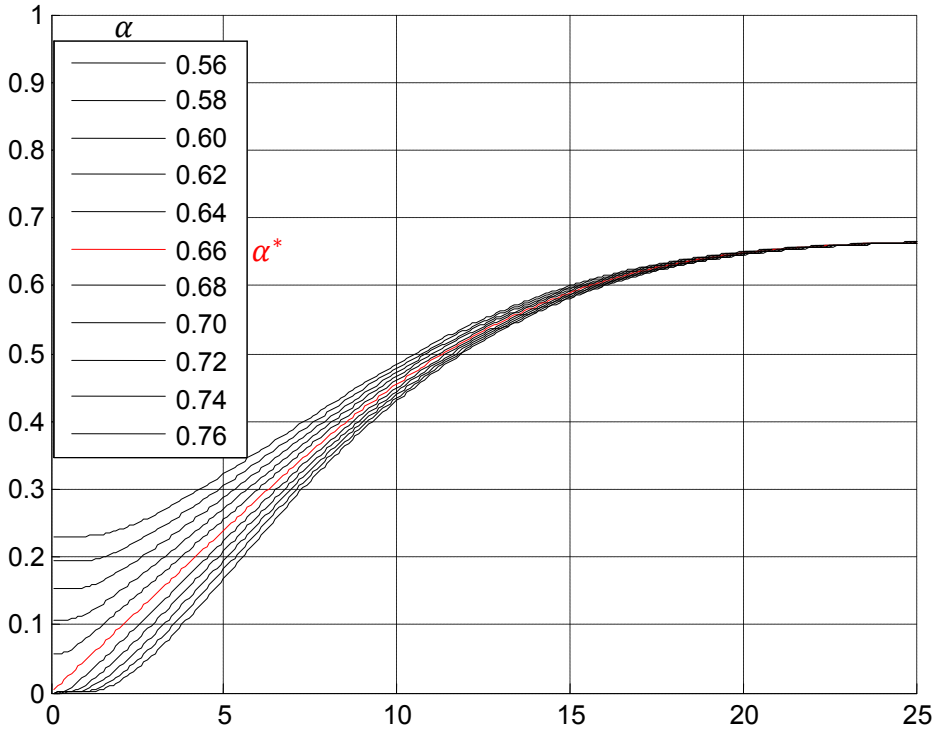
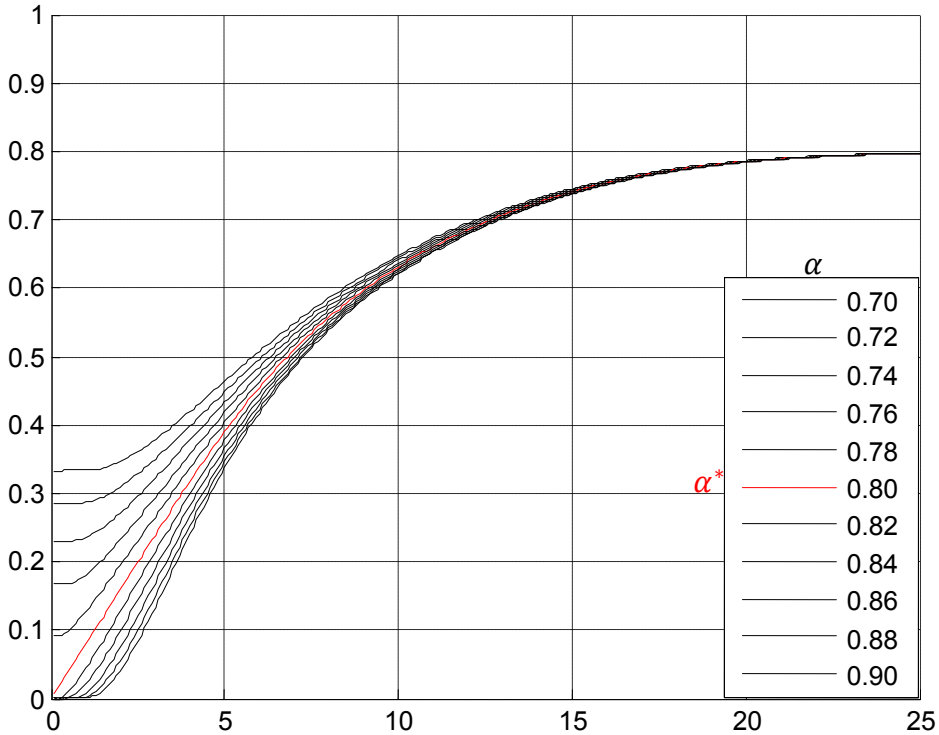


Figure C-1: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 2}$ and for a fixed value $\boldsymbol{\Delta = 50}$.



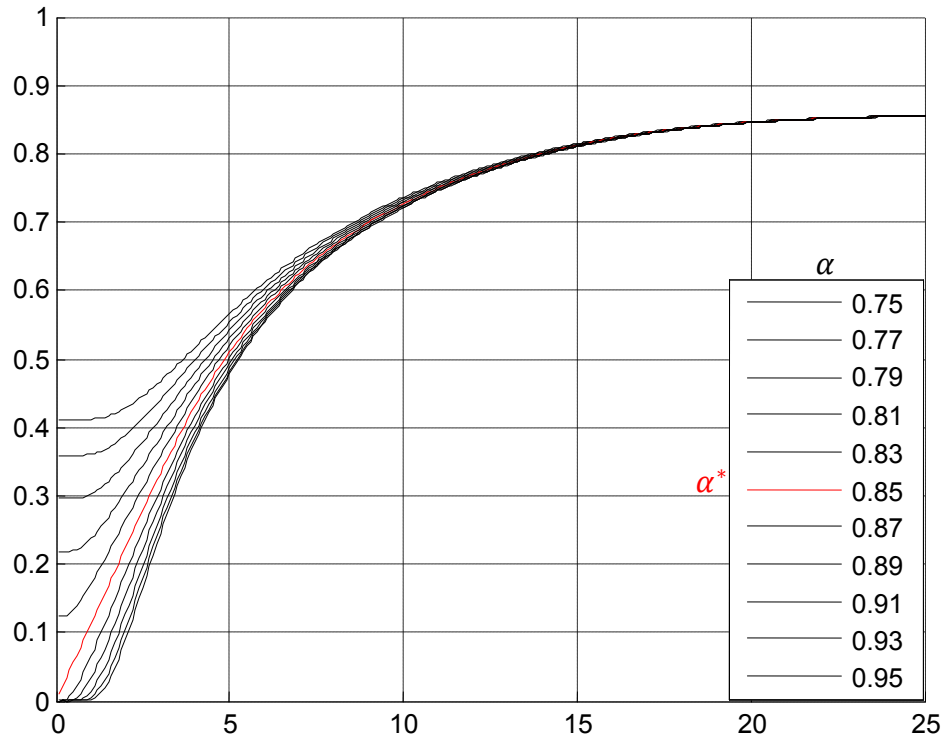
$m = 3$

Figure C-2: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 3}$ and for a fixed value $\Delta = \mathbf{50}$.



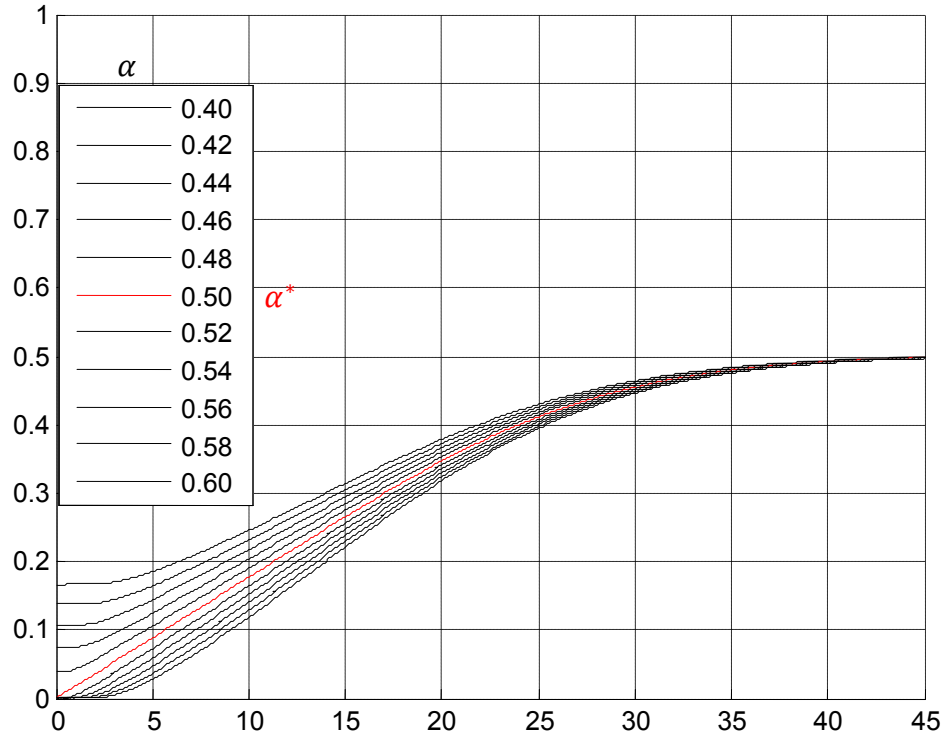
$m = 5$

Figure C-3: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 5}$ and for a fixed value $\Delta = \mathbf{50}$.



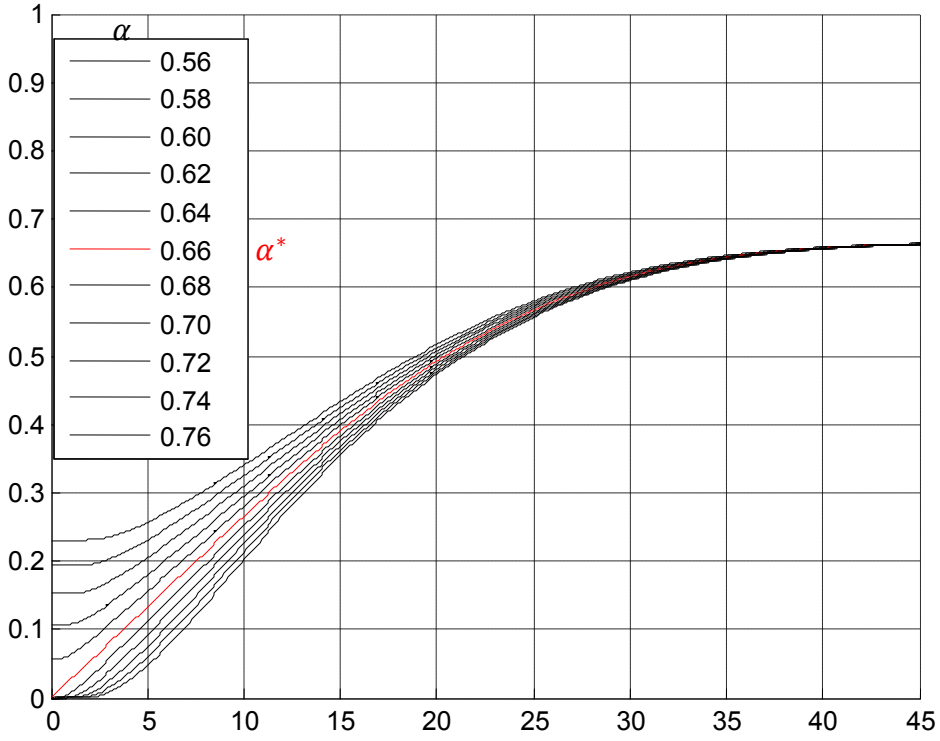
$m = 7$

Figure C-4: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 7}$ and for a fixed value $\Delta = \mathbf{50}$.



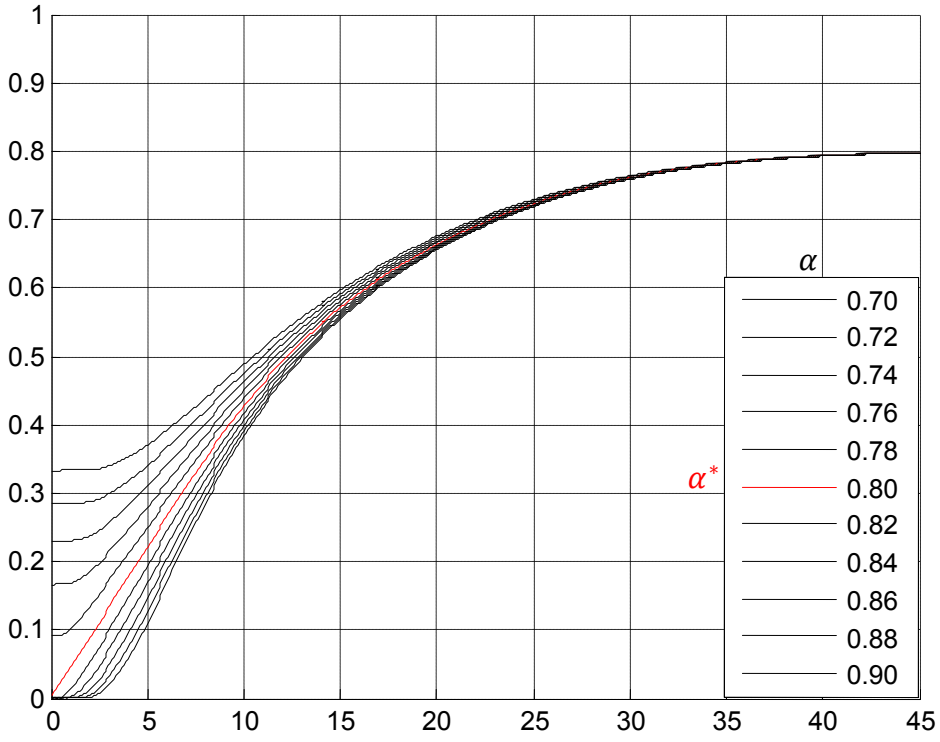
$m = 2$

Figure C-5: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 2}$ and for a fixed value $\Delta = \mathbf{90}$.



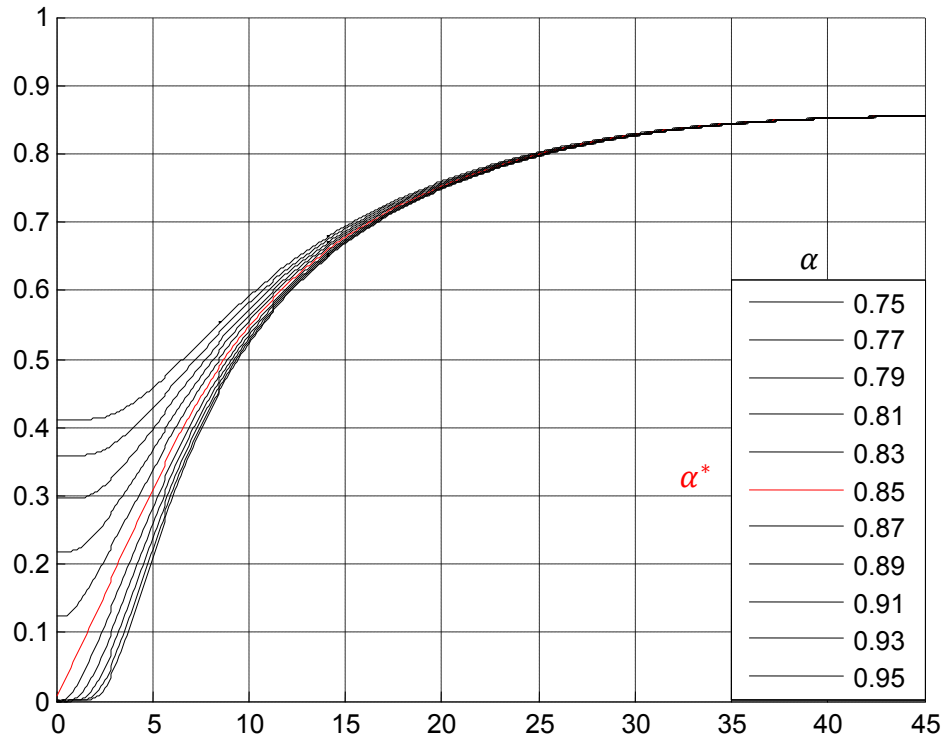
$m = 3$

Figure C-6: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 3}$ and for a fixed value $\boldsymbol{\Delta = 90}$.



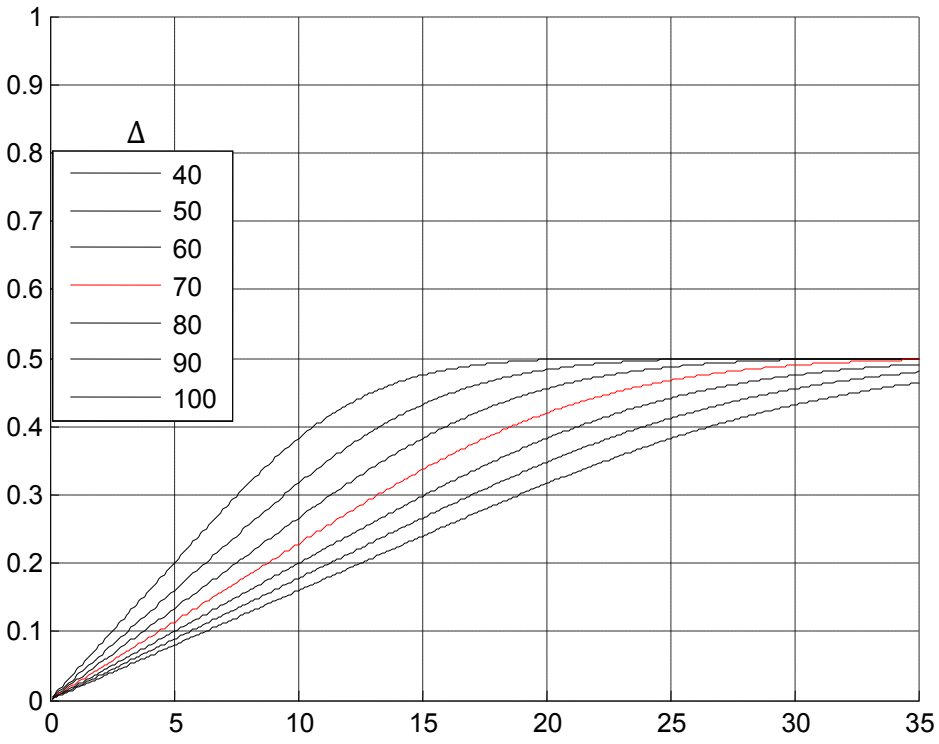
$m = 5$

Figure C-7: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 5}$ and for a fixed value $\boldsymbol{\Delta = 90}$.



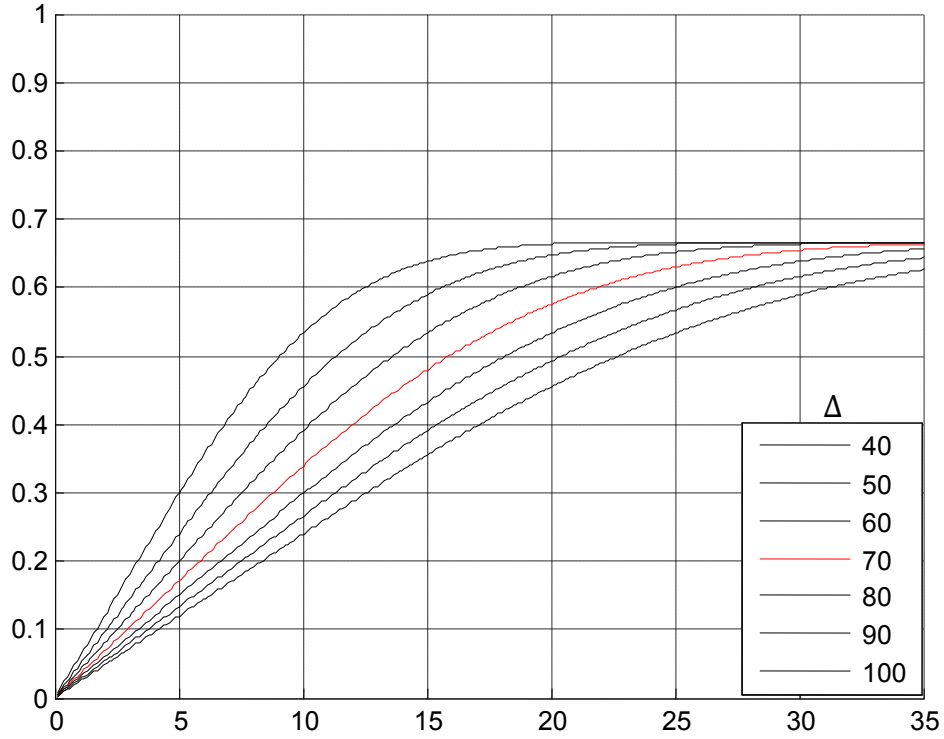
$m = 7$

Figure C-8: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for 11 values of $\boldsymbol{\alpha}$, for four values of $\boldsymbol{m = 7}$ and for a fixed value $\Delta = \mathbf{90}$.



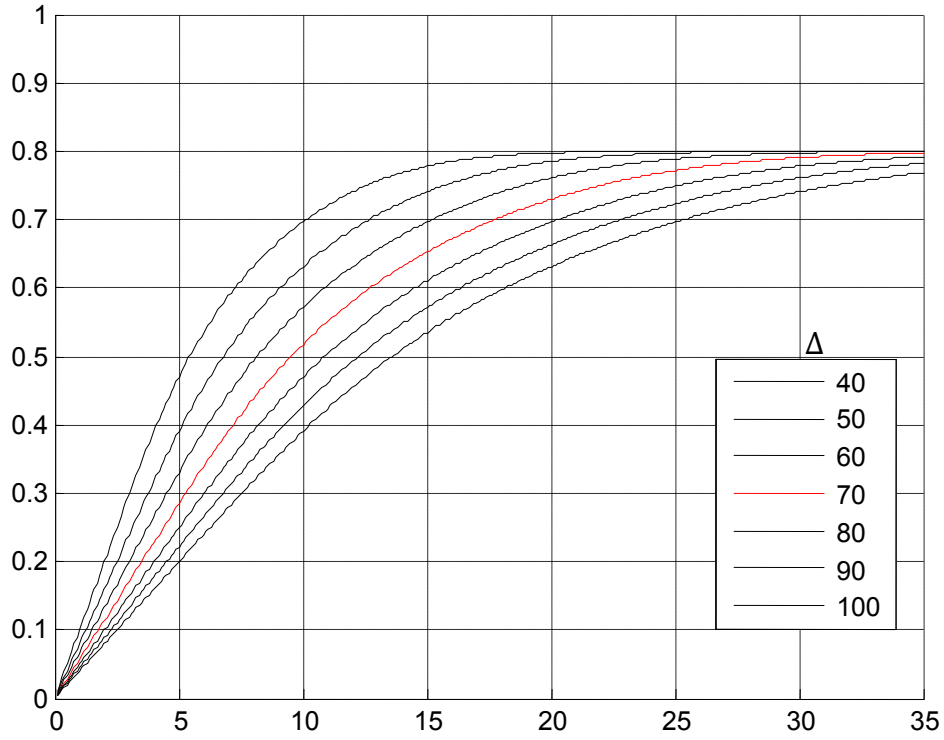
$m = 2, \alpha = \alpha^* = 0.5$

Figure C-9: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 2}$ and $\boldsymbol{\alpha = \alpha^*}$.



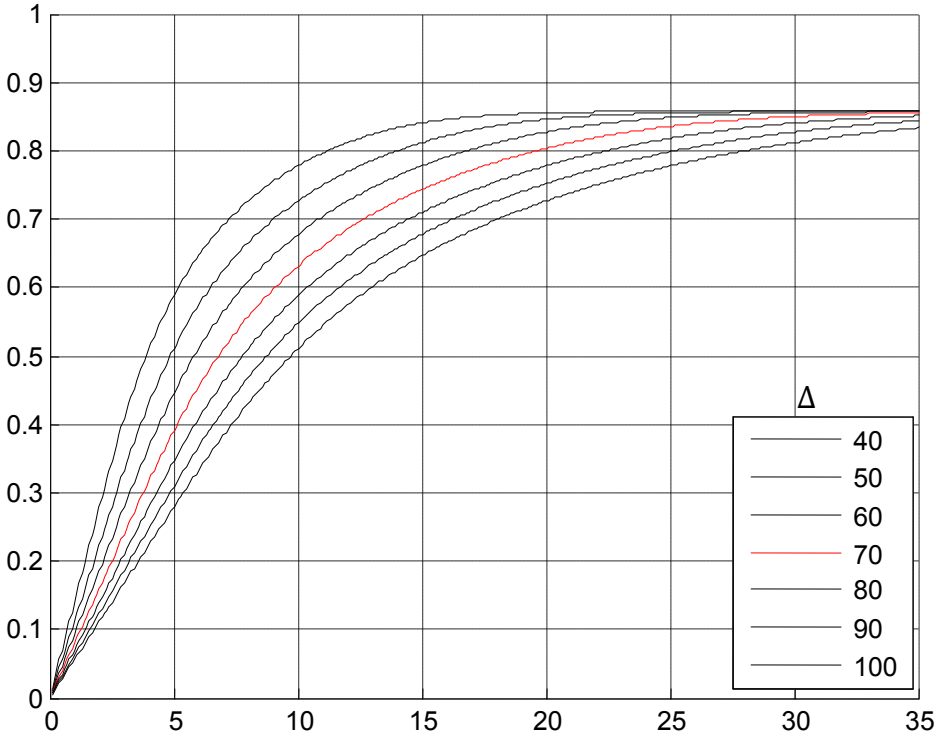
$m = 3, \alpha = \alpha^* = 0.66$

Figure C-10: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 3}$ *and* $\boldsymbol{\alpha = \alpha^*}$.



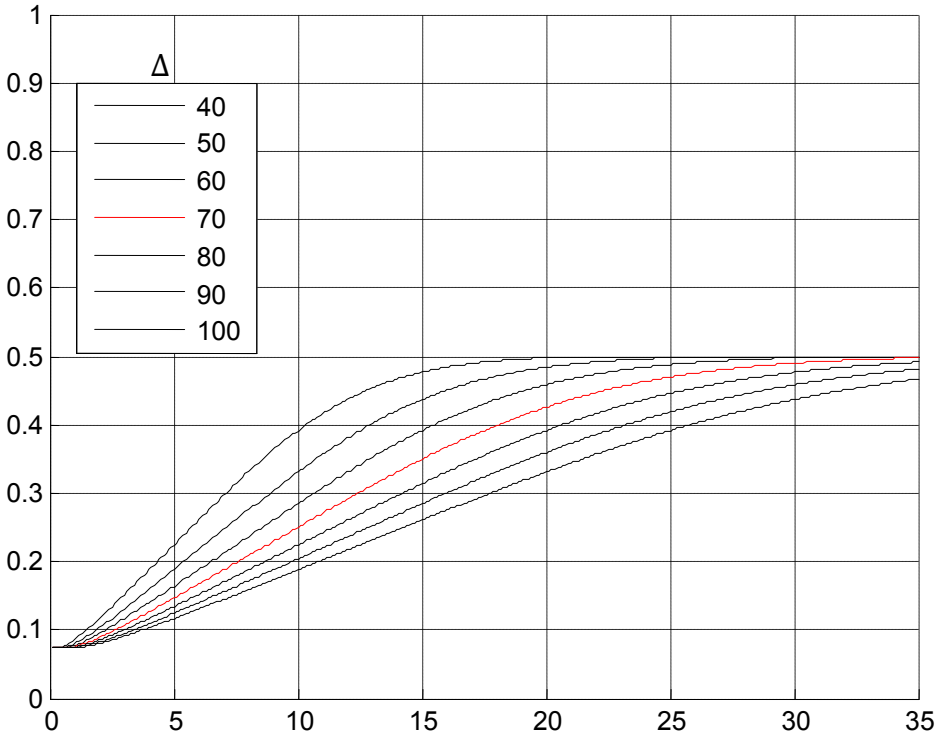
$m = 5, \alpha = \alpha^* = 0.8$

Figure C-11: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 5}$ *and* $\boldsymbol{\alpha = \alpha^*}$.



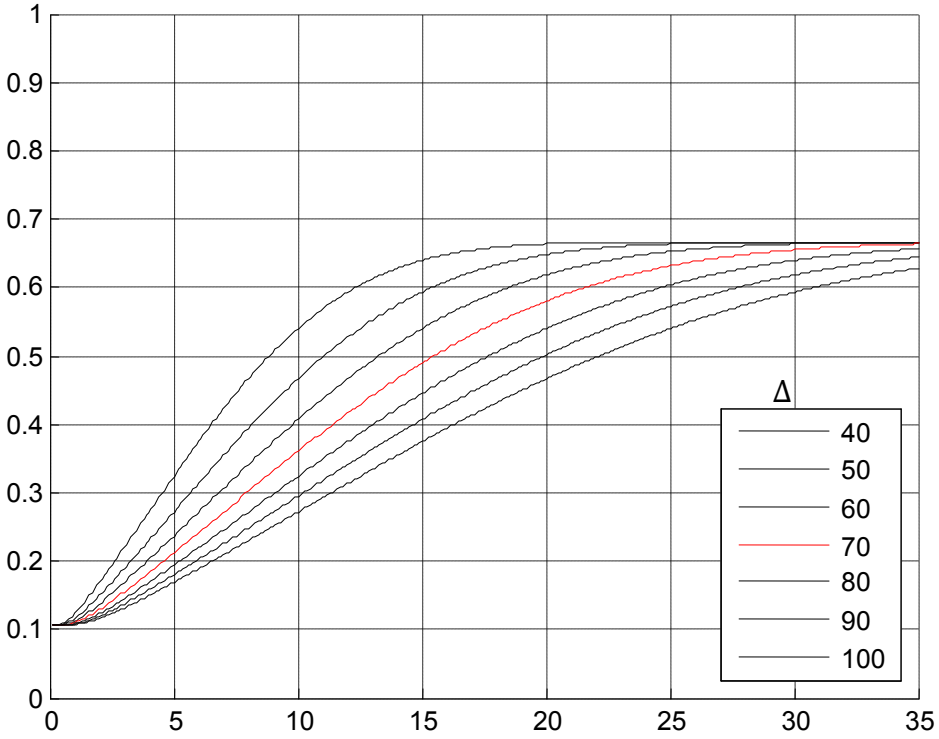
$m = 7, \alpha = \alpha^* = 0.85$

Figure C-12: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 7}$ *and* $\boldsymbol{\alpha = \alpha^*}$.



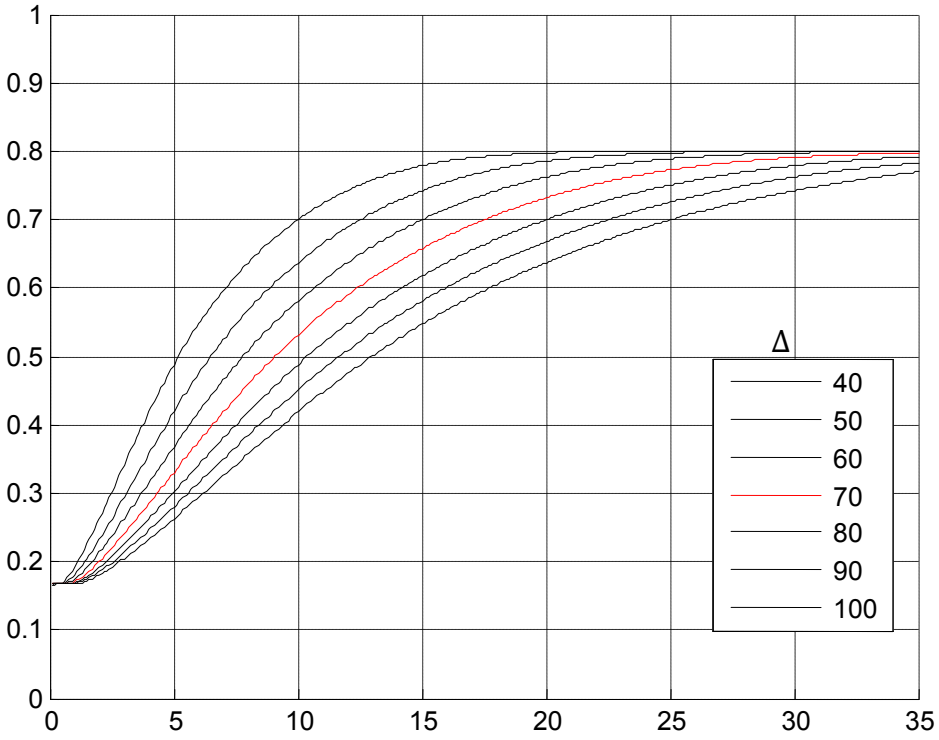
$m = 2, \alpha = \alpha^* - 0.04 = 0.46$

Figure C-13: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 2}$ *and* $\boldsymbol{\alpha = \alpha^* - 0.04}$.



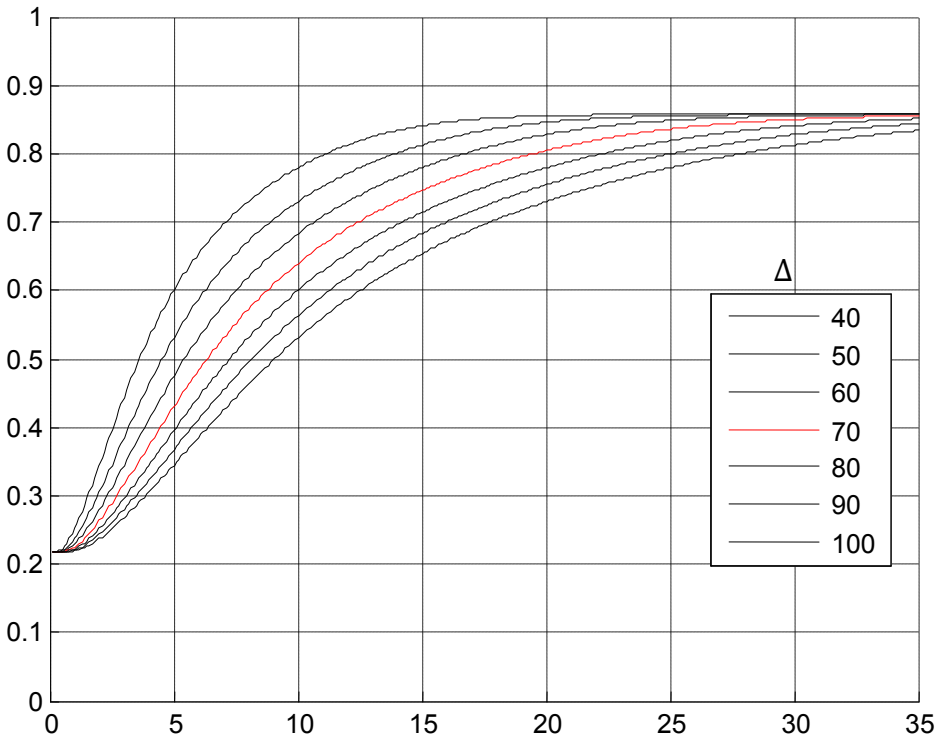
$m = 3, \alpha = \alpha^* - 0.04 = 0.62$

Figure C-14: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 3}$ and $\boldsymbol{\alpha = \alpha^* - 0.04}$.



$m = 5, \alpha = \alpha^* - 0.04 = 0.76$

Figure C-15: $\boldsymbol{P_e}$ as a function of $\boldsymbol{\sigma}$, for $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 5}$ and $\boldsymbol{\alpha = \alpha^*}$.



$m = 7, \alpha = \alpha^* - 0.04 = 0.81$

Figure C-16: $\boldsymbol{P_e}$ *as a function of* $\boldsymbol{\sigma}$*, for* $\Delta \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}\}$, $\boldsymbol{m = 7}$ *and* $\boldsymbol{\alpha = \alpha^* - 0.04}$.

D. Related collaborative R&D Projects

European project: SPY (Surveillance imProved sYstem) project funded by ITEA2 [WEB02]:

The aim of the SPY project is to design, develop and test a new automated and smart system architecture for assistance and surveillance, adapted to a mobile environment, see Figure D-1. This project optimises the use of mobile infrastructures (wireless communications network) and sensors (video cameras, microphones) to assist units in the field (police and fire services) with their day-to-day missions. The SPY project is based on applications such as advanced algorithms to provide users with the most pertinent possible representation of the situation and an automated decision-making help system in changing and unforeseeable contexts. All of this will contribute to improving the reactivity and effectiveness of security and emergency forces in their surveillance, intervention and control missions.

Today, fixed video surveillance cameras are everywhere and police vehicles are increasingly equipped with these devices, however information is only processed at the control centre and is often underused. Understanding the situation is a critical factor, particularly in complex and fast-moving environments, while decisions must be taken rapidly, sometimes based on uncertain information. Security forces need to be able to access reliable and relevant information via optimized media such as video and the interactive display of data to assist them in their day-to-day missions. New solutions for the interactive display of information are made available in SPY, to provide users with an intuitive, understandable and adapted view of the situation.

Figure D- 1: SPY system design.

French project: MEDIEVAL (waterMarking et Embrouillage pour la DIffusion et les Echanges Vidéos et Audios Legalisés) project funded by ANR:

The main idea of this project (see Figure D-2) consists in associating an original technique of partial encryption to watermarking solutions for audio and video digital media, in order to ensure a high security end-to-end transmission. Compared to the watermarking-encryption schemes usually proposed in the state of art, the proposed solution presents several advantages. First, at the client side, the operations of decryption and the watermark insertion are performed in an absolutely inseparable way. Therefore, at no time, the only-watermarked or only-encrypted contents are available. Second, the most complex watermarking operations (computation of the watermark itself and of the way it is inserted) are carried out at the server side on a very small part of the digital content. Hence, very little resources are required at the client terminal. Third, the partial encryption associated to the watermarking allows personalizing a very small part of the media. The major part, common to all users, may then be diffused over super-distribution (or peer-to-peer) networks.
That project objectives yield many technical challenges. Those related to transactional watermarking consist first in generating a specific watermark for every single transaction. This watermark has then to be inserted in the small part of the media that is partially encrypted. The different insertion ways are studied to select the most robust one ensuring the watermark invisibility. A particular effort is dedicated to the global system robustness against hacker attacks in order to jointly optimize the watermarking and the partial encryption. The researches and developments carried out during this project results into a prototype ready to be converted into an industrial product.

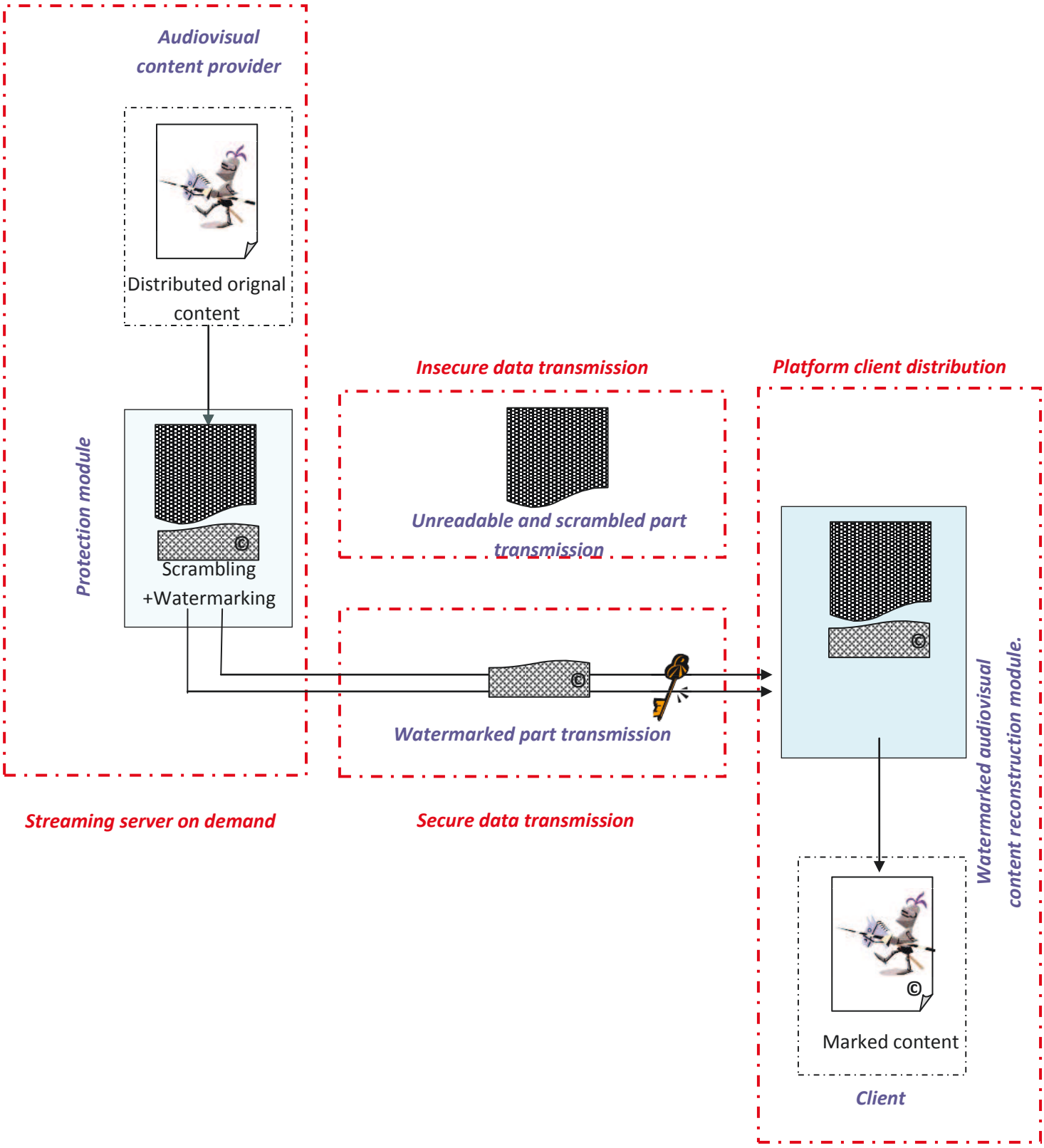


Figure D- 2 : MEDIEVAL architecture.

List of Publications

Journals

1 M. Hasnaoui, M. Mitrea, "Multi-symbol video watermarking", Singal Processing: Image Comminucation, vol. 29(1), pp. 107-127, 2013 (http://dx.doi.org/10.1016/j.image.2013.07.007).
2 M. Mitrea, M. Hasnaoui, "Semi-fragile watermarking between theory and practice", Proceeding of the Romanian Academy, Series A, vol. 14, Special Issue on Criptology Science, pp. 328-337, 2013, (**http://www.acad.ro/sectii2002/proceedings/doc2013-3s/06-MITREA.pdf**)

International conferences

1 M. Hasnaoui, M. Mitrea, "Drift-free MPEG-4 AVC semi-fragile watermarking", accepted for SPIE Electronic Imaging - Media Watermarking, Security, and Forensics, San Francisco, 2014; to be published in Proc SPIE vol. 9028.
2 M. Ammar, M. Mitrea, and M. Hasnaoui, "Saliency map computation in the MPEG-4 AVC compressed stream", accepted for SPIE Electronic Imaging – Human Vision and Electronic Imaging XIX, San Francisco, 2014; to be published in Proc SPIE vol. 9014.
3 M. Hasnaoui, M. Mitrea, "Semi-fragile watermarking for video surveillance application", Proceeding of the European Signal Processing, pp. 1782–1786, Bucharest, 2012
4 M. Hasnaoui, M. Belhaj, M. Mitrea, and F. Preteux, "MPEG-4 AVC stream watermarking by *m*-QIM technique", Proc. SPIE, vol. 7881, 2011.
5 M. Hasnaoui, M. Belhaj, M. Mitrea, and F. Preteux, "MPEG-4 AVC stream watermarking by ST-*m*DM technique", IEEE International Conference on Electronic, Circuits, and Systems, pp. 487–490, Athens, 2011.
6 M.Hasnaoui, M. Mitrea, "Protection du contenu de la vidéo surveillance par tatouage semi-fragile", Proceeding of Ateliers de travail sur le traitement et l'analyse de l'information, pp. 1-8, Hammamet, Tunisie, 2011.
7 M. Hasnaoui, M. Mitrea, M. Belhaj, and F. Preteux, "Visual Quality assessment for motion vector watermarking in the MPEG-4 AVC domain", Fifth International Workshop on Video Processing and Quality Metrics , Scottsdale, U.S.A, 2010. (invited paper).

Oral presentation

1 M. Hasnaoui, M. Mitrea, "Vérification d'intégrité du flux MPEG-4 AVC pour des applications de vidéosurveillance", GDR ISIS, Journée extraction de preuves multimédia: détection de manipulation-identification et authentification de contenus ou personnes, 2013.

R&D projects deliverables (contribution under the supervision of M. Mitrea)

- **MEDIEVS French project**

1 D2.1.1.2- Technique pour la protection de copie: tatouage robuste (May 2010).
2 D3.1- MEDIEVALS – Architecture logicielle et matérielle (October 2011).

- **SPY European project**

1 D5.1.1- Network management: state of the art (August 2011)
2 D5.2.1- Network management: data coding: specification (January 2012)
3 D5.4.3- Integrity techniques: Benchmarking Results (August 2013)

References

[AIT10] K. Ait. Saadi, A. Bouridane, and A. Guessoum, “Combined fragile watermarking and digital signature for H.264/ AVC video authentication,”Proc. Eusipco, pp. 24-28, Scotland, (Aug 2010).

[ALA03] A. M. Alattar, E. T. Lin, and U. M. Celik, “Watermarking low bit-rate advanced simple profile MPEG-4 bitstreams”, IEEE. Trans on Circuits and Systems for Video Technology, Vol. 13 (8), pp. 787-800, (Aug 2003).

[AVC01] I. Avcibas, “Image quality statiscs and their use in steganalysis and compression,” PhD Thesis, Bogazici University, Turkey (2001).

[BAS01] P. BAS, N. V. Boulgouris, F. D. Koravos, J. M. Chassery, M. G. Stintizis, and B. Macq, “Robust watermarking of video objects for MPEG-4 applications,” Proc. SPIE, Vol. 85, (2001).

[BEL10] M. Belhaj, M.Mitrea, S. Duta, and F. Prêteux, “MPEG-4 AVC robust video watermarking based on QIM and perceptual masking,” IEEE International Conference on Communications, pp. 477–480, Bucharest, (Jun 2010).

[BUC94] M. Buckland, F. Gey, “The relationship between Recall and Precision,” Journal of the American Society for Information Science, Vol. 45 (1), pp. 12-19, (Jan 1994).

[CRA98] S. Craver, N. Memon, B.-L. Yeo, and M. Yeung. “Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks and implications”. IEEE Trans. on Selected Areas of Communications , Vol. 16(4), pp. 573–586, (1998).

[CHA00] L. Chang, D. Lou, and T. Chen, “Image authentication and integrity verification via content-based watermarks and public key cryptosystem,” Proc. Image Processing, vol. 3, pp. 649-697, (2000).

[CHE98] B. Chen, G. W. Wornell, “Digital watermarking and information embedding using dither modulation,” IEEE Workshop on Multimedia Signal Processing, pp. 273–278, CANADA, (1998).

[CHE08] S. Chen, H. Leung, “Chaotic Watermarking for Video Authentication in Surveillance Applications,” IEEE Trans on Circuits and Systems For Video Technology, Vol. 18 (5), pp. 704–709, (May 2008).

[COS83] M. Costa, “Writing on dirty paper, “IEEE Transactions on Information Theory, Vol. IT-29, pp. 439 441 (1983).

[COX00] I.J. Cox, M.L. Miller and J.A. Bloom, “Watermarking applications and their properties” Proc. Information Technology: Coding and Computing, pp. 6-10, (2000).

[COX99] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, “Digital Watermarking and Steganography,” 2Nd Ed. ISBN: 978-0123725851

[COX01] I.J. Cox, M.L. Miller and J.A. Bloom, Digital watermarking principles and practices, (2001)

[COX02] I.J. Cox, M.L. Miller, and J.A. Bloom, “Digital Watermarking,” Academic Press, (2002).

[COX07] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, “Digital Watermarking and Steganography,” Morgan Kaufmann, (2007).

[COX08] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, and T. Kalker, “Digital Watermarking and Steganography,” Second edition, Morgan Kaufmann, Burlington, MA, (2008).

[DES12] J. Desai, k. G. Maradia and B. Gohli, “Performance analysis of modulation techniques in GA assisted CDMA wireless communication system with AWGN Rayleigh fading channel,” international journal of scientific & technology research, vol. 1(4), pp. 59-62, (May 2012).

[DIG03] Digimarc Corporation, "Digimarc Comments to USPTO Regarding Technological Protection Systemd for Digitized Copyrighted Works", (2003).

[DUR05] J.Durand.**http://www.liberation.fr/page.php?article=315627**. La caméra, nouvelle arme des policiers européens, (2005).

[EGG13] J. J. Eggers, J. K. Su, and B. Girod, “Scalar costa scheme for information embedding,” IEEE Trans on Signal Processing, Vol. 51 (4), pp. 1003–1019, (2003).

[ESK95] A. M. Eskicioglu, S. P. Fisher, “Image Quality Measures and their performance,” IEEE Trans on Communications, Vol. 43 (12), pp. 2959–2965, (1995)

[EQU08] Equancy&Co, Impact économique de la copie illégale des biens numérisé en France, (2008).

[FIL13] Filmamaker magazine, “Audio watermarks for the second Screen,” (2013)

[FIT00] M. Fitton, “Principles of digital modulation,” TOSHIBA-TREL (berk.tc/combas/digital_mod.pdf).

[GER99] Z. J. Geradts, J. Bijhold, “Forensic Video investigation with real time digitized uncompressed video image sequences,” Proc. SPIE, Investigation and Forensic Science Technologies, vol. 3576, p. 154-164, Boston, (Feb 1999).

[GOL07] J. A. Golikeri. P. Nasiopoulos, and Z. J. Wang, “Robust digital video watermarking scheme for H.264 advanced video coding Standard,” J. Electron. Imaging, Vol. 16 (4), (Jun 2007).

[GON08] X. Gong, H. Lu, “Towards fast and robust watermarking scheme for H.264 video,” IEEE Symp on Multimedia, 649–653, (2008).

[HAA98] G. Haan, E. Bellers, “Deinterlacing: an overview,” Proc. IEEE, Vol. 86 (9), pp. 1839-1857, (Jun 1998).

[HUO11] W. Huo, Y. Zhu, H. Chen, “A controllable error-drift elimination scheme for watermarking algorithm in H.264/AVC stream,” IEEE Signal Processing Letters, 18(9), 535-538, (2011).

[KAL99] T. Kaller, J. Haitsma and M. Maes, “The VIVA project: digital watermarking for broadcast monitoring,” Proc. ICIP, Vol. 2, pp. 202-205, (1999).

[KAL01] T. Kalker, “Considerations on watermarking security,” in Proc. MMSP, pp. 201–206, (2001).

[KER83] A. Kerckhoffs, "La Cryptographie militaire," Journal des Sciences militaires, vol. IX, pp. 5–38, (1883).

[KIM12] T. Kim, K. Park, and Y. Hong, “Video watermarking technique for H.264/AVC,” Opt. Eng, Vol. 51 (4), (2012).

[LAN98] G. Langelaar, R. Lagendijk, and J. Biemond, “Real-time labeling of MPEG2 compressemd video,” Journal of Visual Communication and Image Representation, Vol. 9, pp. 256-270, (Dec 1998).

[LEV01] K. L. Levy, R. S. Hiatt, G. B. Rhoads, User-friendly rights management systems and methods, U.S. Patent Application 10/017,679 (2001).

[LIN99] E.T. Lin, E.J. Delp, “A Review of Fragile Image Watermarks,” Proc. ACM of the Multimedia and Security Workshop, pp. 25-29, Orlando, (Oct 1999).

[MA09] X. Ma, Z. Li, J. Lv, W. Wang, “Data hiding in H.264/AVC streams with limited intra-frame distortion drift,” Computer Network and Multimedia Technology, 1-5, (2009).

[MA10] X. Ma, Z. Li, H. Tu, B. Zhang, “A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift,” IEEE Trans. Circuits and Systems for Video Technology, 20(10), 1320-1330, (2010).

[MAN74] J. Mannos, D. Sakrison, “The effect of a visual fidelity criterion on the encoding of images,” IEEE Trans on Information Theory, vol. 20 (4), pp. 525-536, (Jul 1974).

[MAT13] B. Mathon, P. Bas, F. Cayre and B. Macq, “Impacts of watermarking security on tardos-based fingerprinting,” IEEE. Trans on information forensics and security, vol. 8 (6), pp. 1038-1050, (2013).

[MEM98] N. Memon and P. W. Wong, “Protecting digital media content,” Communications of the ACM, vol. 41(7), (1998).

[MIT07] M. Mitrea, O. Dumitru, F. Prêteux, and A. Vlad, “Zero memory information sources approximating to video watermarking attacks,” Proc. ICCSA, Vol. 4707, pp. 445–459, (2007).

[MIT07] M. Mitrea, F. Prêteux, and J. Nunez, (for SFR and GET), “Procédé de tatouage d’une séquence video”, French patent no. 05 54132 (29/12/2005), and EU patent no. 1804213 (04/07/2007).

[MOR77] H. P. Moravec, “Towards Automatic Visual Obstacle Avoidance,” Proc. IJCAI, Vol. 2, pp. 584-590, San Francisco, (1977).

[NOO05] M. Noorkami, R. M. Mersereau, “Compressed-domain video watermarking for H.264,” Proc. ICIP, pp. 890–893, Atlanta, (Sep 2005)

[JAC67] I. Jacobs, “Comparison of M-ary modulation systems,” Bell Syst. Tech. J., vol. 46, pp. 843-864, (Jun 1967).

[PET98] F. Peticolas, R. Anderson, and M. Kuhn, “Attacks on copyright marking systems,” LNCS, Vol. 1525, pp. 218-238, (1998).

[PET00] F. Peticolas, “Watermarking schemes evaluation,” IEEE Signal Processing Magazine, vol. 17 (5), pp. 58-64, (Sep 2000).

[PRO01] G. J. Proakis, Digital Communications, 4th Ed., McGraw-Hill, (2001).

[PRO05] D. Proforck, H. Richter, M. Schlauweg, and E. Muller, “H.264/AVC video authentication using skipped macroblocks for an erasable watermark,” Proc. SPIE, Vol. 5960, pp. 59–60, China, (2005).

[QUE98] P. M. Queue, “Toward robust content based techniques for image authentication,” Proc. IEEE workshop on Multimedia signal processing, pp. 297-302, Redondo, (Dec 1998)

[RIC03] I. Richardson, H.264 and MPEG-4 Video Compression, the Robert Gordon University, Aberdeen, (2003).

[SAM09] R. Samtani, “Ongoing innovation in digital watermarking,” Computer, Vol. 42 (3), pp. 92-94, (2009).

[SER02] S. V. Serdean, Spread Spectrum-Based Video Watermarking Algorithmsfor Copyright Protection, PhD thesis, university of Plymouth, (2002)

[SHA48] C.E. Shannon, "A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, pp. 379-423, July 1948.

[SHA58] C. E. Shannon, “Channels with Side Information at the Transmitter”, IBM Journal, pp. 289 293, (1958).

[SHA02] K. R. Sharma, S. Decker, “Practical challenges for digital watermarking applications,” EURASIP Journal on Applied Signal Processing, Vol. 2, pp. 133-139, (2002)

[SON09] C. Song, S. Sudirman, and M. Merabti, Recent advances and classification of watermarking techniques in digital images, ISBN: 978-1-902560-22-9, (2009).

[SPA87] Al. Spataru, Fondements de la théorie de la transmission de l’information, Presse Polytechnique, Laussane, ISBN 2-88074-133-0, (1987).

[THI05] S. Thiemert, H. Sahbi, and M. Steinebach, “Applying interest operators in semi-fragile video watermarking,” Proc. SPIE on Electronic imaging: Security, Steganography, and watermarking of Multimedia Contents, vol. 5681, pp. 353-362, San Jose, (Jan 2005).

[THI06] S. Thiemert, H. Sahbi, and M. Steinebach, “Using entropy for image and video authentication watermarks,” Proc. SPIE on Electronic imaging: Security, Steganography, and watermarking of Multimedia Contents, Vol. 6072, pp. 218–228, USA, (Jan 2006).

[TIT99] J. Titman, A. Steinmetz, and R. Steinmetz, “Content based digital signature for motion pictures authentication and content fragile watermarking,” IEEE International Conference on Multimedia computing and systems, vol. 2, pp. 209-213, Florence, (Jun 1999).

[UPA11] S. Upadhyay, K. Singh, “Video Authetication – An Overview,” International Journal of Computer Science & Engineering Survey (IJCSES), vol. 2, No. 4, (Nov 2011).

[WAL02] E. R. Walpole, H. R. Myers, L. S. Myres, and k. Ye, Probability & Statistics for Engineers and Scientists, Pearson Educational International, (2002).

[WAN04] Z. Wang, C. A. Bovik, R. H. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” Trans. IEEE on Image Processing, vol. 13 (4), pp. 600–612, (Apr 2004).

[WAN08] T-Y. Chen, T-H. Chen, and D-J. Wang, “H.264 Video Authentication Based on Semi-Fragile Watermarking,” International Conference on Intelligent Information Hiding and Multimedia Signal Procession, pp. 659–662, Harbin, (Aug 2008).

[WAN10] C. C. Wang, Y. C. Hsu, “Fragile watermarking scheme for H.264 video stream authentication,” Opt. Eng. Vol. 49 (2), (2010).

[WAT93] A. B. Watson, “DCT Quantization Matrices Optimization for individual images,” Proc. SPIE, Vol. 1913, pp. 202–216, (1993).

[WAT01] A. Watson, J. Hu, and J.F. McGowman, “Digital video Quality Metric Based on Human Vision,” Journal of Electronic Imaging , Vol.10, pp. 20-29, (2001).

[WEB01] www.webcamendirect.net

[WEB02] www.ppsl.asso.fr/eng/spy.php

[XU05] X. Xu, M. Tomlinson, M. Ambroze, and M. Ahmed, “Techniques to provide robust and high capacity data hiding of id badges for increased security requirement”, Proc. SETIT, pp. 1-5, Tunisia, (2005).

[YAN02] L. L. Yang, L. Hanzo, “Performances of generalized multicarrier DS-CDMA over Nakagmi-m fading channels,” IEEE. Trans on communications, vol. 50(60), pp. 956-966, (Jun 2002).

[ZAN06] J. Zang, A. T. S. Ho, “Efficient Video Authentication for H.264/AVC,” Proc. ICICIC. Vol. 3, pp. 157-164, London, (2006).

[ZHA10] L. Zhang, Y. Zhu, L.-M. Po, “A novel watermarking scheme with compensation in bit-stream domain for H.264/AVC,” Proc IEEE. ICASSP, 1758–1761, (2010).

[ZOU08] D. Zou, J. Bloom, “H.264/AVC stream replacement technique for video watermarking,” Proc. ICASSP, pp. 1749–1752, Las Vegas, (2008).

List of Acronyms

AAD	Absolute Average Difference
AVC	Advanced Video Coding
ASO	Arbitrary Slice Ordering
B	Bidirectional
BER	Bit Error Rate
bps	bit per second
CABAC	Context Adaptive Binary Arithmetic Coding
CPU	Central Processing Unit
CAVLC	Context Adaptive Variable Length Coding
DCT	Discrete Cosine Transform
DM	Dither Modulation
DFT	Discrete Fourier Transform
DRM	Digital Rights Management
DVQ	Digital Video Quality
FMO	Flexible Macroblock Ordering
GHz	Giga Hirtz
HD	High Definition
HDD	Hard Disk Drive
I	Intra
IF	Image Fidelity
Inf	Inferior
IPR	Intellectual Properties Rights
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
MPEG	Moving Picture Experts Group
m-QIM	Multi symbol Quantization Index Modulation
ms	milli seconde
MSE	Mean Squared Error

Kbits	Kilo bits
NAL	Network Abstraction Layer
NCC	Normalized Cross Correlation
NNZ	Number of Non Zero coefficients
pdf	Probability Density Function
P	Prediction
PSNR	Signal to Noise Ratio
Q	Quantization
QIM	Quantization Index Modulation
RAM	Random Access Memory
RTP	Real-time Transportation Protocol
S1	Scenario 1
S2	Scenario 2
SC	Structural Content
SD	Standard Definition
SI	Side Information
Sup	Superior
ST	Spread Transform
ST-DM	Spread Transform Dither Modulation
SPY	Surveillance imProved sYstem
T	Transformation
UVLC	Universal Variable Length Coding
VEG	Video Expert Group
VLC	Variable Length Coding