Noise sources in robust uncompressed video watermarking

Corneliu Octavian Dumitru

Ecole Doctorale EDITE

Thèse présentée pour l'obtention du diplôme de
DOCTEUR DE L'INSTITUT NATIONAL DES TELECOMMUNICATIONS

Doctorat délivré conjointement par
L'Institut National des Télécommunications et l'Université Pierre et Marie Curie - Paris 6

Spécialité : Télécommunications

DUMITRU CORNELIU OCTAVIAN

Les sources de bruit dans le tatouage robuste de vidéo non-compressée

Soutenue le 11 Janvier 2010 devant le jury composé de :

Pr. Patrick GALLINARI	Président
Pr. Frédéric TRUCHETET	Rapporteur
Pr. Peter SCHELKENS	Rapporteur
Pr. Patrick GALLINARI	Examinateur
Pr. Christine GRAFFIGNE	Examinateur
Dr. Gérard MOZELLE	Examinateur
Pr. Inge GAVAT	Invitée
Pr. Françoise PRETEUX	Directeur de thèse
MdC. Mihai MITREA	Co-encadrant

Thèse n° 2010TELE0001

PhD THESIS
UNIVERSITY PIERRE and MARIE CURIE

Noise Sources in Robust Uncompressed Video Watermarking

Corneliu-Octavian DUMITRU

"The mind is like an umbrella - it only works when it is open."

Sir James Hopwood Jeans (1877-1946)

(British physicist, astronomer and mathematician, Royal Astronomical Society, 1922)

Table of Contents

Acknowledgement:

First, I would like to thank my supervisor, Professor Françoise PRÊTEUX, for giving me the opportunity to join the ARTEMIS group at Télécom SudParis (ex. INT) and to do the research in her laboratory for the last three years and a half.

Second, I would also to thank to my co-supervisor, Dr. Mihai MITREA, for bringing the problem of natural video and watermarking attacks modelling to my attention, for his guidance, and good discussions.

Next, I would like to express my gratitude to Professor Frédéric TRUCHETET and Professor Peter SCHELKENS for the time spent in reviewing this thesis, and for their helpful comments on improving its content. I am also thankful to Professor Patrick GALLINARI, Professor Christine GRAFFIGNE and Dr. Gérard MOZELLE that have agreed to be members of the examination jury.

My foremost acknowledgment must go to my Professor Inge GAVĂT, invited member of the jury, for her support in every aspect from the beginning of my career as a young researcher in 2001. Professor Inge GAVĂT was the one who persuaded me to continue my studies and inspired me with her enthusiasm on research, her experience (in the field of signal processing especially speech/audio recognition but also in the field of information theory), and her good advice (scientific and personal). In particular, I thank her for the opportunity she gave me to work in a team so valuable as that Human Machine Interface.

My thank also goes to Dr. Cătălin FETIȚA, Dr. Marius PREDA, Dr. Titus ZAHARIA and Dr. Nicolas ROUGON, Associate Professors at the ARTEMIS Department for their stimulating and unconditional help.

I am grateful to Ms. Evelyne Taroni, for her continuous support and making the administrative work seems easy.

I thank all my former and actual ARTEMIS Department colleagues: Adriana, Afef, Amaury, Andrei, Ashutosh, Benôit, Blagica, Franck, Ivica, Khaled, Kuang Che, Louis-Arnaud, Maher, Marc, Margarete, Marwen, Matthieu, Perrine, Thomas, Sameh, Son-Minh, Srgjan, Van Luyen for making the life in the laboratory enjoyable and sometimes fun. Special thanks to my two officemate Sorin and Veronica. Of course, it is impossible to mention everyone; I apologize to those whom I omitted.

Without giving names, I would like to thank to all special people from the *"Romanian Group"* I met here in France (especially in Paris, Limours and Evry).

I owe special gratitude to my parents and my family for their continuous, unconditional love and support, and for giving me the strength and motivation to overcome all difficulties especially in the last year.

Abstract:

The process of imperceptibly and persistently inserting into a digital content of some additional information is referred to as *watermarking*. Watermarking is a viable solution to a large variety of multimedia applications like authentication, broadcast monitoring, copy protection, e-Commence, rights management, *etc*.

The original content may be a still image, an audio or video excerpt, some 3D data, a text... The additional information generally refers to some copyright information (a logo, time stamp, *etc*) and will be referred as watermark.

The digital watermark needs to be transparent (imperceptible) and robust against the attacks (which are transforms designed by malicious users in order to damage or eliminate the watermark from the watermarked content).

From the theoretical point of view, a watermarking system can be modelled as a noisy channel. The watermark is a sample from the information source and should be recovered at the detection side. The elements that make the watermark detection difficult can be modelled as the channel noise, namely: the original content (be it represented into some transform domain or not) and the attacks. The watermarking procedure itself plays the role of the modulation technique.

This model brings into evidence that the design of a reliable watermarking technique cannot be achieved without an in-depth knowledge of the noise (*i.e.* original content and attacks) statistical model. Despite suspicion, reticence and even contradictory results, in the absence of any sound theoretical result, such a model is generally assumed to be Gaussian.

The present thesis clarifies this problem: it focuses on natural video and attack modelling for uncompressed video watermarking purposes.

By reconsidering a statistical investigation combining four types of statistical tests, the thesis starts by identifying with accuracy the drawbacks and limitations of the popular Gaussian model in watermarking applications. Further on, an advanced statistical approach is developed in order to establish with mathematical rigour:

1. that a mathematical model for the original video content and/or attacks exists;

2. the model parameters.

From the theoretical point of view, this means to prove for the first time the stationarity of the random processes representing the natural video and/or the watermarking attacks.

These general results have been already validated under applicative and theoretical frameworks. On the one hand, when integrating the attack models into the *IProtect* watermarking method patented by Institut Télécom/ARTEMIS and SFR, a speed-up by a factor of 100 of the insertion procedure has been obtained. On the other hand, accurate models for natural video and attacks allowed the increasing of the precision in the computation of some basic information theory entities (entropies and capacity).

Résumé

Le tatouage numérique s'impose aujourd'hui comme une solution unificatrice pour un large éventail d'applications multimédia, depuis la protection des contenus numériques jusqu'au contrôle de leur diffusion sur tout type de support.

Le principe du tatouage consiste à insérer une information additionnelle, appelée marque, au contenu du média d'origine, qu'il s'agisse des données image, audio, vidéo ou graphique 2D/3D.

Les enjeux scientifiques et méthodologiques du tatouage, illustrés figure AF.1, concernent :

1. la taille de la marque, *i.e.* la quantité d'information optimale à insérer,
2. la transparence, *i.e.* l'imperceptibilité des données additionnelles par rapport au contenu d'origine,
3. la robustesse, *i.e.* la capacité à détecter la marque tout au long du cycle de vie du contenu marqué et en particulier après tout type d'attaque.

Du point de vue de la théorie de l'information, un système de tatouage peut être modélisé comme un canal de bruit (figure AF.1). La marque est un échantillon de la source d'information, qui doit être retrouvée à la détection. Tous les autres éléments, *i.e.* le contenu d'origine (représenté dans le domaine d'une transformée ou non), ainsi que les attaques (malveillantes ou non) correspondent au bruit de canal. La procédure du tatouage elle-même joue le rôle de la modulation.

Dans ce cadre, concevoir une procédure de tatouage efficace nécessite de modéliser le bruit de canal.

Cette thèse traite de ce verrou théorique pour des vidéos naturelles. Les contributions scientifiques développées dans la thèse ont permis :

1. De réfuter mathématiquement le modèle gaussien en général adopté dans la littérature pour représenter le bruit de canal.
2. D'établir pour la première fois, le caractère stationnaire des processus aléatoires représentant le bruit de canal, la méthode développée étant indépendante du type de données, de leur traitement et de la procédure d'estimation.
3. De proposer une méthodologie de modélisation du bruit de canal à partir d'un mélange de gaussiennes pour une transformée aussi bien en cosinus discrète qu'en ondelette discrète et pour un large ensemble d'attaques (filtrage, rotation, compression, StirMark, …). L'intérêt de cette approche est entre autres de permettre le calcul exact de la capacité du canal alors que la littérature ne fournissait que des bornes supérieures et inférieures.
4. Les contributions technologiques concernent l'intégration et l'implémentation de ces modèles dans la méthode de tatouage *IProtect* brevetée par l'Institut Télécom/ARTEMIS et SFR avec un gain en temps d'exécution d'un facteur 100 par rapport à l'état de l'art.

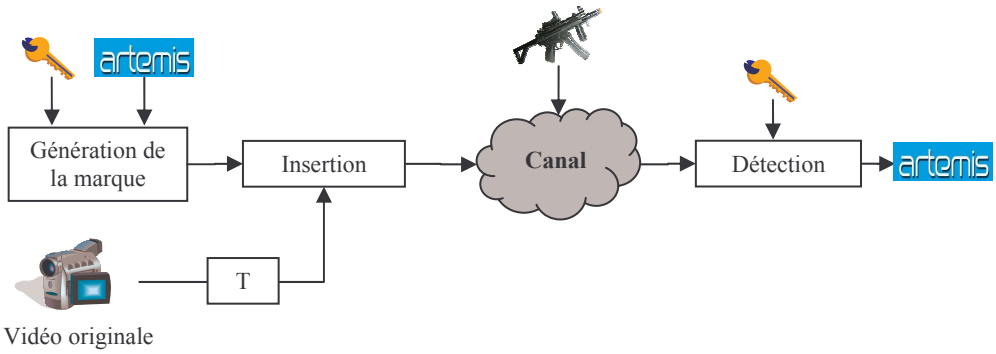


Figure AF.1. Schéma de tatouage modélisé par un canal bruité.

A noter que toutes les activités de cette thèse ont été réalisées au sein du département ARTEMIS et en liens plus on moins étroits avec les partenaires industriels de différents projets du département comme, SFR (dans le cadre des projets *TAMUSO* et *DWD*) et HD3D SAS (dans le cadre du projet *HD3D-IIO* du pôle de compétitivité – *CapDigital*).

La thèse est structurée en quatre parties :

I. Partie I - Introduction

La première partie à caractère introductif est organisée en deux chapitres.

Le premier présente le contexte du tatouage numérique, depuis les enjeux économiques liés aux pertes financières considérables dues au piratage numérique, jusqu'à l'offre hétérogène de produits commerciaux, encore inadaptés aux besoins des applications. Il en ressort que l'approche pragmatique actuellement mise en place reste peu efficace et montre la nécessité de disposer d'un cadre théorique fiable et unitaire pour le tatouage numérique.

Le deuxième chapitre traite de ce défi et décrit les fondements théoriques sur lesquels reposent les méthodes et les expériences développées dans la deuxième et troisième partie de la thèse. Ainsi, définitions, propriétés et modèles mathématiques relevant des variables et processus aléatoires, de la théorie de l'information et des statistiques sont ils rappelés et discutés au regard du problème posé.

II. Partie II – Modélisation de la vidéo naturelle

La deuxième partie de la thèse (Chapitre 3) est consacrée à la modélisation mathématique de vidéo naturelle. Après avoir identifié les limitations théoriques et applicatives des méthodes de l'état de l'art, un nouvel algorithme *ART.MOD-V* est introduit. Il s'agit d'une procédure statistique originale qui permet de gérer mathématiquement les difficultés générales liées à la dépendance des données vidéos et à leur non-stationnarité *a priori*. Les modèles, pour les transformées les plus couramment utilisées (TOD – transformée en ondelette discrète et TCD – transformée en cosinus discrète), sont

calculés et discutés. Nous avons établi qu'ils peuvent être considérés comme des valeurs de référence pour la vidéo naturelle : les modèles ne dépendent pas de la séquence vidéo sur laquelle ils ont été calculés, ils sont indépendants de la procédure d'estimation, et leur précision est très élevée (erreurs relatives moyennes entre 5% et 10%).

II.1. Description de l'algorithme

ART.MOD.V est structuré en deux parties: calcul du modèle et validation du modèle.

Les étapes (*étape A* et *étape B*) du calcul du modèle sont illustrées sous forme d'organigramme figure AF.2.a. L'*étape A* est dédiée au pré-traitement de la vidéo et à la transformation TOD / TCD afin de disposer du vecteur des coefficients après échantillonnage. *L'étape B* concerne la définition et le calcul du modèle par mélange de gaussiennes : l'*étape B* est appliquée successivement à chaque *rang* $r = 1, 2, ..., R$ obtenu à l'*étape A*.

L'*étape C* correspond à la validation du modèle (figure AF.2.b) pour en préciser la généralité, *i.e.* son indépendance par rapport à la

- *C – (1) :* mesure de similarité et la définition de l'erreur (entrée : le modèle $\hat{p}(x)$ et les D estimations $\hat{p}^i(x)$; sortie : confirmation/infirmation du modèle $\hat{p}(x)$).
- *C – (2) :* séquence vidéo (entrée : le modèle $\hat{p}(x)$ et les D estimations $\hat{p}^i(x)$; sortie : confirmation/infirmation du modèle $\hat{p}(x)$).
- *C – (3) :* procédure d'estimation - appliquée pour chaque *rang* (entrée : le modèle $\hat{p}(x)$ et les R vecteurs *rang* ; sortie : confirmation/infirmation du modèle $\hat{p}(x)$).

II.2. Résultats expérimentaux

Le corpus expérimental est composé de 10 séquences vidéos, issues de différents films d'environ 25 minutes (35000 trames) chacune. Les contenus sont hétérogènes. Chaque vidéo est codée en DivX en basse et haute qualité :

- trames de 192x80 pixels codées à 64 kbit/s, correspondant à un débit pour un réseau mobile ; c'est le cas illustré dans ce résumé ;
- trames de 640x480 pixels codées à 512 kbit/s correspondant à une distribution par Internet ; ce cas est détaillé dans le mémoire.

En l'absence de toute preuve mathématique concernant la stationnarité/ergodicité de la vidéo naturelle, ces 10 séquences ont été considérées soit individuellement, soit sous forme combinée, selon trois schémas :

- IVM (Individual Video Modelling) : une seule séquence a été choisie (aléatoirement) et a été traitée dans les *étapes A* et *B* de l'algorithme ; les *nr_vid* = 9 (*nr_vid* - number of video). Les autres séquences ont servi à la validation, étape *C – (2)*.

- MVM (Multiple Video Modelling) : les *étapes A* et *B* de l'algorithme sont appliquées à une séquence vidéo composée des quatre-quarts de quatre séquences naturelles. Les validations de type *C – (2)* sont réalisées avec des séquences construites à partir des quarts d'autres séquences (dans les expérimentations de ce type, *nr_vid* = 3).
- MSVM (Multiple Shuffled Video Modelling) : cette expérimentation est similaire à la précédente, à la différence que les trames des 10 séquences naturelles sont préalablement mélangées aléatoirement (ici, *nr_vid* = 3).

Ces trois types de modélisation ont un caractère incrémental, les expérimentations s'arrêtant dès lors que le modèle est validé pour les trois méthodes à l'*étape C*.

Dans ce résumé, les résultats sont présentés selon la transformation appliquée à la vidéo (TOD ou TCD) et selon le type de modélisation (IVM, MVM et MSVM).

Résultats concernant la modélisation des séquences vidéos dans le domaine de la TOD.

Le tableau AF.1.a présente les modèles IVM, *i.e.* $P(k)$, $\mu(k)$ et $\sigma(k)$ décrivant un mélange de 10 lois gaussiennes $\hat{p}(x) = \sum_{k=1}^{K} P(k)\, p_k(x)$, $p_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2})$ (les lignes de ce tableau correspondent aux 3 rangs, r = 1, 100, 200). La figure AF.3 illustre la forme de cette densité de probabilité et la compare à l'histogramme des coefficients.

Bien que l'erreur ε dans le calcul de ce type de modèle soit petite (de l'ordre de 4%, *cf.* tableau AF.2, col 1), il ressort que le modèle ne présente aucune généralité, les erreurs de type ε' et ε'' étant assez élevées (*cf.* tableau AF.2).

En conséquence, on peut affirmer qu'un tel modèle IVM (*i.e.* calculé à partir d'une seule séquence vidéo) ne permet pas de représenter une autre séquence vidéo.

L'analogie se poursuit avec le calcul des modèles de type MVM (*cf.* tableau AF.1.b et figure AF.4). Cette fois-ci, on peut admettre qu'on a obtenu un modèle général : les erreurs de calcul (tableau AF.3, col 1) sont inférieures à 5% tandis que les erreurs de validation ε' (tableau AF.3, col 2) et ε'' (tableau AF.3, col 3-5) sont inférieures à 10%, E_{HL} et E_{DL} étant proches de zéro.
En outre, dans presque tous les cas, ces probabilités calculées via MVM appartiennent à 95% aux limites de confiance statistique (*étape C – (3)* dans la validation du modèle $\hat{p}(x)$).

Tableau AF.1.a. IVM : hiérarchie des coefficients de la TOD (9,7) d'une vidéo en basse qualité.

Rang	Paramètres du modèle										
1	*P(k)*	0,075	0,070	0,074	0,134	0,071	0,131	0,151	0,098	0,060	0,137
	μ(k)	0,575	0,405	0,646	0,497	0,544	0,535	0,627	0,628	0,654	0,850
	σ(k)	0,112	0,076	0,138	0,089	0,100	0,098	0,062	0,133	0,140	0,125
100	*P(k)*	0,064	0,101	0,056	0,070	0,090	0,080	0,216	0,073	0,043	0,207
	μ(k)	0,086	0,094	0,084	0,130	0,090	0,090	0,090	0,090	0,100	0,106
	σ(k)	0,033	0,025	0,035	0,039	0,026	0,026	0,024	0,025	0,040	0,015
200	*P(k)*	0,058	0,178	0,057	0,083	0,099	0,056	0,068	0,054	0,157	0,189
	μ(k)	0,041	0,036	0,040	0,059	0,043	0,041	0,043	0,041	0,032	0,033
	σ(k)	0,013	0,014	0,013	0,019	0,012	0,013	0,012	0,013	0,007	0,009

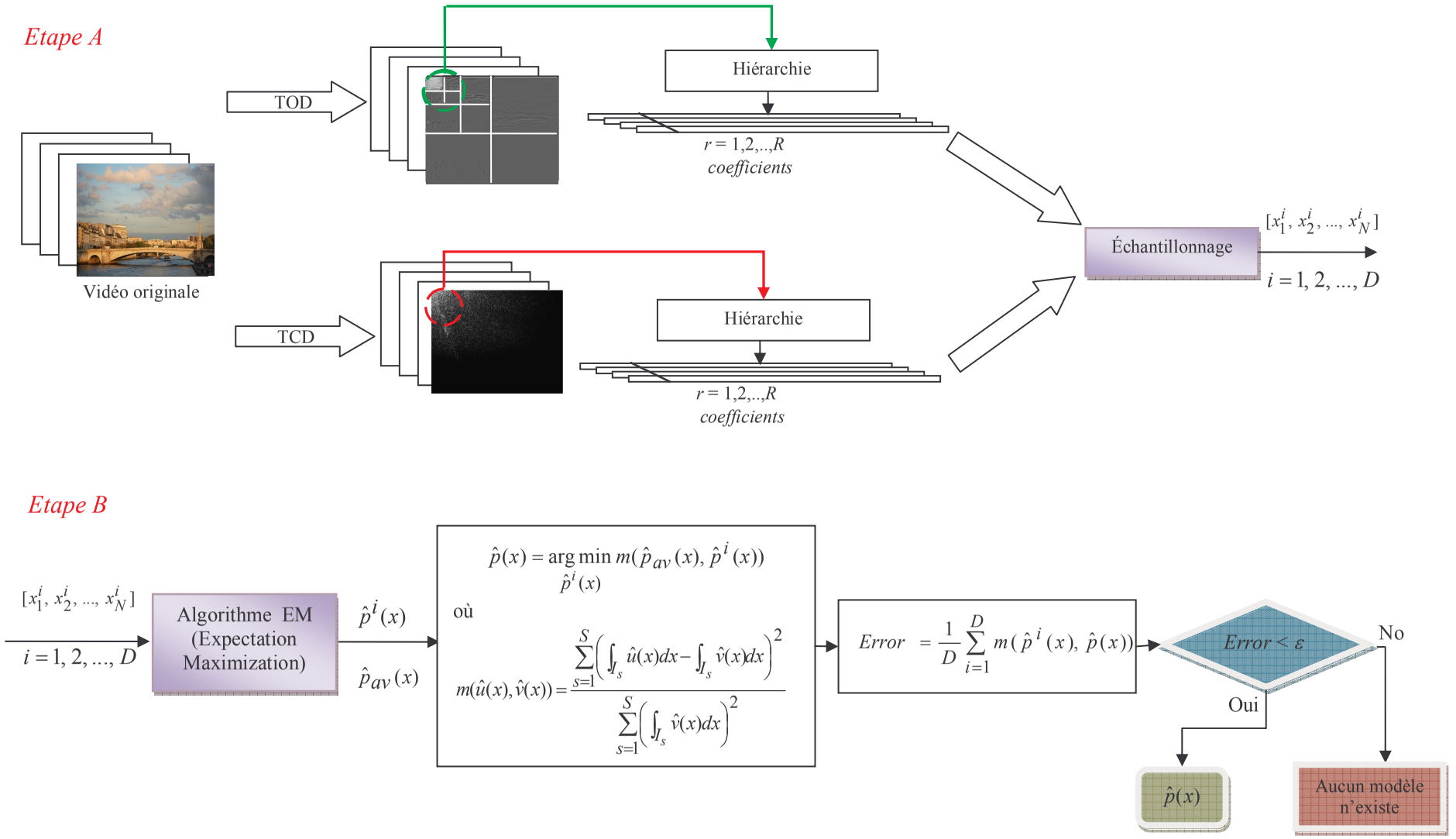


Figure AF.2.a. Calcul du modèle selon l'algorithme *ART.MOD-V* développé (*étape A* et *B*).

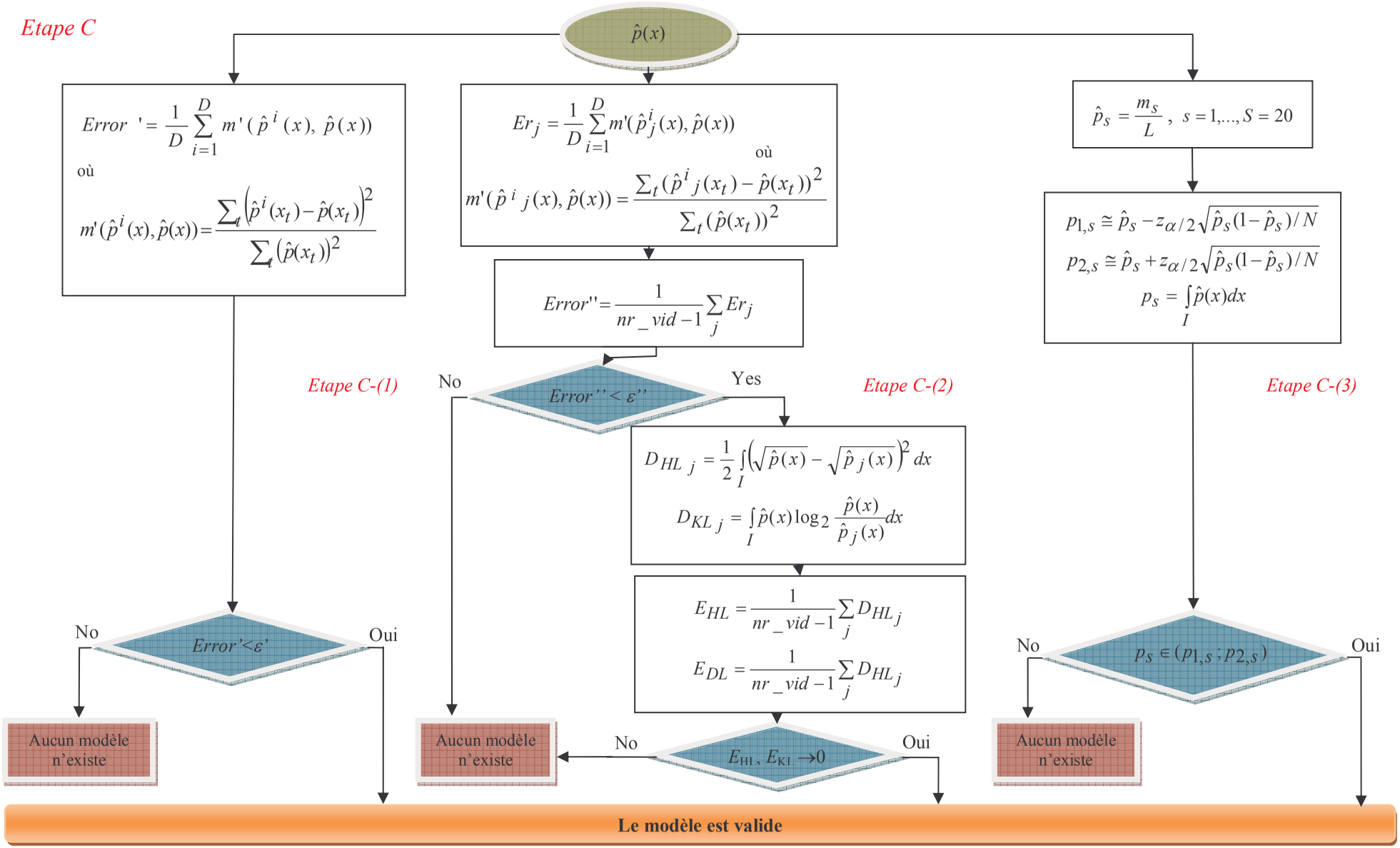


Figure AF.2.b. Validation du modèle calculé selon l'algorithme *ART.MOD-V* (*étape C*).

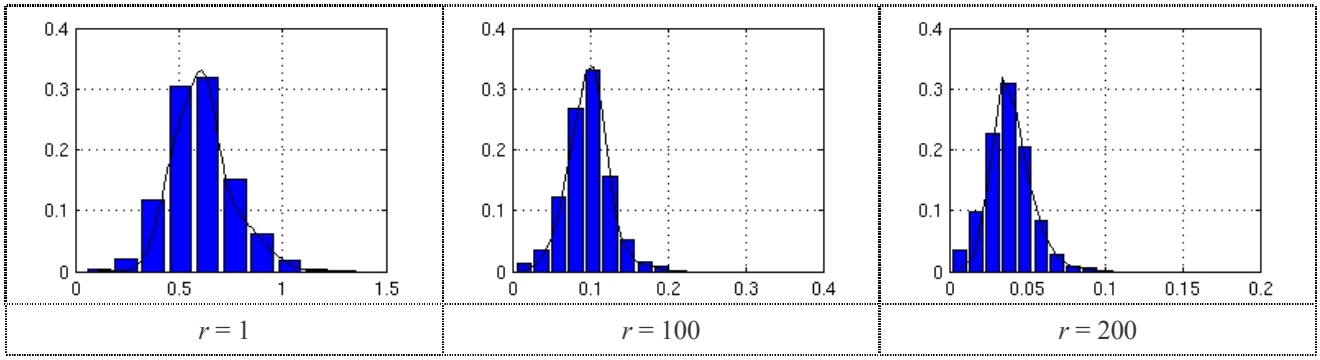


Figure AF.3. Modèle IVM et histogrammes calculés pour trois rangs sur la TOD d'une vidéo en basse qualité.

Tableau AF.1.b. MVM : hiérarchie des coefficients de la TOD (9,7) pour une vidéo en basse qualité.

Rang	Paramètres du modèle										
1	*P(k)*	0,129	0,046	0,100	0,133	0,110	0,116	0,060	0,133	0,060	0,110
	μ(k)	0,574	0,685	0,443	0,566	0,314	0,383	0,515	0,514	0,431	0,299
	σ(k)	0,125	0,226	0,142	0,128	0,108	0,111	0,141	0,141	0,139	0,105
100	*P(k)*	0,114	0,110	0,129	0,127	0,069	0,020	0,077	0,129	0,119	0,105
	μ(k)	0,070	0,053	0,025	0,091	0,091	0,013	0,079	0,049	0,057	0,107
	σ(k)	0,018	0,018	0,011	0,023	0,023	0,001	0,024	0,016	0,018	0,019
200	*P(k)*	0,026	0,061	0,066	0,154	0,143	0,086	0,251	0,072	0,127	0,013
	μ(k)	0,030	0,032	0,030	0,022	0,045	0,032	0,009	0,033	0,025	0,014
	σ(k)	0,009	0,010	0,009	0,006	0,007	0,010	0,004	0,010	0,006	0,001

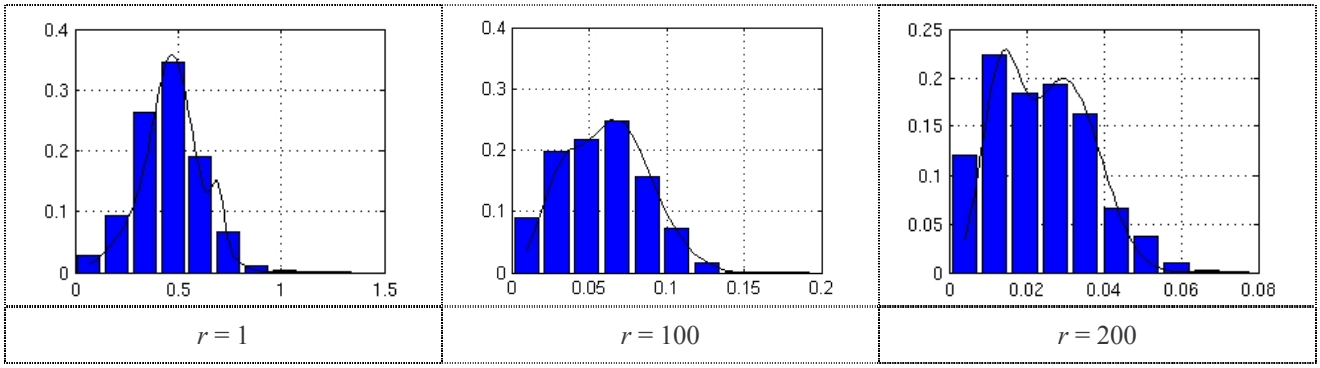


Figure AF.4. MVM et histogrammes calculés pour trois rangs sur la TOD d'une vidéo en basse qualité.

Tableau AF.2. Erreurs de calcul et de validation - IVM.

Rang	Etape B	Etape C- (1)	Etape C- (2)		
	Erreur ε	***Erreur ε'***	***Erreur ε''***	E_{HL}	E_{KL}
1	0,039	0,107	0,119	0,176	0,566
100	0,002	0,084	0,097	0,104	0,445
200	0,010	0,061	0,060	0,074	0,223

Tableau AF.3. Erreurs de calcul et de validation - MVM.

Rang	Etape B	Etape C- (1)	Etape C- (2)		
	Erreur ε	***Erreur ε'***	***Erreur ε''***	E_{HL}	E_{KL}
1	0,038	0,076	0,047	0,012	0,031
100	0,036	0,056	0,081	0,039	0,191
200	0,043	0,047	0,042	0,019	0,178

Résultats concernant la modélisation des séquences vidéo dans le domaine de la TCD.

L'investigation statistique est répétée maintenant dans le domaine de la TCD, qu'elle soit appliquée sur des trames complètes (illustrée dans ce résume) ou sur des blocs 4x4 et 8x8 (illustrée dans le mémoire).

Le calcul des modèles de type IVM (tableau AF.4 et figure AF.5) a mis en évidence le même comportement que dans le cas de la TOD : des erreurs de calcul très petites, mais des erreurs de validation assez grandes ($\varepsilon' < 10\%$, $\varepsilon'' > 75\%$ - tableau AF.5).

L'étape suivante (*i.e.* le calcul des modèles de type MVM, *cf.* tableau AF.6 et figure AF.6) a démontré que l'hypothèse de stationnarité ne peut pas encore être acceptée : les erreurs de validation ($\varepsilon' < 10\%$, $\varepsilon'' < 75\%$ - tableau AF.7) restent grandes.

En conséquence, dans le domaine de la TCD, le calcul des modèles de type MSVM a été obligatoire (tableau AF.8) : c'est seulement ainsi qu'une même densité de probabilité a pu représenter (dans le sens antérieurement discuté) l'intégralité du corpus vidéo : les erreurs de validation (*cf.* tableau AF.9) ε', ε'' et E_{HL} et E_{DL} étant acceptables.

Dans presque tous les cas, ces probabilités calculées en utilisant la MSVM appartiennent à 95% aux limites de confiance statistique.

Tableau AF.4. IVM : hiérarchie des coefficients de la TCD (trame complète) d'une vidéo en basse qualité.

Rang	Paramètres du modèle										
1	$P(k)$	0,065	0,129	0,065	0,177	0,114	0,099	0,129	0,071	0,064	0,085
	$\mu(k)$	5,646	3,361	5,951	4,817	4,717	4,927	8,096	5,710	6,411	5,215
	$\sigma(k)$	1,141	6,894	1,258	5,233	9,309	1,027	9,070	1,170	1,323	8,906
100	$P(k)$	0,098	0,088	0,095	0,103	0,055	0,111	0,077	0,073	0,209	0,091
	$\mu(k)$	0,670	0,567	0,568	0,603	0,546	0,532	0,450	0,567	0,603	0,680
	$\sigma(k)$	0,103	0,120	0,118	0,108	0,124	0,079	0,114	0,121	0,033	0,064
200	$P(k)$	0,093	0,171	0,121	0,119	0,126	0,021	0,066	0,102	0,160	0,034
	$\mu(k)$	0,382	0,369	0,386	0,418	0,366	0,089	0,376	0,386	0,271	0,353
	$\sigma(k)$	0,069	0,031	0,034	0,062	0,048	0,022	0,069	0,069	0,040	0,068

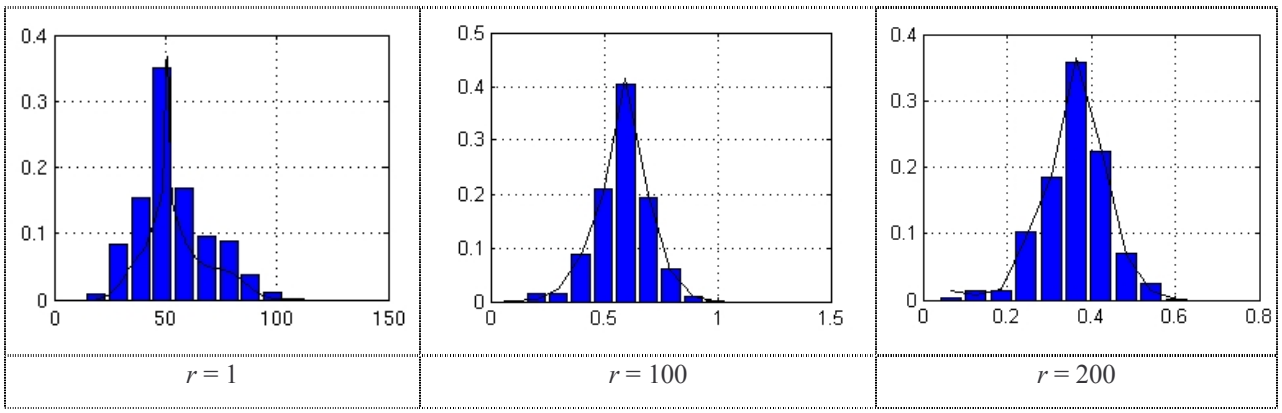


Figure AF.5. IVM et histogrammes calculés pour trois rangs sur la TCD d'une vidéo en basse qualité.

Tableau AF.5. Erreurs de calcul et de validation - IVM.

Rang	Etape B	Etape C- (1)	Etape C- (2)
	Erreur ε	***Erreur ε'***	***Erreur ε''***
1	0,026	0,074	0,579
100	0,030	0,078	0,945
200	0,036	0,070	0,901

Tableau AF.6. MVM : hiérarchie des coefficients de la TCD (trame complète) d'une vidéo en basse qualité.

Rang	Paramètres du modèle										
1	*P(k)*	0,106	0,096	0,112	0,033	0,177	0,081	0,119	0,092	0,031	0,152
	μ(k)	4,364	4,761	4,490	2,333	1,516	4,704	4,437	5,011	2,727	3,475
	σ(k)	8,410	9,726	8,300	1,439	2,999	9,528	8,387	1,021	1,546	5,113
100	*P(k)*	0,007	0,040	0,164	0,036	0,043	0,114	0,083	0,113	0,148	0,252
	μ(k)	1,025	0,616	0,346	0,610	0,508	0,445	0,506	0,675	0,425	0,298
	σ(k)	0,001	0,021	0,106	0,008	0,099	0,109	0,099	0,064	0,076	0,092
200	*P(k)*	0,097	0,168	0,062	0,080	0,108	0,053	0,108	0,165	0,081	0,078
	μ(k)	0,222	0,385	0,456	0,126	0,218	0,296	0,276	0,212	0,295	0,253
	σ(k)	0,060	0,033	0,105	0,042	0,060	0,080	0,076	0,061	0,079	0,068

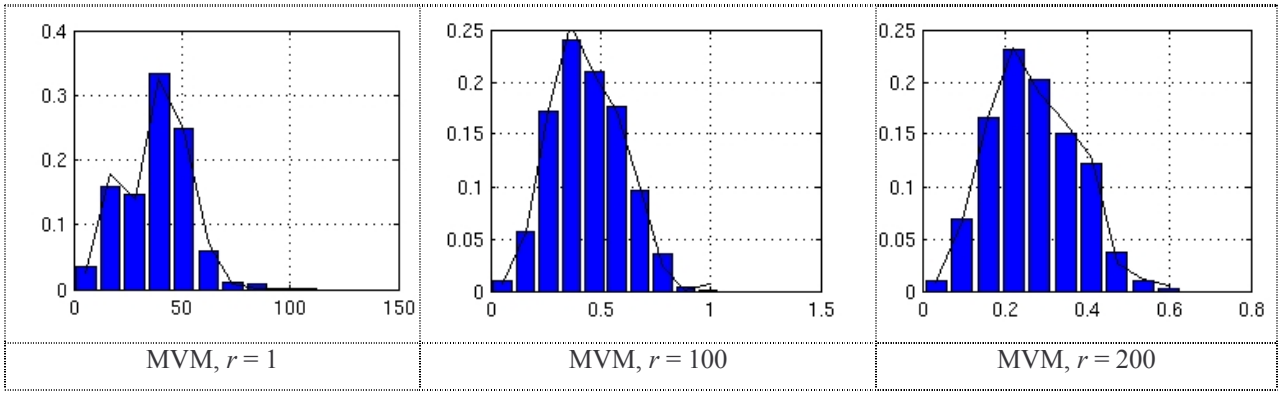


Figure AF.6. MVM et histogrammes calculés pour trois rangs, sur la TCD d'une vidéo en basse qualité.

Tableau AF.7. Erreurs de calcul et de validation - MVM.

Rang	Etape B	Etape C- (1)	Etape C- (2)
	Erreur ε	***Erreur ε'***	***Erreur ε''***
1	0,030	0,078	0,678
100	0,044	0,078	0,661
200	0,034	0,100	0,783

Tableau AF.8. MSVM : hiérarchie des coefficients de la TCD d'une vidéo en basse qualité.

Rang	Paramètres du modèle										
1	*P(k)*	0,047	0,384	0,280	0,013	0,055	0,082	0,001	0,053	0,029	0,056
	μ(k)	3,336	0,141	0,141	2,348	4,533	5,063	0,149	7,526	5,011	5,745
	σ(k)	3,524	0,058	0,058	1,469	1,930	3,968	0,058	9,989	8,723	7,853
100	*P(k)*	0,027	0,074	0,047	0,214	0,115	0,067	0,054	0,161	0,025	0,216
	μ(k)	0,782	0,205	0,434	0,187	0,195	0,079	0,568	0,186	0,255	0,596
	σ(k)	0,033	0,057	0,108	0,057	0,058	0,007	0,075	0,056	0,078	0,060
200	*P(k)*	0,148	0,047	0,081	0,138	0,090	0,115	0,089	0,070	0,106	0,116
	μ(k)	0,154	0,302	0,278	0,372	0,396	0,332	0,268	0,351	0,265	0,313
	σ(k)	0,040	0,087	0,066	0,021	0,076	0,092	0,044	0,089	0,042	0,091

Tableau AF.9. Validation du modèle MSVM, TCD et quatre rangs.

Rang	Etape B	Etape C- (1)	Etape C- (2)		
	Erreur ε	***Erreur ε'***	***Erreur ε''***	E_{HL}	E_{KL}
1	0,004	0,028	0,301	0,122	0,378
100	0,058	0,100	0,308	0,293	0,589
200	0,057	0,096	0,154	0,106	0,449

II.3. Conclusion

Cette deuxième partie de la thèse a permis de déterminer pour la première fois rigoureusement quel est le périmètre de validité de l'hypothèse de stationnarité dans les séquences vidéos. Chaque séquence considérée, qu'elle le soit à partir d'un modèle IVM, ou MVM ou encore MSVM, a abouti à la création d'un modèle statistique. Toutefois, dans le cas IVM, la régularité s'arrête là : les différents modèles mathématiques sont obtenus pour des séquences vidéos différentes. Dans les cas MVM et MSVM, le modèle mathématique obtenu est indépendant des données sur lesquelles il a été calculé. Par conséquent, parler d'ergodicité de séquences vidéo, impose de considérer des séquences vidéos combinées (MVM -TOD) ou des séquences vidéo combinées et mélangées (MSVM - TCD) plutôt que des séquences vidéos individuelles (IVM).

Ces modèles généraux (MVM - TSD ou MSVM - TCD) ont été validés pour un large éventail d'outils: mesures de similarité et d'erreurs, estimation de la limite de confiance, distance Hellinger et divergence Kullback-Leibler

L'intérêt de ce modèle général est d'être explicitable directement dans de nombreuses applications :

- Compression / débruitage

  Ici, il s'agit de disposer d'un modèle unique qui peut être utilisé sur des larges séquences vidéos. Par conséquent, notre procédure d'estimation doit être appliquée à la séquence vidéo combinée (pour TOD) et à la séquence des vidéos combinées et mélangées (pour TCD).

- Indexation / classification

  Dans ce cas, il est nécessaire que les modèles soient capables d'identifier les particularités des données. Par conséquent, la procédure d'estimation doit être appliquée sur des séquences vidéos individuelles (pour les deux transformées TOD et TCD).

- Tatouages

  Pour cette classe d'applications, les deux modèles combinés et individuels peuvent être utiles pour TOD et TCD.

  Pour les méthodes d'étalement du spectre, les techniques d''insertion/détection sont indépendantes à l'égard du contenu vidéo ; un modèle combiné devrait prouver son efficacité.

  Pour les méthodes par information de bord, la technique d'assemblage doit exploiter les caractéristiques de la vidéo originale : un modèle estimé sur cette vidéo particulière s'avère adapté et utile.

Enfin, l'utilité de ces modèles est illustrée par une estimation de l'entropie de la vidéo naturelle (tableau AF.10 pour TOD et tableau AF.11 pour TCD).

Tableau AF.10. Entropies de MVM dans le domaine de la TOD (9,7) et vidéo en basse qualité.

Rang	Gaussien	modèle DWT
$r = 1$	-0,143	-0,547
$r = 100$	-2,007	-6,162
$r = 200$	-3,299	-8,852
$r = 300$	-4,472	-5,050

Tableau AF.11. Entropies de MSVM dans le domaine de la TCD et vidéo en basse qualité.

Rang	Gaussien	modèle DCT
$r = 1$	11,918	3,510
$r = 100$	-0,488	-0,804
$r = 200$	-0,279	-2,590
$r = 300$	-0,188	-1,915

III. Partie III – Modélisation des attaques de tatouage

III.1. Présentation générale

La troisième partie de la thèse (Chapitre 4) est dédiée à une analyse statistique approfondie des attaques de vidéos tatouées. Après avoir réfuté statistiquement l'hypothèse gaussienne pour les attaques, l'algorithme *ART.MOD-A* que nous avons développé est décrit.

Son principe repose sur un comportement additif des attaques et étudie les différences entre trames attaquée et originale, après représentation dans un domaine de transformée.

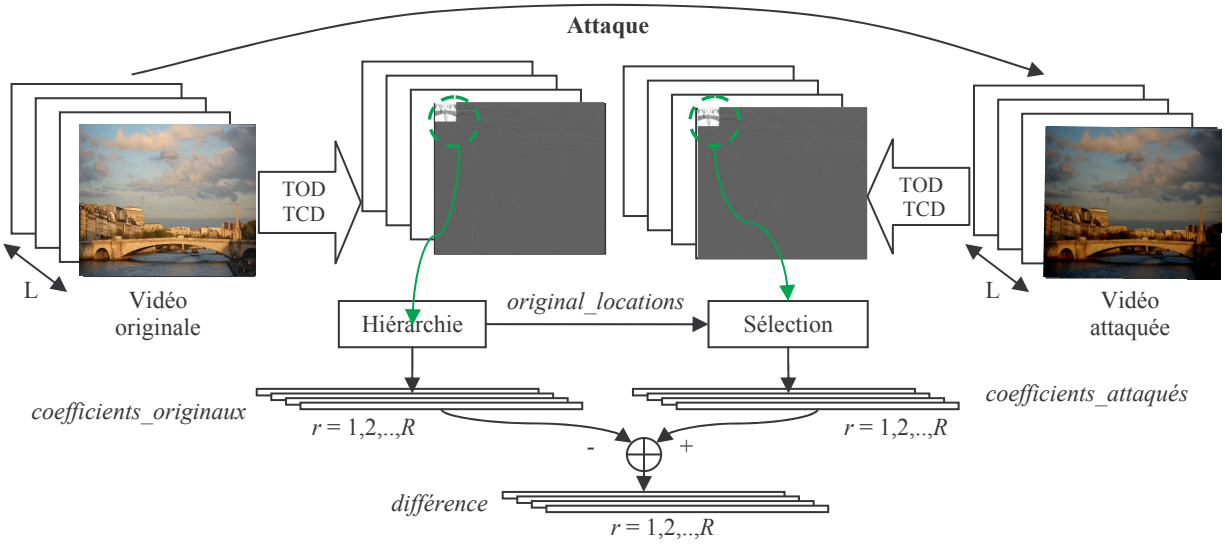


Figure AF.7. Synopsis de l'étape de pré-traitement des données dans l'algorithme *ART.MOD-A*.

III.2. Résultats expérimentaux

L'étude expérimentale conduite sur le corpus précedemment décrit a pris en compte plusieurs types d'attaques : filtrage gaussien, compression JPEG, rotation +2°/-2° /+5°, filtrage médian, rehaussement et attaque StirMark. Dans le calcul des modèles pour les attaques, seules des séquences individuelles (IVM) ont été considérées (donc, *nr_vid* = 10 dans toutes les expérimentations).

Résultats concernant la modélisation des séquences vidéo dans le domaine de la TOD.

Soit une séquence vidéo aléatoirement choisie parmi les 10 dans le corpus. Les modèles obtenus pour le plus grand rang (r =1) sont présentés tableau AF.12 et illustrés figure AF.8. La tableau AF.13 démontre que ces modèles ont une grande généralité : les erreurs de calcul ε, de validation ε' et ε'' sont respectivement inférieures à 5%, 10% et 15% (tableau AF.13 – col 2-3) et E_{HL} et E_{DL} sont proches de zéro (tableau AF.13 – col 4-5).

Les valeurs obtenues dans les expériences mettent en évidence qu'à chaque fois (pour chaque type d'attaque et de rang), le modèle appartient à 95% aux limites de confiance statistique correspondantes.

Tableau AF.12. IVM : hiérarchie des coefficients de la TOD (9,7), $r = 1$ et vidéo en basse qualité.

Attaques	Paramètres des modèles										
Filtrage gaussien	*P(k)*	0,075	0,024	0,270	0,063	0,069	0,056	0,072	0,118	0,20	0,022
	μ(k)	-0,087	-0,157	-0,048	-0,078	-0,069	-0,115	-0,088	-0,051	-0,057	-0,078
	σ(k)	0,019	0,021	0,014	0,018	0,018	0,003	0,019	0,015	0,016	0,001
JPEG	*P(k)*	0,103	0,007	0,183	0,060	0,111	0,180	0,069	0,080	0,114	0,092
	μ(k)	-0,002	-0,207	0,002	-0,058	-0,004	-0,008	-0,009	0,003	-0,001	-0,028
	σ(k)	0,021	0,001	0,003	0,027	0,005	0,007	0,022	0,020	0,021	0,016
Rot +2°	*P(k)*	0,086	0,065	0,068	0,089	0,094	0,105	0,111	0,105	0,136	0,140
	μ(k)	-0,686	-0,007	-0,541	-0,541	-0,260	-0,969	-0,404	-0,789	-0,234	0,583
	σ(k)	0,334	0,076	0,338	0,337	0,269	0,268	0,248	0,315	0,134	0,169
Rot -2°	*P(k)*	0,102	0,143	0,122	0,142	0,091	0,112	0,120	0,007	0,114	0,048
	μ(k)	-0,699	-0,629	-0,583	-0,443	-0,247	-0,020	-0,639	-2,067	-0,681	-0,140
	σ(k)	0,297	0,296	0,285	0,181	0,182	0,141	0,297	0,001	0,298	0,009
Rot +5°	*P(k)*	0,096	0,061	0,127	0,086	0,096	0,098	0,097	0,101	0,092	0,147
	μ(k)	-0,571	-0,023	-0,527	-0,835	-0,628	-0,754	-0,584	-0,519	-0,802	-0,462
	σ(k)	0,217	0,104	0,175	0,254	0,254	0,268	0,115	0,194	0,261	0,150
Médian	*P(k)*	0,064	0,078	0,075	0,122	0,170	0,072	0,091	0,205	0,071	0,050
	μ(k)	0,014	-0,032	-0,028	-0,011	-0,026	-0,012	-0,101	-0,017	-0,055	0,010
	σ(k)	0,056	0,044	0,044	0,025	0,024	0,053	0,032	0,021	0,042	0,006
Rehausse ment	*P(k)*	0,112	0,041	0,074	0,135	0,128	0,128	0,104	0,103	0,098	0,077
	μ(k)	0,184	0,245	0,020	0,113	0,045	0,088	0,054	0,051	0,019	0,053
	σ(k)	0,017	0,012	0,072	0,022	0,038	0,048	0,060	0,063	0,072	0,062
StirMark	*P(k)*	0,070	0,377	0,084	0,079	0,087	0,026	0,084	0,085	0,078	0,027
	μ(k)	-0,108	-0,028	-0,037	-0,033	-0,024	-0,509	-0,338	-0,114	-0,049	-0,172
	σ(k)	0,073	0,047	0,053	0,119	0,118	0,255	0,127	0,073	0,122	0,002

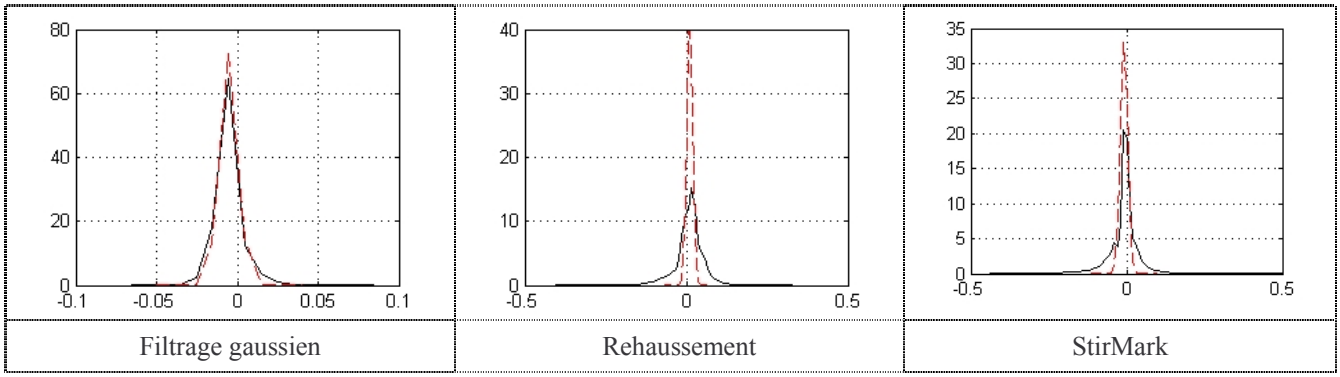


Figure AF.8. Modèles d'attaques (ligne continue) et distributions gaussiennes correspondantes (pointillés), pour le rang $r = 1$, une vidéo en basse qualité, et les attaques par filtrage gaussien, par rehaussement et par StirMark.

Tableau AF.13. Validation du modèle IVM, TOD et rang $r = 1$.

Rang	Etape B	Etape C- (1)	Etape C- (2)		
	Erreur ε	***Erreur ε'***	***Erreur ε''***	E_{HL}	E_{KL}
Filtrage gaussien	0,035	0,063	0,168	0,089	0,549
JPEG	0,011	0,061	0,992	0,078	0,202
Rot +2°	0,045	0,119	0,083	0,093	0,599
Rot -2°	0,054	0,114	0,089	0,076	0,465
Rot +5°	0,040	0,085	0,126	0,153	0,353
Médian	0,024	0,185	0,156	0,028	0,093
Rehaussement	0,032	0,097	0,095	0,023	0,131
StirMark	0,018	0,088	0,037	0,008	0,093

Résultats concernant la modélisation des séquences vidéo dans le domaine de la TCD.

L'investigation statistique sur l'effet des attaques dans le domaine de la TCD a exhibé un comportement similaire au précédent dans le cas de la TOD : $\varepsilon < 5\%$, $\varepsilon' < 10\%$, $\varepsilon'' < 10\%$ avec quelques exceptions (tableau AF.15 – col 3) et E_{HL} et E_{DL} proches de zéro (tableau AF.15 – col 4 et 5).

Les valeurs obtenues dans les expériences mettent en évidence qu'à chaque fois (pour chaque type d'attaque et de rang), le modèle appartient à 95% aux limites de confiance statistique correspondantes.

Tableau AF.14. IVM : hiérarchie des coefficients de la TCD, $r = 1$ et vidéo en basse qualité.

Attaques	Paramètres des modèles										
Filtrage gaussien	$P(k)$	0,118	0,062	0,091	0,102	0,105	0,101	0,047	0,054	0,212	0,109
	$\mu(k)$	0,465	0,502	0,284	0,255	0,316	0,151	0,426	0,349	0,216	0,121
	$\sigma(k)$	0,089	0,174	0,120	0,118	0,119	0,099	0,170	0,075	0,039	0,092
JPEG	$P(k)$	0,195	0,139	0,094	0,077	0,089	0,071	0,137	0,109	0,044	0,044
	$\mu(k)$	-0,016	-0,067	-0,005	0,008	-0,068	-0,006	-0,098	-0,048	-0,219	-0,130
	$\sigma(k)$	0,044	0,059	0,104	0,102	0,092	0,105	0,063	0,053	0,056	0,004
Rot +2°	$P(k)$	0,101	0,112	0,099	0,078	0,099	0,104	0,093	0,105	0,105	0,104
	$\mu(k)$	-0,408	-1,888	1,216	-0,802	-0,541	-1,070	-1,234	-0,481	-0,953	1,505
	$\sigma(k)$	0,956	1,098	0,645	1,087	0,994	1,163	0,648	0,971	1,111	0,824

Tableau AF.14. *(Suité).*

Rot -2°	*P(k)*	0,131	0,084	0,101	0,085	0,108	0,098	0,101	0,091	0,084	0,115
	μ(k)	-0,814	-1,813	-1,554	0,013	1,151	-0,494	-0,894	-0,436	0,385	-1,705
	σ(k)	0,556	0,999	1,016	0,883	0,667	0,976	1,054	0,965	0,828	0,977
Rot +5°	*P(k)*	0,078	0,088	0,095	0,084	0,071	0,122	0,162	0,081	0,084	0,136
	μ(k)	1,379	-2,651	-3,688	-2,059	1,267	-3,004	-0,675	-0,404	-1,082	-1,191
	σ(k)	2,437	2,868	2,667	2,954	2,462	1,117	1,569	1,708	2,013	1,720
Médian	*P(k)*	0,106	0,156	0,142	0,079	0,106	0,029	0,146	0,056	0,087	0,092
	μ(k)	-60,51	-59,59	-89,95	-55,61	-60,34	-39,42	-60,02	-36,29	-76,15	-67,05
	σ(k)	10,19	6,797	6,792	9,509	1,102	5,806	7,097	5,662	1,325	1,287
Rehauss ement	*P(k)*	0,101	0,037	0,157	0,088	0,079	0,078	0,072	0,167	0121	0,099
	μ(k)	-1,169	-3,381	-1,937	-1,246	-1,436	-1,201	-1,400	-1/809	-2,707	-1,965
	σ(k)	0,870	0,537	0,503	0,883	0,090	0,876	0,329	0,475	0,299	0,521
StirMark	*P(k)*	0,144	0,169	0,121	0,078	0,114	0,089	0,108	0,068	0,056	0,054
	μ(k)	-0,201	-0,278	-0,276	-0,384	-0,196	-0,510	-0,163	-0,334	-0,667	0,300
	σ(k)	0,293	0,264	0,300	0,296	0,294	0,162	0,284	0,350	0,423	0,554

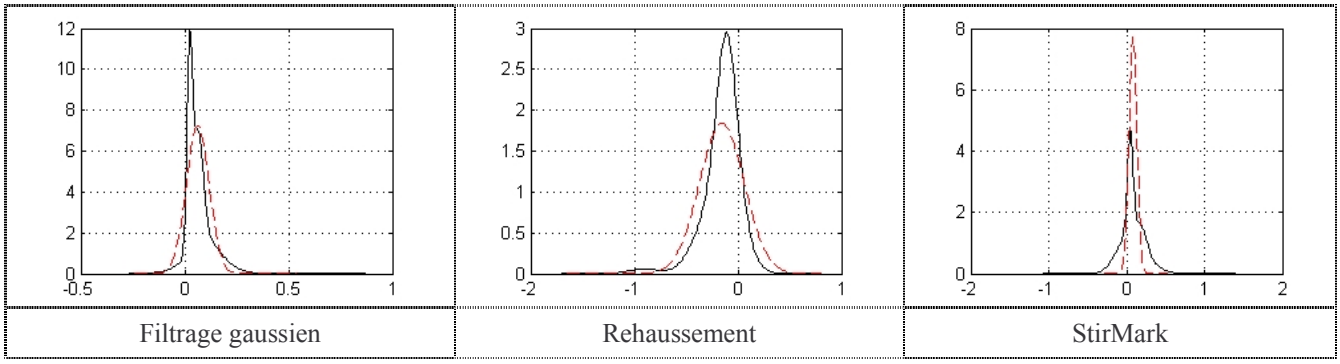


Figure AF.9. Modèles d'attaques (ligne continue) et distributions gaussiennes correspondantes (pointillés), pour le rang $r = 1$, une vidéo en basse qualité, et trois attaques.

Tableau AF.15. Validation du modèle IVM, TCD et rang $r = 1$.

Rang	Etape B	Etape C- (1)	Etape C- (2)		
	Erreur ε	***Erreur ε'***	***Erreur ε''***	E_{HL}	E_{KL}
Filtrage gaussien	0,041	0,067	0,365	0,072	0,247
JPEG	0,028	0,069	0,092	0,002	0,010
Rot +2°	0,029	0,053	0,065	0,029	0,204
Rot -2°	0,029	0,047	0,074	0,042	0,174
Rot +5°	0,032	0,044	0,074	0,056	0,263
Médian	0,020	0,033	0,801	0,010	0,742
Rehaussement	0,021	0,060	0,073	0,221	0,477
StirMark	0,039	0,079	0,040	0,028	0,199

En guise de validation, ces modèles ont été impliqués tout d'abord dans l'évaluation de la capacité (tableaux AF.16 et AF.17).

Au lieu de limites supérieure et inférieure pour la capacité, il est maintenant possible de calculer les valeurs exactes avec la nouvelle méthode *ART.CAP* que nous avons développée. Comme ces valeurs ont été obtenues par une méthode numérique, il est nécessaire d'en évaluer la précision. A cet effet,

l'algorithme Blahut-Arimoto (proposé par J. Dauwels) a été considéré en raison de sa simplicité et rapidité. La pertinence des résultats quantitatifs a été discutée au-delà du domaine strict du tatouage pour des applications de *richmedia*.

Tableau AF.16. Les valeurs de la capacité pour la TOD : Gaussien - C_{Gauss}, limite de Shannon - $C_{non\text{-}Gauss}$, méthode *ART.CAP* - $C_{ART.CAP}$, and Blahut–Arimoto - $C_{B\text{-}A}$.

r	Attaques / C	Filtrage gaussien	Rehaussement	Rotation de +2°	Rotation de -2°	Rotation de +5°	StirMark
$r = 1$	C_{Gauss}	0,05	0,02	0,01	0,01	0,01	0,01
	$C_{non\text{-}Gauss}$	(0,31 ; 1,51)	(0,04 ; 0,57)	(0,01 ; 0,90)	(0,01 ; 0,90)	(0,01 ; 1,26)	(0,03 ; 0,55)
	$C_{ART.CAP}$	**0,58**	**0,44**	**0,36**	**0,37**	**0,36**	**0,38**
	$C_{B\text{-}A}$	**0,41**	**0,37**	**0,31**	**0,31**	**0,30**	**0,32**

Tableau AF.17. Les valeurs de la capacité pour la TOD : Gaussien - C_{Gauss}, limite de Shannon - $C_{non\text{-}Gauss}$, méthode *ART.CAP* - $C_{ART.CAP}$, and Blahut–Arimoto - $C_{B\text{-}A}$.

r	Attaques / C	Filtrage gaussien	Rehaussement	Rotation de +2°	Rotation de -2°	Rotation de +5°	StirMark
$r = 1$	C_{Gauss}	2,49	0,42	0,69	0,69	0,21	1,99
	$C_{non\text{-}Gauss}$	(3,63 ; 3,65)	(1,26 ; 1,72)	(0,29 ; 0,46)	(0,29 ; 0,46)	(0,28 ; 0,48)	(2,39 ; 2,42)
	$C_{ART.CAP}$	**2,57**	**1,33**	**0,30**	**0,30**	**0,28**	**2,31**
	$C_{B\text{-}A}$	**2,42**	**1,27**	**0,23**	**0,23**	**0,26**	**2,18**

Des modèles généraux pour des attaques peuvent être exploités également dans de nombreuses applications. Nous l'avons intégré à la méthode de tatouage *IProtect*.

L'analyse de la répartition moyenne des temps d'exécution entre les différents modules de la chaîne de tatouage (prétraitement, insertion, post-traitement, détection), met en évidence que l'étape d'insertion compte pour 90% (figure AF.10.b - gauche), dont plus de 99% liés à prendre en compte le comportement réel des attaques. Une méthode efficace pour réduire ces derniers est de remplacer lors de l'insertion les attaques (figure AF.10.a - gauche) par leur simulation (figure AF.10.a - droite).

La contribution proposée est d'intégrer au schéma de tatouage un module de simulation des attaques par générateurs Monte Carlo. L'étude des performances (figure AF.10.b – droite) ainsi obtenues a montré un gain d'un facteur 100 tout en préservant les propriétés initiales de transparence et de robustesse du tatouage hybride.

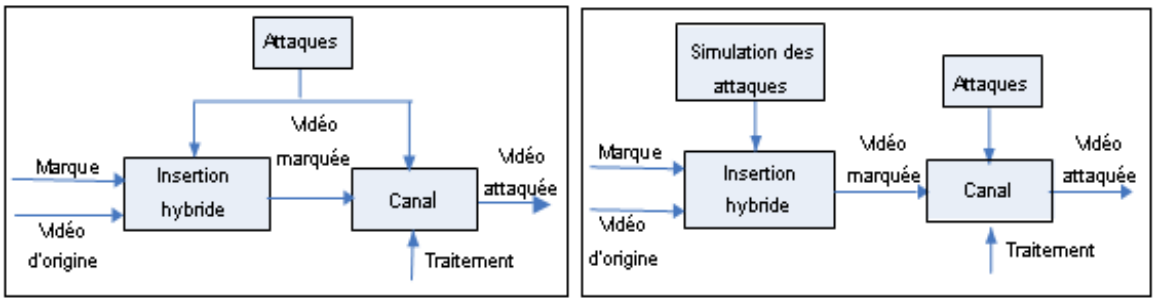


Figure AF.10.a. Méthode de tatouage hybride : version initiale (à gauche) et optimisée (à droite) par introduction d'un simulateur d'attaques.

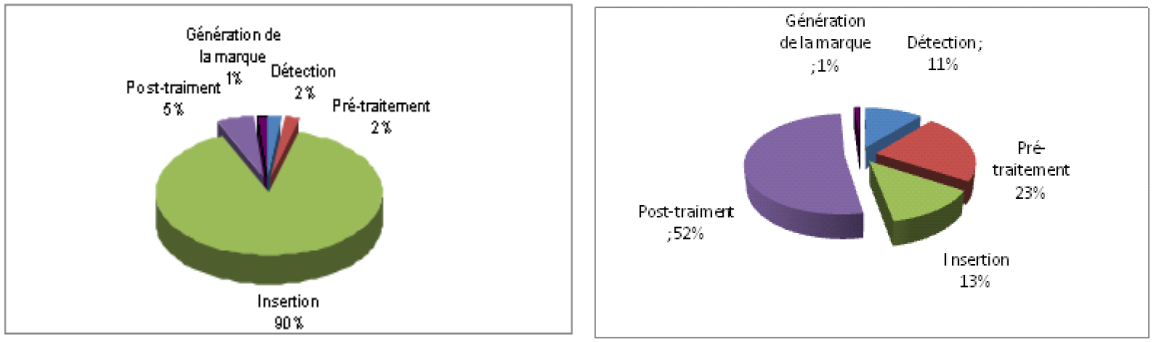


Figure AF.10.b. Temps d'exécution des différentes opérations de la chaîne de tatouage.

III.3. Conclusion

L'étude présentée dans la première partie du chapitre calcule pour la première fois des modèles statistiques pour plusieurs types d'attaques de tatouage à partir d'une hiérarchie des coefficients de TOD ou TCD. Parallèlement à l'estimation de la densité de probabilité pour les attaques, cette étude apporte des preuves en faveur d'une hypothèse de stationnarité : il a été vérifié que les modèles de calcul sont indépendants par rapport à la mesure de similitude, à la séquence vidéo considérée, et à la procédure d'estimation.

Les modèles précis pour les attaques peuvent être aussi le point de départ pour l'optimisation de la règle de détection des applications de tatouage. A noter qu'une grande classe de méthodes rapportées dans la littérature réalise la détection de la marque par des moyens de corrélation, dont l'optimalité est prouvée lorsque le bruit est distribué d'une manière gaussienne. Pour le bruit non-gaussien, de nouvelles règles doivent être élaborées.

IV. Partie IV – Conclusion et perspectives

La dernière partie de la thèse est consacrée aux observations finales et ouvre des perspectives tant méthodologiques qu'appliquées.

De façon synthétique, les contributions de cette thèse sont présentées dans les figures suivantes, figures AF.11.a - b.

	Stationnarité	Modélisation
Vidéo naturelle	**État de l'art** - Toujours supposée.	**État de l'art** - Différentes distributions (gaussienne, gaussienne généralisée, laplacien) sélectionnées d'une façon empirique pour une application. - Quelques tests statistiques sont pris en compte.
	Verrou : - Concept mathématique très sophistiqué, qui nécessite de grandes analyses statistiques. - Absence d'une procédure mathématique d'analogie.	**Verrou :** - Manque de modèle pour la stationnarité.
	Contributions de la thèse	**Contributions de la thèse**
	Algorithme *ART.MOD-V*	
	- 1ere preuve de la stationnarité au premier ordre.	- Modèle gaussien réfuté mathématiquement. - Proposition de modèle précis et générale pour la vidéo naturelle, avec calcul de l'entropie continue sur celui-ci.
	Perspectives - Analyse statistique, concernant la stationnarité d'ordre supérieur.	**Perspectives** - Utilisation des modèles développés pour la compression, le débruitage, l'indexation, le classement, ...

Figure AF.11.a. Tableau synthétique mettant en regard de l'état de l'art nos contributions et leur perspective pour la modélisation de vidéo naturelle.

Attaques de tatouage	Stationnarité	Modélisation
	État de l'art - Implicitement incluse dans l'hypothèse de bruit blanc gaussien additif, mais jamais discutée et / ou investiguée.	**État de l'art** - Hypothèse de bruit blanc gaussien additif toujours considérée, malgré sa simplicité.
	Verrou - Manque d'une procédure d'analogie. - Enorme travail expérimental.	**Verrou** - Manque de modèle pour l'investigation. - Aucune méthode fiable de calcul de la limite théorique de tatouage.
	Contributions de la thèse	**Contributions de la thèse**
	Algorithme *ART.MOD-A*	
	- 1ere preuve de la stationnarité du premier ordre des processus aléatoires correspondant aux attaques de tatouage.	- Réfutation mathématique du modèle gaussien. - Estimation pour la première fois des modèles précis pour une large classe d'attaques (sous l'hypothèse d'additivité). - Validation des modèles à travers leur intégration dans une méthode de tatouage existante : accélération d'un facteur de 100 du temps d'exécution par rapport à la méthode initiale.
		ART.CAP
		- Identification des limites théoriques du tatouage par calcul des valeurs précises de la capacité - Analyse dans une application *richmedia*.
	Perspectives - Analyse de la stationnarité d'ordre supérieur.	**Perspectives** - Optimisation de la règle de détection sous contraintes de bruit non-gaussien.

Figure AF.11.b. Tableau synthétique mettant en regard de l'état de l'art nos contributions et leur perspective pour de modélisation des attaques de tatouage.

"The science of today is the technology of tomorrow."

Edward Teller (1908 - 2003)

(Hungarian-American nuclear physicist, The Legacy of Hiroshima, 1962)

Preface

The process of imperceptibly and persistently inserting into a digital content of some additional information is referred to as *watermarking*. Watermarking is a viable solution to a large variety of multimedia applications like authentication, broadcast monitoring, copy protection, e-Commence, rights management, *etc*.

The original content may be a still image, an audio or video excerpt, some 3D data, a text... The additional information generally refers to some copyright information (a logo, time stamp, *etc*) and will be referred as watermark.

The digital watermark needs to be transparent (imperceptible) and robust against the attacks (which are transforms designed by malicious users in order to damage or eliminate the watermark from the watermarked content).

Figure P.1 shows a watermarking example. A frame from an original video sequence is presented in Figure P.1.a. After inserting the ARTEMIS logo, the corresponding watermarked frame is shown in Figure P.1.b: note that no visual differences can be perceived between the original and the watermarked frame, *i.e.* the procedure feature transparency. In robust watermarking, the inserted logo should be detected even when the StirMark attack (emulating the in-theatre camcorder recording) is applied to the watermarked video (Figure P.1.c).

Figure P.1. Example of a watermarking application: (a) original frame; (b) watermark frame (left side) and logo inserted (right side); (c) attacked frame (left side) and recovered logo after attack (right side).

From the theoretical point of view, a watermarking system can be modelled as a noisy channel. The watermark is a sample from the information source and should be recovered at the detection side. The elements that make the watermark detection difficult can be modelled as the channel noise, namely: the original content (be it represented into some T transform domain or not) and the attacks. The watermarking procedure itself plays the role of the modulation technique.

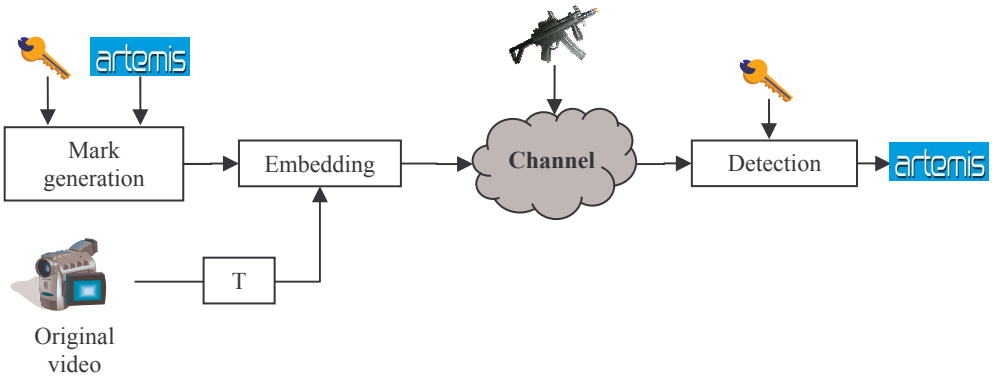


Figure P.2. The watermarking as a noisy channel.

This model brings into evidence that the design of a reliable watermarking technique cannot be achieved without an in-depth knowledge of the noise (*i.e.* original content and attacks) statistical model. Despite suspicion, reticence and even contradictory results, in the absence of any sound theoretical result, such a model is generally assumed to be Gaussian.

The present thesis is meant to clarify this problem: it focuses on natural video and attack modelling for uncompressed video watermarking purposes.

By reconsidering a statistical investigation combining four types of statistical tests, the thesis starts by identifying with accuracy the drawbacks and limitations of the popular Gaussian model in watermarking applications. Further on, **an advanced statistical approach is developed in order to establish with mathematical rigour:**

1. **that a mathematical model for the original video content and/or attacks exists, and**
2. **the model parameters.**

From the theoretical point of view, this means to prove for the first time the stationarity of the random processes representing the natural video and/or the watermarking attacks.

These general results have been already checked-up under applicative and theoretical frameworks. On the one hand, when integrating the attack models into a watermarking method patented by ARTEMIS and SFR, a speed-up by a factor of 100 of the insertion procedure has been obtained. On the other hand, accurate models for natural video and attacks allowed the increasing of

the precision in the computation of some basic information theory entities (entropies and capacity). However, it should be emphasised that these validations are presented here just as first-hand validations for the original video/attack modelling: future theoretical and experimental work is required in order to design the watermarking method completely exploiting all the benefits of these models (*i.e.* the watermarking method reaching the capacity limit).

The thesis is structured into four parts, as follows.

The first part contains two chapters. The first one starts by presenting the main watermarking properties, and goes further by facing their potential applications to the commercial products available on the market. The second one creates a fundamental basis for the methods and the experiments developed in the second and third part of the thesis.

The second part of the thesis (Chapter III) is devoted to the mathematical modelling of the natural video. After identifying the inner limitations in the state-of-the-art coefficient modelling, the *ART.MOD-V* algorithm is presented and applied to a large and heterogeneous corpus of natural video. *ART.MOD-V* is an original statistical investigation making it possible to properly handle the general difficulties connected to the video data dependency and to their *a priori* non-stationarity. The models computed for the most intensively transforms nowadays in use (the DWT and the DCT) are computed and discussed. It is proved that they can be considered as reference values for the natural video: they do not depend on the particular video sequence on which they were computed, they are independent with respect to the estimation procedure, and their precision is assed. The quantitative results were obtained out of processing a heterogeneous corpus. Finally, the model utility is illustrated by natural video entropy estimation.

The third part of the thesis (Chapter IV) keeps the same structure, this time focussing on the watermarking attacks. In this respect, after statistically refuting the Gaussian hypothesis for the attacks, the algorithm *ART.MOD-A* (paired designed with *ART.MOD-V*) is described and applied to the same corpus. Here again, reference models for the investigated noise source (the watermarking attacks) are obtained for the first time. The utility of these models is exemplified by a real life watermarking application and by increasing the accuracy in watermarking capacity evaluation. Note that the latter issue relies on an original method for capacity evaluation, denoted by *ART.CAP*.

The last part of the thesis is devoted to concluding remarks and opens the perspectives for future work.

Note that all the development in the thesis beneficed of the continuous and iterative feed-back of the ARTEMIS partners, namely the SFR (Vodafone) mobile service provider in France (under the framework of the ***TAMUSO*** and ***DWD*** industrial contracts) and HD3D SAS (the start-up emerged from the ***HD3D-IIO*** project of the CapDigital competitiveness cluster in Ile de France).

"One way to plug the analog hole is through the use of watermarks … some government action will be needed to require appropriate detection of and response to the watermark."

Richard Parsons (1948 - ___)

(CEO, AOL Time Warner; Senate Judiciary Hearing, 2002)

Chapter I

The Watermarking Challenge

In this first chapter of the thesis, after having identified the huge economic impact of the multimedia piracy, the main watermarking definitions are presented and discussed. Then, the watermarking applications and products are surveyed. This state-of-the-art analysis points out to the need for a sound theoretical framework as a unique solution to stop the multimedia piracy.

Contents

I.1. Piracy in the Information Society

Any piece of digital content may be *a priori* easily plundered: in the digital world, the copies are perfect and unlimited!

In 2009, the report of the IIPA (International Intellectual Property Alliance) evaluates the losses caused by copyright infringement between 2007 and 2008 at more than 38 billion US dollars [Ipa 09]. Details in this respect (losses per year and per media type: motion pictures, sound recordings & musical software, business software, entertainment software, and books) can be found in Table I.1 and Figure I.1.

Each year, IIPA classifies the countries in four categories (the so-called IIPA lists) *priority watch list, watch list, monitoring list,* and *other special mention list*. This partition is based on three criteria: the importance of their legitimate market, the local trend in piracy, and the degree of action of their governments.

Table I.1. Estimated trade losses (in millions of US dollars) connected to copyright piracy in 2000–2008.

Industry	2000	2001	2002	2003	2004	2005	2006	2007	2008
Motion pictures	1,242	1,288	1,322	1,528	1,800	2,913	N. a.	N. a.	N. a.
Records & Music	1,835	2,034	2,142	2,260	2,657	2,456	2,374	2,239	1,969
Business software	2,490	2,653	3,539	N. a.	6,448	8,684	10,345	14,695	16,442
Entertainment software	1,658	1,767	1,690	1,549	1,847	2,652	1,951	2,655	N. a.
Books	675	636	514	499	603	600	582	529	N. a.
Total	7,903	8,379	9,208	5,838	13,356	17,307	15,252	20,119	18,411

Note: N.a. stands for "not available" values.

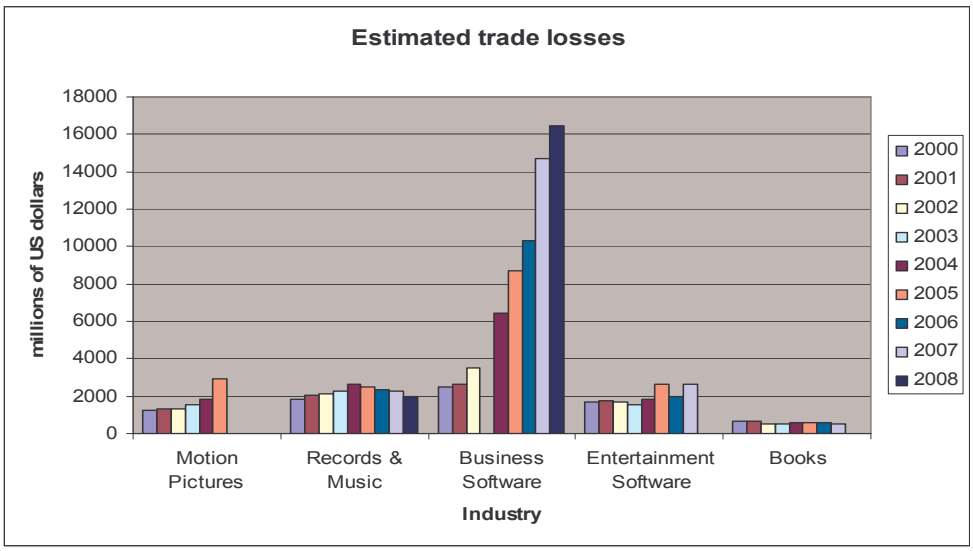


Figure I.1. Estimated trade losses (2000–2008) due the copyright piracy.

In 2009, 48 countries around the world are listed with different recommendations (13 countries in the *priority watch list*, 25 countries in the *watch list*, one country in the *monitoring* and 9 countries in the *special mention list*) [Ipa 08].

As the piracy losses continuously increase, to coordinate, train and demonstrate the benefit of taking effective actions against piracy becomes a crucial issue in the Information Society. Alongside with legislative efforts, technical solutions enabling piracy fighting should be designed and deployed at once.

In this respect, the last year trends pointed to watermarking as a framework potentially outperforming the DRM (Digital Right Management): flexibility, generality, and virtually the zero-cost on the user side are just three of its main advantages.

By *persistently* and *imperceptibly* associating some additional data (*a mark*) with the original content, watermarking aims at solving the piracy issue: the philosophy consists in no longer restricting the copy mechanism by software/hardware devices but in tracking–down the copy maker and in allowing the identification of the true owner [Cox 00].

I.2. Robust Watermarking Framework

The process of imperceptibly and persistently inserting into a digital content of some additional information is referred to as *watermarking* [Cox 02], [Cox 08].

The original content may be a still image, an audio or video excerpt, some 3D data or a text. The additional information generally refers to some copyright information (owner, time stamp, *etc*).

In this definition, *imperceptibility* means that the human observer is not able to detect the existence of the mark (to "see" it in the visual content or to "hear" in the audio content).

The *persistency* refers to the possibility of detecting the mark even when strong malicious operations were applied to the marked content. This property is also referred to as *robustness*. In this thesis only the issue of robust watermarking is considered.

I.2.1. Watermarking Properties

The main properties characterising any watermarking application are: the *data payload,* the *transparency,* the *robustness,* the *obliviousness,* the *false positive rate,* the *key size,* and the *cost* [Cox 08].

The trade-off among these general watermarking properties is to be reached according to the particular application requirements.

Data payload

The data payload represents the amount of information (in bits) which is inserted into the original content. The data payload may vary from 1 bit (a simple watermarked or unwatermarked

decision) to a large quantity of information (*e.g.* a binary logo or even a colour image). Note that, the inserted information may be public.

Transparency

The transparency property expresses the human perceptual impact of the artefacts induced in the original content by the mark insertion.

A watermark is said to feature *fidelity* if the degradation is very difficult to be perceived by a human observer.

When the watermarked product has noticeable yet undisturbing artefacts, the product is said to be of a good *quality*. Note that in this definition, the artefacts may be induced by the watermarking procedure or by other content pre–processing (*e.g.* compression).

Robustness

A watermark is said to be robust if it survives common signal processing operations (for example: digital–to–analog and analog–to–digital conversions, lossy compression, and geometric transformations – rotation, translation) as well as hostile *attacks* (*i.e.* malicious transforms specifically designed so as to turn the mark undetectable).

Obliviousness

A watermarking technique is considered *oblivious* when the original unmarked content is not required during the detection. This property is alternatively called *blind verification*.

False positive rate

A false positive is a detection of a watermark in a content that was not watermarked. The false positive rate is represented by the relative number of false positives expected to appear after a number of detections. The relevance of this rate is connected to the particular application (*example*: for proof of ownership, the false positive rate should be less than 10^{-6} [Cox 08]).

The key size

In the most general case, the key represents the only secret information in a watermarking system. Consequently, system security partly depends on the key size and management [Cox 08].

For a viable application, the general key requirements stated in cryptography should be preserved: the key space should be as large as to avoid an exhaustive search, the key should be randomly chosen, there should be no dependency between the key and the content to be protected, *etc*. Sometimes, additional constraints concerning the maximum amount of secret information may be imposed.

In 1998, France was the only Western European country with out free use of encryption. In March 23, 1998 [Tce 98] a law gave the limits to encryption based on the maximum key length of

40 bits (Decree 98-207 of March 23, 1998). In January 19, 1999 [Fer 99] the French Prime Minister Jospin announced the liberation of the domestic cryptography legislation, using the key lengths of up to 128 bits (Decree 99-200 of March 17, 1999). For the operations between the members of the European Community, the key length must to be up to 56 bits (Decree 2007-663 of May 2, 2007) [Leg 07].

Cost

Different applications require different embedder and detector speeds (*example:* for ownership the detector is valuable even if it takes more than one day to find a watermark, because such a detector will be used only during a dispute. However, for broadcast monitoring, the embedders and detectors must work in real time).

The wide variation in financial cost and speed requirements means that there is a big variety in the required computational efficiency (depending by the application) of watermark field [Cox 00].

I.2.2. Watermarking Attacks

According to the target application, several types of attacks can be identified. Two types of classification are achieved in the literature, [Vol 01], [Cox 08].

Following the classification in [Vol 01], the attacks are divided in three categories: *(A) content attacks*, *(B) cryptographic attacks*, and *(C) protocol attacks*, see Figure I.2.

(A) Content attacks are grouped in two sub-categories: *removal attacks* (*A.1*) and *geometrical attacks* (*A.2*).

(A.1) Removal attacks remove a watermark from the watermarked data. These approaches consider the inserted watermark as noise with a given statistics and estimate the original, non watermarked data from watermarked data. Many types of attacks are included in this category: *denoising*, *lossy compression*, *quantization*, *remodulation*, *averaging*, *collusion* and *mosaic*.

Denoising, *lossy compression* and *quantization attacks* exploit the basic idea that the watermark resembles a noise which can be statistically modelled. By estimating the original, non-watermarked cover data based on an available copy of the data, an attacker can achieve the watermark removal.

Remodulation attacks are transforms specially designed for watermarking purposes. Two schemes for remodulation attack are presented in [Vol 01]. The first is the Langelaar [Lan 98] scheme, where the watermark is first predicted through subtraction of the median filter of the watermarked image. The predicted watermarked is further high-pass filtered, truncated, and finally subtracted from the watermarked image. In real life application, as the watermarks are not always high-pass, poor performances are obtained. The second scheme proposed by Holliman [Hol 99] is based on weighted mean prediction and was applied with success to remove the watermark inserted according to [Pit 96]. The experiment was also successful in attacking the commercial products of Digimarc [Pit 96].

Averaging attacks are attacks in which many instances of a given data set, each time signed with a different key or watermark, are averaged to compute the attacked data. If the number of data sets is large enough, the embedded watermark may not be detected anymore assuming that on average it will yield a zero mean.

For *collusion attack*, many replicas of the same data are also available, but in this case the attacked data set is generated by taking only small part of each data set and rebuilding a new attacked data set from these parts.

The *mosaic attack* doesn't try to remove the watermark using some signal processing methods, but rather it aims at creating problems in retrieving the watermarked media for the detector by dividing the watermarked image on small fragments which can be stored in different files and concatenated only for display.

(A.2) Geometrical attacks don't remove the embedded watermark but turn it undetectable, generally by desynchronizing the detector (*e.g.* changing the position of watermark within the media). Examples from this class are the *global & local warping*, the *global & local transforms*, and the *jittering*.

For the case of still image watermarking, the most well-known integrated software of these attacks (*global & local warping*, *global & local transforms*, *jittering*) are StirMark and Unzign (for more details see the next section).

StirMark introduces global and local geometrical distortions: rotation, scaling, change of aspect ratio, affine transformations, *etc*. This software also includes the line/column removal and cropping/translation.

Unzign introduces *local pixel jittering* and is very efficient in attacking spatial domain watermarking schemes.

(B) Cryptographic attacks are meant to render the watermarking system useless by exploiting any fault in the key management. These attacks are very similar to the attacks used in cryptography.

There are *brute force attacks* which aim at finding secret information through an exhaustive search. As all watermarking schemes use a secret key it is important to use keys with secure length.

Another attack defined in this category is the *Oracle attack*, which can be used to create unwatermarked data when the pirate controls a detector.

(C) Protocol attacks try to generate protocol ambiguities in the watermark process thus discrediting it and referring it from deployed. The first type of protocol attacks is the *inversion attack,* where the hacker inserts his/her watermark in the marked media, so that two watermarks are present: he/she may declare then his/she is the owner of the data and that the original watermark was added at a later date. The second type is the *copy attacks*. Here the idea is to find a solution to estimate the watermark from the watermarked data, creating an ambiguity on the legitimacy of the mark presence.

Although the above classification [Vol 01] makes it possible to have a clear separation among the different classes of attacks, it is necessary to remind that often a malicious attacker applies not only a single attack, but rather a combination of attacks. For instance, in all benchmarking tools, the geometrical transformations are implemented.

According to [Cox 08] the attacks are classified *active* and *passive*.

In the former case, the hacker tries to remove the watermark or to make it undetectable. The latter case corresponds to the solution in which the hacker is not trying to remove the watermark, but rather to determine whether a watermark is present or not. Such an attack can become very dangerous in conjunction with other types of attacks.

In order to summarize this classification, Figure I.2 is a combination of both classifications [Vol 01], [Cox 08]; each attack (sub-categories of attacks) in Voloshynovskiy classification is represented in figure by dotted line square and is attached to one of the attacks on the Cox classification by a colour code: *active attacks* are in red and *passive attacks* are in blue.

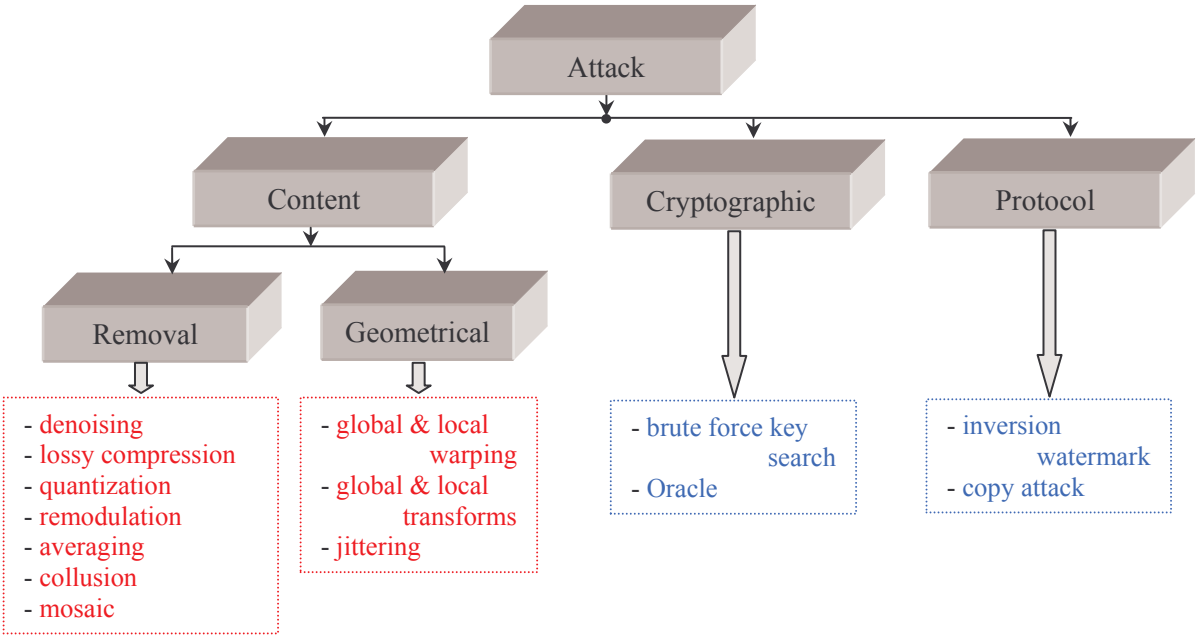


Figure I.2. Classification of the watermarking attacks.

I.2.3. Benchmarking Tools for Watermarking

StirMark, *CheckMark*, *OptiMark*, *Unzign* and *CertiMark* are benchmarking tools for the robustness of the digital watermarking technologies for images. In order to verify if the embedded watermark can still be detected, these tools generate a number of modified images from the watermarked one [Nik 02], [Vol 01b].

StirMark [Dww 08a] has been developed by Fabien Peticolas [Pet 98], [Pet 00] at Cambridge University – United Kingdom in 1997. StirMark is currently the most used benchmarking tool for digital watermarking technologies.

The image attacks implemented in the StirMark (version 4.0 released in 2000) are classified as: cropping, rotation and rotation-scale, FMLR (Frequency Mode Laplacian Removal), sharpening, Gaussian filtering, linear transformations, aspect ratio modification, scale change, line removal, colour reduction, JPEG compression and StirMark. These operations can be applied individually or grouped, according to the user choice.

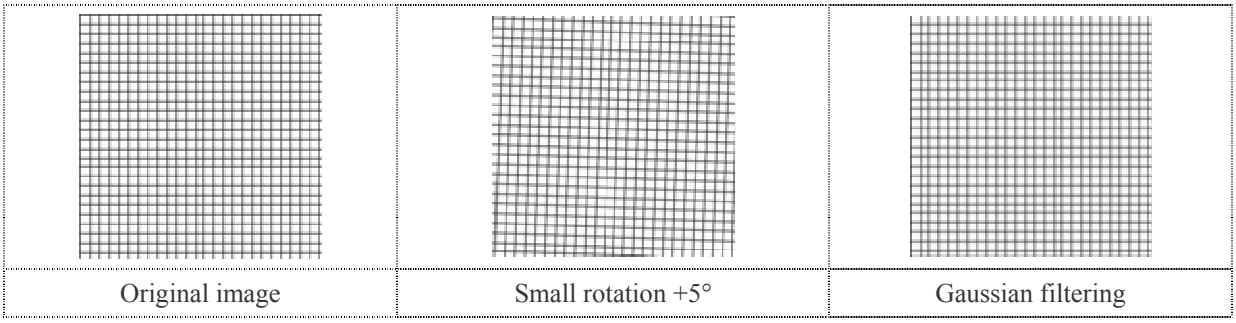
For example, in the case of cropping, rotation, linear transformation, other geometric transformations, *etc* the attacked images are obtained with and without JPEG compression with a quality factor of 90% [Vol 01b].

Some details about the most intensively considered attacks are further presented:

- The JPEG is one of the most common compression algorithms for images and all the watermarking scheme should face it. JPEG is one of the most used formats for storing and transmitting photographic images on the Web. [Kut 99] experimentally showed that the minimum quality fact for JPEG is 70%.
- The rotations with a small angle (Figure I.3.a) are often combined with cropping. In almost cases, they do not modify the commercial value of the image, but they can make the watermark undetectable [Kut 99]. Note that rotations derive not only from the malicious attacks, but they may be part of some mundane image processing like the realignment of an image after scanning.
- The FMLR works using the characteristics of the Laplacian (LP) 3×3 convolution mask (normally used for simple edge detection in image processing). The image to be attacked is processed using an LP convolution mask; the image thus obtained is called negative Laplacian (L_n). The Laplacian convolution mask is applied once more, this time to the L_n; the results of this operation is the positive Laplacian (L_p) [Bar 98]. Be S is the marked image, S' is the attacked image, and α the strength of the attack (α takes values in the interval $[0.05, 0.15]$) then $S' = S - \alpha(L_p - L_n)$.
- The low pass filtering includes linear and non-linear filters, like the Gaussian or the median filter [Kut 99].

  A Gaussian filter is meant to give no overshoot to a step function input while minimizing the rise and fall time (Figure I.3.a).

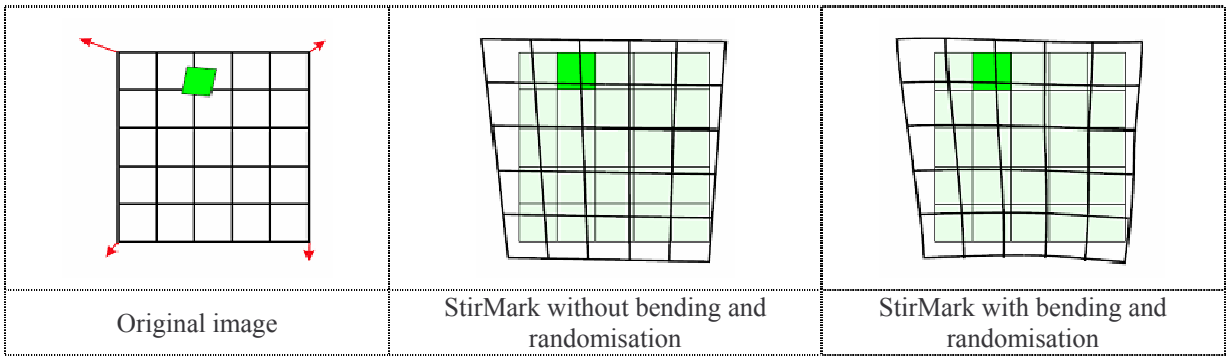
  The median filter is a non-linear operation used to remove noise from images (reduces the speckle or salt and pepper noise). The idea is to examine a sample of the input and to decide whatever it is representative of the signal. This investigation is realized using a window consisting in an odd number of samples. The values in the considered window are sorted into ascending order; the median value is selected as the output and the sample in the centre of the window is replaced.



Original image	Small rotation +5°	Gaussian filtering

Figure I.3.a. Original image and two attacks: rotation with 5° and Gaussian filtering.

- The rows / columns removal is very efficient against any straightforward implementation of spread-spectrum techniques in the spatial domain [Kut 99]. Removing k samples at regular intervals in a pseudo random sequence typically divides by k the amplitude of the cross correlation peak with the original sequence.
- The sharpening filter is a function used by the large majority of photo software products. These filters can be used as an effective attack to watermarking scheme [Kut 99] due to their property of detecting high frequency noise introduced by many digital watermarking schemes/software. The attacks are based on the LP operator [Bar 98], the attacked image being $S' = S - \alpha\nabla^2(\nabla^2 I - I)$, where α has the same signification like in the FMLR attack.
- The random geometric distortion attack (StirMark attack [Pet 98]) applies a combination of minor geometric distortions, *i.e.* the image is slightly stretched, sheared, shifted, and/or rotated by an unnoticeable random amount and finally re-sampled (Figure I.3.b). In the last decade, StirMark attack has always been the most defeating attack.



Original image	StirMark without bending and randomisation	StirMark with bending and randomisation

Figure I.3.b. The StirMark attack [Pet 98].

CheckMark [Dww 08b], [Per 01] has been implemented at the Computer Vision Group – University of Geneva, Switzerland.

It re-implements all the attacks available in StirMark and contains some new attacks: wavelet compression (JPEG 2000 based on Jasper), projective transformations, warping, copy, template

removal, denoising (midpoint, trimmed mean, soft and hard thresholding, Wiener filtering), non-linear removal, and collage.

CheckMark also computes some transparency metrics in order to evaluate the image quality, like PSNR (Peak Signal to Noise Ratio) and the Watson's measure.

OptiMark [Dww 08c] was developed at the Aristotle University of Thessaloniki – Greece and was partiality supported by EU projects CERTIMARK and INSPECT [Sol 01].

The attacks included in this software tool are: cropping, line and column removal, scaling, shearing, horizontal flip, rotation (rotation + autocropping or rotation + autocropping + autoscale), sharpening, Gaussian & median filtering, and JPEG compression.

Unzign [Vol 01a] is a software for attacking watermarked images in the JPEG format. For the first version (version 1.1) of Unzign the developer introduced pixel jittering in combination with a slight image translation. In many cases the Unzign arrives to remove the watermark, but this version introduces severe distortion. To reduce distortions, an improvement version (version 1.2) was developed but watermarking destruction capability is reduced.

CertiMark (CERTIfication for waterMARKing techniques) [Cer 08] is a benchmarking platform designed within the framework of an European project, having the following partners: Belgium (Université Catholique de Louvain), France (Institut Eurecom, Institut National de l'Audiovisuel, NETIMAGE, THOMSON), Germany (Fraunhofer-Gesellschaft zur Förderung der augewandten Forschung, Media Technologies Gmb), Greece (Aristotle University of Thessaloniki), Netherland (Philips, Technische Universiteit Delft), Spain (Universidad de Vigo), and Switzerland (Digital Copyright Technologies, Ecole Polytechnique Fédérale de Lausanne, University of Geneva).

The object of this project was to simulate real attacking scenarios (unintentional or intentional) and to compare the performance of different algorithm taking into account their possible applications (proof of ownership, broadcast and distribution monitoring, fingerprinting, authentication and identification, *etc*). The attacks for each category are:

- Unintentional attacks:
  - for image – compression, printing/scanning, filtering, geometric transforms, cropping, multiple marking, *etc;*
  - for video – compression, format conversion, analog–digital and digital–analog conversion, geometric transforms, jitter, multiple marking, *etc*.
- Intentional attacks: removal and interference attacks, de-synchronization, cryptographic attacks, protocol attacks, *etc*.

I.3. Watermarking Applicative Panorama

In this moment, digital watermarking is one of the most expansive technologies to protect digitized copyright content (Figure I.4).

Initially devoted to fighting piracy, the watermark may be nowadays the core of a large variety of applications for each type of media content, and for both analogous and digital formats, including secure documents, photos, audio, video (film, news, commercials and home video), *etc* [Dig 03].

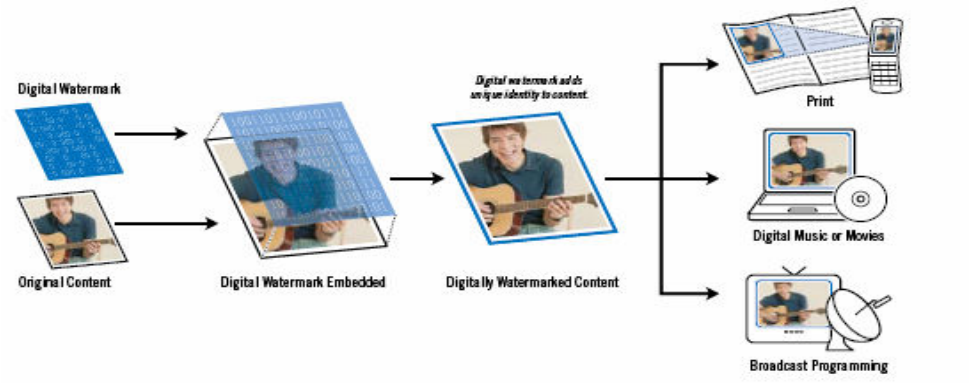


Figure I.4. Digital watermarking enables content identification applications for print, movies, and broadcast programming [Dig 03].

Figure I.5 presents the potential watermarking application fields which are group in three categories:

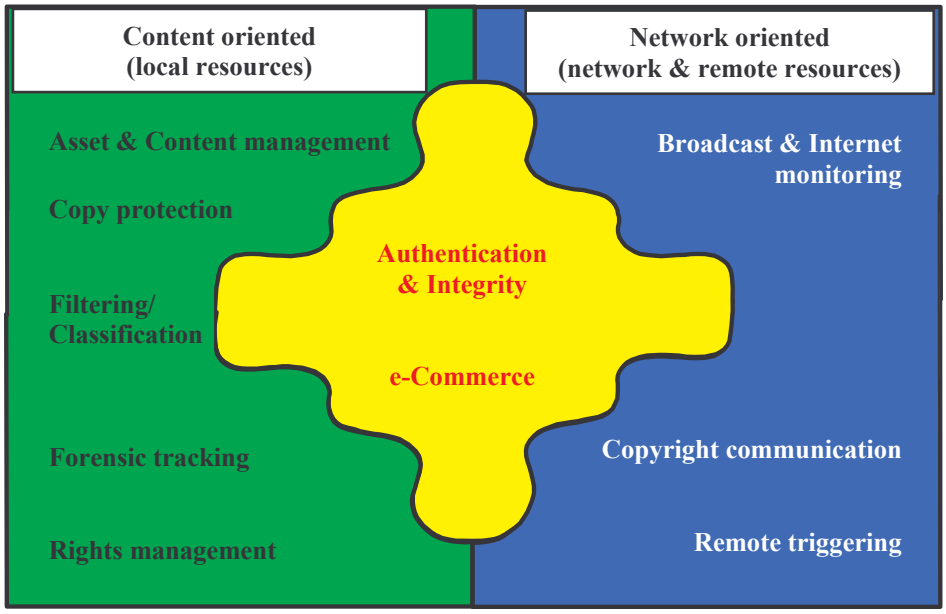


Figure I.5. Applications based on watermarking.

1. ***Content oriented (local resources) applications*** (in green colour)*;*

Asset & Content management

Digital watermarks can be used as a persistent media asset tag, acting as a key into a digital asset management system (DAM). Any piece of tagged content can lead back to the original which is stored in the DAM system. Tagged content can also be linked to metadata in the DAM system, such as keywords, rights, and permissions. The digital watermark carries the content (and possibly the distribution) identification. The related database links the content identification to the content.

Copy protection

Digital watermarks enable a mean to embed and play copy control instructions within content. These copy control instructions might indicate that play out is allowed, that a single copy can be realized, or that no copies are permitted. Actually, this is a DRM tool implemented *via* the watermarking paradigm.

Filtering/Classification

Digital watermarks enable content to be identified, classified and filtered. The systems are enabled to selectively filter potentially inappropriate content, such as corporations and parents restricting viewing of adult content or of other objectionable material.

Forensic tracking

The content owners or service providers can detect the point where the content left the authorized distribution. The digital watermark identifies the authorized person of the content. The detector and the database are managed by the content owner and by the device manufacturer.

Rights management

Digital watermarking enables digital rights management systems (DRM) to connect digital content outside the DRM back to the DRM: linking the content to usage rules, billing information, *etc*. The digital watermark contains the content identification and distribution identification (which is optionally). The related database links the content identification to the content owner, usage rules, and billing information, and the distributor identification to the method of distribution.

2. ***Network oriented (network & remote resources) applications*** **(**in blue colour)*;*

Broadcast & Internet monitoring

Digital watermarks enable content owners and distributors to track into their traditional digital content (radio and TV) and/or Internet broadcasting. The original content is embedded with a unique identifier (owner distributor, data/time information). Detectors are usually placed in all markets where the broadcast is distributed. The digital watermark is detectable and used to reference a database, resulting each time in a report to the owner or distributor.

Copyright communication

Content often circulates anonymously, without the identify of its owner, or without an easy mean to contact the distributor to obtain the usage rights. Digital watermarks enable copyright holders to communicate their ownership and offer links to copyright information.

Remote triggering

Digital watermarking identifies the content and causes automatic action during content distribution. The watermark can trigger the insertion of local or regional advertisements or service announcements, can point to a unique database at any individual sites where the digital watermark is found, *etc*.

3. Of course, authentication & integrity and e-Commerce are two watermarking ***applications transversal to this classification*** **(**in yellow colour).

Authentication & Integrity

Digital watermarks can verify that the content is genuine and/or that it has not been falsified or alerted. Such an operation can also be carried out as an additional layer of security (*e.g.* the integrity of an encrypted content).

e-Commerce

Digital watermarks enable access to information about a product and its vending condition. They can also include the content and distributor identifications which are managed by a centralized database.

I.4. Applications vs. Watermarking Properties

The previous sections discussed the watermarking properties (Section I.2) and several of their potential applications (Section I.3). Table I.2 synoptically represents the inter dependency between these two different points of view (*i.e.* the way in which the practical solutions require the *transparency*, *robustness,* and/or *data payload*).

Table I.2. Digital watermarking properties *vs.* practical solutions (for each property three levels are chosen: L – low, M – medium, and H – high).

Applications	Transparency	Robustness	Data payload	Examples / Observations
Asset & Content management	H	H	H	The Smart Toy system [Dig 09a] contains about 100 toys each of them 50 manufactures. The data payload to be inserted in each toy is 22 bits (7 bits for toy ID, 6 bits for manufacturer, 3 bits for minimum age and other 6 bits for future use).
Copy protection	H	H	H	Between 4 and 8 bits should be inserted in every 10 seconds of the audio files and in every 5 minutes for the video files [Cox 08].
Filtering/ Classification	H	H	M	The data payload is minimum 1 bit/frame which mean 24 bits/seconds [Pat 02].
Forensic tracking	H	L	H	21bits inserted in any 90 seconds of video may protect video on optical disks [Phi 08d].
Rights management	H	H	H	In [Dig 09b], 31 bits are inserted in the MPEG-4 compressed domain in order to assure the DRM.
Broadcast & Internet monitoring	H	M	M	The data payload imperative is minimum 24 bits/seconds [Cox 08].
Copyright communication	H	H	H	For a music delivery system, the data payload size is 55bits (8 bits corresponding to ID content, 30 bits for approximately 1 billion different clients and 17 for redundancy checking and dependency allocation) [Pat 05].
Remote triggering	H	L	H	WaterCast [Phi 09] allows 12 to 24 bits to be inserted with this purpose.
Authentication & Integrity	H	H	H	The watermark of a security card or driving licenses required between 150 and 200 bits of information (owner name, date of birth, license number, and other information) [Xu 05].
e-Commerce	H	M	M	The data payload is about 96 bits [Pat 02].

I.5. Commercial Products

Trying to reach the trade-off between the huge economic (Table I.1) & applicative (Table I.2) interests the industrial players offer a large variety of "ready to use" watermarking solutions. Unfortunately, the market landscape is still very fragmented and information concerning objective performances is very scarce.

Note that even the nowadays emerging consortia do not reach yet a real convergence and coherence in their efforts. For instance, Digimarc, Hitachi, Macrovision, NEC, Philips, Pioneer and Sony created *Video Watermarking Group* (VWM) [Dig 03] joining companies from motion picture, computer and consumer electronics industries. The goal of this group is to develop digital video watermarking techniques that protect the illegitimate copying. In September 2006, the *Digital Watermarking Alliance* [Dwa 08] was created in order to build awareness of the values of the digital watermarking for content owners, industry and consumers. The members of the DWA are: AquaMobile, Cinea, DataMark Technologies, Digimarc, GCS Research, Gibson, ISAN, Jura, MarkAny, MediaGrid, MSI, Nielsen, Philips, Signum, Streamburst, Teletrax, Thomson, Université Catholique de Louvain, Verance, Verimatrix, and Widevine.

The rest of this section will summarize the results of an Internet – based study on the commercial products, Table I.3. The presentation, which has no advertising purpose, is structured according to the company and follows an alphabetical order.

Table I.3. Digital watermarking solutions offer by different companies.

Applications	Companies
Asset & Content management	**Digimarc, Philips, Teletrax**
Copy protection	**AlpVision, AiS WPP, Alpha Tec, IGD, MarkAny, Sarnoff Co., Verimatrix**
Filtering/Classification	**Digimarc, Verimatrix**
Forensic tracking	**Activated Content, Cinea, Digimarc, Philips, Pixeltools, Signum, Thomson, Widevine**
Rights management	**Digimarc, MarkAny, Philips, Pixeltools, Sarnoff Co., Verimatrix**
Broadcast & Internet monitoring	**Easy Watermarking Creator, IGD, Philips, Pixeltools, Teletrax, Thomson, Verimatrix**
Copyright communication	**Digimarc, Philips, Teletrax, Signum**
Remote triggering	**Philips**
Authentication & Integrity	**DataMark Technologies Pte., Digimarc, MarkAny, Pixeltools, Signum**
e-Commerce	**Digimarc, Philips, Signum**

Activated Content

Activated Content claims to provide a robust, imperceptible and secure audio watermarking technology addressing both the digital and the traditional analog technologies [Act 08].

The following products are proposed by the company:

- *Activator* is a software which combines forensic tracking and content monitoring. The product is available for *Windows & Windows Mobile*. The *C* implementation is also provided in order to allow its integration with customisable watermarking functions even at a lower level.
- *Webstore* software supports .WAV and .MP3 files. The watermarking is done on the client Windows 2003 server while the meta-data are implicitly stored on the Activator Server. The meta-data can be optionally stored on the client server.

Activated Content costumers include major and independent record labels, recording studios, mastering studios, duplication houses, radio and television broadcasters, and online content distribution services worldwide (*example:* G-Unit Records, Masterpiece Duplication, MediaDISC, SonyBMG, Universal Music, Viacom, Warner Bros., *etc*).

Alpha Tec

In order to protect image, audio, video and 3D objects ALPHA TEC LTD [Alp 08a] provides 7 products, namely *AlphaCrawler*, *AudioMark*, *EikonaMark*, *PolyMark*, *MeshMark*, *VideoMark* and *VolMark*. No particular hardware and software resources are required for these applications.

- *AlphaCrawler* is a web crawler[1] searching from an initial web page all linked media content for existing marks embedded by one of the company softwares [Alp 08b]. The results consist in lists of all investigated files and of watermarked files

  This solution can be used for image copyright protection and authentication.
- *AudioMark* is a software designed for inserting imperceptible watermarks into the audio content [Alp 08c] and working in real-time.

  The mark is supposed to survive to digital audio processing operation (filtering, re-sampling and re-quantization) and MPEG audio compression up to a low quality factor.
- *EikonaMark* is devoted to inserting & detecting imperceptible watermarks in digital images. The mark is expected to be robust against JPEG compression up to a low quality factor and for different digital image processing operations. The watermarking detection is oblivious.

  The applications of this software consist in copyright protection, and authentication [Alp 08d].
- *MeshMark* is a software for obliviously embedding imperceptible watermark into 3D objects (3D graphics models, 3D animation character models, 3D games and 3D terrain models for geographic information systems) [Alp 08e]. The mark should be robust against 3D rotation, translation and uniform scaling and suitable for small and big 3D meshes.

[1] A web crawler is a software which browses the www in a methodical and automated manner.

The program works only with VRML2 files.

MeshMark is applicable to copyright identification and copy protection of 3D meshes.

- *PolyMark* is a software for watermarking vectorial two–dimensional computer graphics (digital maps – 2D maps, cartoons and vector computer graphic files, files of image contours, edges and image region borders, *etc*.) [Alp 08f].

  This version of the software has included a module for opening, reading and writing Geographic Information System (GIS) files. The mark is inserted in the polygonal lines existing in this type of files.

  The main application of the software is connected to copyright protection.

- *VideoMark* uses the imperceptible watermarking inserted into the video content. The software supports five embedding levels which are associated to the strength of modifications performed for the watermark embedding [Alp 08g].

  The system was designed to be robust to MPEG2 encoding up to compression ratios 1:50 (5Mbps compressed video) and provide copyright protection.

- *VolMark* is a software for embedding imperceptible 3D watermarks into digital greyscale images, and 3D images (volumes) [Alp 08h].

  The mark is supposed to be robust against and different digital image processing operations (*e.g.* JPEG compression). The watermarking detection is oblivious. The application of the software is in the copyright protection of the 3D content.

AlpVision

AlpVision was created in June 2001 as a start-up coming out from the Signal Processing Institute, Ecole Polytechnique Fédérale de Lausanne. The goal of the team is to develop security technologies applied to brand protection (protect consumer products against counterfeiting and fraudulent imports) and document security (fighting against criminal acts related to printed documents) [Alv 08].

AlpVision does not have direct products for sale but it licenses its technology to third parties. AlpVision filed US and EU registered patents (for *example*: patent US6785332 and EP0997042 *"Method for watermarking a compressed digital video signal"*).

AiS Watermarker

AiS Watermark Picture Protector (AiS WPP) is a software developed in order to protect digital images (about 40 different file formants) using perceptible watermarking. The software is devoted to image galleries, designers, computer painters and banner makers who want to send their production and to contest through the Internet [Ais 08].

The *AiS Watermark Picture Protector* is accessible as a standalone program for *Windows*; it is also possible to integrate this program in customer software written in C++, VB, Delphi, *etc*.

Cinea

Cinea, Inc. (part of Dolby Laboratories) is an anti-piracy pioneer focused on developing and commercializing solutions for studios, distributors, exhibitions and vendors [Cin 08]. The team holds 20 patents in content protection and forensic watermarking.

Among its clients, both content producers (Walt Disney, Sony Pictures Entertainment, New Line Cinema and Miramax Films) and vendors (Cinram and Deluxe Media Service) can be mentioned.

One of its products, *RUNNING MARKS* was deployed in over 200,000 homes and successfully protected more then 700 Hollywood titles. The product is integrated in the DVD systems.

The system uses an imperceptible watermark. The insertion is realized into a compressed content and the detection of the watermark is oblivious.

The data payload (64 bits) represents the playback device ID, the date & time, and other information according to the specific application.

DataMark Technologies Pte

DataMark Technologies Pte (DMT) is a subsidiary of Singapore Technologies Electronics and is a provider of patented digital watermarking solutions [Dat 08]. The security technologies and patented digital watermarking technologies address the copyright and authentication issues.

- *StegMark ImageLITE* is a software (working on-line) for authentication of an images captured by HP scanners. Users are able to digitize their certificates and documents (after the scanning operation) and to insert their copyright information (*example*: name, date and time) onto digital content to ensure the ownership and the authenticity of the images.
- *StegMark Digital Camera* deals with the SLR (Single–Lens Reflex) digital cameras that contain a block chip for copyright and picture integrity protection using watermarking technologies. With this new camera, the photographs are instantaneously watermarked and protected, this operation being done transparently and in real–time (Figure I.6). Note that, in such a case, there is practically no unwatermarked image (the original no longer exists).

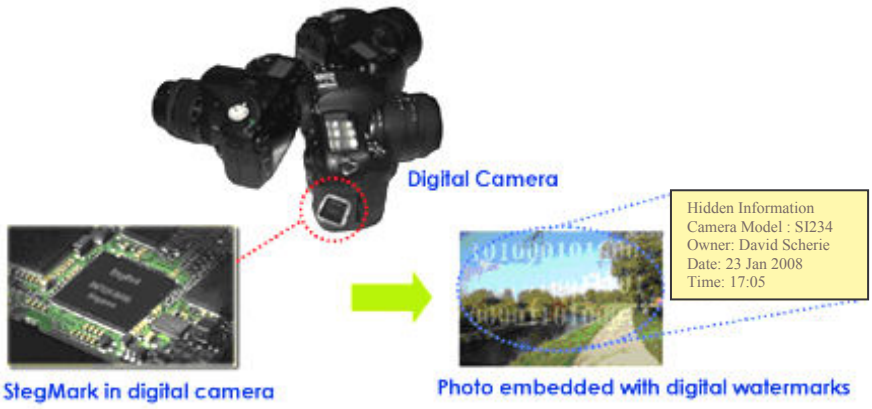


Figure I.6. StegMark Digital Camera schema [Dat 08].

The watermark is robust against common image manipulations: JPEG compression, format conversion, contrasts adjustments, *etc*. A stamp is also inserted in the form of fragile watermarks so that any image tampering can be detected.

- *StegMark CF Module* is a flash memory storage device with the additional functionality of a StegMark Digital Watermarking engine that is transparent to the user. The copyright information to be inserted as robust watermarks is user-defined and can be set through a simple user interface.
- *StegMark SDK* is provided in the form of a Component Object Model API. This allows programmers working in C++, MFC, Visual Basic and other high level languages to easily develop watermarking applications. The software is Windows 2000 or XP compatible.
- *StegMark Video* is an authentication solution for video surveillance using digital video watermarking techniques to secure video content obtained from the video cameras. The invisible watermark is inserted in each frame of the video. The information to be embedded in the video content are the frame number, the camera ID, and the recording date & time. This robust watermark is designed to resist to different frame modifications: removal, addition, averaging and swapping (Figure I.7).

Figure I.7. StegMark Video schema [Dat 08].

Digimarc

Digimarc is the owner of more than 250 patents for all the topics of multimedia watermarking (both analog and digital format) [Dig 08a]: images (pictures), video (movies, TV and broadcast), audio (music and radio), mobile (mobile computing & commerce), financial & identification documents, *etc*.

The main products developed by Digimarc are: *ImageBridge*, *MyPictureMarc* and *IDMarc*.

- *ImageBridge* is a solution to protect the images and brands [Dig 08b] and is applicable to tracking and Internet monitoring.

  The *ImageBridge* is used by major stock photo agencies and leading brands (like manufactures, e-tailers, movie studios and other companies) to protect their digital images and to track image distribution.

  The advantages of the product are claimed to be automatic image monitoring on the Internet, unauthorised usage identification, on-line channel programs management, copyright communication and image ownership proving.

- *MyPictureMarc* was designed to ensure the watermark robustness against usual image manipulation and is a solution for copyright communication application (to create online portfolio for photographers, artists or Web designers). The product has multiple layers of security allowing to communicate copyright and contact information. Image back-up and sharing facilities are also provided.

  The product offers: covert (imperceptible) watermarking to embed the copyright information into images, visible watermarking to prove ownership, on-line linking of copyright information and commerce capabilities, on-line tracking & watching image usage [Dig 08c].
- The *IDMarc* ensures the authentication of the ID documents like driving licences, identity cards, passports *etc* [Dig 08d]. Imperceptible for the human eye, *IDMarc* can easily be read by most available document scanners used in the field. These scanners read and decode the watermark in order to detect counterfeit IDs (photo swapping, data modification, *etc*).

  In 2007, more than 100 million of circulating driver licences are protected with *IDMarc* in 24 U.S. States. Starting from 2008, the *IDMarc* is incorporated into almost all new driver licenses produced in United States.

Besides these large market products, Digimarc is also active on some specialised market. For instance, its US patent *EP1142190 - Counterfeit deterrence system*, protects banknotes or other security document-images (passports, visas, postal stamps, travellers' checks, and a variety of tickets) using digital watermark.

In 2004, the CBCDG (Central Bank Counterfeit Deterrence Group which is a group of 27 central banks) proposes a system in cooperation with Digimarc to *"prevent personal computers and digital imaging tools from capturing or reproducing the image of a protected banknote"* by using the watermarking technology [Cdc 08], [Mur 04].

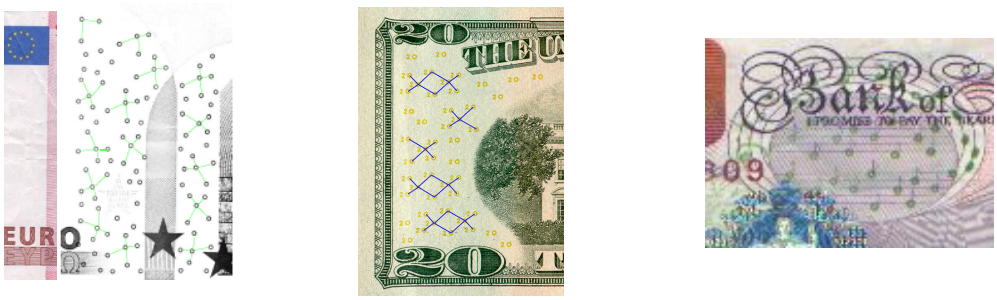


Figure I.8. These images, from the 10€, 20$ and 20£ contain the entire *EURion constellation* [Kat 08].

The *EURion constellation* is a pattern of symbols found on a number of banknotes (Australian dollar, Canadian dollar, Danish krone, Euro, Japanese yen, Norwegian krone, Pound sterling, Swedish krone, US dollar, *etc*) designed in 1996 by Markus Kuhn [Kuh 02]. *EURion*, consists of a pattern of

five small circles, by different colours, which are repeated across areas of the banknote at different orientations. Technical details regarding the *EURion constellation* are kept secret by its inventors (patent by OMRON Corporation [Kat 08] called "Omron anti-photocopying feature") and users (*e.g.* CBCDG).

Easy Watermarking Creator

Easy Watermark Creator 2.1 [Eas 08] is a tool for adding perceptible watermarks to images. This software can be applied to protection from unauthorized use of the images published on the Web.

Its the characteristics are: accepted image files are BMP, JPEG, PSD, PCX, PNG, TGA, TIFF, GIF; the watermark may represent text, image or a line stamp; and allows to apply a selected list of watermarks to a group of image files (batch wizard). The tool is compatible with all the *Windows* versions.

Fraunhofer Institut für Graphische Datenverarbeitung – IGD

The academic research carried out at the Franunhofer Institut für Graphische Datenverarbeitung in Darmstadt – Germany resulted in a prototype technology to protect audio files like MPEG Layer–3 (MP3) and MPEG AAC (Advanced Audio Coding) [Igd 08a].

Certimark, Geomark, Octalis, Okapi, SysCoP, Talisman and *Wedelmusic* are projects [Igd 08b] developed in the Security Technology department at IGD which guarantee the quality of the marked data and provide robustness of the embedded watermark:

- *Certimark* is a complete benchmark for image and video watermarking technologies addressing a wide selection of watermarking applications (broadcast monitoring, authentication & integrity, fingerprinting).
- *Geomark* is a system capable to embed private watermarks in 3D geometric data. The system is robust against translation, scaling, rotation, removing of faces, re-meshing operations, polygon simplification and addition of noise.
- *SysCoP* is a collection of software for embedding digital watermarks into images and video streams for copyright protection. The invisible mark survives a wide range of image processing modifications.
- An application of the *Talisman* project is broadcast monitoring, where the dissemination of digital video streams can be tracked. The product was tested with success during the football world championship in France (1998).
- *Wedelmusic* was designed to protect distribution the digital music (speech, audio files or video clips) through Internet.

Kodak

With new Kodak's new watermarking technology, every screening of every digitally projected movie may have its own unique code (Figure I.9).

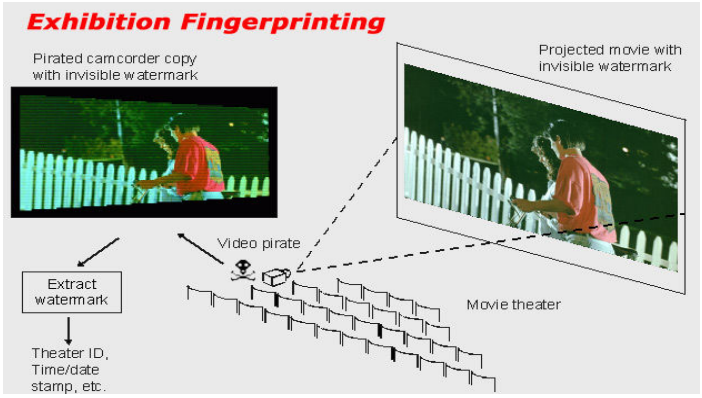


Figure I.9. Exhibition fingerprinting [Kod 08].

In this respect, the imperceptible watermark should include a unique ID code that identifies the cinema and the date and time of the movie showing [Kod 08].

Recent testing made by Kodak has demonstrated robust watermark extraction, using 32 bits per frame – more than enough to assign a unique ID for every showing of every movie that has been–or will be–produced.

MarkAny

MarkAny is a South Korea company that develops products against multimedia piracy. It provides solutions for copyright protection, authentication and rights management [Mka 08] and holds more than 200 patents.

MarkAny has provided DRM solutions to more than 500 companies and governmental agencies throughout the world:

- *MAIM* (MarkAny Image) imperceptibly inserts watermark into image in order to protect digital libraries, Internet shopping web–sites, image providers, *etc*.

  The features of the *MAIM* are: variable number of inserted logos, robustness against image compression and mundane image processing, real-time operating. The software works under *Windows OS*.
- *MAO* (MarkAny audiO) is implemented for audio files and target service providers (including MP3 files), CD producers, sound-source producer, and audio on demand (AOD) service providers.
- *MAVI* (MarkAny VIdeo) is meant to protect video distribution across Internet against illegal copying.

  MAVI is a real-time watermarking solution capable to embed up to 64bits of information in each frame and surviving to cropping and resizing.

  It accepts only the following file formats: *.m2v* (MPEG-2 video stream files), *.avi* (Audio Video Interleave), *.asf* (Advanced Streaming Format), and *.wmv* (Windows Media Video). The

supported data rate is 300kbps and video formats are CCIR80 1 – 720x480, VGA 640x480, S-VGA 1280x768, SIF 360x240 and HDTV 1920x1035. It is available only for *Windows*.

The target areas are: video on demand (VOD), broadcasting service providers, digital video libraries, on-line education, and Internet& satellite broadcasting.

Philips

Philips products are connected to watermarking and fingerprinting applications like: media identification (*CineFence, VTrack, CompoTrack* and *RepliTrack*), broadcast monitoring (for video monitoring there is a partnership with Teletrax), piracy deterrence (Philips integrates a chip in all camcorders, DVD players that can identify which content can be seen/listened and which content cannot), Internet monitoring (in partnership with MediaHegde, Philips provides content identification, management and monitoring services) [Phi 08a].

- *CineFence* aims to stop piracy in digital cinema. Compatible with the DCI (Digital Cinematography Initiative) standard, *CineFence* is able to link the captured movies back to the cinema where the content was recorded. The watermark is robust and can survive to camcorder and microphone capturing in the cinema, as well as to the main compression techniques [Phi 08b].

  Concerning the video, it allows 35 bits to be inserted into a sequence of 5 minutes. *CineFence* is available to embed in 2K and 4K resolutions with 24 and 48 frames per second.

  For the audio, the same data payload (35 bits) can be inserted into a sequence of 5 minutes of audio content. The sampling frequency is 48 kHz or 96 kHz.

  The *CineFence Video* detector can work with all common video formants and *CineFence Audio* detector is able to perform in all audio formats after conversion to WAV (WAVeform audio) format.
- *RepliTrack* may be used for uniquely marking and identifying multimedia content stored on CDs and DVDs [Phi 08d]. The *RepliTrack* inserts 21 bits in 90 seconds of video. The system can automatically operate: reading the source of DVD/MPEG file, embedding, duplicating, auto CD/DVD labelling (printing a perceptible mark and notice on the disk) and embedding directly in ISO 9660 image [Iso 88].

  The technology is applicable to rights protection and monitoring and the product is compatible with all the watermarking products and components developed by Philips using *CompoTrack* technology.
- *VTrack* is a solution to uniquely mark and identify the content delivered to individual PayTV subscribes. It is developed for video on demand (VOD), PayTV, Hotel TV and IPTV[2]

[2] IPTV (Internet Protocol Television) is a system where a digital television service is delivered by using Internet connection.

[Phi 08f]. Accepted formats are SD (Standard – Definition), HD (High – Definition), MPEG–2, MPEG–4 Part 2, and MPEG–4 Part 10 (H.264/AVC) and VC-1.

The mark is robust to camcorder copying and severe quality degradation. The detection of watermarking is oblivious. The embedding is in real – time and uses a unique ID that contains the date/time and operator identification cod.

The technology is applicable to broadcast and Internet monitoring.

- *CompoTrack* is a package with three components: *CompTrack MPEG*, *CompoTrack H.264* and *CompoTrack WAV* [Phi 08c].
  - *CompTrack MPEG* is a solution to mark in the video content. The data payload is 21 bits and is inserted in any 90 seconds (2700 frames). The mark would survive to shifting, cropping, scaling, recompression to MPEG-1, DivX, WMV, noise addition, digital/analog conversion, noise and median filtering.

    The mark can be directly embedded into the MPEG-2 SD (Standard – Definition) and HD (High – Definition) stream as well as into SD ISO 9660 DVD images.
  - *CompTrack H.264* has the same performance as *CompoTrack MPEG* (data payload and robustness) and was designed for MPEG–4 Part 10 (H.264/AVC) format.

    The technology for video watermarking is applicable to rights protection, tracking and monitoring and the *CompTrack MPEG* and *CompoTrack H.264* are fully compatible with all other Philips *CompoTrack* and *RepliTrack* video watermarking products.
  - *CompTrack WAV* is devoted to audio protection. The data payload is 37 bits and is inserted in 45 seconds; the mark is embedded in real-time in WAV format and broadcast WAV stream. The detection is robust against digital-analog and analog-digital conversion, MP3 and additional AAC (Advanced Audio Coding**)** encoding/decoding, filtering (all-pass and band-pass), echo addition, amplitude compression, re-sampling, speed change and noise addition. The applications are rights protection, tracking and monitoring.

PixelTools

PixelTools is a group dedicated to image processing with the goal of developing the highest quality video compression software for video editors, animators, and broadcasters [Pix 08].

PixelTools' patented *MPEGEscort* watermarking software slightly alters a compressed MPEG video to insert a visually indiscernible digital data file. This software watermarks any MPEG-1 or MPEG-2 elementary video stream. It may be used to extract or verify the presence of a watermark or of a digital signature. The data payload is quite large (dozens to hundreds of bits per 720x480 frame). The insertion is in real-time. The software is available for Mac, PC and UNIX platforms.

Sarnoff Co.

The *iTrace* was designed in order to transparently watermark the Digital Cinema content [Sar 08]. The watermark survives to camcorder capture, copying, compression, Internet distribution, cropping and other modifications. The *iTrace* watermarking is compatible with both analog and digital media including VHS tapes, DVD, DigiBeta professional tapes, digital cinema, and internet files (*example*: Windows Media).

The *iTrace* is a watermarking system for HD and SD content that work in real-time.

Signum

The Signum products [Sig 08] are used by many well known companies (France Press, BBC, British Museum, IBM, HP, Toyota, Mitsubishi, *etc*):

- *SureSign Entreprise* is an automated batch-processor watermark embedding and detection application for image protection. The accepted file formats are: JPG, TIFF, BMP, PCX, PNG, Targa and PICT-RGB, CMYK or greyscale with the minimum size 150x150 pixels. The product is available only for *Windows OS*.
- *SureSign SDK* can be integrated in many software, firmware and hardware products. These cover a large area of applications such as: image and photo database, operating system software, digital SLR cameras, e-commerce systems, 2D maps, GIS systems, medical imaging systems, *etc*. The SDK is available for *MacOS*, *Linux* and *UNIX* (Solaris), but versions for other *OS*s are available upon request. The test images and the source code/library (C or C++) are included in the package offered to the potential developers.
- *SureSign Plug-ins* is designed to work with Adobe Photoshop or Corel Photopaint to embed and read watermarks in digital images. It works with all the RGB, CMYK and greyscale images with minimum size of 150x150 pixels. There are two versions of the software, one for *Windows* and one for *Mac*.

Teletrax

Teletrax is a company that offers a watermarking solution for broadcast monitoring [Tel 08]. The watermark is robust to standard conversions applied to the video content.

Currently, Teletrex monitor about 1500 broadcast channels (ABS Television, BBC, CBS Television, Fox Broadcasting Company, NBC News Channel, NBC Agency, Reuters Television, *etc*) from more than 50 countries.

The system implemented by Teletrax for the broadcast verification service is based on the digital watermarking technology developed at Philips.

Thomson

Thomson offers solutions to track and secure digital content throughout production and post-production stage, distribution, *etc*. In April 2008, it announced its collaboration with IBM and the

ISAN-IA (International Standard Audiovisual Number International Agency) for content tracking and security solution.

NexGuard, *NexTracker* and *Nextamp* are products developed by Thomson for several video applications (forensic, content protection, broadcast and Internet monitoring) [Tho 08].

- *NexGuard* provide solutions for content protection and for forensic marking devoted to the media and entertainment industries.
  - *NexGuard – content protection* claims to assure confidentiality and conserve the ownership of the content. It is conceived as a package of professional solutions: for post-production, for limited distribution, for the jury in different cinema festivals or for marketing campaigns.

    The product is available for *Windows* and for *Mac OS*s.

    NexGuard – forensic is a robust watermarking solution using an imperceptible mark. This is a solution applied to digital cinema exhibition, media distribution and VOD.

    Depending on the application, four version of this product are offered:

    a) NexGuard Forensic Marking Masters and Files (Figure I.10) also exploits the discussion if recipients of a given content know each individual copy of the video can be tracked, the temptation to illegally reuse the content is sharply reduced.

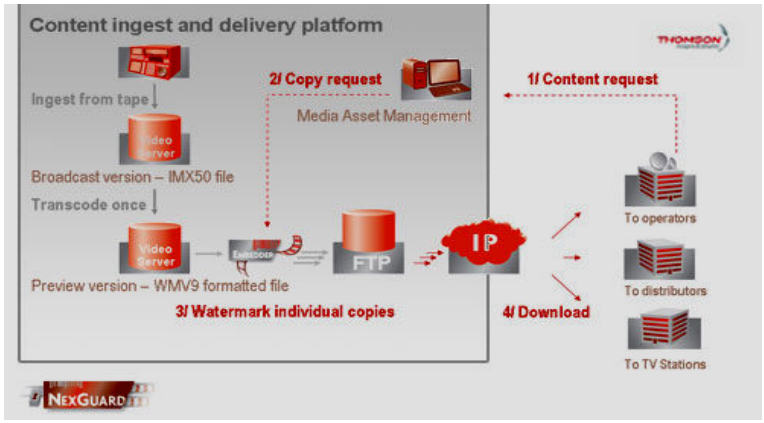


Figure I.10. *NexGuard Forensic Marking Masters and Files* schema [Tho 08].

The product is intended to secure internal validation and primary distribution of content, and to ease and secure the workflow for content distributors, broadcasters and web portals. The watermarking is claimed to resist to attacks at any level in the production chain [Tho 08]. The various file formats are accepted, including the MPEG/IMX and the WMV9/VC-1 formats. There are two versions of the product, for *Windows* and one for *Linux*.

b) NexGuard Forensic Marking Screener ensures the traceability of individual DVDs, tapes and files in audio/video format, without affecting the viewer experience.

The product can imperceptibly watermark each DVD with identifying information (*e.g.* a unique ID per copy). This ability helps secure the delivery of DVDs screeners and trace illegally copied content to its original source.

c) NexGuard Forensic Marking Set-top Boxes (Figure I.11)

Forensic watermarking extends content security beyond the STB (Set-top boxes) and acts as a deterrent against piracy, as subscribers are alerted of that technical capability.

The watermark embedding is entirely executed on this STB and is performed right after decoding. It is applicable to live broadcast being independent from the video codec used to transport the video content (MPEG–2, MPEG–4 Part 10 (AVC–H.264), VC-1). Note that both the digital HDMI and the analog YPbPr HD video output interfaces of the STB contain the imperceptible watermark

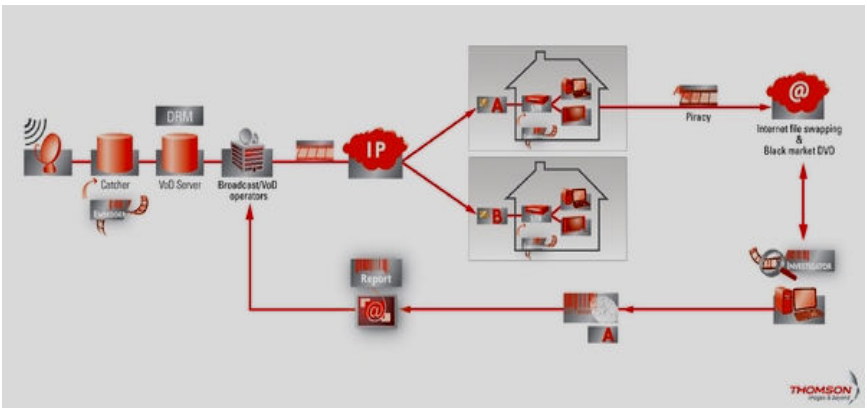


Figure I.11. *NexGuard Forensic Marking Set-top Boxes* schema [Tho 08].

For VOD and Pay TV the main characteristics are:

- it operates with any compression scheme (MPEG–2, MPEG–4 Part 10 (H.264), VC-1),
- VOD content traceability on a per-transaction basis,
- enhances the security of cable and IPTV VOD streaming distribution, satellite or terrestrial push – VOD,
- identifies subscriptions involved in illicit redistribution of TV signal.

d) NexGuard Forensic Marking Theatrical-DCI (Figure I.12)

The watermarking product is a solution provided by Thomson to embed the date, time and origin of projection into a digital motion picture visual and audio contents.

The product is integrated in most d-Cinema[3] and e-Cinema[4] [Ctm 08] server manufacturers. The watermark is imperceptible and the embedding can be made for MPEG/IMX and WMV9/VC-1 formats.

[3] d-Cinema means Digital Cinema: it is distinct from high-definition television, and is use to distribute and project motion pictures.

[4] e-Cinema means all the techniques of digital on the big cinema. The equipment are used to project advertising and alternative content (concerts, sport events, documentaries, *etc*).

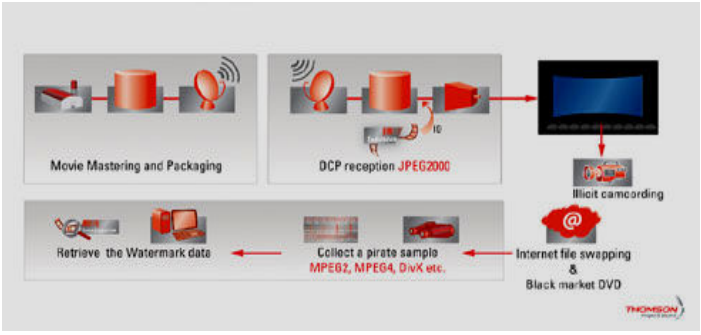


Figure I.12. *NexGuard Forensic Marking Theatrical-DCI* schema [Tho 08].

e) NexGuard Forensic Marking VOD Server is specifically designed for cable and IP network VOD services (*e.g.* IPTV). It inserts the mark in the compressed stream based on a unique transactional ID.

- *NexTracker* provides reliable broadcast (radio/TV and/or Internet).
  - *NexTracker – broadcast monitoring* (Figure I.13)

    This product is meant to verify the delivery of the media content. An illustrative example is the TV monitoring (promos, commercial and sponsorship).

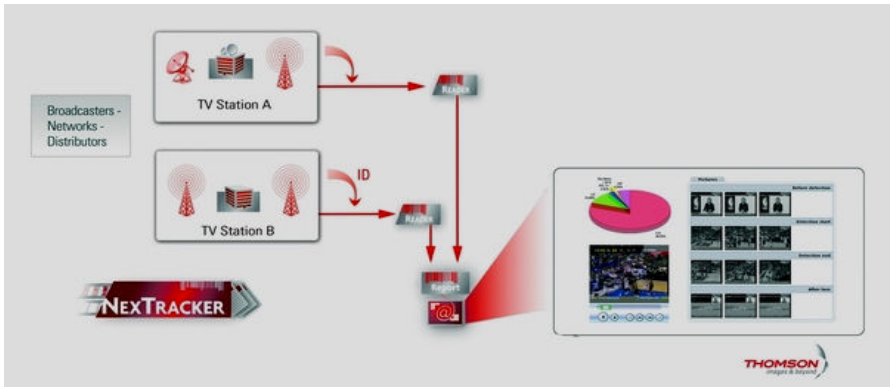


Figure I.13. *NexTracker* – broadcast monitoring schema [Tho 08].

  - *NexTracker – web monitoring*

    The product identifies and tracks the content on the Internet. The product would camcording, editing the replica of an original content obtained by mashups, parodies, sketches, *etc*.

- *Nextamp* used to be a company specialized in protection of video content based on the watermarking technology and was bought by Thomson in 1995.

  Nowadays *Nextamp* refers to a product commercialized by Thomson. In order to have a "content tracking and tracing using a virtual barcode" [Nex 08] mentions the following *Nextamp* properties: 1) an imperceptible mark (containing identifiers, date and time stamp) is embedded in compressed video (MPEG–2) or in compressed video; 2) the watermark is robust

against format conversion, editing compression or logo insertion; 3) generation of tracking report with transition storyboards.

Verimatrix

Verimatrix offers a tool to fight against video piracy with its *VideoMark*. The product is a solution for content security in pay–TV networks [Vmt 08].

VideoMark forensic watermarking technology was integrated in AmiNet125 set–top box for commercial market.

The watermark is robust and will survive to digital–analog and analog–digital conversion, camcorder capture, *etc.*

Widevine

Widevine Technologies provides content protection and forensic watermarking solutions for telecommunications, internet, mobile, cable and satellite service providers worldwide (over 130 operators use Widevine products and holds six patents) [Wid 08].

Widevine Mensor is a solution that invisibly watermarks content at multiple points in a video distribution network and employs session-based watermarks in real-time to protect the content that is streamed from a server to a client.

- *Widevine Mensor* identifies and tracks assets distributed to each operator or third-party service providers. This enables them to track the effectiveness of an operator's content protection system across all networks, while offering a way to quickly track and identify authorized and unauthorized distribution (see Figure I.14).

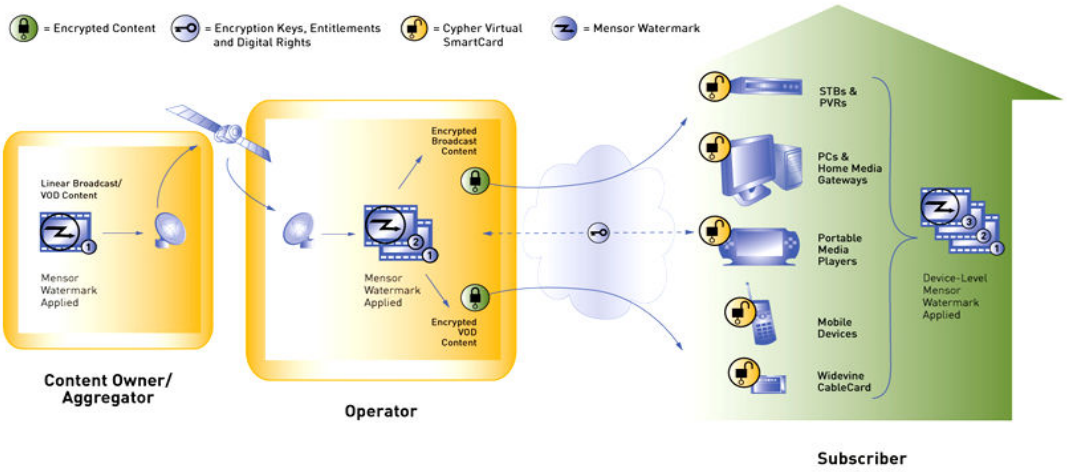


Figure I.14. *Mensor* identifies content at three key points in the video delivery, from content owner and aggregator, to operator and finally at the device [Wid 08].

Depending on content ownership requirements, *Widevine Mensor* utilizes watermarking algorithms offered by Cinea and/or by Thomson:

- watermark is robust against digital and analog conversion, transcoding and degradation;
- the embedding is realised at the studio, operator or on consumer devices;

the product works on STBs, PVRs (Personal Video Recorder), PCs, portable media players or Widevine CaleCard.

I.6. Conclusion

Some years ago a kind of esoteric handcraft, watermarking is nowadays an effervescent industrial field. Unfortunately, its landscape is very fragmented and heterogeneous: each "wise" protection procedure is followed by a "clever" attack, thus making it practically impossible to ascertain which are the watermarking limits and which are the tomorrow trends. Hence, in the fight against the multimedia piracy, to define a unitary, reliable and generic watermarking framework becomes a stringent necessity.

I.7. References

[Act 08] Activated Content – home page
http://www.activatedcontent.com

[Ais 08] Ais Watermarking Pictures Protector (AiS WPP) – home page
http://www.watermarker.com/watermark-protector/

[Alp 08a] ALPHA TEC LTD – home page
http://www.alphatecltd.com/watermarking/watermarking.html

[Alp 08b] ALPHA TEC LTD product – ALPHACRAWLER
http://www.alphatecltd.com/watermarking/alphacrawler.html

[Alp 08c] ALPHA TEC LTD product – AudioMark
http://www.alphatecltd.com/watermarking/audiomark/audiomark.html

[Alp 08d] ALPHA TEC LTD product – EIKONAMARK
http://www.alphatecltd.com/watermarking/eikonamark/eikonamark.html

[Alp 08e] ALPHA TEC LTD product – MeshMark
http://www.alphatecltd.com/watermarking/meshmark/meshmark.html

[Alp 08f] ALPHA TEC LTD product – PolyMark
http://www.alphatecltd.com/watermarking/polymark/polymark.html

[Alp 08g] ALPHA TEC LTD product – VideoMark
http://www.alphatecltd.com/watermarking/videomark/videomark.html

[Alp 08h] ALPHA TEC LTD product – VolMark
http://www.alphatecltd.com/watermarking/volmark/volmark.html

[Alv 08] AlpVision – home page
http://www.alpvision.com/video-watermarking.html

[Bar 98] R. Barnett, D.E. Pearson, *"Frequency mode LR attack operator for digitally watermarked images"*, Electronics Letters, Vol. 34, September 1998, pp. 1837 – 1839.

[Cdc 08] Central Bank Counterfeit Deterrence Group
https://www.rulesforuse.org

[Cer 08] CertiMark: Certification of Watermarking Techniques
http://www.certimark.org

[Ctm 08] CTM Solutions – D-Cinéma et D-Cinéma
http://www.ctmsolutions.com

[Cox 00] I. J. Cox, M.L. Miller, J.A. Bloom, *"Watermarking Applications and their Properties"*, Proceedings of the International Conference on Information Technology: Coding and Computing – ITCC2000, 2000, pp. 6 – 10.

[Cox 02] I. J. Cox, M.L. Miller, J.A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2002.

[Cox 08] I. J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers, 2008.

[Dat 08] DataMark TechnologiesPte Ltd – home page
http://www.datamark.com.sg

[Dig 03] Digimarc Corporation, *"Digimarc Comments to USPTO Regarding Technological Protection Systems for Digitized Copyrighted Works"*, January 14, 2003.

[Dig 08a] Digimarc – home page
https://www.digimarc.com/tech/applications.asp

[Dig 08b] Digimarc product – ImageBridge
https://www.digimarc.com/comm/imagebridge.asp

[Dig 08c] Digimarc product – MyPictureMarc
https://www.digimarc.com/comm/mypicturemarc.asp

[Dig 08d] Digimarc product – IDMarc https://digimarc.com/govt/idmarc.asp

[Dig 09a] R.K. Sharma, S. Decker, *"Practical Challenges for Digital Watermarking Applications"*, Digimarc – Publish Papers,
www.digimarc.com/appdev/published_papers.asp .

[Dig 09b] A.M. Alattar, E.T. Lin, M.U. Celik, *"Watermarking Low Bit-rate Advanced Simple Profile MPEG-4 Bitstreams",* Digimarc – Publish Papers,
www.digimarc.com/resources/docs/tech_papers/dmrc_mpeg4_compressed.pdf

[Dum 06] C.O. Dumitru**,** S. Duță, M. Mitrea, F. Prêteux, *"Commercial and Standard in Robust Multimedia Watermarking"*, WSEAS Transactions on Communications, Issue 9, Vol. 5, September 2006, pp.1667 – 1673.

[Dum 07] O. Dumitru, S. Duță, M. Mitrea, F. Prêteux, *"Gaussian Hypothesis for Video Watermarking Attacks: Drawbacks and Limitations"*, EUROCON 2007, Warsaw–Poland, September 2007, pp.849 – 855.

[Dwa 08] Digital Watermarking Alliance (DWA) – home page http://www.digitalwatermarkingalliance.org/membership.asp

[Dww 08a] Digital Watermarking World: StirMark http://www.watermarkingworld.org/stirmark/stirmark.html

[Dww 08b] Digital Watermarking World: CheckMark http://www.watermarkingworld.org/checkmark/checkmark.html

[Dww 08c] Digital Watermarking World: OptiMark http://www.watermarkingworld.org/stirmark/index.html

[Eas 08] Easy Watermark Creator http://www.easyimagetools.com/products/watermark/

[Fer 99] French Encryption Regulation by Yves Le Roux

[Fra 08] Fraunhofer IIS – home page http://www.iis.fraunhofer.de/EN/bf/amm/projects/index.jsp

[Hol 99] M.Holliman, N. Memon, M.Yeung, *"Watermarks estimation through local pixel correlation",* SPIE Electronic Imaging, San Jose, USA, January 1999, pp.134-146.

[Ipa 09] International Intellectual Property Alliance (IIPA) – home page http://www.iipa.com

[Ipa 08] International Intellectual Property Alliance (IIPA) – report http://www.iipa.com/special301_TOCs

[Igd 08a] Fraunhofer Institut Graphische Datenverarbeitung (IGD) – home page http://www.iis.fraunhofer.de/EN/bf/amm/index.jsp

[Igd 08b] Fraunhofer Institut Graphische Datenverarbeitung (IGD) – projects http://www.igd.fhg.de/igd-a8/en/projects/watermarks/index.html

[Iso 88] International Organisation for standardization ISO 9660 http://www.iso.org/iso/search.htm?qt=iso+9660&searchSubmit=Search&sort=rel&type=simple&published==on

[Kat 08] M. Katoh et al, *"Image Processing Deice and Method for Identifying an Input Image, and Copier Scanner and Printer Including Same"*, OMRON Corporation, US Patent 5845008.

[Kod 08] Kodak – home page http://www.kodak.com/country/US/en/corp/researchDevelopment/productFeatures/cinema.shtml

[Kun 01] D. Kundur, D. Hatzinakos, *"Diversity and Attack Characterization for Improved Robust Watermarking"*, IEEE Transactions on Signal Processing, Vol. 49, No. 10, October 2001, pp. 2382 – 2396.

[Kun 02] M. Kuhn, *"The EURion Constellation"*, 8 February 2002.
http://www.cl.cam.ac.uk/~mgk25/eurion.pdf

[Kut 99] M. Kutter, F.A.P. Petitcolas, *"A Fair Benchmark for Image Watermarking Systems"*, Electronic Imaging 1999, Vol. 3657, San Jose–USA, January 1999.

[Lan 98] G.C. Langelaar, R.L. Lagendijk, J. Biemond, *"Removing Spatial Spread Spectrum Watermarks by Non-linear Filtering"*, EUSIPCO 1998, Rhodes, Greece, September 1998.

[Leg 07] Décret n°2007-663 de 2 mai 2007 pris pour l'application des articles 30, 31 et 36 de la loi n°204-575 du 21 juin 2004 pour la confiance dans l'économie numérique et relatif aux moyens et aux prestations de cryptologie.
http://legifrance.gouv.fr

[Mka 08] MarkAny – home page
http://www.markany.com/en/index.asp

[Mit 08] M. Mitrea, C.O. Dumitru, S. Duta, F. Prêteux, A. Vlad, *"A Comparative Study on Video Watermarking Capacity",* COMMUNICATION, Bucharest, Romania, June 2008, pp. 335 – 338.

[Mpp 08] mp3PRO – new generation of mp3 proposed by Thomson – description
Http://www.all4mp3.com/info/mp3pro_about.html

[Mur 04] S.J. Murdoch, Software Detection of Currency, 2004
http://www.cl.cam.ac.uk/~sjm217/projects/currency/

[Mus 08] Music Trace – ContentMark solution
http://www.musictrace.de/products/contentmark.en.htm

[Nik 02] N. Nikolaidis, V. Solachidis, A. Tefas, V. Arguriou, I. Pitas, *"Benchmarking of Still Image Watermarking Methods: Principles and State of The Art"*, Proc. of Electronic Imaging & the Visual Arts (EVA 2002), Florence-Italy, March 2002.

[Nex 08] Nextamp Media Marking Solutions – home page
http://www.nextamp.com/en/watermarking/papers.htm

[Pat 02] Patent WO/2002/050760 – Audio/Video Commerce Application Architectural Framework.

[Pat 05] Patent WO/2005/071610 – Method of Allocating Payload Bits of a Watermark.

[Per 01] S. Pereira, S. Voloshynovski, M. Madueño, S. Marchand-Maillet and T. Pun, *"Second Generation Benchmarking and Application Oriented Evaluation",* Information Hiding Workshop III, Pittsburgh, PA, April 2001.

[Pet 98] F. Petitcolas, R. Anderson, M. Kuhn, *"Attacks on Copyright Marking Systems"*, Lecture Notes in Computer Science (LNCS), Vol. 1525, 1998, pp. 219 – 239.

[Phi 08a] Philips – home page
http://www.business-sites.philips.com/contentidentification/Section-14117/Index.html

[Phi 08b] Philips product – CineFence
http://www.business-sites.philips.com/contentidentification/products/section-14098/index.html

[Phi 08c] Philips product – CompoTrack
http://www.business-sites.philips.com/contentidentification/products/section-14097/index.html

[Phi 08d] Philips product – RepliTrack
http://www.business-sites.philips.com/contentidentification/products/section-14096/index.html

[Phi 08e] Philips product – Video Fingerprinting
http://www.business-sites.philips.com/contentidentification/Products/Section-14244/Index.html

[Phi 08f] Philips product – VTrack
http://www.business-sites.philips.com/contentidentification/products/section-14423/index.html

[Phi 09] Philips product – WaterCast
http://www.teletrax.tv/site/tech_spec.pdf

[Pit 96] I. Pitas, *"A Method for Signature Casting on Digital Images",* IEEE International Conference on Image Processing, Lausanne, Switzerland, September 1996, pp.215 – 218.

[Pix 08] PixelTools: MPEG Video Watermark Tool – home page
http://www.pixeltools.com/mpg_escort.html

[Sar 08] Sarnoff product – news
http://www.sarnoff.com/press-room/news/2005/02/11/dr-jeffrey-lubin-to-talk-on-sarnoffs-itrace-watermarking-technology-for-digital-cinema

[Tce 98] RU10: Technical Communication and Encryption
http://www.tc-forum.org/topicus/ru10tech.htm

[Sig 08] Signum – home page
http://www.signumtech.com/

[Sol 01] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, I. Pitas, *"A Benchmarking Protocol for watermarking Methods"*, IEEE International Conference on Image Processing (ICIP'01), Vol. 3, Thessaloniki – Greece, October 2001, pp. 1023 – 1026.

[Tel 08] Teletrax – home page

http://www.teletrax.tv/index.php/System

[Tho 08] Thomson – home page
http://www.thomson.net/GlobalEnglish/Pages/default.aspx
http://www.thomson.net/GlobalEnglish/Solutions/content-tracking-security/Pages/default.aspx

[Uma 08] uMark watermarking software – home page
http://www.uconomix.com/Products/uMark/Default.aspx

[Ver 08] Verance – home page
http://www.verance.com/products/index.php

[Vmt 08] Verimatrix – home page
http://www.verimatrix.com/solutions/forensic_watermarking.php

[Vol 01a] S. Voloshynovskiy, S. Pereira, V. Iquise, T. Pun, *"Attacks Modelling: Towards a Second Generation Watermarking Benchmark"*, Signal Processing Vol. 81, 2001, pp. 1177 – 1214.

[Vol 01b] S. Voloshynovskiy, S. Pereira, T. Pun, J.J. Eggers, J.K. Su, *"Attacks on Digital Watermarking: Classification, Estimation-based Attacks and Benchmarks"*, Communication Magazine, Vol. 39, No. 8, August 2001, pp. 118 – 126.

[Wid 08] Widevine Technologies – home page
http://www.widevine.com/cypher.html, http://www.widevine.com/mensor.html

[Xu 05] X. Xu, M. Tomlinson, M. Ambroze, M. Ahmed, *"Techniques to Provide Robust and High Capacity Data Hiding of ID Badges for Increased Security Requirements"*, the 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2005), Tunisia, March 2005.

"What is a scientist after all? It is a curious man looking through a keyhole, the keyhole of nature, trying to know what's going on."

Jacques Yves Cousteau (1910 - 1997)

(French naval officer, inventor, explorer and researcher, Christian Science Monitor, 1971)

Chapter II

Theoretical Background

This chapter gathers the fundamental concepts (definitions, properties, models) necessary to offer a fundamental basis for the methods and experiments developed in the thesis: random variables (Section II.2) and processes (Section II.3), information theory (Section II.4) and applied statistics (Section II.5).

Contents

II.1. Introduction

A communication system (Figure II.1) is meant to send information from an information source to a destination (be it a human observer or a device). In its largest acceptation, the system consists in five blocks [Sha 80]: (a) an information source which produces the message to be communicated to the receiver; (b) an transmitter which associates the message to a signal to be transmitted over the channel; (c) the channel which is the medium (a coaxial cable, a pair of wires, a band of radio frequencies, *etc*) used to transmit the signal from the transmitter to the receiver and which imposes some modifications (errors) on the transmitted signal; (d) the noise source which establishes the way in which the channel alters the transmitted information; (e) the receiver which realizes the inverse function of the transmitter, reconstituting the message from the received signal although errors might have been occured.

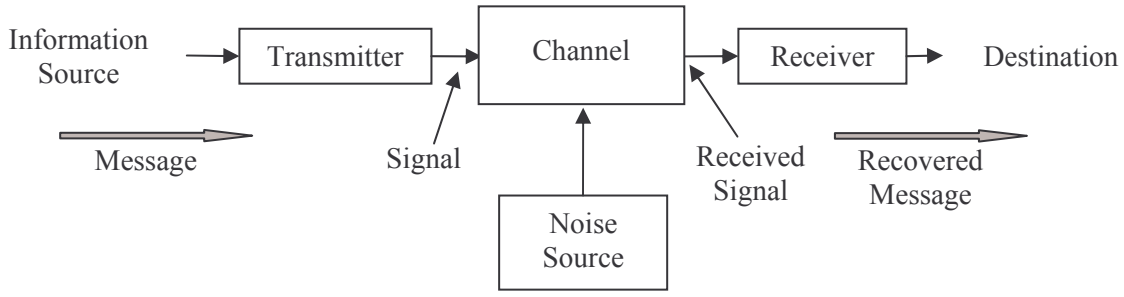


Figure II.1. Diagram of a general communication system.

The present thesis is focused on the mathematical models for the noise sources involved in the watermarking applications namely the natural video (see Chapter III) and the attacks (see Chapter IV).

In order to obtain accurate mathematical models for these sources, basic concepts from random variables (see Section II.2), random process (see Section II.3), information theory (see Section II.4) and statistics (see Section II.5) are to be exploited.

Note: in the rest of this chapter, the definitions of general concepts will follows the famous works in the literature [Bar 05], [Cov 06], [Dem 77], [Mcl 97], [Pap 91], [Pet 95], [Red 84], [Sha 48] [Sha 80] as well as the books used as student at POLITEHNICA University of Bucharest [Ciu 05], [Gav 03a], [Gav 03b], [Mur 98], [Spa 83], [Spa 87], [Ver 99], [Vla 99].

II.2. Random Variables

II.2.1. Probabilistic Space

A probabilistic space (H, $\boldsymbol{A}$, P) is composed of three elements: the sample set H, the event field $\boldsymbol{A}$, and the probability measure P.

- The **sample set** H is the multitude of all possible results η of a random experiment.

Example: The 6 possible faces obtained when throwing a die, or the incoming call numbers corresponding to a telephonic switching central.

- The **event field** $\boldsymbol{A}$ is a non-empty set, containing sub-sets of H with the following properties:
  1. $H \in \boldsymbol{A}$,
  2. $A \in \boldsymbol{A} \Rightarrow \overline{A} \in \boldsymbol{A}$,
  3. $A_1, A_2, ... \in \boldsymbol{A} \Rightarrow \bigcup_{i=1}^{n} A_i \in \boldsymbol{A}$.

  In this definition, $\overline{A}$ denotes the complement of A with respect to H, *i.e.* all elements of H witch are not in A; the integer n may be finite or infinite. A result $\eta_i \in H$ may be part of several events.

  Example: When throwing a die, the sample set is $H = \{\eta_1, \eta_2, ..., \eta_6\}$, where η_i is the occurance of face i. One possible event field is $\boldsymbol{A} = \{\phi, \{\eta_1\}, \{\eta_2, \eta_3, \eta_4, \eta_5, \eta_6\}, H\}$, where ϕ stands for the empty set.

- Given the sample space H and an event field $\boldsymbol{A}$, according to the Kolmogorov axioms a **probability function**, $P(\cdot)$ associates to the event A a real number such that:
  1. $P(A) \geq 0$, $\forall \ A \in H$,
  2. $P(H) = 1$,
  3. $P(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i)$ where A_i are some countable mutually exclusive events $(A_i \bigcap A_j = \phi, i \neq j)$.

After defining the concept of probability, the next step is to present its useful properties:

1. For any event A, the probability is $P(A) \leq 1$.
2. ϕ is called the *"impossible event"* and has a zero probability $P(\phi) = 0$.
3. $P(\overline{A}) = 1 - P(A)$.
4. If A and B are two events defined on H, $P(A \bigcup B) = P(A) + P(B) - P(A \bigcap B)$.
5. If $A_1, A_2, ... A_n$ are mutually exclusive events and if $H = A_1 \bigcup A_2 \bigcup A_3 \bigcup ... \bigcup A_n$ then $P(H) = P(A_1) + P(A_2) + ... + P(A_n) = 1$.

Conditional probability and independent events are now to be defined. Let A and B be two events. The conditional probability of A given B, $P(A/B)$ is defined as $P(A/B) = P(A \bigcap B)/P(B)$. If the A and B are independent events, then $P(A/B) = P(A) \Leftrightarrow P(A \bigcap B) = P(A)P(B)$.

According to the Bayes' rule, if $A_1, A_2, ..., A_n$ are *n* mutually exclusive events, the union of which is the sample set *H*, $H = A_1 \cup A_2 \cup ... \cup A_n$, then for every event *A*: $P(A_k|A) = P(A_k \cap A)/P(A)$, where $P(A_k \cap A) = P(A_k)P(A|A_k)$, $k = 1, 2, ..., n$; since $P(A|A_k) = P(A_k|A)/P(A_k)$, the total probability of *A* can be computed as: $P(A) = P(A|A_1)P(A_1) + ... + P(A|A_n)P(A_n)$.

An alternative definition of the probability may come from practice. If an experiment is repeated *N* times under the same conditions, and if the event *A* appear N_A times, the probability of the event *A* may be defined as a relative frequency: $P(A) = \frac{N_A}{N}$, when the number of experiments is sufficiently large. According to the distribution of large numbers this empiric definition becomes meaningfully from the mathematical point of view when $N \to \infty$.

II.2.2. Random Variables - Definition

The random variables are a representation of the sample set *H* on the real axis R. Supposing (*H, **A**, P*) is a probabilistic space, a random variable is defined as a real finite measurable function $X : H \to R$ with the following condition: $\forall x \in R$, $\{\eta \in H | X(\eta) \le x\} \in \boldsymbol{A}$.

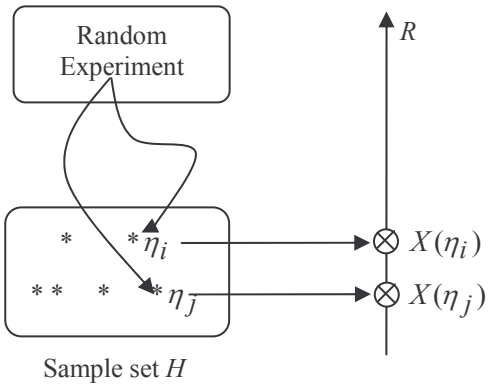


Figure II.2. A random variable $X(\eta)$ assigns to every experimental outcome a real number [Gav 03b].

II.2.3. Probability Density Function and Moments

II.2.3.1 Probability Density Function

Be there *X* a random variable and be *x* a real number. The probability of event $X(\eta) \le x$ [Bra 99] is a function of *x* and is called the *cumulative distribution function* (*cdf*) of *X*:

$$F_X(x) = P(X(\eta) \le x) . \tag{II.1}$$

The first derivative of a *cdf* is called *probability density function* (*pdf*) of *X*:

$$f_X(x) = F'_X(x) = \frac{dF_X(x)}{dx}. \tag{II.2}$$

The inverse relationship between the *cdf* and the *pfs* is:

$$F_X(x) = \int_{-\infty}^{x} f_X(u)du. \tag{II.3}$$

If X can take only a finite number of discrete values it is called discrete random variable, and its *cdf* (see Figure II.3.a) is a step function. The random variable X is called a continuous random variable if the set of its value is continuous; hence, its *cdf* may be represented in Figure II.3.b.

If the *cdf* is continuous and differentiable, than the *pdf* is a function in the classical sense (*cf.* Figure II.3.b). If not (*i.e.* the *cdf* has jumps), than the *pdf* should be considered in the distribution sense (a sum of functions and δ *Dirac* distributions, Figure II.3.a).

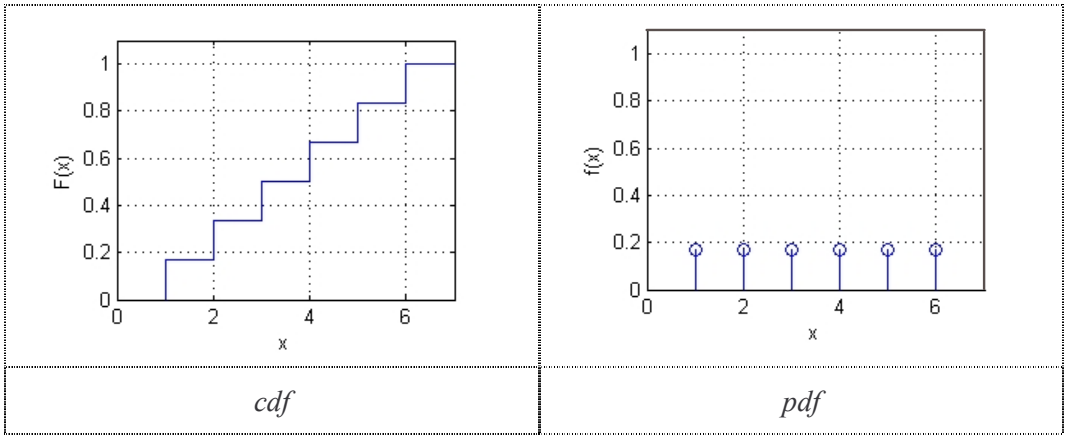


Figure II.3.a. Example of graphical representations for the *cdf* and *pdf* of a discrete random variable.

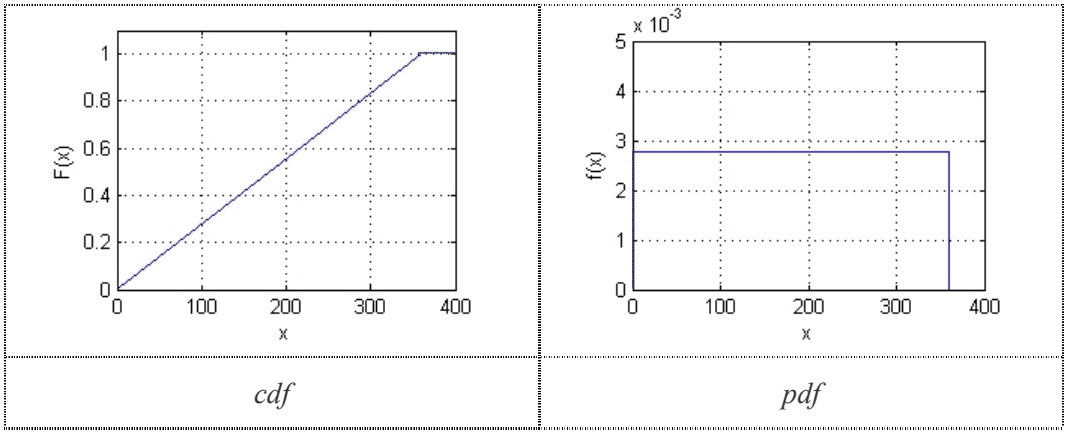


Figure II.3.b. Example of graphical representations for the *cdf* and *pdf* of a continuous random variable.

Any *cdf* hs the following properties:

1. A *cdf* is always monotonic and non-decreasing;

2. $\lim_{x\to\infty} F(x) = \lim_{x\to\infty} P(X \le x) = P(H) = 1$;

3. $\lim_{x\to-\infty} F(x) = \lim_{x\to-\infty} P(X \le x) = 1 - \lim_{x\to-\infty} P(X \ge x) = 0$.

4. In the continuous case (which will be further consider in the thesis) the following computing relations are true:

$$P(x \le a) = \int_{-\infty}^{a} f_X(x)dx = F_X(a),$$

$$P(a < x \le b) = \int_{a}^{b} f_X(x)dx = F_X(b) - F_X(a),$$

$$f_X(x) \ge 0,\ \forall x \in R,\ \int_{-\infty}^{\infty} f_X(x)dx = 1.$$

The probability distributions which are relevant for the thesis are detailed in Appendix 0.

II.2.3.2 Moments

A very important concept in the theory of probability and statistics is the *mathematical expectation*, or *expected value*, or *mean value* of a random variable *X*. The mean value of a random variable is denoted by m_1, $E[X]$ or by $\overline{X}$. If *X* is a continuous random variable with the *pdf* $f_X(x)$, the mean value is define by:

$$m_1 = E[X] = \overline{X} = \int_{-\infty}^{\infty} x f_X(x)dx. \tag{II.4}$$

Note that if *c* is a constant, then $E[cX] = cE[X]$.

The *n*-th moment of the random variable *X* is defined as $m_n = E[X^n] = \int_{-\infty}^{\infty} x^n f_X(x)dx$. For a particular value $n = 2$, the second moment of *X* or *mean-square value* is obtained:

$$m_2 = E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)dx. \tag{II.5}$$

The *n*-th central moment of the mean is defined as: $E[(X - E[X])^n] = \int_{-\infty}^{\infty} (x - \overline{X})^n f_X(x)dx$ for a particular value, $n = 2$ the *variance* is defined as:

$$\sigma_X^2 = E[(X - E[X])^2] = E[X^2] - (E[X])^2 = \int_{-\infty}^{\infty} (x - \overline{X})^2 f_X(x)dx. \tag{II.6}$$

The *standard deviation* is denoted by σ_X and is always positive.

II.2.4. Two and Multi – Dimensional Random Variables

If X and Y are two continuous random variables defined on the same probabilistic space, the *joint cumulative density function* (*jcdf*) is:

$$F_{XY}(x,y) = P(X \le x, Y \le y), \tag{II.7}$$

The *joint probability distribution function* (*jpdf*) of X and Y is defined by the following equation:

$$f_{XY}(x,y) = \frac{\partial^2 F_{XY}(x,y)}{\partial x \partial y}. \tag{II.8}$$

The probability that X and Y jointly lies between x_1 and x_2 and y_1 and y_2, respectively is given by:

$$P(x_1 < X < x_2, y_1 < Y < y_2) = \int_{x_1}^{x_2}\int_{y_1}^{y_2} f_{XY}(x,y)dxdy. \tag{II.9}$$

The *jcdf* and *jpdf* have the following properties:

1. $0 \le F_{XY}(x,y) \le 1$;
2. $F_{XY}(\infty,\infty) = 1$, $F_{XY}(-\infty,-\infty) = F_{XY}(x,-\infty) = F_{XY}(-\infty,y) = 0$;
3. $F_{XY}(x,\infty) = F_X(x)$, $F_{XY}(\infty,y) = F_Y(y)$;
4. $f_{XY}(x,y) \ge 0$; $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)dxdy = 1$;
5. $F_X(x) = \int_{-\infty}^{x}\int_{-\infty}^{\infty} f_{XY}(u,v)dudv$; $F_Y(y) = \int_{-\infty}^{y}\int_{-\infty}^{\infty} f_{XY}(u,v)dvdu$;
6. $f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x,y)dy$; $f_Y(y) = \int_{-\infty}^{\infty} f_{XY}(x,y)dx$.

If the random variables are independent, their *jcdf* is $F_{XY}(x,y) = F_X(x)F_Y(y)$ and the related *jpdf* is $f_{XY}(x,y) = f_X(x)f_Y(y)$.

These concepts of joint distributions can be extended from two random variables (X, Y) to n random variables ($X_1, X_2, ..., X_n$):

$$F_{X_1,X_2,...,X_n}(x_1,x_2,...,x_n) = P(X_1 \le x_1, X_2 \le x_2, ..., X_n \le x_n), \tag{II.10}$$

$$f_{X_1,X_2,...,X_n}(x_1,x_2,...,x_n) = \frac{\partial^n F_{X_1,X_2,...,X_n}(x_1,x_2,...,x_n)}{\partial x_1 \partial x_2 ... \partial x_n}, \tag{II.11}$$

The *joint moment* of k and l order between two random variables X and Y is defined by:

$$m_{kl} = E[X^k Y^l] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^k y^l f_{XY}(x,y)dxdy \,. \tag{II.12}$$

The correlation $Cor(XY)$ between X and Y is a particular case of the *joint moments* obtained for $k = l = 1$:

$$Cor(XY) = m_{11} = E[XY] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_{XY}(x,y)dxdy \,. \tag{II.13}$$

Of course, if $k = 1$ and $l = 0$ the mean value of a one-dimensional random variable is obtained: $m_{10} = E[X] = m_X$ or $m_{01} = E[Y] = m_Y$.

The *joint central moment* between two random variables X and Y can be defined through analogy as:

$$\mu_{kl} = E[(X - m_X)^k (Y - m_Y)^l] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - m_X)^k (y - m_Y)^l f_{XY}(x,y)dxdy \,. \tag{II.14}$$

If $k = 1$, $l = 1$ the *covariance* of X and Y ($Cov_{XY} = \mu_{11}$) is obtained and if $k = 2, l = 0$ or $k = 0, l = 2$ the variance for X or Y is found:

$$\mu_{20} = E[(X - m_X)^2] = \sigma_X^2 \,, \quad \mu_{02} = E[(Y - m_Y)^2] = \sigma_Y^2 \tag{II.15}$$

The *correlation coefficients* between X and Y is defined as:

$$\rho_{XY} = \frac{E[(X - m_X)\ (Y - m_Y)\ ]}{\sigma_X \sigma_Y} = \frac{Cov_{XY}}{\sigma_X \sigma_Y} \,, \quad -1 \le \rho_{XY} \le 1 \,. \tag{II.16}$$

1. If $\rho_{XY} = 0$, then X and Y are uncorrelated;
2. If X and Y are linearly dependent then, $X = aY$, then $\rho_{XY} = 1,\ a > 0$ or $\rho_{XY} = -1,\ a < 0$.

II.3. Random Processes

II.3.1. Definitions

The random variables were basically defined in Section II.1 as a mapping of the elements of the sample set H into points of the real axis. In the case of random processes, the sample space would be mapped into a family of time functions.

Associating a time function to each element of the *H* results in a family of time functions called the *ensemble* (the ensemble is the set of sample functions with the associated probabilities). Note that the random process is devoted by $X(\eta;t)$, where η is an elementary sample of the *H*, see Figure II.4. The notation $X(t)$ is sometime used for simplicity.

There are different types of random processes, according to the characteristics of *t* and of the random variable $X(t) = X$ at time *t*:

1. *Continuous-state and continuous-time*: for this case both $X(t)$ and *t* have a continuous values; $X(t)$ is said to be a *continuous random process*.
2. *Discrete-state and continuous-time*: $X(t)$ assumes a discrete set of values and *t* is continuous; such a random process is referred to as a *discrete random process*.
3. *Continuous-state and discrete-time*: $X(t)$ assumes a continuum of values and *t* is a discrete set of values; such a process is called a *continuous random sequence*.
4. *Discrete-state and discrete-time*: both $X(t)$ and *t* assume a discrete set of values; such a random process is called a *discrete random sequence*.

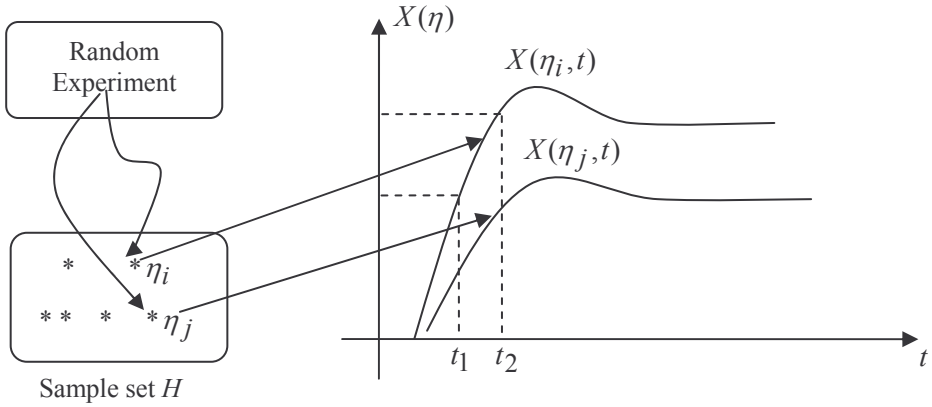


Figure II.4. Mapping of a sample space into sample functions [Gav 03b].

II.3.2. Statistic Description

If the time *t* is fixed, the random process is transformed in a random variable, and it is possible to define the corresponding *cdf* and *pdf*.

$$F_X(x;t) = P[X(\eta,t) \le x], \tag{II.17}$$

$$f_X(x;t) = \frac{dF_X(x;t)}{dx}. \tag{II.18}$$

The entities in eq. (II.17) and eq. (II.18) give the first order statistical description of the random process.

The second order distribution function of a random process is the *joint distribution* (*jcdf* and *jpdf*) of the two random variables $X(t_1)$ and $X(t_2)$ obtained by sampling the process at two time moments t_1 and t_2:

$$F_{X_1,X_2}(x_1,x_2;t_1,t_2) = P[X(t_1) \le x_1, X(t_2) \le x_2], \tag{II.19}$$

$$f_{X_1,X_2}(x_1,x_2;t_1,t_2) = \frac{\partial^2 F_{X_1,X_2}(x_1,x_2;t_1,t_2)}{\partial x_1 \partial x_2}. \tag{II.20}$$

In the same way can be defined the distribution of the n^{th} order:

$$F_{X_1,X_2,..,X_n}(x_1,x_2,...,x_n;t_1,t_2,...,t_n) = P[X(t_1) \le x_1, X(t_2) \le x_2,..., X(t_n) \le x_n], \tag{II.21}$$

$$f_{X_1,X_2,..,X_n}(x_1,x_2,...,x_n;t_1,t_2,...,t_n) = \frac{\partial^n F_{X_1,X_2,..,X_n}(x_1,x_2,...,x_n;t_1,t_2,...,t_n)}{\partial x_1 \partial x_2 ... \partial x_n}. \tag{II.22}$$

Two random processes $X(\eta;t)$ and $Y(\eta;t)$ defined on the same probabilistic space are independent if for any moments $[t_{11},...,t_{1n}]$ and $[t_{21},...,t_{2m}]$, the random variables $X(\eta;t_{11}),...,X(\eta;t_{1n})$ are statistically independent with respect to the random variables $Y(\eta;t_{21}),...,Y(\eta;t_{2m})$.

$$\begin{aligned} F_{XY}(x,y;t_1,t_2) &= F_X(x;t_1)F_Y(y;t_2), \\ f_{XY}(x,y;t_1,t_2) &= f_X(x;t_1)f_Y(y;t_2). \end{aligned} \tag{II.23}$$

The above equalities should hold $\forall x = [x_1,...,x_n]$ and $y = [y_1,...,y_m]$ arbitrarily sampled vector from R^n and R^n respectively, and $\forall t_1 = [t_{11},...,t_{1n}]$ and $\forall t_2 = [t_{21},...,t_{2n}]$ arbitrarily chosen time moments.

II.3.3. Average Values

For any random process two types of average values can be defined: statistical mean values defined for all time moments, and temporal mean values defined for each particular sample (Figure II.5).

The **statistical mean** is actually the mean of the random variables considered at some moment of time, thus becoming a functions of time. In many problems, only the first and second-order statistics are considered in order to characterize a random process. These values are defined in the same way as the mean of random variables:

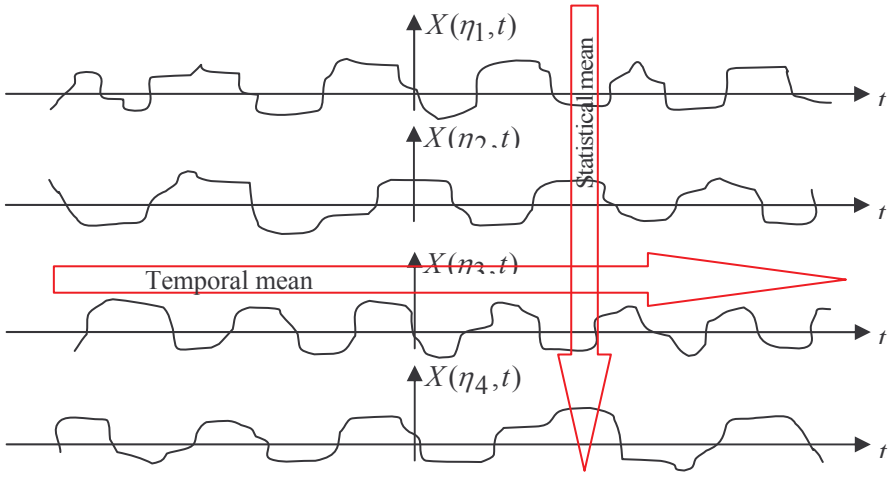


Figure II.5. Statistical and temporal mean values of a random process.

The *autocorrelation function* is defined by eq. (II.27):

$$B_{XX}(t_1,t_2)=E[X(t_1)X(t_2)]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x_1x_2f_{X_1X_2}(x_1,x_2;t_1,t_2)dx_1dx_2\,, \tag{II.27}$$

The *cross-correlation function* of two random processes $X(t)$ and $Y(t)$ is defined as:

$$B_{XY}(t_1,t_2)=E[X(t_1)Y(t_2)]\,. \tag{II.28}$$

The most relevant properties for *autocorrelation function* are:

1. $B_{XX}(t_2,t_1)=B_{XX}(t_1,t_2)$;
2. $B_{XX}(t_1,t_1)=m_X{}^{(2)}\geq 0$;
3. $\left|B_{XX}(t_1,t_2)\right|\leq\sqrt{B_{XX}(t_1,t_1)B_{XX}(t_2,t_2)}$.

The **temporal means** are computed on a sample $X(\eta;t)$, thus depending on the particular η from the sample space.

- The *mean value* represents the continuous component of the $X(\eta;t)$ and it is itself a random variable because it depends on the η; but it does not depend on the origin of time t_0.

$$m_{x\eta}\overset{not}{=}\overline{\overline{X_\eta(t)}}\overset{\Delta}{=}\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}X(\eta;t)dt=\overline{\overline{X_\eta(t_0+t)}}\,, \tag{II.29}$$

- The *mean-square value* is defined as:

$$\overline{\overline{[X_\eta(t)]^2}}=\lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}[X(\eta;t)]^2dt=\overline{\overline{[X_\eta(t_0+t)]^2}}\,, \tag{II.30}$$

and is independent with respect to the time origin.

The *temporal autocorrelation function* is defined as:

$$R_{XX,\eta}(t_1,t_2) = \overline{\overline{X_\eta(t_1+t)X_\eta(t_2+t)}} = \lim_{T\to\infty}\int_{-T/2}^{T/2} X_\eta(\eta;t_1+t)X_\eta(\eta;t_2+t)dt = R_{XX,\eta}(t_2-t_1)\,. \quad \text{(II.31)}$$

The *temporal cross-correlation function* is defined as:

$$R_{XY,\eta}(t_1,t_2) = \overline{\overline{X_\eta(t_1+t)Y_\eta(t_2+t)}} = \lim_{T\to\infty}\int_{-T/2}^{T/2} X_\eta(\eta;t_1+t)Y_\eta(\eta;t_2+t)dt = R_{XY,\eta}(t_2-t_1)\,. \quad \text{(II.32)}$$

It can be demonstrated that: $R_{XY,\eta}(\tau) \neq R_{XY,\eta}(-\tau)$ and $R_{XY,\eta}(\tau) = R_{YX,\eta}(-\tau)$; $\tau = t_2 - t_1$.

II.3.4. Stationarity and Ergodicity

A random signal is stationary if its *n*-th order statistical descriptions are invariant with respect to the origin of time. Two signals are simultaneously stationary if everyone is stationary and if their statistical properties are simultaneously invariable of any translation of time.

Stationarity may be strictly or wide-sense.

- *Strict sense stationarity*: A random process $X(\eta;t)$ is stationary in the strict sense (or hard) if the *cdf* is invariant to any translation of time τ.

$$F_n(x_1,x_2,...,x_n;t_1,t_2,...,t_n) = F_n(x_1,x_2,...,x_n;t_1+\tau,t_2+\tau,...,t_n+\tau)\,,\ \forall\,\tau,n\in N^*\,. \quad \text{(II.33)}$$

- *Wide-sense stationarity*: A random process $X(\eta;t)$ is stationary in the wide-sense (or weak) if the following equations are satisfied.

$$\begin{aligned} m_X^{(1)}(t) &= m_X^{(1)} = ct\,, \\ B_{XX}(t_1,t_2) &= B_{XX}(\tau)\,,\ \tau = t_2 - t_1\,. \end{aligned} \quad \text{(II.34)}$$

A random process $X(\eta;t)$ is *ergodic* if all of its statistics can be determined, with probability one, from a unique sample function of the process. Consequently, the statistical averages also equal the corresponding time averages with probability one. In many cases, a weaker form of ergodicity is investigate, namely the mean and autocorrelation function ergodicity.

A random process $X(\eta;t)$ is *ergodic in the mean* if the time averaged mean value is equal to the statistical mean value:

$$\lim_{T\to\infty} P\left\{\left|E(X_\eta(t)) - \overline{\overline{[X_\eta(t)]}}\right| < \varepsilon\right\} = 1\,,\ \forall\ \varepsilon > 0\,. \quad \text{(II.35)}$$

A $X(\eta;t)$ is *ergodic in the autocorrelation* if equation (II.36) is true, the equality being defined in the sense of probability convergence:

$$\lim_{T\to\infty} P\left\{\left|E(X_\eta(t+\tau)X_\eta(t)) - \overline{\overline{X_\eta(t+\tau)X_\eta(t)}}\right| < \varepsilon\right\} = 1,\ \forall\ \varepsilon > 0. \tag{II.36}$$

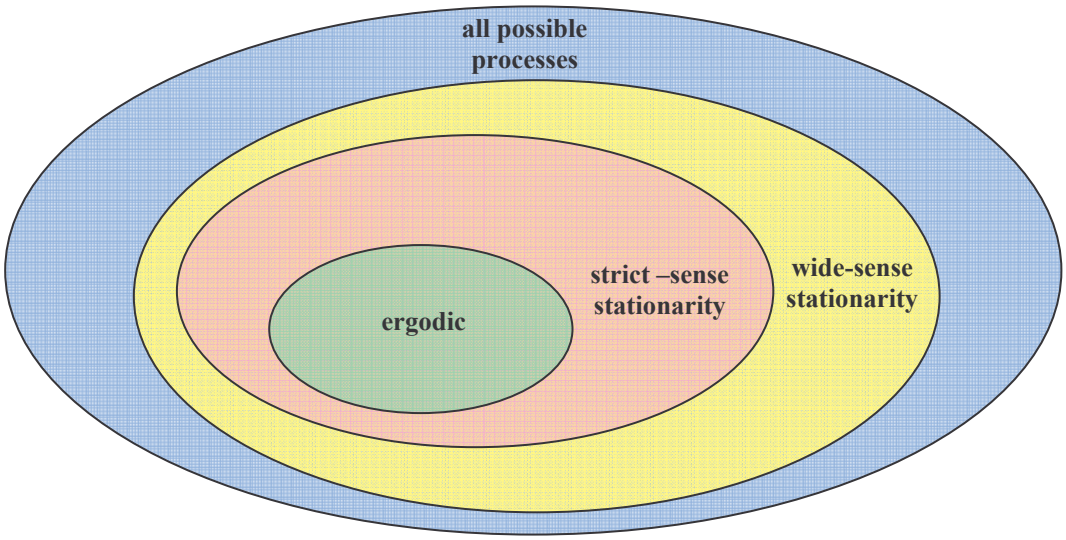


Figure II.6. Set of different classes of processes.

Note that for the next Sections and Chapters the *pdf* will be denoted by $p_X(x)$ or $p(x)$ instead of $f_X(x)$.

II.4. Information Theory: Models and Description

Fundamented as a science by the Shannon's works, information theory quickly established strong and fruitfully synergies with a large variety of application, Figure II.7.

In the present thesis, the fundamental issues of information theory will be approached from the watermarking point of view. In this respect a special emphasis will be put on the noise modelling and capacity evaluation. Some illustrations concerning the entropy are also presented.

II.4.1. Information Source

Intuitively, information source is a mechanism to choose in an unpredictable way among all possible messages a particular one to be sent to a receiver. The particular samples of the source are called symbols.

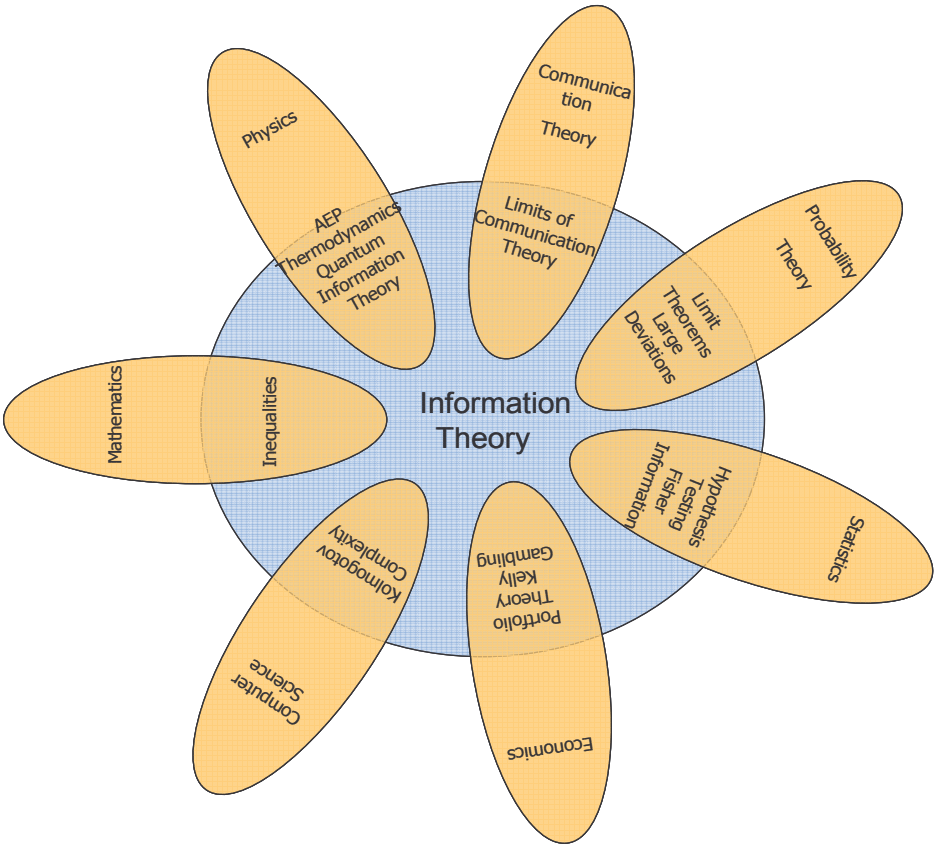


Figure II.7. Mutual synergies between information theory and other scientific fields [Cov 06].

From the theoretical point of view, an information source is a random process defined on a probabilistic space (*H, **A**, P*).

A *discrete information source* sends messages at a discrete time moments and each of these messages is one of a finite set of symbols.

The *symbol* is defined as an irreducible element which contains information.

The *alphabet* represents the totality of all the symbols and the finite succession of the symbols that represent a message is called *word*; the *language* is the totality of all the words formed with an alphabet.

A *discrete memoryless information source* is a source for which the probability of apparition of a symbol does not depend on the previous symbols.

An *ergodic information source* is a source for which the underlying random process is ergodic.

For an *ergodic and memoryless discrete information source* the value of the information for a symbol (in bit) is defined as:

$$i(x_i) = -\log_2 p(x_i), \tag{II.37}$$

where the alphabet is $[X] = [x_1, ..., x_n]$ and the probabilities $[P] = [p(x_1), ..., p(x_n)]$.

The *entropy* (in bits/symbol) of an ergodic and memoryless information source X having the probabilities $p(x_i),\ i=\overline{1,...,n}$ is expressed as:

$$H(X)=\sum_{i=1}^{n} i(x_i)p(x_i)=-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)\,. \tag{II.38}$$

The entropy of an *ergodic memoryless continuous information source* is defined as:

$$H(X)=-\int_{-\infty}^{\infty} p_X(x)\log_2 p_X(x)dx\,, \tag{II.39}$$

where $p_X(x)$ is the *pdf* of the source X.

In the continuous case, the entropy properties are:

1. If X is limited to a certain volume in its space and the *pdf*s are constant in the volume ($p_X(x)=1/v$), then the entropy of X is maximum and equal with $H(x)=\log_2 v$.
2. The maximum entropy among all distributions of the given variance σ^2 is provided by a Gaussian distribution.
3. The entropy of a one-dimensional Gaussian distribution of σ standard deviation is $H(X)=\log_2\sqrt{2\pi e\sigma}$.
4. If x is limited to a half line $p(x)=0,\ x\le 0$, and the first moment of x is fixed at $a=\int_0^{\infty} x\,p(x)\,dx$, the maximum entropy is $H=\log_2 ea$, when $p(x)=\frac{1}{a}e^{-(x/a)}$.

II.4.2. Noisy Channel

The noisy channel concept was already briefly presented (see Figure II.1 and II.8): during its transmission through the channel, the information coming from the X input source is corrupted by an N noise source.

Hence, there are two statistical processes at work (Figure II.8): the input source (denoted by X) and the noise source (denoted by N). These two processes are combined in order to compute the Y output information source.

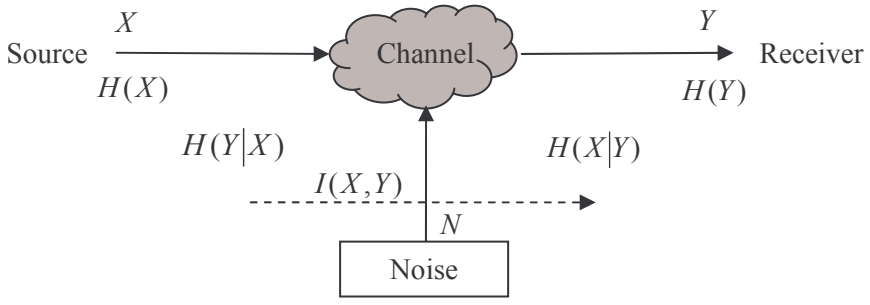


Figure II.8. Schematic diagram of the communication system.

The joint entropy of the input and output is:

$$H(X,Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XY}(x,y)\log_2 p_{XY}(x,y)dxdy\,, \tag{II.40}$$

where $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XY}(x,y)dxdy = 1$, $p_{XY}(x,y) = p_X(x)\cdot p_{Y|X}(y|x) = p_Y(y)\cdot p_{X|Y}(x|y)$ and

$p_X(x) = \int_{-\infty}^{\infty} p_{XY}(x,y)dy$, $p_Y(y) = \int_{-\infty}^{\infty} p_{XY}(x,y)dx$.

The average error is a measure of the incertitude of the output Y when the input X is known:

$$H(Y/X) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XY}(x,y)\log_2 \frac{p_{XY}(x,y)}{p_X(x)}dxdy\,. \tag{II.41}$$

Equivocation is a measure of the ambiguity of the input X when the output Y is known:

$$H(X/Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XY}(x,y)\log_2 \frac{p_{XY}(x,y)}{p_Y(y)}dxdy\,. \tag{II.42}$$

If X and Y are independent $p_{XY}(x,y) = p_X(x)p_Y(y) \Rightarrow H(X,Y) = H(X) + H(Y)$, else $H(X,Y) = H(X) + H(Y|X)$ or $H(X,Y) = H(Y) + H(X|Y)$.

The relative entropy between two *pdf*s, $p(x)$ and $q(x)$ is defined as:

$$D(p(x),q(x)) = \int_{-\infty}^{\infty} p(x)\log_2 \frac{p(x)}{q(x)}dx\,, \tag{II.43}$$

The $D(p(x),q(x)) = 0$ if the two *pdf*s are identically.

In the presence of noise, it is not possible to reconstruct the original message with certainty by any operation on the received signal. The proper correction to be applied to the amount of information transmitted is the amount of the information which is missing in the received signal, *i.e.* the uncertainly introduced by the noise.

A stationary noisy channel is described by the conditional *pdf* $p_{Y|X}(y|x)$ which is time-invariant (*i.e.* that expresses the probability of observing the output y given that x was sent). This function affords the computation of the information sent through the channel which is the mutual information, denoted by $I(X,Y)$:

$$I(X,Y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XY}(x,y)\log_2\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}dxdy\,, \tag{II.44}$$

where $p_{XY}(x,y)=p_X(x)\cdot p_{Y|X}(y|x)$.

Alternatively, it can be proved that

$$I(X,Y)=H(X)-H(X|Y)=H(Y)-H(Y|X)=H(X)+H(Y)-H(X,Y)\,. \tag{II.45}$$

The relationship between input, output and conditional entropies can be expressed by a Venn diagram (Figure II.9).

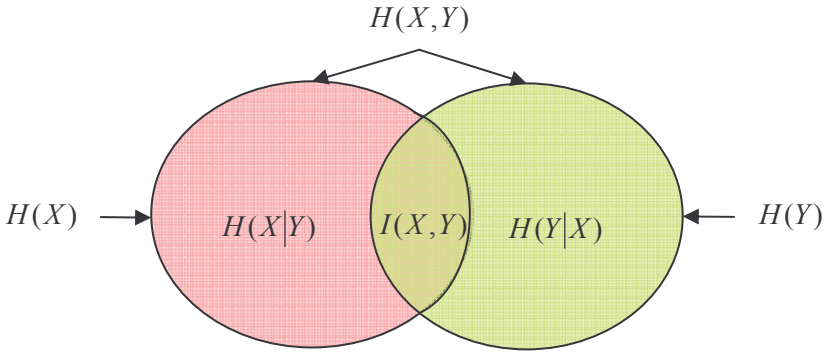


Figure II.9. Relationship between entropy and mutual information.

The continuous channel capacity is the maximum value of the mutual information over all possible probability distributions of the input:

$$C=\max_{p_X(x)} I(X,Y)=\max_{p_X(x)}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{XY}(x,y)\log_2\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}dxdy= \tag{II.46.a}$$

$$=\max_{p_X(x)}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_X(x)p_{Y|X}(y|x)\log_2\frac{p_{Y|X}(y|x)}{\int_{-\infty}^{\infty}p_X(x)p_{Y|X}(y|x)dx}dxdy\,. \tag{II.46.b}$$

As it can be seen, each entity involved in channel description requires an accurate knowledge of the noise *pdf*. Practical tools allowing an experimenter to estimate the noise *pdf* are further presented.

II.5. Basic Tools in Probability Estimation

Nowadays different methods aimed at estimating *pdf*s coexist, and, as each method has its good and bad points, there is no consensus in selecting the "best" one [Ege 95], [Roe 97], [Sco 80], [Weg 72].

The first approach is parametric density estimation. Unfortunately, in most of the cases, some *a-priori* choice about the *pdf* model is not adequate because it may result in a false representation of the true *pdf* [Arc 06].

The second approach relies on nonparametric *pdf* estimators [Sco 80] (*e.g.* the *Parzen window* estimator [Par 62]). An overview of nonparametric density estimation techniques is presented in [Weg 72], [Ege 95].

Let us exemplify by the *Parzen window* estimator [Arc 06]. Consider a continuous random variable X and by $p(x)$ its *pdf*; denote by $\{x_k\}_{k=1}^{K}$ the values obtained when sampling X. The $\hat{p}(x)$ estimation is computed by using a kernel function on each data point and by determining a common width σ (the smoothing parameter). In practice the most used kernel is the Gaussian kernel:

$$N(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2}), \tag{II.47}$$

where μ and σ are the kernel centre and the kernel width respectively. After that the estimated *pdf* is defined as:

$$\hat{p}(x) = \frac{1}{K}\sum_{k=1}^{K} N(x|x_k,\sigma). \tag{II.48}$$

In order to estimate the optimal value of the kernel width, two methods are proposed in the literature: *Silverman's plug-in principle* and *leave-one-out estimator* [Arc 06].

The third approach in *pdf* estimation is based on the semi-parametric models [Roe 97], where no assumption is made about the form of the *pdf*. The complexity is fixed in advance in order to avoid the increasing of the number of the parameters with the size of the data.

The finite Gaussian mixtures are used for this purpose. A commonly technique for approximating the MLE (Maximum Likelihood Estimate) of *pdf* is the EM algorithm (Expectation–Maximization) proposed for the first time by [Dem 77].

As the main contribution of the thesis is found on such a method, it will be further described in detail.

II.5.1. Finite Gaussian Mixtures

The Expectation – Maximization algorithm is an approach intensively considered in order to iteratively compute the maximum likelihood estimates. At each iteration of the EM algorithm, there are two steps: the *expectation step* or *E-step* and the *maximization step* or *M-step* [Dem 77].

After many years, the EM algorithm becomes a standard tool in the statistical field. The interest in this Section is a parametric family of finite mixture densities, *i.e.* the family of *pdf*s defines by [Red 84]:

$$p(x|\Phi) = \sum_{k=1}^{K} P(k)\, p_k(x|\phi_k)\,,\ x = (x_1, x_2, ..., x_n)^T \in R^n, \tag{II.49}$$

where the weights $P(k)$ are nonnegative and $\sum_k P(k) = 1$, and $p_k(x|\phi_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp(-\frac{(x-\mu_k)^2}{2\sigma_k^2})$ are Gaussian probability density functions. The parameters necessary to be estimated are $\Phi = (P(1), P(2), ..., P(K), \phi_1, \phi_2, ..., \phi_K) \overset{not}{=} (P(k), \phi_k)$, where $\phi_k = (\mu_k, \sigma_k)^T$ the mean μ_k and the variances σ_k, $k = 1, ..., K$.

Finite mixture densities can be interpreted as densities associated with a statistical population which is a mixture of *K* component populations with associated components densities $p_k(x/\phi_k)$ and mixing weights $P(k)$. Such densities appear as fundamental models in domains as statistical pattern recognition, classification, and clustering [Mcl 97].

II.5.1.1. Maximum Likelihood Estimation

The goal of this Section is to define a procedure for computing maximum likelihood estimates for the parameters for the Gaussian mixture densities.

In general terms, the maximum likelihood estimate (MLE) of a parameter which determines a *pdf* is a choice of the parameter which maximizer the introduced *pdf* of a given sample of observation, called in this context the *likelihood function* [Red 84].

Be there a family of mixture densities, eq. (II.49) and a particular parameter Φ^* is the *true* parameter value to be estimated and $S = \{x_i\}_{i=1,...,N}$ an independent sample of *N* observation on the mixture, *i.e.* a set of *N* observations which are *i.i.d.* (*i*ndependent and *i*dentically *d*istributed) random variables with the $p_i(x|\Phi^*)$ *pdf* [Red 84].

The likelihood function of a sample is the probability density function of the random sample. Often in literature [Bil 98] for likelihood function the *log likelihood function* is used:

$$L(\Phi) = \sum_{i=1}^{N} \log p(x_i|\Phi)\,. \tag{II.50}$$

Supposed [Red 84] that each component density $p_k(x|\phi_k)$ is differentiable with respect to ϕ_k and the parameters ϕ_k are mutually independent variables.

The classical way to determine a maximum likelihood estimate is to arrive at a system of *likelihood equations* satisfied by the MLE and then to try to solve these equations. The likelihood equations are found by considering the partial derivates of the $L(\Phi)$ with respect to Φ:

$$\nabla_{\phi_k} L(\hat{\Phi}) = 0\,,\ k = 1,...,K\,, \tag{II.51}$$

where $\hat{\Phi} = (\hat{P}(k),\ \hat{\phi}_k)$ and ∇_{ϕ_k} is the gradient of the partial derivatives with respect to the ϕ_k;

and

$$0 \geq \nabla_P L(\hat{\Phi})^T (P - \hat{P})\,, \tag{II.52}$$

where $\hat{P} = (\hat{P}(1), \hat{P}(2),..., \hat{P}(K))^T$ is the likelihood equation, and $P = (P(1), P(2),..., P(K))^T$.

Finally [Red 84], the likelihood equations for the weights are:

$$\hat{P}(k) = \frac{1}{N} \sum_{i=1}^{N} \frac{\hat{P}(k)\, p_k(x_i|\hat{\phi}_k)}{p(x_i|\hat{\Phi})}\,,\ k = 1,...,K\,. \tag{II.53}$$

The equations (II.51) and (II.53) together constitute a full set of likelihood equations for the maximum likelihood estimate.

II.5.1.2. Expectation Maximization algorithm

In this Section, our objective is to present how the parameters of a mixture density ($\Phi = (P(k), \phi_k)$) are estimated by using EM algorithm. At the end of the section we'll have a set of equations that will allow us to apply the EM algorithm in order to estimate the parameters of the Gaussian mixtures distribution (weights, means, and variances).

Considering the *pdf*s $f(y|\Phi)$ and $g(x|\Phi)$; for a given real value $x \in R$ and $y \to x(y) = x$, the EM algorithm needs to maximize the log likelihood function $L(\Phi) = \log g(x|\Phi)$ by evaluating the relationship between the $f(y|\Phi)$ and $g(x|\Phi)$ [Dem 77].

The conditional density, $l(y|x,\Phi)$ is given by $f(y|\Phi) = l(y|x,\Phi)g(x|\Phi)$. For two sets of parameters (Φ and Φ'), the log likelihood function is: $L(\Phi) = Q(\Phi|\Phi') - H(\Phi|\Phi')$, where $Q(\Phi|\Phi') = E(\log f(y|\Phi)|x,\Phi')$ and $H(\Phi|\Phi') = E(\log l(y|x,\Phi)|x,\Phi')$. Note that the $E(\Phi)$ denotes the expectation using parameter Φ.

The EM algorithm can be written as [Dem 77]:

Given a current approximation Φ^c *of a maximizer of* $L(\Phi)$*, obtain a next approximation* Φ^+ *as:*

E-step: *Determine* $Q(\Phi|\ \Phi^c)$. (II.54)

$$\textit{M-step:} \qquad \textit{Choose } \Phi^{+} \in \arg\max_{\Phi} Q(\Phi|\Phi^{C}). \tag{II.55}$$

where $\arg\max_{\Phi} Q(\Phi|\Phi^{C})$ denotes the set of values Φ which maximize $Q(\Phi|\Phi^{C})$.

In order to calculate $Q(\Phi|\Phi^{'})$ for the *E-step*, eq. (II.56), the following two *pdf*s are introduced $f(y|\Phi)=\prod_{i=1}^{N} P(k_i)p_{k_i}(x_i|\phi_{k_i})$, and $g(x|\Phi)=\prod_{i=1}^{N} p(x_i|\Phi)$, alongside with the conditional density $l(y|x,\Phi^{'})=\prod_{i=1}^{N}\frac{P^{'}(k_i)p_{k_i}(x_i|\phi^{'}_{k_i})}{p(x_i|\Phi^{'})}$ [Red 84].

$$\begin{aligned} Q(\Phi|\Phi^{'}) &= \sum_{k_1=1}^{K}\cdots\sum_{k_N=1}^{K}\sum_{i=1}^{N}\log P(k_i)p_{k_i}(x_i|\phi_{k_i})\prod_{i=1}^{N}\frac{P^{'}(k_i)p_{k_i}(x_i|\phi'_{k_i})}{p(x_i|\Phi')}=\ldots= \\ &= \sum_{k=1}^{K}\Big[\sum_{i=1}^{N}\frac{P'(k)p_k(x_i|\phi'_k)}{p(x_i|\Phi')}\Big]\log P(k)+\sum_{k=1}^{K}\sum_{i=1}^{N}\log p_k(x_i|\phi_k)\frac{P'(k)p_k(x_i|\phi'_k)}{p(x_i|\Phi')}. \end{aligned} \tag{II.56}$$

where $\Phi=(P(k),\phi_k)$, $\Phi^{'}=(P^{'}(k),\phi'_k)$; $x=[x_1,\ldots,x_N]$ and $y=[y_1,\ldots,y_N]$.

The next step (*M-step*) is to maximize the equation which was determined in the previous step (*E-step*). The maximization is separated in two parts: in the first part in relation with the weights $P(1),P(2),\ldots,P(K)$ and in the second part in relation with the rest of the parameters witch are $\phi_1,\phi_2,\ldots,\phi_K$.

For $S=\{x_i\}_{i=1,\ldots,N}$ and $\phi_1,\phi_2,\ldots,\phi_K$ mutually independent variables, if $\Phi^{C}=(P^{C}(k),\phi_k^{C})$, $k=1,\ldots,K$ is a maximizer of $L(\Phi)$, then in order to verify that the next approximate maximizer $\Phi^{+}=(P^{+}(k),\phi_k^{+})$, $k=1,\ldots,K$ prescribed by the *M-step* satisfies:

$$P^{+}(k)=\frac{1}{N}\sum_{i=1}^{N}\frac{P^{C}(k)p_k(x_i|\phi_k^{C})}{p(x_i|\Phi^{C})},\ k=1,\ldots,K, \tag{II.57}$$

$$\phi_k^{+}\in\arg\max_{\Phi}\sum_{i=1}^{N}\log p_k(x_i|\phi_k)\frac{P^{C}(k)p_k(x_i|\phi_k^{C})}{p(x_i|\Phi^{C})},\ k=1,\ldots,K. \tag{II.58}$$

Note that each weight $\frac{P^{C}(k)p_k(x_i|\phi_k^{C})}{p(x_i|\Phi^{C})}$ is the *posteriori* probability that x_i originated in the k^{th} component, given the current maximum likelihood estimate Φ^{C}.

II.5.1.3. Properties of the EM algorithm

The EM algorithm performances are critically summarized below [Mcl 97].

Advantages:

- The EM algorithm is stable with each iteration (*E–step* and *M–step*) increasing the likelihood function.
- The EM algorithm has reliable global convergence; that is, starting from an arbitrary point in the parameter space, convergence is nearly to a local maximizer.
- The EM algorithm is easily to be implemented (easy to program, small storage space).
- The cost *per* iteration is low, thus allowing a large numbers of iterations for the EM algorithm in comparison with the other algorithms. It is easy to monitor convergence and programming errors.

Disadvantages:

- The EM algorithm may converge very slowly.
- The algorithm doesn't guarantee the convergence to a global maximum when there are multiple maxima. In this case, the estimate obtained depends upon the initial value.
- In some cases, the *E–step* may be analytically intractable, although in such situations there is the possibility of performing it via a Monte Carlo approach, [Wei 90].

II.5.2. Confidence Limits

Confidence limits is a tool for probability estimation in discrete case. It can be applied for the continuous case by dividing the continuous distribution in intervals and after that by integrating and estimating this probability in each interval.

II.5.2.1. Presentation

Be $[x_1, x_2, ..., x_N]$ the N observation values resulted after recording the success of the event A define on the events field $\boldsymbol{A}$. The probability of the apparition of the event A is $p = P(A)$.

- This observation results $[x_1, x_2, ..., x_N]$ can be considered derived by the independent and successive measures of a binary random variable X with the distribution:

$$X \to \begin{cases} 1, & P(1) = p \\ 0, & P(0) = 1 - p = \overline{p} \end{cases}. \tag{II.59}$$

- The experimental results can be considered like a particular realization of the sample $[X_1, X_2, ..., X_N]$, associated with the random variable X. This means that $X_1, X_2, ..., X_N$ are *i.i.d.* random variable with the same distribution function, eq. (II.59).

- The probability estimator $\hat{\boldsymbol{p}}$ is defined by the equation $\hat{\boldsymbol{p}} = \frac{\boldsymbol{m}}{N}$, where $\boldsymbol{m}$ is the number of the successes of the event A in all N observations.
  - $\boldsymbol{m}$ is a random variable having a binomial distribution of parameters N and $\boldsymbol{p}$ $\boldsymbol{m} \rightarrow Bi(N, \boldsymbol{p})$.
  - The mean and variance of the probability estimator are: $E[\hat{p}] = p$ and $\sigma_{\hat{p}}^2 = E[(\hat{p} - p)^2] = \frac{p(1-p)}{N}$.

II.5.2.2. Probability Estimation

In order to solve the problem, the probability estimation is necessary to know the distribution function for the estimator $\hat{p}$. If the following conditions are true, then the distribution function for $\hat{\boldsymbol{p}}$ will be a normal (Gaussian) distribution with mean p and variance $\frac{p(1-p)}{N}$.

$$Np(1-p) >> 1 \text{ and } |m - Np| <\approx \sqrt{Np(1-p)} \tag{II.60}$$

Before to investigate the probability it is necessary to define the P_1 like the probability of the absolute difference between the estimation $\hat{\boldsymbol{p}}$ and theoretical probability p, as:

$$P_1 = P\{|\hat{p} - p| < \varepsilon\} = \int_{N(p-\varepsilon)}^{N(p+\varepsilon)} \frac{1}{\sqrt{2\pi Np(1-p)}} \exp\left(-\frac{(x-Np)^2}{2Np(1-p)}\right) dx . \tag{II.61}$$

Changing $\frac{x - Np}{\sqrt{Np(1-p)}}$ with z, the final equation for the P_1 is:

$$P_1 = \int_{-N\varepsilon/\sqrt{Np(1-p)}}^{N\varepsilon/\sqrt{Np(1-p)}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz . \tag{II.62}$$

The goal is to find an interval (the confidence limits) where the probability belongs. The steps are:

Step 1: The probability estimation is computed by the equation: $\hat{\boldsymbol{p}} = \frac{\boldsymbol{m}}{N}$.

Step 2: For a fixed value $P_1 = 1 - \alpha$, the $-z_{\alpha/2}$ and $z_{\alpha/2}$ with $\alpha/2$ superior and inferior percentile points of the Gaussian distribution are obtained. From the eq. (II.62) it results $\varepsilon = z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{N}}$.

The unknown probability p takes values in the interval

$$I = (\hat{p} - \varepsilon; \hat{p} + \varepsilon), \tag{II.63}$$

around the $\hat{p}$ *value with a confidence of* $(1-\alpha)$.

II.6. Conclusion

This chapter presented the theoretical background of the thesis, namely definitions, properties, and models from random variables and processes, information theory, and statistics. All these related yet so different issues are to be further synergistically combined in order to devise a new algorithm for natural data modelling and to grant the coherency in the quantitative evaluations.

Chapter III refers to natural video modelling for watermarking purposes and exploits the random variables, random processes and *pdf* estimation concepts. In the end of the chapter, the continuous entropy of natural video was under investigation.

Chapter IV is focussed on an algorithm paired-designed with the one presented in Chapter III, this time devoted to watermarking attack modelling. Here the information theory application is the continuous capacity evaluation. Additionally, the attack models are the starting point in some Monte Carlo approaches designed in order to speed up the state-of-the-art watermarking methods by a factor of 100.

II.7. References

[Arc 03a] C. Archambeau, J.A. Lee, M. Verleysen, *"On Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures"*, Proc. ESANN (European Symposium on Artificial Neural Networks, Bruges, Belgium, April 2003, pp. 99 – 106.

[Arc 03b] C. Archambeau, M. Verleysen, *"Fully Nonparametric Probability Density Function Estimation with Finite Gaussian Mixtures Models"*, Proc. ICAPR (5th International Conference on Advances in Pattern Recognition, Calcultta, India, December 2003, pp. 81 –84.

[Arc 06] C. Archambeau, M. Valle, A. Assenza, M. Verleysen, *"Assessment of Probability Density Estimation Methods: Parzen Window and Finite Gaussian Mixtures"*, Proc. IEEE International Symposium on Circuits and Systems, Kos, Greece, May 2006.

[Bar 05] M. Barkat, *Signal Detection and Estimation*, 2nd Edition, Artech House, Boston 2005.

[Bil 98] J.A. Bilmes, *"A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models"*, Technical Report,

Department of Electrical Engineering and Computer Science, Berkeley University, California, April 1998.

[Bra 99] S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*, Springer-Verlang, N.Y., 1999.

[Ciu 05] M. Ciuc, C. Vertan, *Prelucrarea Statistica a Semnalelor,* Ed. MatrixRom, 2005.

[Cov 06] T.M. Cover, J.A. Thomas, *Elements of Information Theory,* 2nd Edition, Wiley-Interscience, July 2006.

[Dem 77] A.P. Dempster, N.M. Laird, D.B. Rubin, *"Maximum Likelihood from Incomplete Data via the EM Algorithm"*, Journal of the Royal Statistical Society, Series B, Vol. 39, No. 1, 1977, pp. 1 – 38.

[Ege 95] O. Egecioglu, A. Srinivasan, *"Efficient Non-parametric Estimation of Probability Density Functions"*, Technical Report, Department of Computer Science, University of California Santa Barbara, CA, October 1995.

[Gav 03a] I. Gavăt, *Teoria Informației şi a Codării*, Course notes, Politehnica University Bucharest, 2003 (https://www.lpsv.pub.ro).

[Gav 03b] I. Gavăt, *Detecție şi Estimare în Prelucrarea Informației*, Course notes, Politehnica University Bucharest, 2003 (https://www.lpsv.pub.ro).

[Mcl 97] G.J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, John Wiley & Sons, Inc., N.Y., 1997.

[Mur 83] A.T. Murgan, I. Spânu, I. Gavăt, I. Sztojanov, V.E. Neagoe, A. Vlad, *Teoria Transmisiunii Informației – probleme,* Ed . Didactică şi Pedagocică, Bucharest, 1983.

[Mur 98] A.T. Murgan, *Principiile Teoriei Informatiei in Ingeneria Informatiei si a Comunicatiilor*, Ed. Academiei Romane, Bucharest, 1998.

[Pap 91] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill International Ed., Singapore, 1991.

[Par 62] E. Parzen, *"On Estimation of a Probability Density Function and Mode"*, Annals of Mathematical Statistics, Vol. 33, No. 3, September 1962, pp. 1065 – 1076.

[Pet 95] V. Petrov, *Limit Theorems of Probability Theory*, Clarendon Press, Oxford, 1995.

[Red 84] R.A. Redner, H.F. Walter, *"Mixture Densities, Maximum Likelihood and the EM Algorithm"*, Society for Industrial and Applied Mathematics, Vol. 26, No. 2, April 1984, pp. 195 – 239.

[Roe 97] K. Roeder, L. Wasserman, *"Practical Bayesian Density Estimation using Mixtures of Normals",* Journal of the American Statistical Association, Vol. 92, No; 439, September 1997, pp. 894 – 902.

[Sco 80] D.W. Scott, R.A. Tapia, J.R. Thompson, *"Nonparametric Probability Density Estimation by Discrete Maximum Penalized-Likelihood Criteria"*, The Annals of Statistics, Vol. 8, No. 4, 1980, pp. 820 – 832.

[Sha 80] C.E. Shannon, W. Weaver, *The Mathematical Theory of Communication,* 8th ed., University of Illinois Press, 1980.

[Spa 83] Al. Spătaru, *Teoria Transmisiunii Informației*, Ed. Didactică şi Pedagocică, Bucharest, 1983.

[Spa 87] Al. Spătaru, *Fondements de la Théorie de la Transmission de l'Information*, Presses Polytechniques Romandes, Lausanne, 1987.

[Tra 02] L. Trailovic, L. Pao, *"Variance Estimation and Ranking of Gaussian Mixture Distribution in Target Tracking Applications"*, Proc. of the 41st IEEE Conference on Decision and Control, Las Vegas – Nevada, USA, 2002, pp. 2195 – 2201.

[Ver 99] C. Vertan, I. Gavăt, R. Stoian, *Variabile şi Procese Aleatoare: Principii şi Aplicații*, Ed. Printech, Bucarest, 1999.

[Vla 99] A. Vlad, B. Badea, M. Mitrea, *Metode Statistice în Prelucrarea Informației: Compediu şi Aplicații*, Ed. Metropol, Bucharest, 1999.

[Vla 02] A. Vlad, M. Ferecatu, M. Mitrea, *Teoria Transmisiunii Informatiei II: Elemente teoretice illustrate in Mathcad*, Ed. Paideia, Bucharest, 2002.

[Weg 72] E.J. Wegman, *"Nonparametric Probability Density Estimation: I. A Summary of Available Methods"*, Technometrics (Journal of Statistics for physical, chemical and engineering), Vol. 14, No. 3, August 1972, pp. 533 – 546.

[Wei 90] G.C.G. Wei, M.A. Tanner, *"A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms"*, Journal of the American Statistical Association, Vol. 85, 1990, pp. 699 – 704.

"Science is wonderfully equipped to answer the question 'How?' but it gets terribly confused when you ask the question 'Why?'."

Erwin Chargaff (1905 - 2002)

(Prof. of Biological Chemistry, Columbia Forum, 1969)

Chapter III

Natural Video Modelling for Watermarking Purposes

While keeping a watermarking oriented approach, this chapter bridges the gap between the huge interest in an accurate mathematical model for the natural video and the incomplete and/or improper models already reported in the literature.

In this respect, Section III.1 summarise the state-of-the-art approaches and identifies their current limitations. Section III.2 presents a new method for natural video modelling, ART.MOD-V, which was design in order to handle with mathematical rigour the natural data dependency and their a priori *lack of stationarity and which is able to precise the accuracy of the results. ART.MOD-V is further deployed in order to obtain the first general models for natural video represented in 2D-DWT and 2D-DCT domains (Section III.3). Section III.4 deploys the models to grant a mathematical basis for the estimation of the entropy of the natural source generating the video sequences. Conclusions are drawn and perspectives are opened in Section III.5.*

Contents

III.1. Previous Results

Finding a distribution modelling the 2D–DWT (bi–Dimensional Discrete Wavelet Transform) coefficients or the 2D-DCT (bi-Dimensional Discrete Cosine Transform) coefficients has always been a challenging research topic [Pod 01]; in this respect, a concise state-of-the-art is presented in Sections III.1.1. and III.1.2.

III.1.1. State-of-the-art on 2D-DWT coefficient modelling

The wavelet based tools and methods proved their efficiency and are nowadays intensively involved in a large variety of image processing schemes. Regardless the targeted application (compression [Mal 89], [Ant 92], [Buc 99], denoising [Kiv 99], [Das 02], [Sch 06], classification, texture detection, analysis, restoration [Sim 97], and synthesis [Sri 03]) a special attention has been paid to the statistical model of the DWT coefficients.

Several algorithms have been proposed to compute the DWT coefficients such as the Mallat algorithm [Mal 89], the *"a trous"* algorithm of Holshneider [Hol 89], and the Shensa algorithm [She 92] as a unified approach of the previous two.

In the very fundamental paper on wavelets [Mal 89], the applications related to the image compression, texture discrimination and fractal analysis are considered.

For the first application, the natural images are special kinds of 2D signals. The statistical investigation starts by noting that as the pixels of the detail images are decomposition coefficients of the original image into an orthonormal basis, no correlation exists among them. Then, it is brough into evidence that practically for all images, for all resolutions and orientations, the histograms are symmetrical peaks centred on zero.

The author proposed for the detail image histograms to be modelled by an generalized Gaussian distribution: $h(u) = Ke^{-(|u|/\alpha)^{\beta}}$, where β is a parameter for decreasing the rate of the peak and α models the variance. Note that $\beta = 2$ corresponds to the Gaussian distribution. The parameters α and β can be computed by measuring the first and second moments of the detail image histogram [Mal 89]. This model was built by studying the histograms of several different images decomposed on four resolution levels each.

The constant *K* used in the previous equation is adjusted in order to have the normalised *pdf* approximating the histograms.

[Buc 99] advances a Markov (non-Gaussian) model for a large variety of images: photographic, graphical and medical. The test made with this model was for image compression applications. The image database contained 13 images of 512x512 pixels, encoded at 8 bits/pixel (*Baboon*, *Bark, Boats, Brain, Brownie, Cantrell, Earth, Flowers, Goldhill, Lena, Mt. Will, Vein* and *Wedding*).

To demonstrate the impact of the model, an image EPWIC coder (Embedded Predictive Wavelet Image Coder) was built. The sub-band coefficients are encoded one bitplane at a time using a non-adaptive arithmetic encoder that extracts directly from the model the conditional probabilities. The decoder uses the statistical model in order to predict the value for the coefficients based on the bits it received. In the end of the investigation, the rate-distortion performance of the coder (for only 5 of the 13 images) is shown to outperform image coders available in the literature.

In [Sch 06] and [Dui 07], the wavelet denoising strategies for multicomponent images are considered. For such an application, three models are discussed: Bayesian, generalized Laplacian and Gaussian Scale Mixture (GSM). For Landsat multispectral images, the GSM model is elaborated and validated by simulations.

In [Kiv 99], the DWT coefficients are modelled like *i.i.d.* random variables with generalized Gaussian distribution. The proposed model is applied to image denoising with illustration on *Lena* and *Barbara*.

[Das 02] also models the wavelet coefficients by a Gaussian distribution. A new scheme of wavelet thresholding for noise removal is developed. The idea of this approach is to take the significant wavelet coefficients as signal energy and let the rest as a noise. In order to make this discrimination the Chi-square test is iteratively run. The algorithm was tested on a number of synthetic real-life signals, but in the paper the authors presents the results only for two signals: a synthetic signal (*Bump* with 2048 data samples) and a real-life signal (biomedical signal with 1024 data samples).

In [Sim 97] a parametric statistical model for visual images in the DWT domain was presented. The authors characterize the joint densities of coefficient magnitudes at adjacent spatial locations, orientations and scales. The model accounts for the statistics of a wide variety of image applications: compression, restoration, and synthesis. The test was realized on the *Einstein* and *Boats* images but also on some textures.

For example, for the noise removal problem (a restoration application) [Ros 78] the classical solution is the Wiener filter, which assumes an image model of independent Gaussian distributions of the coefficients in Fourier domain. The non-Gaussian marginal statistics of the filtered sub-bands were used only at the noise removal stage. The images are split into sub-bands, and after that the coefficients are thresholded to suppress the values of low-amplitude. Starting from the high-amplitude values, they use an over complete tight frame represented by four oriented sub-bands at each scale.

Having in view to watermarking applications [Mit 04b], [Dum 07b] carried out a new coefficients investigation which can also establish at what extent the ergodicity hypothesis holds. In this respect, the restrictions of the Gaussian distribution in modelling the DWT coefficients selected according to their decreasing order have been established.

A concise presentation of the proposed procedure [Mit 04a] follows:

(1) Be there a video sequence sampled from the 2D random process representing the video.

(2) Consider the video sequences as a set of L successive frames.

(3) Compute the DWT to the V component of each frame, sort de coefficients in a decreasing order and record the largest R coefficients.

(4) Partition the L values corresponding to an $r = 1,...,R$ rank, into D classes by using a fixed period sampling of period $D = 250$ and by shifting the sampling origin.

(5) Apply for each class the Chi-square test on concordance, the Ro test on correlation, the Fisher test on equality between two variances, and the Student test on equality between two means; all these tests are applied at an $\alpha = 0.05$ significance level.

Note that if D is large enough, the elements in each class are independent.

The results are illustrated in Figure III.1.a and Figure III.1.b. In Figure III.1.a the statistical investigation was applied for three bi-orthogonal DWTs, namely (2,2), (4,4), and (9,7) DWT [Dau 92], [Mit 04c]. The abscissa corresponds to the investigated rank and the ordinate corresponds to the relative number of the Chi-square tests which are not passed.

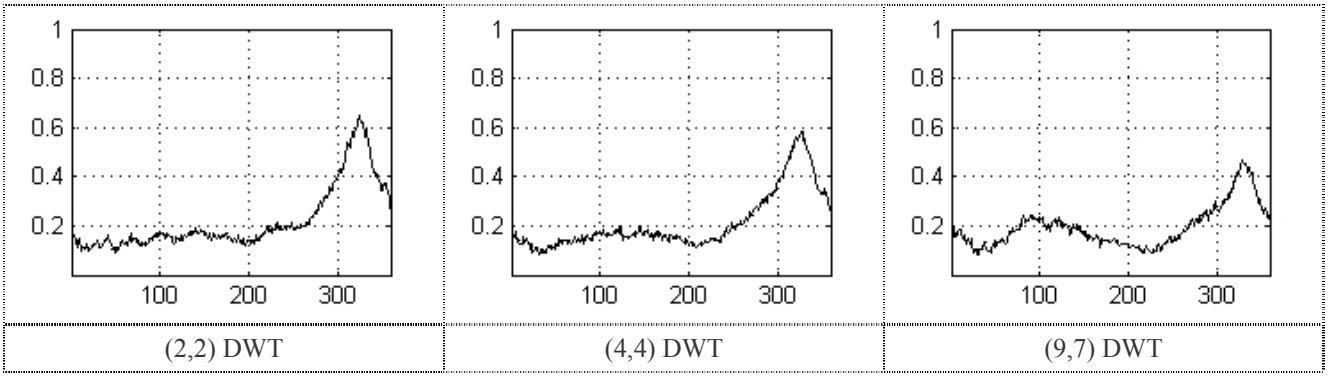


Figure III.1.a. The relative number of Chi-square tests on concordance with the Gaussian distribution which are not passed ($R = 360$ coefficients, $L = 35000$ frames, and $D = 250$ frames).

It can be seen that for the considered DWTs, more than 75% of tests are passed only when:

- $r \in [10; 50] \cup [150; 250]$ in the (9,7) DWT case;
- $r \in [0; 100] \cup [170; 230]$ in the (4,4) DWT case;
- $r \in [0; 210]$ in the (2,2) DWT case.

For others ranks the Gaussian behaviour has been refuted.

When the Chi-square tests were not passed, the Ro, Fisher and Student tests cannot be properly run (such tests are mathematically proved only for Gaussian data). However the very high ratio of the Ro tests which are passed are considered as an encouraging hint in data independency and stationarity: the mean value and the variance are independent with respect to a translation on the time axis.

The Ro, the Fisher and Student tests were applied and the results are illustrated in Figure III.1.b for the (9,7) DWT. The figure axes correspond to the investigated ranks *vs.* the relative number of tests (Ro, Fisher and Student) which are not passed.

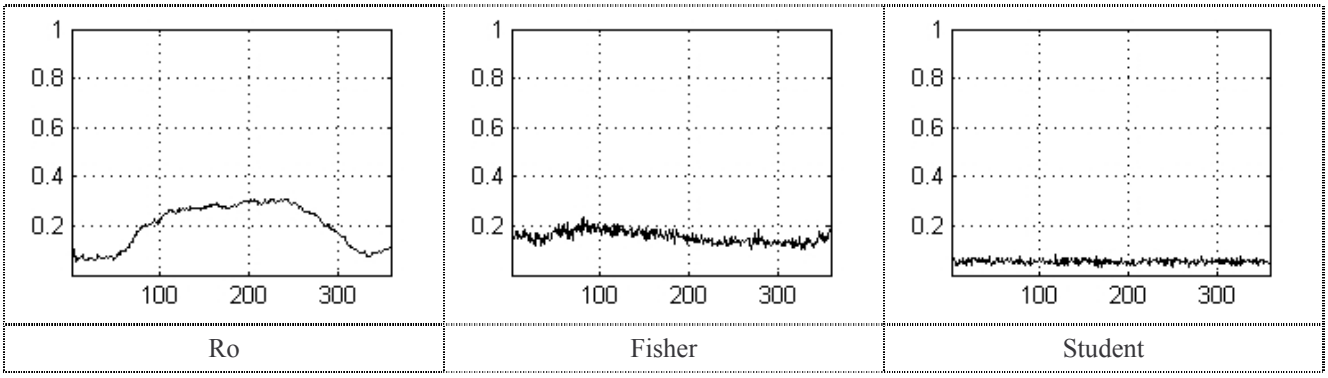


Figure III.1.b. The results of the Ro, Fisher, and Student tests in (9,7) DWT domain ($L = 35000$ frames, $R = 360$ coefficients, and $D = 250$ frames).

III.1.2. State-of-the-art on 2D-DCT coefficient modelling

The core of the most intensively normalised image and video compression schemes, the DCT remains a host research topic.

Reininger and Gibson [Rei 83] consider the DCT coefficients computed on block sizes of 8x8, 16x16, and 32x32 pixels from 5 gray levels images (*Girl, Couple, Moon, X-Ray* and *Aerial*) of 256x256 pixels, 8 bits/pixel. The authors take into investigation different distributions for both DC and AC coefficients, as the Gaussian, Laplacian, Gamma and Rayleigh *pdf*s, for instance. For each image and each block, the Kolmogorov–Smirnov test is applied on the ten high-energy coefficients in the upper left-hand corner of the transform block. The results obtained on these 5 images show that the DC coefficients are modelled by the Gaussian distributions while the AC coefficients are modelled by Laplacian distributions.

In his paper [Mul 93], Müller deals only with blocks of 8x8 pixels extracted from 5 test images of 512x512 pixels (*Lena, Basket, Boats, Clown* and *Baboon* from the USC– SIPI Image Database [Usc 08]), encoded at 8 bits/pixel. He runs the Chi-square test and models the AC coefficients for 4 images with generalised Gaussian distribution [Sun 04]. Concerning the *Baboon* image, he points out to the Laplacian distribution as a better solution. In order to run the statistical tests, the law parameters (mean values and variances) are estimated by a maximum likelihood approach.

The results reported in Joshi and Fisher [Jos 95] consider only 2 images (*Lena* and *Baboon*) of the same type and same block sizes as Müller. Although this paper does not make any new assumptions for AC coefficient behaviour, it confirms the results previously obtained and validates them for image compression applications.

By using a double stochastic model for images, Lam and Goodman [Lam 00] offer some *a priori* reasons for starting the investigation on coefficient *pdf*s with the Laplacian distribution.

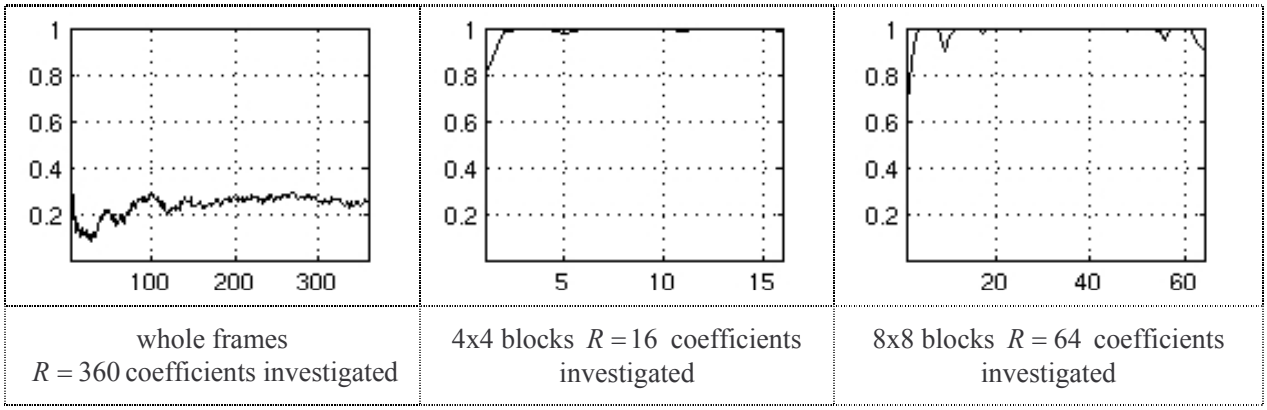
The study in [Mit 04a] follows a different approach: the DCT is computed on whole frames and the coefficients thus obtained are sorted into a decreasing order of their values. Further on, an investigation procedure combining four types of statistical tests (Chi-square, Ro, Fisher, and Student) is applied to a corpus of video sequences (natural & simulated) or medical images. This study identifies with mathematical rigour the validity perimeter for the Gaussian and stationarity hypotheses. However, it is not suitable for mathematical modelling the non-Gaussian behaviours. In [Dum 08a] the authors resumed the study in [Mit 04a] considering three common ways of applying the DCT, namely on whole frames, on 4×4, and 8×8 blocks.

The investigation procedure is paire-designed with the one considered for 2D–DWT (Section III.1.1).

To verify whatever the elements in the considered class are Gaussian distributed or not, the Chi-square test was applied (see the illustrations in Figure III.1.c). The abscissa corresponds to the investigated rank and the ordinate corresponds to the relative number of the Chi-square tests on concordance with the Gaussian distribution which were not passed. It can be observed that more than 80% of tests are passed only when the DCT is applied on the whole frames and when $r \in [10;50]$. For others ranks the Gaussian behaviour has been refuted. Similar results have been obtained on 10 video sequences of $L = 35000$ frames (about 25 minutes each).

Some illustration concerning the independence and stationarity are presented below for the whole frame case. In order to verify whether the sampling period is large enough so as to ensure the data independency in each class, the Ro tests were applied and the results are displayed in Figure III.1.d (left). The axes are the same: the relative number of Ro tests which are not passed *vs.* the rank in the hierarchy.

The Fisher and Student tests (Figure III.1.d – centre and right column) were applied in order to verify the stationarity and pointed out to a very fine homogeneity: each and every time, more than 95% of the Fisher and Student tests are passed. The results concerning the data homogeneity are similar to those obtained for DWT.



whole frames $R = 360$ coefficients investigated	4x4 blocks $R = 16$ coefficients investigated	8x8 blocks $R = 64$ coefficients investigated

Figure III.1.c. The relative number of Chi-square tests on concordance with the Gaussian distribution which are not passed. The video sequence has $L = 35000$ frames and $D = 250$ frames (*i.e.* 10s).

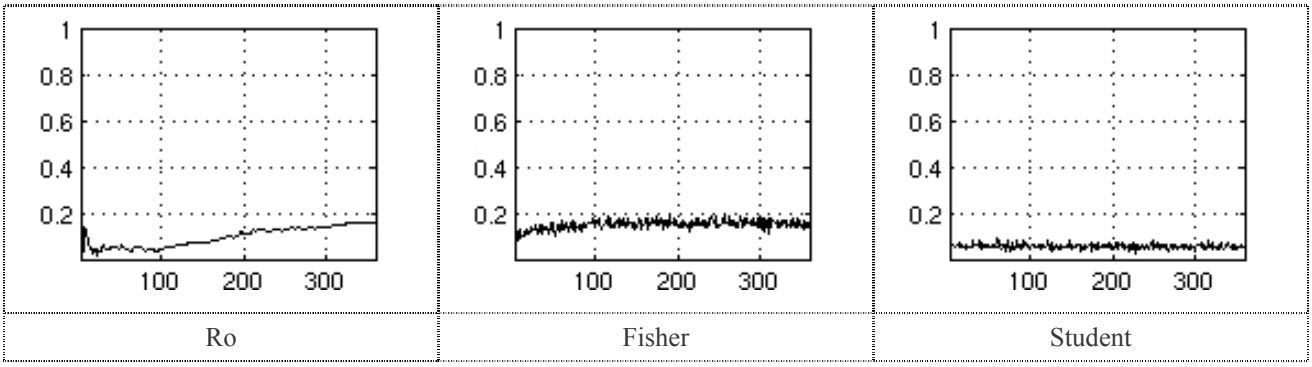


Figure III.1.d. The results of the Ro, Fisher, and Student tests in the case of DCT applied on whole frames ($L = 35000$ frames, $R = 360$ coefficients, and $D = 250$ frames).

The present thesis takes a completely different direction: instead of forcing the coefficients to stick to some distributions and check the goodness of fit, it develops an origin statistical approach able to estimate the true distributions.

III.2. New Method for Video Modelling (ART.MOD – V)

As it can be seen from the previous section, despite the huge variety of approaches to natural video modelling, no method was yet designed in order to:

- **Mathematically handle the *inter* and *intra* frame dependency.** Note that all the common statistical inferences (goodness-of-fit tests, estimation, …) assume the independence of the data they are applied to. However, sampling such independent data from natural video is not a trivial task, because of the inter and intra frame dependency.
- **Carry out an investigation on natural video stationarity/ergodicity**. As explained in Section II.3.4, stationarity and ergodicity are very sophisticated mathematical concepts ensuring that the statistical/temporal description is independent with respect to the way in which the data are sampled. Although never investigated in literature, this property is always assumed in video modelling thus raising suspicions about the mathematical meaning of the results. Actually, this is a crucial issue and any sound video modelling procedure should verify whether the same statistical description can be obtained from different data sets sampled from the same video sequence. Should it be the case, a hint to stationarity is obtained. Should it not be the case, the stationarity is refuted and it is brought to light that no mathematical model exists. Moreover, note that such a behaviour may characterise a particular video sequence (or a class of video sequence) but not all the natural video sequence: hence, the existence of a mathematical model should be incrementally investigated, starting from individual video sequences toward a heterogeneous corpus of video sequences.

In order to accurately model the natural video, while properly dealing with natural video inter and intra frame dependency and with its *a priori* lack of stationarity, the *ART.MOD-V* algorithm was first presented in [Dum 08a], as an extension of the algorithm in [Mit 07a].

From the theoretical point of view, natural video can be considered as a 2D random process. The underlying random experiment imposes the frame content. Consequently, a particular video sequence is obtained by sampling this random process. As a complete investigation on such a 2D random process is computationally unfeasible, the present Section focuses on its first order description (see the flowcharts in Section III.2.1); this task is accomplished by defining some random variables on the same random experiment as the original process (Section III.2.2 – algorithm *step A*) and by further investigating them using *pdf* estimation tools (Section III.2.3– algorithm *step B*). Section III.2.4 (algorithm *step C*) is dedicated to model validation.

III.2.1. Algorithm ART.MOD–V

This Section is a flowcharts overview of the proposed modelling algorithm; the details will be presented in the next Sections.

The algorithm is structured in two parts: model computation and model validation. The two corresponding flowcharts are structured into three columns. The left columns point to the theoretical background, the central columns are dedicated to the algorithm itself, while the right columns offer additional information.

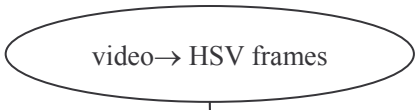
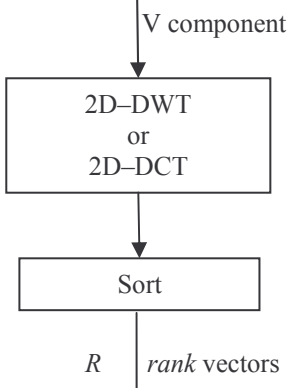
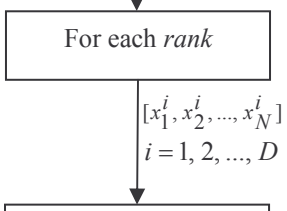
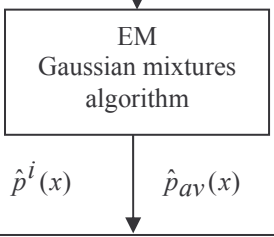
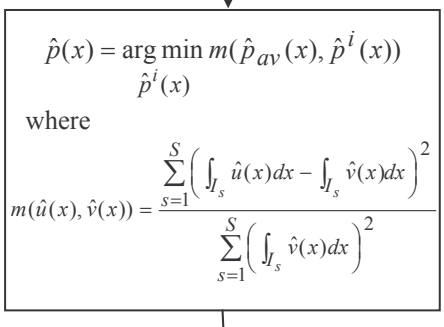
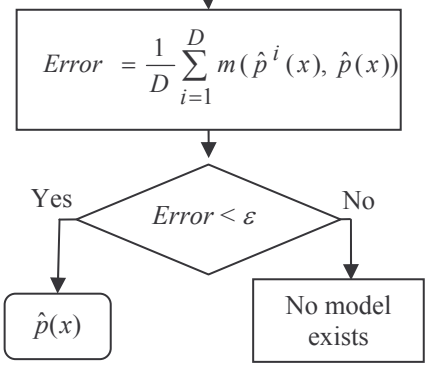
The model computation flowchart summarizes the steps which are necessary in order to compute the model:

- some pre-processing operations representing the video sequence in the space in which it will be investigated;
- extracting different independent data sets from the same sequence in order to be able to properly apply the estimation tools and estimating by means of Gaussian Mixtures the models for each of these independent data sets;
- an evaluation of the variability of the obtained model with respect to the independent data set in order to check-up the inner stationarity of the investigated sequence.

The model validation flowchart precises the generality of the model obtained by this procedure, namely:

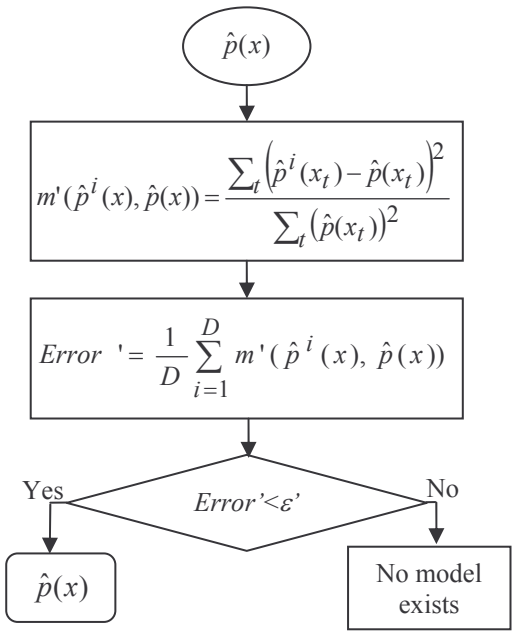
(1) the independency of the model with respect to the similarity measure and error definition;
(2) the independency of the model with respect to the video sequence;
(3) the independency of the model with respect to the estimation procedure.

Note: When reading the algorithm presentation (be it as the following two flowcharts or as the textual details), the reader should be focussed on the way in which data dependency in natural video is overcame, on the estimation procedure and on the stationarity/ergodicity investigation (*i.e.* on the way in which multiple data *i.i.d.* data sets are sampled from the same video sequence and further processed). All the other issues (the colour space in which the natural video is represented, the type of pixel transformation – DWT/DCT, the coefficient selection mode – according to their special frequencies/absolute value hierarchy, the number of investigated coefficients, the sampling period, *etc*) are just simple algorithm parameters, which can be adapted according to any particular application. For instance, in the sequel, these parameters will be set so as to be as general as possible while keeping in mind the thesis targeted application, *i.e.* the watermarking in the DWT/DCT coefficient hierarchy.

Model computation flowchart			
Step A	Video pre-processing		- The video sequence consists of L frames, and each frame is represented in the HSV space.
	2D transformation and sorting		- Apply the transform to the V component: ▪ DWT (applied at N_r resolution level); record the coefficients corresponding to LH and HL sub – bands. ▪ DCT (applied on whole frames, 4x4 or 8x8 blocks). - Sort the coefficients and take the largest R values. - Obtain R *rank* vectors, each of them with L component.
	Sample and obtain *i.i.d.* data		- Periodically sample the *rank* vector (r) with a large period D; - Shift the sampling origin - Obtain D *i.i.d.* data sets with $N=L/D$ components.
Step B	*pdf* estimation using Gaussian mixtures		- Estimate *pdf*s using a Gaussian mixture. - Apply the EM algorithm to each D *i.i.d.* data obtaining $\hat{p}^i(x)$. - Apply the same algorithm to the whole *rank* to obtain the average model $\hat{p}_{av}(x)$.
	Model definition		- Define the model $\hat{p}(x)$ which is the *i.i.d.* estimate closest to the $\hat{p}_{av}(x)$ in the sense of a similarity measure $m(\cdot)$ between *pdf*.
	Model computation		- Compute the average error among and between the $\hat{p}(x)$ model and the D $\hat{p}^i(x)$ estimate. - The model $\hat{p}(x)$ is considered as a general model, if the *Error* is lower than a prescribed threshold ε.

Model validation (1) flowchart: independency of the model with respect to the similarity measure

Step C – (1)



- Sample the $[x_{\min}; x_{\max}]$ interval on which $\hat{p}(x)$ is non-zero with a 0.001 step and obtain a set of x_t points

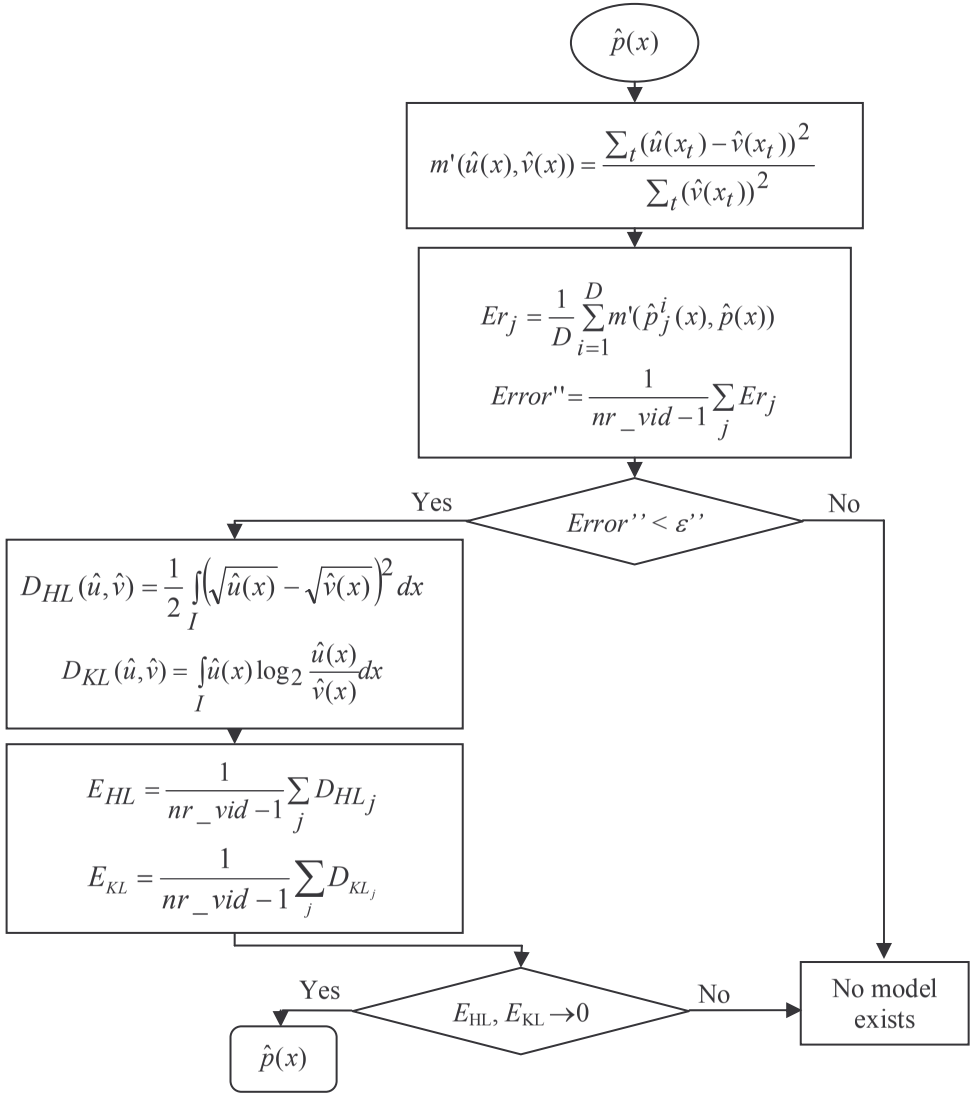
- Define a new $m'(\cdot)$ similarity measure between the *pdf*s.

- Compute the average error between the reference model $\hat{p}(x)$ and the D *pdf*s obtained on the same video sequence.

- The model $\hat{p}(x)$ is validated if the error is lower than the threshold ε'.

Model validation (2) flowchart: independency of the model with respect to the video sequence

Step C – (2)



- Sample the $[x_{\min}; x_{\max}]$ interval on which $\hat{p}(x)$ is non-zero with a 0.001 step and obtain a set of x_t points.

- Define a new similarity measures $m'_i(\cdot)$ between the *pdf*s.

- Compute the error (*Er)* between the reference model $\hat{p}(x)$ and the *D pdf*s obtained for the video sequence *j* ($j = 1,...,nr\_vid - 1$, where *nr_vid* is the number of video sequences available in our corpus).
- Calculate the *Error''*, as an average of all Er_j.

- If *Error''* is lower than the threshold ε'', two new similarity measures D_{HL}, D_{KL} between *pdf*s are introduced.

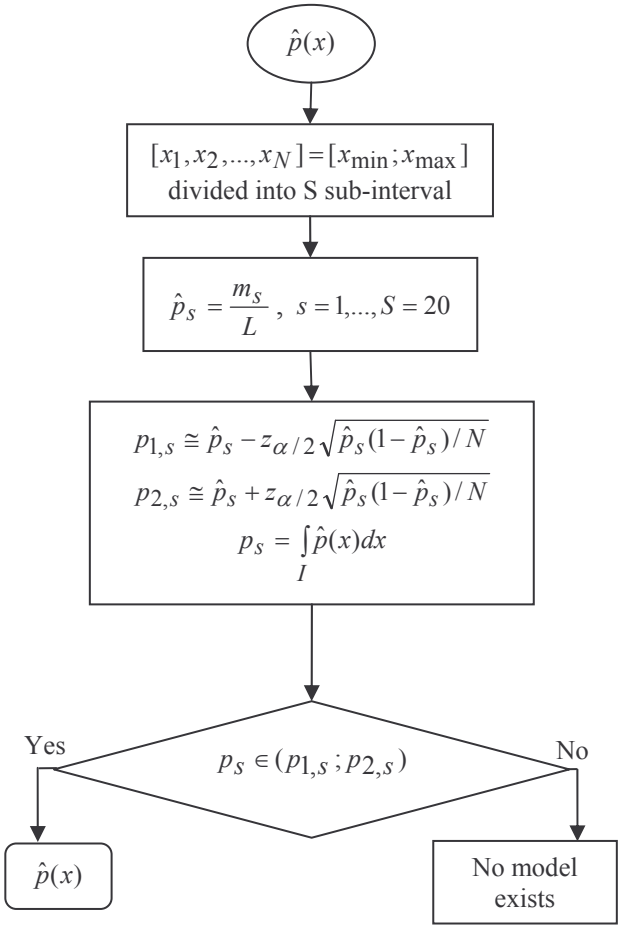
- Evaluate the Hellinger distance and Kullback-Leibler divergence between the reference model $\hat{p}(x)$ and other model computed on a video sequence *j* (one model for each video sequence was computed with the procedure presented in *step A – step B*).

- Calculate the average value (E_{HL}, E_{KL}) of all *j* D_{HL}, D_{KL}.

- The model $\hat{p}(x)$ is validated if the *Error''* is lower than the threshold ε'' and E_{HL}, E_{KL} are near to zero.

Model validation (3) flowchart: independency of the model with respect to the estimation procedure

Step C-(3)



- Consider a division I_S, $s = 1,...,S_{conf}$, of the interval on witch $\hat{p}(x)$ is non-zero.

- Estimate the probability $\hat{p}_S = \frac{m_S}{L}$ for each interval, where m_S is the number of experimental data $\sum_{s=1}^{S} m_S = L$, $\sum_{s=1}^{S} \hat{p}_S = 1$.

- Compute the $p_{1,s}$ and $p_{2,s}$ confidence limits, where $z_{\alpha/2}$ is the $\alpha/2$ - point value of the Gaussian distribution $N(\mu = 0, \sigma = 1)$.

- Compute p_S starting from model

- The model is validated if $p_S \in (p_{1,s}; p_{2,s})$.

The elements in these flowcharts will be further detailed.

Be there a random process whose samples are the frames of the natural video and consider a sequence of L frames.

III.2.2. Step A: Natural Video Representation

The purpose of this step is to extract from the video sequence the salient data to be investigated. Each frame is represented in the HSV (Hue-Saturation-Value) space. Note: this choice was done according to the watermarking methods developed at the ARTEMIS Department; however, it does not conceptually restrict the generality of the results. Actually, all the colour spaces represent their components by some one-to-one conversions of the R, G, and B basic components. Hence, one-to-one mappings can be expressed between any two luminance components corresponding to different spaces (*e.g.* between the V and the Y components) [Sul 03], [Tka 08]. Assuming the *pdf* modelling the V component is known, the *pdf* modelling other luminance components can be computed by linear and/or non-linear *pdf* filtering.

The proposed statistical investigation is carried out on R random variables, derived from the considered random process as follows:

For each frame in the video sequence:

Step A.1: Apply the considered transform (DWT or DCT) to the V (luminance component). The DWT is always applied to the whole frame, at an $N_r = 3$ resolution level. The DCT may be applied on the whole frame or on square blocks (4×4 or 8×8).

Step A.2: Record in a vector the obtained coefficients (Figure III.2).

When applying the DWT only, the values corresponding to the *LH* and *HL* lowest frequency sub-bands are taken into account.

Note: The two frequency sub-band selection was done according to a large class of watermarking applications, [Mit 05] [Kwa 02], [Hie 04], [Par 07], [Red 09]. Should the experimenter be interested in other sub-bands [Xia 98], [Liu 07], [Tin 08], [Mon 09], [Pat 03], our algorithm should be resumed.

When applying the DCT on the whole frame, all the values are considered at this step; when applying it on square blocks, only one block (randomly chosen) per frame is considered.

Step A.3: Build up the coefficient hierarchy by sorting in a decreasing order the vector obtained in the previous step and record the largest R ranks; the resulting vector is denoted by *coefficients*.

In the DWT case $R = 360$ values. When the DCT is applied to the whole frame, only the largest $R = 360$ coefficients are considered; when applying it on square blocks (4x4 and 8x8 blocks), only one block per frame, randomly chosen and the largest $R = 5$ values will be further investigated.

After applying the Steps A.1 - A.3 to each frame in the video sequence, a set of L vectors (each of them with R components) of the same type as *coefficients* is obtained. Should we consider only the values corresponding to an arbitrarily chosen rank r (Figure III.3), a new vector of L components is obtained: $rank = [c_1, c_2, ..., c_L]$. A c_j component ($j = 1, 2, ..., L$) in the *rank* vector is the value corresponding to the rank r in the *coefficient* vector computed on the frame j. The *rank* vector components are random and depend on the original frame content. From the statistical point of view, this means that the *rank* vector components are sampled from a random variable defined on the same probabilistic experiment as the 2D random process representing the video sequences.

Hence, to individually investigate these random variables means to individually investigate r sets for data of the type $[c_1, c_2, ..., c_L]$.

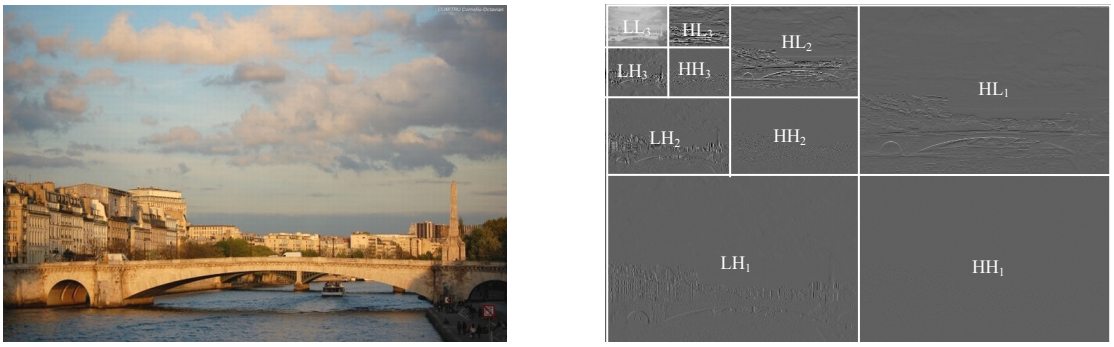


Figure III.2.a. The selected sub-bands for DWT.

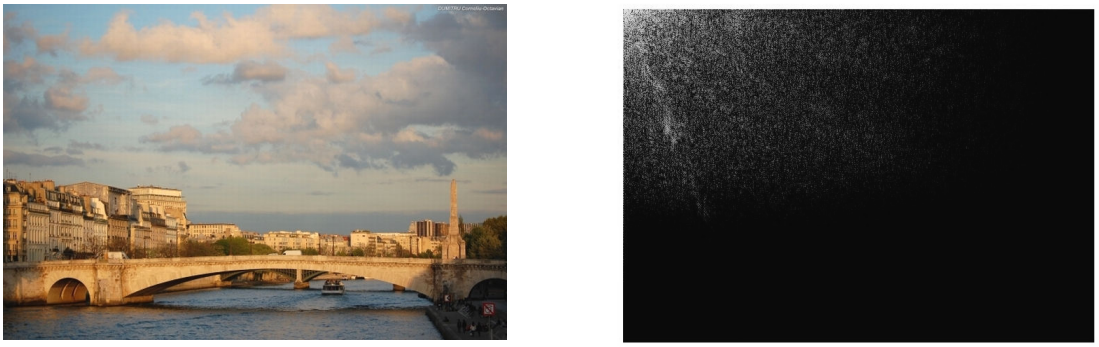


Figure III.2.b. The DCT applied on the whole frame.

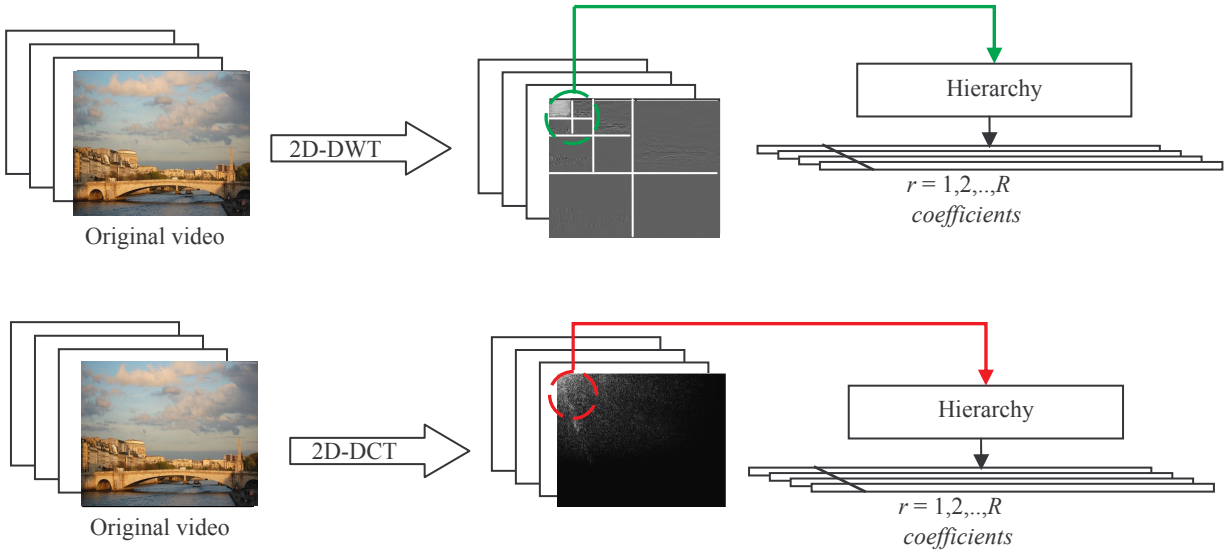


Figure III.2.c. Sampling the random variables modelling the video – in 2D-DWT (applied on the whole frame) or 2D-DCT (applied on the whole frame) case.

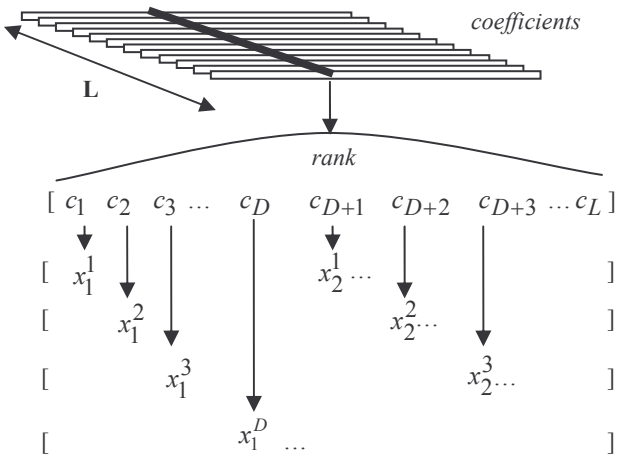


Figure III.3. By shifting the sampling origin, several *i.i.d.* data sets can be obtained from the same rank vector (see *Step A* and *Step B* of the algorithm).

III.2.3. StepB: Pdf Estimation for Natural Video

This step is meant to establish with mathematical rigour:

- The stationarity of the video sequence.

  In other words, is the pdf characterising the values of such a random variable unique or does it depend on the time origin (Figure III.4)? In the sequel, in order to answer this question, several independent data sets, sampled with different time origins should be extracted from the same video sequence. The data independence is supposed to be obtained by having a large enough periodical sampling while the different time origins are obtained by shifting the sampling starting point. Should the differences (in the sense of some similarity measure) among the models estimated on these different data sets be lower than a given threshold, we may consider a positive answer to this question is obtained.
- The mathematical expression for a general model.

  If such a pdf exist, which is its mathematical expression?

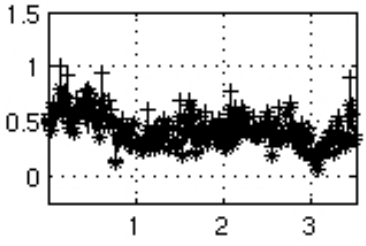


Figure III.4. Samples from the random variables approximating the natural video, for an arbitrarily chosen rank: the 3 plots represent the same video sequence but three different time origins. The values on the ordinate correspond to V components normalised in the [0, 1] interval. The values on the abscissa should be multiplied by 10^4.

Be there $[x_1, x_2, ..., x_N]$ a sample of N experimental data complying with the *iid* (*i*ndependent and *i*dentically *d*istributed data) model.

We suppose that these data are sampled from an X random variable whose $p(x)$ *pdf* is unknown and should be estimated using *Gaussian mixtures* see Section II.5 [Arc 06a], [Arc 06b]. The $\hat{p}(x)$ estimate of the $p(x)$ function can be computed by eq. (III.1):

$$\hat{p}(x) = \sum_{k=1}^{K} P(k)\, p_k(x),$$
$$\text{where } p_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right). \tag{III.1}$$

In this respect, an EM algorithm (considering a maximum likelihood based criterion) is performed (see Section II.5.1.2 where the algorithm was presented its general form *i.e.* $\Phi = (P(k), \phi_k)$, $\phi_k = (\mu_k, \sigma_k)$. Equations (II.57) and (II.58) can be rewritten now as:

- *E* step:
$$p^{(l)}(k|x_n) = \frac{p_k^{(l)}(x_n) P^{(l)}(k)}{\hat{p}^{(l)}(x_n)}, \tag{III.2.a}$$

- *M* step:
$$\mu_k^{(l+1)} = \frac{\sum_{n=1}^{N} p^{(l)}(k|x_n) x_n}{\sum_{n=1}^{N} p^{(l)}(k|x_n)},$$
$$(\sigma_k^2)^{(l+1)} = \frac{\sum_{n=1}^{N} p^{(l)}(k|x_n)\left(x_n - \mu_k^{(l+1)}\right)^2}{\sum_{n=1}^{N} p^{(l)}(k|x_n)}, \tag{III.2.b}$$
$$P^{(l+1)}(k) = \frac{1}{N} \sum_{n=1}^{N} p^{(l)}(k|x_n),$$

where the (l) upper index denotes the current iteration; the total number of iterations is subject to the experimenter choice ($l = 1, 2, ..., N_{iter}$). Concerning the initialisation step, we adopted a random rule matched to the experimental data.

When trying to deploy the statistical tools, the first problem is connected to the data dependency. Note that the components on the *rank* vector are computed on successive frames in the video sequence. In order to extract *iid* data from such dependent frames, the *rank* vector can be sampled with a fixed large period D. Actually, by shifting the sampling origin, D *iid* data sets, each of them with L/D components, can be obtained from the same *coefficient* vector, see Figure III.3. Assuming the stationarity hypothesis is true for natural video sequence, these D data sets would point

to the same mathematical model; hence, in order to check-out the natural video stationarity, the agreement among the D models obtained on these D data sets will be investigated.

The EM Gaussian mixtures estimation may be applied to each of these data sets and D different *pdf* estimates may be obtained. Under the stationarity hypothesis, all these estimates would provide the same information about the investigated rank. However, as no *a priori* information concerning video stationarity is available, the homogeneity of these estimates should be first checked-up and then a selection criterion allowing the definition of a unitary model should be stated. In order to model the coefficients corresponding to a rank (arbitrarily chosen) in the DWT or DCT coefficient hierarchy, the following 6 steps should be gone through:

Step B.1: Periodically sample the *rank* vector, with a large enough period D. Shift the sampling origin and obtain the D *iid* data sets $[x_1^i, x_2^i, ..., x_N^i]$ ($i = 1, 2, ..., D$ and $N = L/D$), *cf.* Figure III.3. This way, all the natural data are further consider into video modelling and not only an arbitrarily chosen $[x_1^i, x_2^i, ..., x_N^i]$ sample.

Step B.2: Apply the EM Gaussian mixture algorithm to each of the D *iid* data sets and obtain the D *pdf* estimates $\hat{p}^i(x)$, $i = 1, 2, ..., D$.

Step B.3: Apply the EM Gaussian mixture algorithm to the whole *rank* vector and obtain the estimate $\hat{p}_{av}(x)$. Note that although $\hat{p}_{av}(x)$ has no inner mathematical meaning (it is computed on the whole *rank* vector, hence on dependent data), it intuitively expresses an average value for the $\hat{p}^i(x)$, $i = 1, 2, ..., D$ estimates.

Step B.4: Define a similarity measure, denoted by $m(.)$, between two arbitrarily chosen $\hat{u}$ and $\hat{v}$ *pdf* estimates, as a normalised sum of squared errors between probabilities:

$$m(\hat{u}(x), \hat{v}(x)) = \frac{\sum_{s=1}^{S} \left(\int_{I_s} \hat{u}(x)dx - \int_{I_s} \hat{v}(x)dx \right)^2}{\sum_{s=1}^{S} \left(\int_{I_s} \hat{v}(x)dx \right)^2}, \qquad \text{(III.3)}$$

where I_s, $s = 1, 2, ..., S$ is a subdivision of the $[x_{\min}; x_{\max}]$ interval on which the $\hat{u}$ and $\hat{v}$ functions take values.

Step B.5: **The model** for the investigated rank is defined as the $\hat{p}^i(x)$ estimate to which $\hat{p}_{av}(x)$ is closest in the $m(.)$ sense; be it denoted by $\hat{p}(x)$:

$$\hat{p}(x) = \arg\min_{\hat{p}^i(x)} m(\hat{p}_{av}(x), \hat{p}^i(x)). \qquad \text{(III.4)}$$

Step B.6: In order to check up the pertinence of the model defined in the previous step, compute the average the measures between each of the *D* estimates and the model:

$$Error = \frac{1}{D}\sum_{i=1}^{D} m(\hat{p}^i(x), \hat{p}(x)) . \qquad \text{(III.5)}$$

The *pdf* defined by eq. (III.4) can be considered as a true model only if the *Error* computed by eq. (III.5) is lower than a prescribed upper limit (for instance, lower than 0.10). This way of defining the model ensures three advantages: (1) has a sound mathematical meaning granted by the Gaussian mixture estimation on *i.i.d.* data; (2) is as close as possible to a model independnet with respect to the time origin; (3) its existence can be considered as a hint to the natural video stationarity.

III.2.4. Step C: Model Validation

The aim of this section is to validate the model thus computed; this will be achieved by successively answering to the following three questions:

(1) Is the model dependent on the similarity measure?

(2) Is the model general or does it depends on the original video sequence?

(3) Is the model dependent on the EM Gaussian mixtures estimation?

(1) Is the model dependent on the similarity measure?

The way in which the model was defined is intimately based on the similarity measure and error definitions expressed by eq. (III.3) and eq. (III.5). A good model should be able to describe all the *D pdf* estimates even when the similarity measure changes. As eq. (III.3) expresses the differences between estimates in probability (*i.e.* between the integral on the *pdf*s), we are now to evaluate the differences between the *pdf*s themselves. In this respect, the $[x_{\min}; x_{\max}]$ domain is evenly sampled with a 0.001 step thus obtaining a set of x_t points. The new similarity measure between any of the $\hat{p}^i(x)$ estimates ($i = 1, 2, ..., D$) and the $\hat{p}(x)$ model is chosen as a normalised sum of squared errors in the x_t points:

$$m'(\hat{p}^i(x), \hat{p}(x)) = \frac{\sum_t (\hat{p}^i(x_t) - \hat{p}(x_t))^2}{\sum_t (\hat{p}(x_t))^2}, \qquad \text{(III.6.a)}$$

Consequently, the new average error is computed according to eq. (III.6.b):

$$Error' = \frac{1}{D}\sum_{i=1}^{D} m'(\hat{p}^i(x), \hat{p}(x)) . \qquad \text{(III.6.b)}$$

It can be considered that the model is independent with respect to the similarity measure when this *Error'* is lower than a prescribe threshold (*e.q.* 0.15).

(2) Is the model general or does it depends on the original video sequence?

In order to decide on that, the following steps should be considered:

- Be there the reference model, $\hat{p}(x)$ computed according to eq. (III.4).
- Compute the *D pdf*s applying the algorithm *ART.MOD-V* (*step A* and *B*) for a new video sequence j ($j = 1,...,nr\_vid$), where nr_vid is the number of the video sequences in the experimenter corpus; be these *pdf*s denoted by $\hat{p}_j^i(x)$, where $i = 1,...,D$.
- Compute the error *Er* between the reference model $\hat{p}(x)$ and the video sequence *j* as:

$$Er_j = \frac{1}{D}\sum_{i=1}^{D} m'(\hat{p}_j^i(x), \hat{p}(x)) \quad . \tag{III.7.a}$$

- Compute the average of the errors (between the reference model $\hat{p}(x)$ and other models) as:

$$Error'' = \frac{1}{nr\_vid - 1}\sum_j Er_j \ . \tag{III.7.b}$$

Should this measure be larger than a prescribed value (*i.e.* 0.2) no model exists. However, if it is lower than the threshold it makes sense to go further in evaluating the differences among the video sequences.

Should suspicions arise about the way in which we defined the $m(.)$ and $m'(.)$ similarity measures, two additional metrics, intensively considered in the literature [Fug 03] in order to bring into light the differences between two *pdf*s may be also taken into account: the Kullback–Leibler divergence (relative entropy) [Kul 51], [Nik 04] and the Hellinger distance [Bas 96].

- Be there $\hat{p}(x)$ the reference model and and $\hat{p}_j(x)$ the model of the video sequence *j* (obtained applying the eq. (III.4) form the *ART.MOD-V* algorithm), respectively.
- Be $I = [\mu_{mix} - 3\cdot\sigma_{mix}, \mu_{mix} + 3\cdot\sigma_{mix}]$ an interval on which the *pdf*s are defined, where μ_{mix} is the mixture mean and σ_{mix} is the mixture variance [Tra 02].
- Compute the Hellinger distance (D_{HL}) [Arc 06b] and the Kullback–Leibler divergence (D_{KL}) [Kul 51], [Arc 03a] as:

$$D_{HL_j} = \frac{1}{2}\int_I (\sqrt{\hat{p}(x)} - \sqrt{\hat{p}_j(x)})^2 dx \ , \tag{III.8.a}$$

$$D_{KL_j} = \int_I \hat{p}(x)\log_2\frac{\hat{p}(x)}{\hat{p}_j(x)}dx \ . \tag{III.8.b}$$

As it can be seen from these definitions, by considering a relative entropy, the Kullback-Leibler divergence somewhat associates an informational meaning to the difference between two models. The side effect consists in its computational complexity. Conversely, the Hellinger distance is often explicitly computable, but is less fine and more illustrative than the Kullback-Leibler.

- Apply the two measures for all the video sequences j, and calculate the average of these two measures, as:

$$E_{HL} = \frac{1}{nr\_vid - 1} \sum_j D_{HL_j} \ , \tag{III.9.a}$$

$$E_{KL} = \frac{1}{nr\_vid - 1} \sum_j D_{KL_j} \ . \tag{III.9.b}$$

The model $\hat{p}(x)$ computed by eq. (III.4) is validated if the average measures E_{HL} and E_{KL} approach zero, otherwise it can be considered that no model exists.

Note: In all the experiments (see the numerical results in Section III.3), the E_{HL} and E_{KL} were never involved in a decision making: actually, if and every time *Error*" was lower than the considered threshold, the E_{HL} and E_{KL} were also lower than their corresponding upper limits. So, in our experiments, they act just as an additional validation.

(3) Is the model dependent on the EM Gaussian mixtures estimation?

The EM Gaussian mixture-based approaches estimate the searched *pdf* by an analytic function whose parameters are computed on the experimental data. The *pdf* estimation with confidence limits follows a different philosophy [Fri 83], [Wal 89]. It starts from the idea that the *pdf* to be estimated is generally employed to compute the probability that the investigated random variable takes values in a given I interval. Consequently, it would be useful to directly estimate this probability and not the *pdf per-se*.

Be there $x_{\min}$ and $x_{\max}$ the extreme values in the considered *rank* vector. Consider a partition of the $[x_{\min}; x_{\max}]$ interval into a set of S_{conf} sub-intervals denoted by I_s, $s = 1,2,...,S_{conf}$.

The confidence limits for the probability that the coefficient of the considered *rank* would take values in an I_S are computed as (see eq. (II.63)) $(\hat{p}_S - \varepsilon_S ; \hat{p}_S + \varepsilon_S)$, where:

- the estimation $\hat{p}_S$ is the ratio of the number of *rank* value in the I_S interval to L;
- $\varepsilon_S = z_{\alpha/2} \sqrt{\frac{\hat{p}_S (1 - \hat{p}_S)}{N}}$, N is the size of an $[x_1, x_2, ..., x_N]$ sample.

Note that this way of computing the confidence limits ensures an estimate $\hat{p}_S$ independent with respect to the D data sets and an ε_S value in concordance with the *i.i.d.* data size considered in maximum likelihood estimation ($N = L / D$).

The confidence limit estimation is considered in this Algorithm just as a validation criterion: should the $\hat{p}(x)$ model computed by eq. (III.4) and eq. (III.5) be general, it is expected for the theoretical probability computed starting from this model (*i.e.* $\int_{I_S} \hat{p}(x)dx$) to belong to the $(\hat{p}_S - \varepsilon_S ; \hat{p}_S + \varepsilon_S)$ confidence interval. In such a case, the EM Gaussian mixture estimation will be granted a statistical confidence mining.

III.3. Experimental Results

The results obtained when applying the *algorithm ART.MOD-V* (*step A – step C*) to a video corpus (presented in Section III.3.1) are illustrated in Section III.3.2 and Appendix I; the complete set of numerical results is available on the accompanying CD-ROM.

The first difficulty in the experimental work is to establish the data on which the investigations procedure should be applied. Hence, at least three points should be clarified:

- whether an individual video sequence, arbitrarily chosen, allows a mathematical model to be defined; such a model will be further denoted by IVM – Individual Video Model;
- whether a virtual video sequence obtained by sampling different video sequences allows a mathematical model to be defined (MVM – Multiple Video Model);
- if the two previous questions provide positive answers, whether the IVM and MVM models are identical.

The models are presented first for the 2D-DWT and then for the 2D-DCT coefficient modelling, in Section III.3.2.1 and Section III.3.2.2, respectively.

III.3.1. Video Corpus

The corpus involved in the experiments is composed of 10 video sequences, belonging to different movies, each of them having about 25 minutes (25 frames/s, *i.e.* $L = 35000$ frames in each video sequence). Their content is heterogeneous, combining indoor and outdoor scenes, instable and arbitrary lighting conditions, still and high motion scenes. In order not to bias the statistical relevance of the results, the black letterboxing bars were not considered in the investigation.

Each of the 10 video sequences are DivX coded at two different rates:

- 64 kbit/s with frame sizes 192×80 pixels (according to mobile network requirements),
- 512 kbit/s with frame size 640×480 pixels (according to Internet distribution requirements).

This video corpus was built up at the ARTEMIS Department in cooperation with its industrial partners, namely SFR (Vodafone mobile service provider in France) and the HD3D SAS (under the framework of the HD3D-IIO project, Cap Digital competitiveness cluster in Ile-de-France).

III.3.2. Statistical models

The modelling refers to the $R = 360$ largest coefficients of the (9,7) DWT (applied at a resolution level $N_r = 3$) or of the DCT applied to the whole frames. In case the DCT is applied to 4×4 or 8×8 blocks only the $R = 5$ largest coefficients are considered (extracted from one block which was randomly chosen from the frame).

The following numerical values are considered in the *Step B*:

- $D = 250$ frame sampling period (*i.e.* 10 seconds);
- $K = 10$ Gaussian distributions (see Section III.3.2.3 for more details);
- $N_{iter} = 100$ iterations in the EM algorithm;
- $S = 20$ evenly distributed intervals (see *Step B.4* of the algorithm).

For confidence limits validation, see *Step C-(3)*, $S_{conf} = 10$ equal length subintervals were considered and $\alpha = 0.05$ ($z_{\alpha/2} = 1.96$).

The model computation for different video sequences from the video corpus is illustrated in Tables III.1 ÷ III.4 for the (9,7) DWT case and in Tables III.21 ÷ III.24 for the DCT case.

The results presented in these tables correspond to two compression rates (namely 64 kbit/s and 512 kbit/s), to DCT (applied on whole frames, or on blocks - 4×4, 8×8) and DWT (applied on the whole frames), and to some selected ranks. Additional quantitative results can be found in Appendix I and in the accompanying CD-ROM.

All the tables provide the numerical values for the $P(k)$, $\mu(k)$ and $\sigma(k)$ parameters which are to be considered in eq. (III.1) in order to obtain the $\hat{p}(x)$ model (eq. (III.4)). The average relative error in model computation, *cf.* eq. (III.5), is also presented.

III.3.2.1. The 2D-DWT coefficient modelling

III.3.2.1.1. Model Computation

In order to illustrate the model computation for the (9,7) DWT applied to an individual video sequence (models denoted by IVM – Individual Video Model), coded at low rate, Table III.1 presents the parameters $P(k)$, $\mu(k)$ and $\sigma(k)$ and the associated *Error* corresponding to four ranks, namely: $r = 1, r = 100, r = 200, r = 300$. The graphical representations for the models computed for $r = 1, r = 100, r = 200$ alongside with their corresponding histograms are displayed in Figure III.5.

Table III.2 and Figure III.6 present the same type of information as Table III.1 and Figure III.5, respectively, this time obtained out of processing high quality video sequence.

The experiments were resumed on video sequences obtained by concatenating four quarters from four different video sequences (models denoted by MVM – Multiple Video Model). The quantitative results are displayed in Tables III.3 and III.4 (corresponding to Tables III.1 and III.2, respectively) and Figures III.7 and III.8 (corresponding to Figures III.5 and III.6).

Table III.1. IVM for (9,7) DWT coefficient hierarchy and low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.075	0.070	0.074	0.134	0.071	0.131	0.151	0.098	0.060	0.137	0.039
	$\mu(k)$	0.575	0.405	0.646	0.497	0.544	0.535	0.627	0.628	0.654	0.850	
	$\sigma(k)$	0.112	0.076	0.138	0.089	0.100	0.098	0.062	0.133	0.140	0.125	
100	$P(k)$	0.064	0.101	0.056	0.070	0.090	0.080	0.216	0.073	0.043	0.207	0.002
	$\mu(k)$	0.086	0.094	0.084	0.130	0.090	0.090	0.090	0.090	0.100	0.106	
	$\sigma(k)$	0.033	0.025	0.035	0.039	0.026	0.026	0.024	0.025	0.040	0.015	
200	$P(k)$	0.058	0.178	0.057	0.083	0.099	0.056	0.068	0.054	0.157	0.189	0.010
	$\mu(k)$	0.041	0.036	0.040	0.059	0.043	0.041	0.043	0.041	0.032	0.033	
	$\sigma(k)$	0.013	0.014	0.013	0.019	0.012	0.013	0.012	0.013	0.007	0.009	
300	$P(k)$	0.450	0.030	0.233	0.032	0.016	0.021	0.032	0.012	0.035	0.034	0.021
	$\mu(k)$	0.007	0.013	0.012	0.013	0.015	0.020	0.013	0.012	0.013	0.020	
	$\sigma(k)$	0.003	0.003	0.004	0.003	0.005	0.006	0.004	0.004	0.003	0.005	

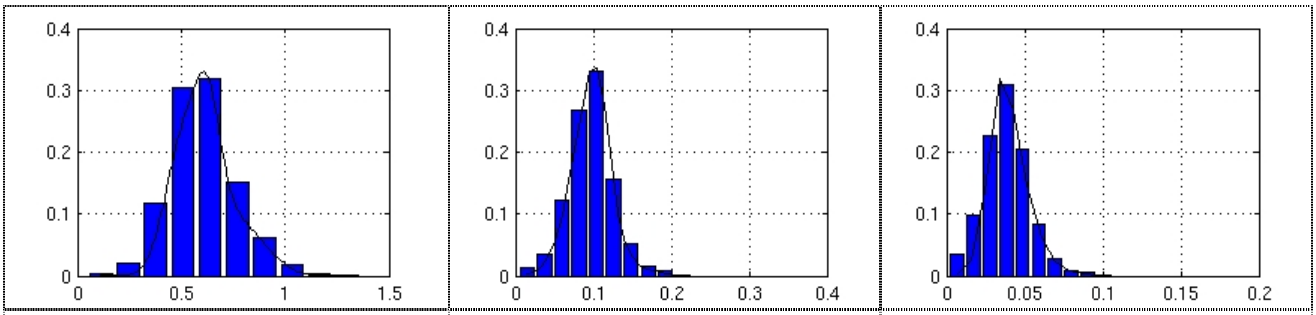


Figure III.5 IVM and histograms computed for three ranks in the (9,7) DWT and low quality video.

Table III.2. IVM for the (9,7) DWT coefficient hierarchy and high quality video.

Rank	Model parameters											Error
1	$P(k)$	0.119	0.122	0.107	0.119	0.129	0.118	0.117	0.014	0.096	0.059	0.028
	$\mu(k)$	0.570	0.713	0.740	0.782	0.508	0.921	0.747	1.730	0.628	0.849	
	$\sigma(k)$	0.124	0.148	0.157	0.175	0.108	0.169	0.159	0.044	0.139	0.109	
100	$P(k)$	0.188	0.047	0.284	0.061	0.059	0.047	0.065	0.079	0.069	0.100	0.033
	$\mu(k)$	0.140	0.226	0.250	0.277	0.227	0.223	0.240	0.310	0.220	0.241	
	$\sigma(k)$	0.032	0.048	0.033	0.070	0.049	0.048	0.060	0.065	0.049	0.034	
200	$P(k)$	0.085	0.083	0.251	0.067	0.108	0.108	0.062	0.094	0.047	0.097	0.030
	$\mu(k)$	0.094	0.180	0.187	0.226	0.170	0.125	0.135	0.173	0.229	0.175	
	$\sigma(k)$	0.013	0.065	0.026	0.063	0.036	0.052	0.054	0.044	0.062	0.041	
300	$P(k)$	0.079	0.073	0.066	0.121	0.086	0.145	0.036	0.076	0.130	0.195	0.032
	$\mu(k)$	0.218	0.0143	0.071	0.113	0.099	0.118	0.133	0.129	0.129	0.159	
	$\sigma(k)$	0.048	0.039	0.001	0.031	0.024	0.047	0.004	0.036	0.035	0.016	

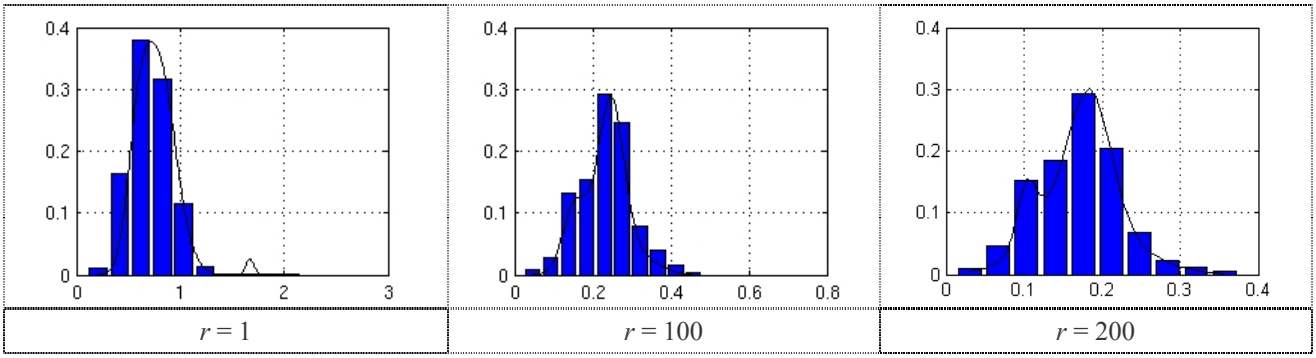


Figure III.6 IVM and histograms computed for three ranks, (9,7) DWT and high quality video.

Table III.3. MVM for the (9,7) DWT coefficient hierarchy and low quality video.

Rank	Model parameters											Error
1	*P(k)*	0.129	0.046	0.100	0.133	0.110	0.116	0.060	0.133	0.060	0.110	0.038
	μ(k)	0.574	0.685	0.443	0.566	0.314	0.383	0.515	0.514	0.431	0.299	
	σ(k)	0.125	0.226	0.142	0.128	0.108	0.111	0.141	0.141	0.139	0.105	
100	*P(k)*	0.114	0.110	0.129	0.127	0.069	0.020	0.077	0.129	0.119	0.105	0.036
	μ(k)	0.070	0.053	0.025	0.091	0.091	0.013	0.079	0.049	0.057	0.107	
	σ(k)	0.018	0.018	0.011	0.023	0.023	0.001	0.024	0.016	0.018	0.019	
200	*P(k)*	0.026	0.061	0.066	0.154	0.143	0.086	0.251	0.072	0.127	0.013	0.043
	μ(k)	0.030	0.032	0.030	0.022	0.045	0.032	0.009	0.033	0.025	0.014	
	σ(k)	0.009	0.010	0.009	0.006	0.007	0.010	0.004	0.010	0.006	0.001	
300	*P(k)*	0.087	0.195	0.181	0.036	0.264	0.042	0.035	0.027	0.098	0.035	0.044
	μ(k)	0.008	0.014	0.006	0.009	0.002	0.009	0.009	0.010	0.006	0.009	
	σ(k)	0.002	0.004	0.001	0.003	0.001	0.003	0.003	0.003	0.002	0.003	

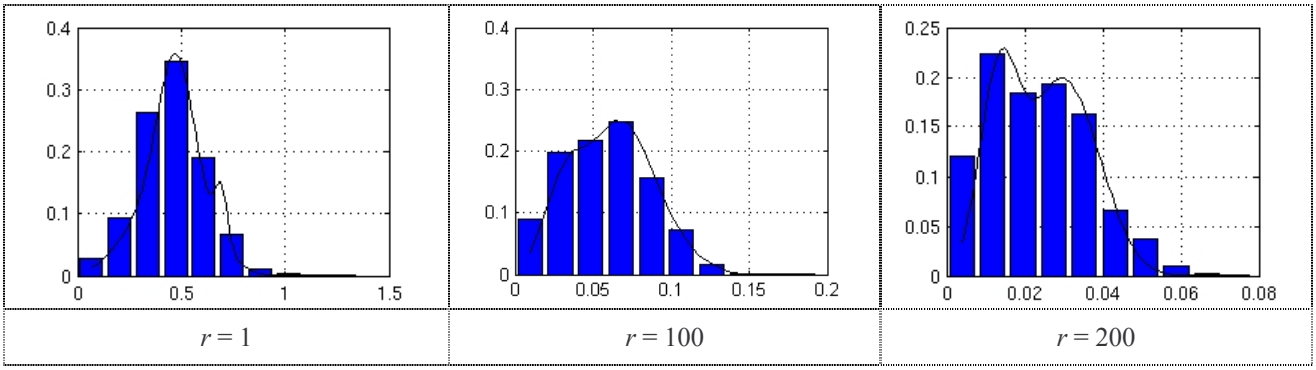
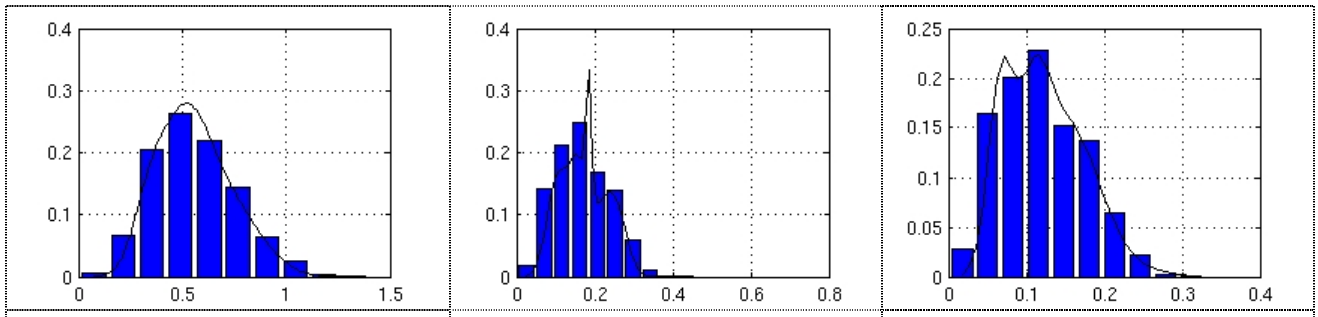


Figure III.7. MVM and histograms computed for three ranks, (9,7) DWT and low quality video.

Table III.4. MVM for the (9,7) DWT coefficient hierarchy and high quality video.

Rank	Model parameters											Error
1	*P(k)*	0.098	0.071	0.119	0.054	0.089	0.115	0.078	0.152	0.114	0.111	0.042
	μ(k)	0.454	0.709	0.508	0.682	0.791	0.527	0.598	0.302	0.435	0.696	
	σ(k)	0.121	0.175	0.120	0.178	0.158	0.120	0.141	0.089	0.117	0.017	
100	*P(k)*	0.262	0.057	0.084	0.103	0.164	0.045	0.028	0.144	0.067	0.046	0.036
	μ(k)	0.090	0.180	0.229	0.180	0.259	0.164	0.173	0.148	0.160	0.175	
	σ(k)	0.029	0.048	0.032	0.008	0.031	0.047	0.048	0.023	0.045	0.048	
200	*P(k)*	0.102	0.141	0.056	0.052	0.090	0.257	0.087	0.066	0.070	0.079	0.038
	μ(k)	0.134	0.103	0.146	0.165	0.103	0.057	0.176	0.199	0.141	0.152	
	σ(k)	0.039	0.022	0.041	0.040	0.023	0.018	0.036	0.060	0.040	0.042	
300	*P(k)*	0.071	0.124	0.079	0.068	0.014	0.125	0.065	0.174	0.148	0.132	0.037
	μ(k)	0.160	0.105	0.088	0.104	0.081	0.149	0.101	0.062	0.034	0.083	
	σ(k)	0.022	0.027	0.022	0.027	0.001	0.024	0.026	0.015	0.011	0.020	



$r = 1$	$r = 100$	$r = 200$

Figure III.8. MVM and histograms computed for three ranks, (9,7) DWT and high quality video.

The experiments in Tables III.1 – III.2 were repeated nine times (for each video sequence in the corpus) while the experiments in Tables III.3 and III.4 were repeated three times (for three random combinations of quarter of video sequences). Each and every time the procedure was applied for each investigated rank and for both compression rates, see Appendix I and the accompanying CD-ROM. As a general conclusion, it can be considered that the model computation was successful, *i.e.* that each type of video sequence allowed an accurate *pdf* estimation to be obtained:

- for all the IVM models, the *Error* values are lower than 5%, with singular exception; for instance, for the video sequence arbitrarily chosen for illustrations, only two exceptions occurred: for $r = 250$ and $r = 350$, the errors were 33.8% and 99.6%, respectively;
- for all investigated MVM cases all *Error* values are lower than 5%, with no exceptions.

III.3.2.1.2. Model Validation

The models previously obtained are now subject to validity investigation, according to the three procedures presented in the *step C* of the algorithm.

Step C-(1): Is the model dependent on the similarity measure?

The values of *Error'* (*cf.* eq. (III.6.a)) are displayed in Table III.5 for all investigated model types: IVM and MVM, low and high quality video. It can be seen that each and every time the error is lower than 10% with singular exceptions ($r = 175$ and $r = 250$). It should be also noticed that the errors corresponding to the MVM models are slightly lower that the corresponding IVM errors.

Table III.5. The average errors according to *step –C (1),* computed for low and high quality video.

Rank	Low quality video		High quality video	
	IVM	**MVM**	**IVM**	**MVM**
	Error'	*Error'*	*Error'*	*Error'*
$r = 1$	0.107	0.076	0.075	0.073
$r = 25$	0.073	0.071	0.084	0.068
$r = 50$	0.117	0.065	0.088	0.071
$r = 75$	0.091	0.084	0.081	0.067
$r = 100$	0.084	0.056	0.058	0.080
$r = 125$	0.049	0.072	0.086	0.082
$r = 150$	0.069	0.055	0.069	0.064
$r = 175$	0.280	0.048	0.057	0.070
$r = 200$	0.061	0.047	0.054	0.056
$r = 225$	0.068	0.046	0.075	0.071
$r = 250$	0.104	0.058	0.056	0.061
$r = 275$	0.048	0.051	0.051	0.057
$r = 300$	0.034	0.069	0.080	0.056
$r = 325$	0.090	0.039	0.069	0.059
$r = 350$	0.150	0.104	0.055	0.059

Step C-(2) Is the model general or does it depends on the original video sequence?

The three similarity measure values between the model and the rest of nine video sequences are computed according to (III.7.b), (III.9.a), and (III.9.b) respectively. The values thus obtained are illustrated in Tables III.6 and III.7, corresponding to low and high quality video, respectively.

Table III.6. The average errors according to *step –C (2),* computed for low quality video.

Rank	**IVM**			**MVM**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.119	0.176	0.566	0.047	0.012	0.031
$r = 25$	0.266	0.118	0.514	0.081	0.023	0.107
$r = 50$	0.187	0.185	0.558	0.126	0.040	0.326
$r = 75$	0.117	0.129	0.430	0.084	0.024	0.280
$r = 100$	0.097	0.104	0.445	0.081	0.039	0.191
$r = 125$	0.537	0.332	0.539	0.037	0.017	0.111
$r = 150$	0.061	0.079	0.252	0.030	0.020	0.077
$r = 175$	0.339	0.273	0.515	0.029	0.018	0.217
$r = 200$	0.060	0.074	0.223	0.042	0.019	0.178
$r = 225$	0.108	0.162	0.636	0.068	0.008	0.158
$r = 250$	0.044	0.069	0.276	0.041	0.024	0.378
$r = 275$	0.031	0.097	0.254	0.012	0.002	0.193
$r = 300$	0.112	0.071	0.453	0.041	0.022	0.296
$r = 325$	0.447	0.084	0.384	0.035	0.008	0.020
$r = 350$	0.220	0.040	0.190	0.050	0.001	0.050

Table III.7. The average errors according to *step –C (2),* computed for high quality video.

Rank	IVM			MVM		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.248	0.033	0.237	0.036	0.009	0.095
$r = 25$	0.256	0.090	0.484	0.108	0.033	0.225
$r = 50$	0.264	0.147	0.525	0.120	0.039	0.113
$r = 75$	0.357	0.179	0.629	0.119	0.039	0.155
$r = 100$	0.306	0.180	0.674	0.133	0.037	0.126
$r = 125$	0.698	0.235	0.901	0.128	0.039	0.151
$r = 150$	0.319	0.206	0.906	0.089	0.031	0.104
$r = 175$	0.347	0.261	0.973	0.115	0.038	0.117
$r = 200$	0.308	0.263	0.924	0.117	0.045	0.251
$r = 225$	0.281	0.229	0.881	0.099	0.032	0.204
$r = 250$	0.285	0.257	0.900	0.076	0.033	0.146
$r = 275$	0.275	0.272	0.931	0.080	0.021	0.018
$r = 300$	0.296	0.226	0.854	0.128	0.054	0.357
$r = 325$	0.500	0.235	0.856	0.100	0.062	0.265
$r = 350$	0.208	0.177	0.734	0.135	0.067	0.399

As a general conclusion, it can be noticed that:

- the IVM are not independent with respect to the video sequence: the three similarity measures are quite high and this is mainly the case for high quality video.
- for the MVM the accuracy in the model estimation is very good (the three types of computed errors being acceptable low): *Error''* was each and every time lower than 15%, the other two values are lower for E_{HL} around 0.04 and for E_{KL} around 0.3.

Hence, it can be ascertain that an accurate model for (9,7) DWT coefficients exists only when considering a video sequence combining parts of different natural videos.

Step C-(3) Is the model dependent on the EM Gaussian mixtures estimation?

The confidence limit estimation procedure was applied for $S_{conf} = 10$ evenly distributed subintervals of the $[x_{\min}; x_{\max}]$ interval and for $\alpha = 0.05$.

In practically all cases (with very few exceptions), these probabilities computed starting form the model, belong to the corresponding 95% confidence limits, as are illustrated in Figure III.9 and Figure III.10 for IVM (low and high quality rates respectively), and in Figure III.11 and Figure III.12 for MVM (low and high quality rates, respectively).

This is a very strong support for the generality of the model: the *pdf* estimation obtained by an maximum likelihood procedure proves itself as being representative also from the confidence estimation meaning.

The same numerical values are obtained for the rest of the ranks as well as for others two investigated DWTs, namely (2, 2) and (4, 4) DWT [Dum 08b].

Note: The experimental interpretation of these results was binary done, *i.e.* it was checked out whether or not the probability computed on the EM Gaussian mixture estimated model belongs to the corresponding confidence limits. However, an in-depth analysis of these plots brings into light that the precision in confidence limit estimation is low. In this respect, values as large as $\varepsilon_S = 8\%$ were sometimes obtained. When going even more into details, it was noticed that the relative error in confidence limit estimation was usually $\varepsilon_{S_r} = \varepsilon_S / \hat{p}_S = 30\%$ but situations in which $\varepsilon_{S_r} > 100\%$ are often met. These results validate our initial choice of considering the EM Gaussian mixture estimation as the main investigation tool. It is known that confidence limit estimation requires very large corpora: for instance, in our application, order to ensure values $\varepsilon_{S_r} < 10\%$, a video corpus 20 to 100 times larger is required (*i.e.* at least 10 sequences of 10 hours each), *cf.* eq. (II.63) and Figures III.9 ÷ III.12. Such a comment holds each and every time the *Step C-(3)* was considered in the thesis (*i.e.* for both natural video and attacks, for all types of encoding rates, and for all considered transforms).

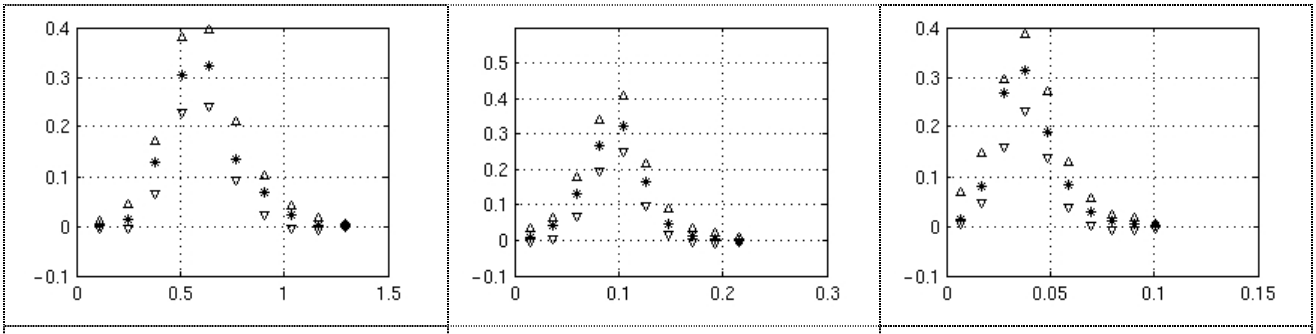


Figure III.9. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the IVM (represented in *). The results correspond to low quality video.

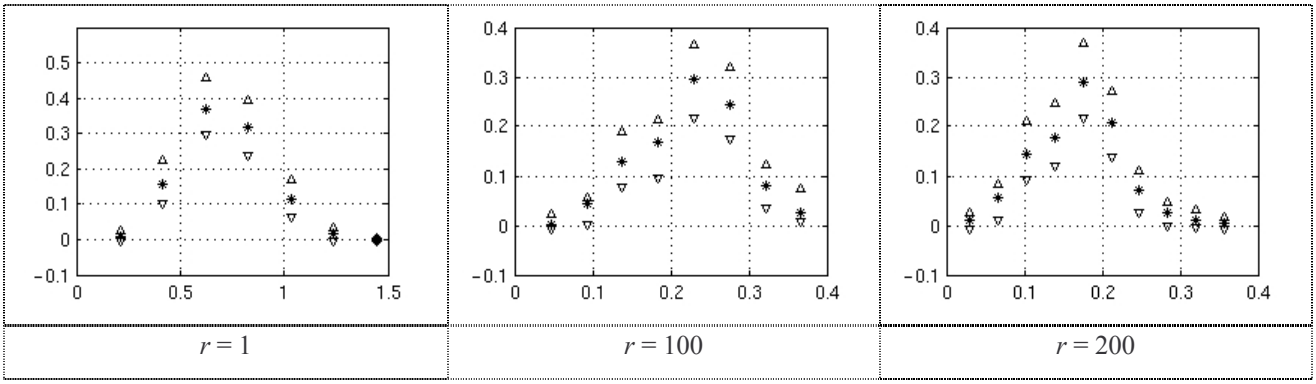


Figure III.10. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the IVM (represented in *). The results correspond to high quality video.

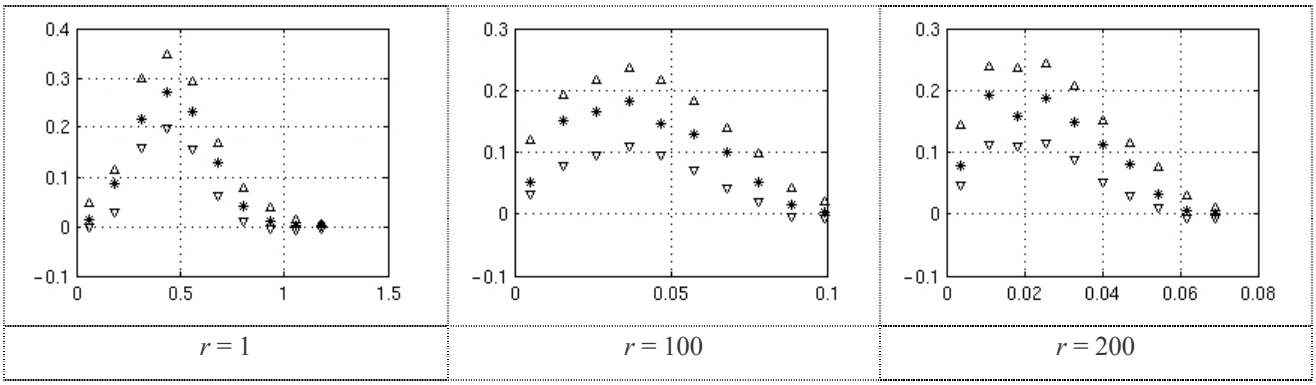


Figure III.11. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the MVM (represented in *). The results correspond to low quality video.

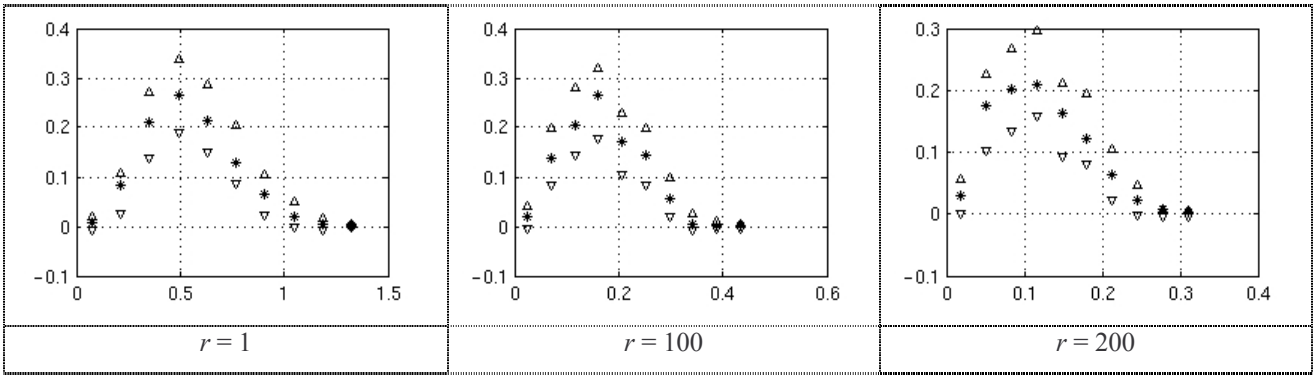


Figure III.12. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the MVM (represented in *). The results correspond to high quality video.

III.3.2.1.3. The Dependency of the Error with the Number of Gaussian Distributions in the Mixture

This section presents an *a posteriori* validation of the K number of mixtures to be included in the video modelling. In this respect, a restraint experimental study (conducted only for video sequences coded at low quality rate and (9,7) DWT) was also realized to see the impact of the K parameter in the model errors [Dum 08b].

The model in Tables III.1 was considered as reference and the errors between it and the rest of the 9 models of IVM type have been computed and then averaged.

This experiment was carried out 15 times, for different values of K = {2, 4, 6, 8, 10, 12, 14, 16, 18 20, 22, 24, 26, 28, 30}; the corresponding results are presented in Table III.8. Figure III.13 displays the error in IVM computation (see eq. (III.5)) and each rank is represented in the figure by a line with different markers and colours.

Figure III.13 and Table III.8 (with the blue colour are represent the lower *Error''* value computed with eq. (III.7.b)) shows that these values can be considered as independent with respect to the K value. Hence, it may be considered that the involved in the video modelling ($K = 10$ chosen for the all the experiments) is large enough so as not to impair the Gaussian mixture estimation approach.

Table III.8: The influence of the *K* parameter for different ranks. The values represent the errors in model computation, eq. (III.7.b).

K \ Rank	$r = 1$	$r = 100$	$r = 200$	$r = 300$
4	0.447	0.558	0.323	0.287
6	0.436	0.581	0.343	0.266
8	0.420	0.547	0.313	0.272
10	0.505	0.525	0.332	0.295
12	**0.405**	0.553	0.347	0.264
14	0.475	0.535	0.322	0.248
16	0.411	0.617	0.327	0.272
18	0.444	0.564	0.351	**0.243**
20	0.437	0.508	**0.308**	0.282
22	0.454	0.589	0.329	0.255
24	**0.405**	0.615	0.344	0.278
26	0.503	0.561	0.323	0.282
28	0.489	**0.496**	0.321	0.276
30	0.521	0.543	0.326	0.275

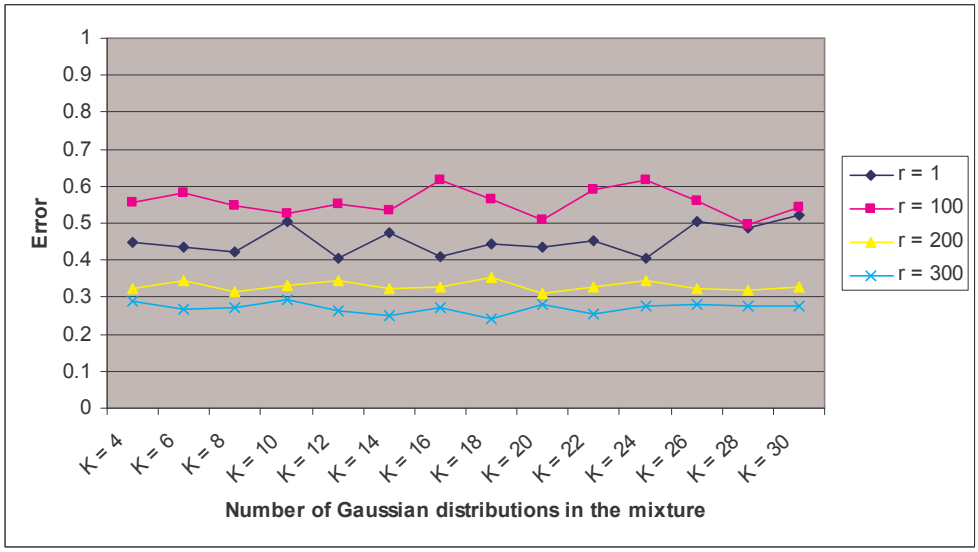


Figure III.13. The influence of the *K* parameter (the number of Gaussian distributions in the mixture) on the model computation. The values represent the errors in model computation, eq. (III.7.b).

III.3.2.2. The 2D-DCT coefficients modelling

III.3.2.2.1. Model Computation

In order to illustrate the model computation starting from an individual video sequence (IVM – Individual Video Model) coded at low quality rate, Table III.9 presents a comparison among the DCT coefficients ($r = 1$) obtained from squared blocks (4×4 and 8×8) and from whole frames.

The graphical representations of these models alongside with their corresponding histograms are presented in Figure III.14.

Table III.10 and Figure III.15 present a comparison among the models corresponding to the whole frames DCT, but to different ranks in the hierarchy: $r = 1, 100, 200, 300$.

Tables III.11–III.12 and Figures III.16 present the same type of information as Tables III.9–III.10 and Figures III.15 respectively, this time obtained out of processing high quality video sequences.

The experiments were resumed on video sequences obtained by concatenating four quarters from four different video sequences (MVM – Multiple Video Model).

The quantitative results are displayed in Tables III.13, III.14, III.15 and III.16 which are in correspondence with the Tables III.9, III.10, III.11 and III.12 respectively. Figures III.17 – III.18 contain the representation of the models with their corresponding histograms for low and high quality video, respectively.

Table III.9. IVM for the DCT coefficient hierarchy and low quality video.

Rank $r = 1$	**Model parameters**											**Error**
4x4 blocks DCT	*P(k)*	0.062	0.089	0.132	0.136	0.078	0.110	0.166	0.065	0.090	0.070	0.077
	μ(k)	1.184	3.759	1.118	0.353	2.371	2.895	0.887	1.983	1.430	2.047	
	σ(k)	0.561	0.165	0.413	0.214	0.545	0.372	0.263	0.581	0.444	0.583	
8x8 blocks DCT	*P(k)*	0.106	0.089	0.113	0.123	0.099	0.071	0.083	0.073	0.135	0.108	0.065
	μ(k)	4.828	4.152	4.998	6.722	2.926	2.479	2.642	3.758	1.019	1.865	
	σ(k)	1.189	1.158	1.189	0.959	0.955	0.877	0.872	1.201	0.634	0.699	
Whole frames DCT	*P(k)*	0.065	0.129	0.065	0.177	0.114	0.099	0.129	0.071	0.064	0.085	0.026
	μ(k)	5.646	3.361	5.951	4.817	4.717	4.927	8.096	5.710	6.411	5.215	
	σ(k)	1.141	6.894	1.258	5.233	9.309	1.027	9.070	1.170	1.323	8.906	

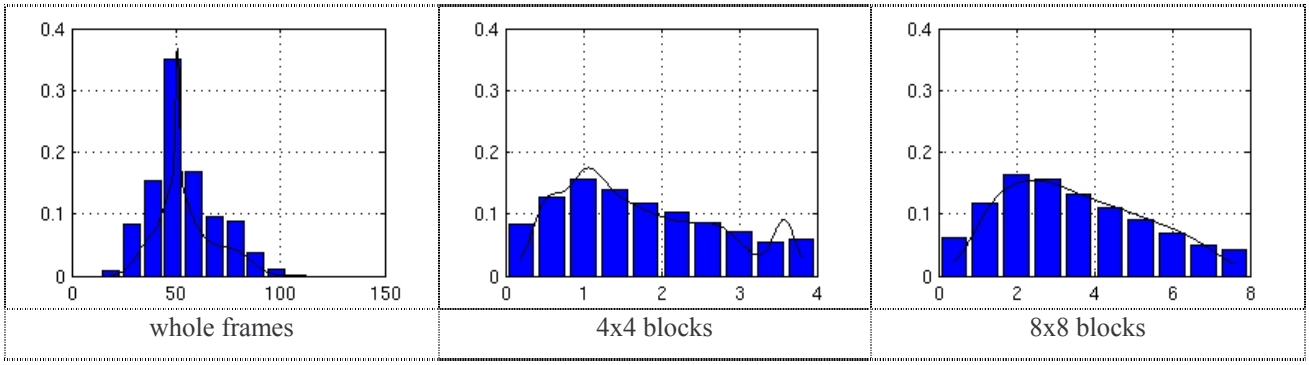
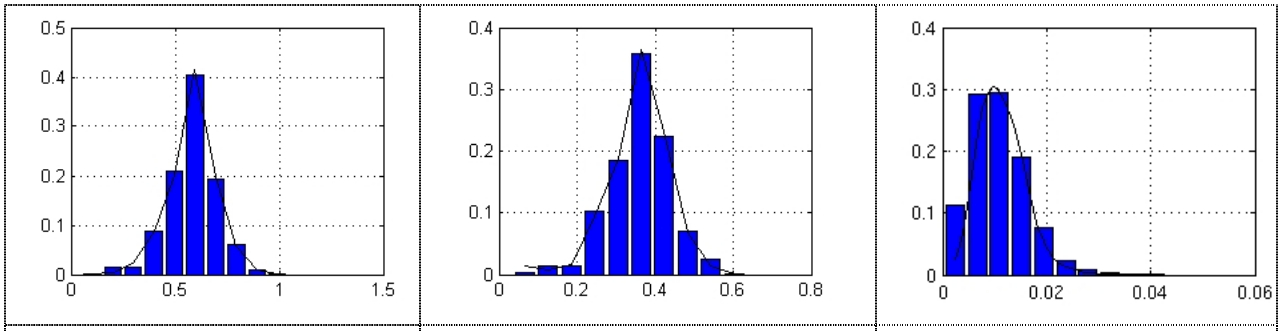


Figure III.14. IVM and histograms computed for rank $r = 1$ in the DCT, low quality video.

Table III.10. IVM for the DCT (applied on the whole frames) coefficient hierarchy and low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.065	0.129	0.065	0.177	0.114	0.099	0.129	0.071	0.064	0.085	0.026
	$\mu(k)$	5.646	3.361	5.951	4.817	4.717	4.927	8.096	5.710	6.411	5.215	
	$\sigma(k)$	1.141	6.894	1.258	5.233	9.309	1.027	9.070	1.170	1.323	8.906	
100	$P(k)$	0.098	0.088	0.095	0.103	0.055	0.111	0.077	0.073	0.209	0.091	0.030
	$\mu(k)$	0.670	0.567	0.568	0.603	0.546	0.532	0.450	0.567	0.603	0.680	
	$\sigma(k)$	0.103	0.120	0.118	0.108	0.124	0.079	0.114	0.121	0.033	0.064	
200	$P(k)$	0.093	0.171	0.121	0.119	0.126	0.021	0.066	0.102	0.160	0.034	0.036
	$\mu(k)$	0.382	0.369	0.386	0.418	0.366	0.089	0.376	0.386	0.271	0.353	
	$\sigma(k)$	0.069	0.031	0.034	0.062	0.048	0.022	0.069	0.069	0.040	0.068	
300	$P(k)$	0.121	0.122	0.106	0.059	0.038	0.035	0.110	0.150	0.222	0.037	0.033
	$\mu(k)$	0.296	0.259	0.315	0.308	0.286	0.082	0.265	0.188	0.274	0.288	
	$\sigma(k)$	0.016	0.035	0.060	0.061	0.055	0.030	0.038	0.021	0.034	0.055	



$r = 100$ | $r = 200$ | $r = 300$

Figure III.15. IVM and histograms computed for three ranks in the DCT (applied on the whole frames), low quality video.

Table III.11. IVM for the DCT coefficient hierarchy and high quality video.

Rank $r = 1$	Model parameters											Error
4x4 blocks DCT	$P(k)$	0.169	0.030	0.021	0.054	0.379	0.020	0.146	0.069	0.036	0.075	0.018
	$\mu(k)$	0.038	0.301	0.241	0.188	0.012	0.417	0.072	0.122	0.712	0.115	
	$\sigma(k)$	0.013	0.026	0.062	0.019	0.007	0.036	0.015	0.037	0.063	0.030	
8x8 blocks DCT	$P(k)$	0.365	0.086	0.059	0.063	0.058	0.014	0.160	0.077	0.058	0.057	0.022
	$\mu(k)$	0.058	0.228	0.436	0.843	0.452	1.732	0.158	0.311	0.213	0.452	
	$\sigma(k)$	0.029	0.057	0.120	0.217	0.117	0.185	0.060	0.084	0.065	0.117	
whole frames DCT	$P(k)$	0.111	0.047	0.093	0.073	0.107	0.111	0.081	0.059	0.112	0.207	0.031
	$\mu(k)$	9.197	127.23	154.61	145.82	158.72	169.18	165.88	143.27	169.84	243.29	
	$\sigma(k)$	11.277	21.180	25.140	20.254	26.491	29.053	38.436	19.121	29.188	36.449	

Table III.12. IVM for the DCT (applied on the whole frames) coefficient hierarchy and high quality video.

Rank	Model parameters											Error
1	*P(k)*	0.111	0.047	0.093	0.073	0.107	0.111	0.081	0.059	0.112	0.207	0.031
	μ(k)	9.197	127.23	154.61	145.82	158.72	169.18	165.88	143.27	169.84	243.29	
	σ(k)	11.277	21.180	25.140	20.254	26.491	29.053	38.436	19.121	29.188	36.449	
100	*P(k)*	0.075	0.093	0.118	0.221	0.065	0.090	0.113	0.078	0.067	0.080	0.029
	μ(k)	2.420	2.112	1.657	2.006	2.327	1.864	1.599	1.962	1.673	1.988	
	σ(k)	0.532	0.111	0.277	0.143	0.303	0.398	0.280	0.414	0.332	0.413	
200	*P(k)*	0.096	0.090	0.100	0.084	0.178	0.009	0.183	0.095	0.110	0.054	0.032
	μ(k)	1.224	1.505	1.257	1.137	1.307	2.089	1.318	0.869	1.164	1.690	
	σ(k)	0.197	0.255	0.184	0.237	0.090	0.219	0.163	0.101	0.226	0.086	
300	*P(k)*	0.037	0.029	0.095	0.118	0.102	0.182	0.078	0.097	0.143	0.119	0.033
	μ(k)	0.614	1.047	0.860	0.865	0.981	1.004	0.927	1.286	0.777	0.891	
	σ(k)	0.006	0.038	0.191	0.192	0.080	0.066	0.207	0.133	0.161	0.199	

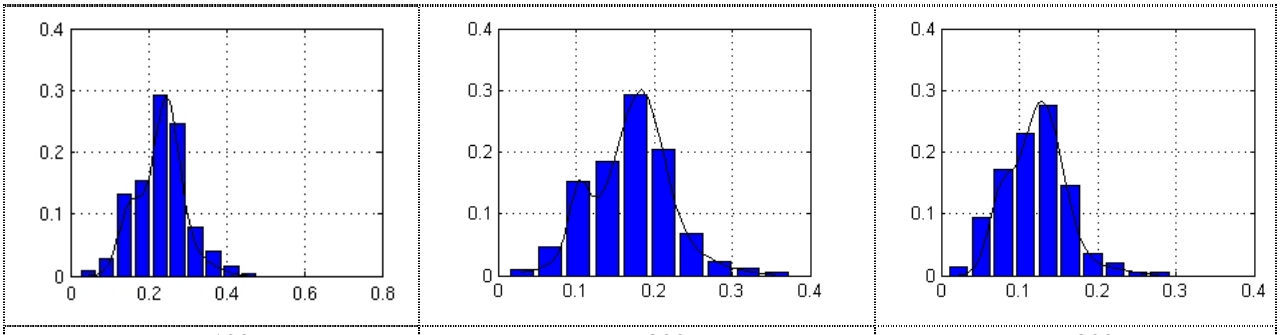


Figure III.16. IVM and histograms computed for three ranks in the DCT (applied on the whole frames), high quality video.

Table III.13. MVM for the DCT coefficient hierarchy and low quality video.

Rank $r = 1$	Model parameters										Error
4x4 blocks DCT	*P(k)*	0.042	0.381	0.116	0.102	0.123	0.014	0.194	0.007	0.007	0.014	0.005
	μ(k)	0.330	0.020	0.070	0.080	0.003	1.054	0.067	0.776	0.654	2.632	
	σ(k)	0.083	0.023	0.113	0.113	0.002	0.108	0.046	0.001	0.001	0.389	
8x8 blocks DCT	*P(k)*	0.165	0.195	0.170	0.044	0.198	0.065	0.007	0.056	0.084	0.016	0.007
	μ(k)	0.188	0.001	0.088	0.522	0.158	0.012	1.199	0.663	0.077	3.701	
	σ(k)	0.153	0.003	0.073	0.223	0.149	0.004	0.001	0.176	0.019	2.081	
Whole frames DCT	*P(k)*	0.106	0.096	0.112	0.033	0.177	0.081	0.119	0.092	0.031	0.152	0.030
	μ(k)	4.364	4.761	4.490	2.333	1.516	4.704	4.437	5.011	2.727	3.475	
	σ(k)	8.410	9.726	8.300	1.439	2.999	9.528	8.387	1.021	1.546	5.113	

Table III.14. MVM for the DCT (applied on the whole frames) coefficient hierarchy and low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.106	0.096	0.112	0.033	0.177	0.081	0.119	0.092	0.031	0.152	0.030
	$\mu(k)$	4.364	4.761	4.490	2.333	1.516	4.704	4.437	5.011	2.727	3.475	
	$\sigma(k)$	8.410	9.726	8.300	1.439	2.999	9.528	8.387	1.021	1.546	5.113	
100	$P(k)$	0.007	0.040	0.164	0.036	0.043	0.114	0.083	0.113	0.148	0.252	0.044
	$\mu(k)$	1.025	0.616	0.346	0.610	0.508	0.445	0.506	0.675	0.425	0.298	
	$\sigma(k)$	0.001	0.021	0.106	0.008	0.099	0.109	0.099	0.064	0.076	0.092	
200	$P(k)$	0.097	0.168	0.062	0.080	0.108	0.053	0.108	0.165	0.081	0.078	0.034
	$\mu(k)$	0.222	0.385	0.456	0.126	0.218	0.296	0.276	0.212	0.295	0.253	
	$\sigma(k)$	0.060	0.033	0.105	0.042	0.060	0.080	0.076	0.061	0.079	0.068	
300	$P(k)$	0.107	0.196	0.217	0.026	0.072	0.138	0.007	0.071	0.055	0.111	0.034
	$\mu(k)$	0.158	0.296	0.126	0.393	0.220	0.153	0.203	0.220	0.218	0.209	
	$\sigma(k)$	0.047	0.033	0.044	0.070	0.051	0.047	0.001	0.051	0.051	0.053	

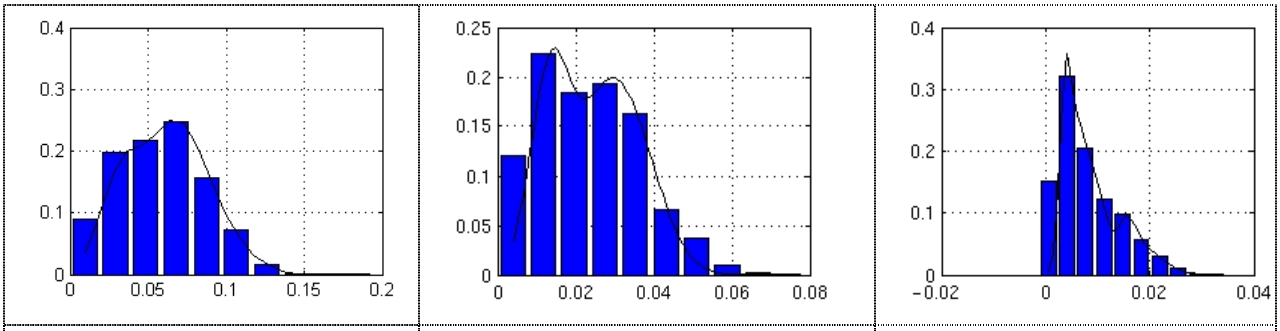


Figure III.17. MVM and histograms computed for three ranks in the DCT (applied on the whole frames), low quality video.

Table III.15. MVM for the DCT coefficient hierarchy and high quality video.

Rank $r = 1$	Model parameters											Error
4x4 blocks DCT	$P(k)$	0.136	0.038	0.011	0.015	0.127	0.071	0.007	0.134	0.284	0.175	0.007
	$\mu(k)$	0.024	0.196	0.359	2.757	0.093	0.309	-0.192	0.049	0.024	0.003	
	$\sigma(k)$	0.079	0.020	0.025	1.149	0.044	0.143	0.001	0.086	0.019	0.005	
8x8 blocks DCT	$P(k)$	0.172	0.179	0.016	0.166	0.046	0.162	0.081	0.071	0.029	0.077	0.006
	$\mu(k)$	0.125	0.119	1.758	0.002	0.736	0.178	0.089	0.001	0.812	0.547	
	$\sigma(k)$	0.113	0.111	0.367	0.006	0.140	0.118	0.009	0.182	0.370	0.039	
whole frames DCT	$P(k)$	0.084	0.098	0.021	0.112	0.086	0.129	0.096	0.126	0.142	0.106	0.029
	$\mu(k)$	56.025	156.37	264.42	145.93	169.81	130.69	131.01	130.99	127.06	50.152	
	$\sigma(k)$	4.466	30.703	8.931	32.519	28.145	25.662	34.621	25.528	25.151	14.989	

Table III.16. MVM for the DCT (applied on the whole frames) coefficient hierarchy and high quality video.

Rank	Model parameters											Error
1	$P(k)$	0.084	0.098	0.021	0.112	0.086	0.129	0.096	0.126	0.142	0.106	0.029
	$\mu(k)$	56.025	156.37	264.42	145.93	169.81	130.69	131.01	130.99	127.06	50.152	
	$\sigma(k)$	4.466	30.703	8.931	32.519	28.145	25.662	34.621	25.528	25.151	14.989	
100	$P(k)$	0.057	0.067	0.101	0.076	0.121	0.157	0.132	0.104	0.090	0.094	0.037
	$\mu(k)$	2.171	2.060	1.843	1.587	1.189	1.057	1.157	1.746	1.729	1.444	
	$\sigma(k)$	0.366	0.055	0.392	0.359	0.324	0.295	0.317	0.397	0.401	0.363	
200	$P(k)$	0.125	0.079	0.089	0.144	0.085	0.116	0.071	0.097	0.106	0.088	0.037
	$\mu(k)$	0.915	0.631	0.908	0.687	0.879	0.757	1.484	1.304	0.827	1.183	
	$\sigma(k)$	0.279	0.222	0.281	0.218	0.279	0.265	0.295	0.080	0.276	0.328	
300	$P(k)$	0.093	0.108	0.029	0.087	0.089	0.165	0.117	0.124	0.165	0.020	0.036
	$\mu(k)$	0.723	0.828	1.089	0.987	1.203	0.575	0.781	0.676	0.404	0.813	
	$\sigma(k)$	0.159	0.174	0.002	0.065	0.216	0.095	0.172	0.148	0.092	0.077	

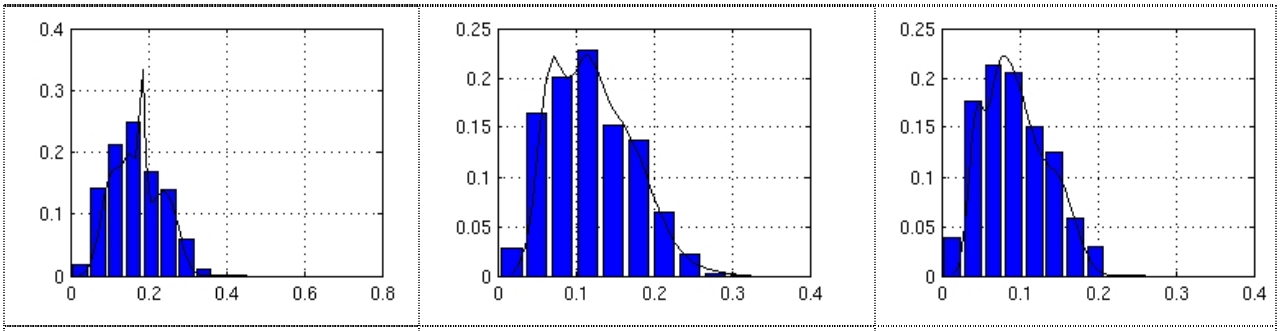


Figure III.18. MVM and histograms computed for three ranks in the DCT (applied on whole frames), high quality video.

The experiments illustrated in Tables III.9 ÷ III.16 were repeated nine times (for each and every video sequence in the corpus) for IVM case and three times for the MVM case (for three random combinations of quarters of four video sequences). The experiments were conducted for each investigated rank, and for two compression rates (see Appendix I and the accompanying CD-ROM).
As a general conclusion, it can be considered that the model computation was successful, *i.e.* that each type of video sequence allowed an accurate *pdf* estimation to be obtained.

The *Error* values for the video coded at low quality rate are:

- 4×4 blocks: for IVM case the *Error*s is about 2% with one exception, $r = 1$ when the values is 7.7%; for MVM case all the *Error*s are lower than 1%, with no exception.
- 8×8 blocks: for IVM the *Error*s are about 2% with one exception $r = 1$ (6.5%) while the MVM are obtained with errors as low as 1%.
- whole frames: for booth cases (IVM and MVM) the *Error*s are higher than the previous two cases but lower than 4%.

The *Error* values for the video coded at high quality rate are:

- 4×4 blocks: for IVM the *Error*s are about 2.3% with no exception, while for MVM the *Error*s are lower than 1.5%.
- 8×8 blocks: the error upper limits are 2.4% for IVM and 1% for MVM, with no exception.
- whole frames: for IVM the *Error*s are lower than 3.7% and for MVM the *Error*s are lower than 4%.

III.3.2.2.2. Model Validation

The models previously obtained are now subject to validity investigation, according to the three procedures presented in the *steps C* of the algorithm.

Step C-(1): Is the model dependent on the similarity measure?

The values of *Error'* for all investigated model types (IVM and MVM) and for low and high quality video, respectively are displayed in Table III.17 and III.18. It can be seen that each and every time the error is lower than 10% with some exceptions (when the DCT was applied on blocks). It should be also noticed that the errors corresponding to the MVM models are slightly lower than the corresponding IVM errors with two exceptions for $r = 1$, when the DCT was applied on blocks.

Table III.17. The average errors according to *step-C(1)*, computed for low quality video.

Rank	IVM			MVM		
	Error''			*Error''*		
	whole frames	4x4 blocks	8x8 blocks	whole frames	4x4 blocks	8x8 blocks
$r = 1$	0.074	0.100	0.084	0.078	0.265	0.307
$r = 2$	0.053	0.135	0.211	0.044	0.071	0.210
$r = 3$	0.067	0.219	0.210	0.061	0.093	0.168
$r = 4$	0.064	0.172	0.143	0.061	0.089	0.124
$r = 5$	0.062	0.191	0.204	0.055	0.117	0.139
$r = 25$	0.080			0.074		
$r = 50$	0.089			0.085		
$r = 75$	0.090			0.100		
$r = 100$	0.078			0.078		
$r = 125$	0.080			0.073		
$r = 150$	0.069			0.060		
$r = 175$	0.078			0.075		
$r = 200$	0.070			0.100		
$r = 225$	0.087			0.084		
$r = 250$	0.069			0.067		
$r = 275$	0.089			0.064		
$r = 300$	0.073			0.067		
$r = 325$	0.075			0.065		
$r = 350$	0.077			0.066		

Table III.18. The average errors according to *step-C(1)*, computed for high quality video.

Rank	IVM			MVM		
	Error''			*Error''*		
	whole frames	4x4 blocks	8x8 blocks	whole frames	4x4 blocks	8x8 blocks
$r = 1$	0.089	0.105	0.092	0.071	0.250	0.297
$r = 2$	0.059	0.139	0.221	0.054	0.092	0.207
$r = 3$	0.046	0.200	0.212	0.044	0.100	0.173
$r = 4$	0.060	0.145	0.148	0.056	0.096	0.131
$r = 5$	0.048	0.198	0.210	0.049	0.123	0.145
$r = 25$	0.067			0.059		
$r = 50$	0.064			0.069		
$r = 75$	0.085			0.075		
$r = 100$	0.066			0.070		
$r = 125$	0.101			0.079		
$r = 150$	0.083			0.081		
$r = 175$	0.097			0.087		
$r = 200$	0.082			0.069		
$r = 225$	0.095			0.092		
$r = 250$	0.165			0.073		
$r = 275$	0.079			0.066		
$r = 300$	0.107			0.106		
$r = 325$	0.096			0.076		
$r = 350$	0.090			0.068		

Step C-(2) Is the model general or does it depends on the original video sequence?

The similarity measure between the model and the rest of 9 video sequences are computed according to (III.7.b). The results are reported in Tables III.19 (low quality video sequences) and III.20 (high quality video sequences). Each of these two tables presents information for both IVM (the three left columns) and MVM (the three right columns). We emphasise that the Table III.19 and III.20 contains only the values corresponding to the *Error*'', eq. (III.7.b). It can be seen that acceptable lower values are obtained only for MVM for 4×4 and for 8×8 blocks. For the rest of the cases (namely for the three types of IVMs and for the MVM corresponding to the whole frames DCT) the error values are very high. This means that these models depend on the investigated video sequence (be it obtained from an individual video or from multiple videos) and cannot be accepted as representing all the video sequences. Note that as *Error''* was large enough, there was no need to compute also the Hellinger distance and the Kullback-Leibler divergence, *cf*. flowchart *step C-(2)*.

Table III.19. The average errors according to *step-C(2)*, computed for low quality video.

Rank	IVM			MVM		
	Error''			*Error''*		
	whole frames	4x4 blocks	8x8 blocks	whole frames	4x4 blocks	8x8 blocks
$r = 1$	0.579	0.995	0.992	0.678	0.004	0.022
$r = 2$	0.187	0.214	0.498	0.658	0.001	0.012
$r = 3$	0.237	0.381	0.450	0.312	0.001	0.001
$r = 4$	0.295	0.008	0.091	0.419	0.002	0.012
$r = 5$	0.387	0.508	0.153	0.399	0.003	0.043
$r = 25$	0.685			0.876		
$r = 50$	0.902			0.953		
$r = 75$	0.913			0.799		
$r = 100$	0.945			0.661		
$r = 125$	0.959			0.989		
$r = 150$	0.973			0.967		
$r = 175$	0.980			0.891		
$r = 200$	0.901			0.783		
$r = 225$	0.878			0.711		
$r = 250$	0.852			0.845		
$r = 275$	0.876			0.844		
$r = 300$	0.843			0.604		
$r = 325$	0.799			0.382		
$r = 350$	0.836			0.257		

Table III.20. The average errors according to *step-C(2)*, computed for high quality video.

Rank	IVM			MVM		
	Error''			*Error''*		
	whole frames	4x4 blocks	8x8 blocks	whole frames	4x4 blocks	8x8 blocks
$r = 1$	0.668	0.991	0.997	0.893	0.010	0.013
$r = 2$	0.269	0.230	0.524	0.790	0.092	0.006
$r = 3$	0.326	0.393	0.477	0.568	0.047	0.001
$r = 4$	0.437	0.013	0.121	0.731	0.004	0.012
$r = 5$	0.491	0.524	0.160	0.651	0.001	0.001
$r = 25$	0.808			0.899		
$r = 50$	0.897			0.973		
$r = 75$	0.912			0.869		
$r = 100$	0.957			0.721		
$r = 125$	0.901			0.964		
$r = 150$	0.906			0.919		
$r = 175$	0.988			0.871		
$r = 200$	0.973			0.661		
$r = 225$	0.965			0.738		
$r = 250$	0.970			0.981		
$r = 275$	0.914			0.932		
$r = 300$	0.994			0.661		
$r = 325$	0.830			0.446		
$r = 350$	0.879			0.295		

III.3.2.2.3. Define a General Model

Tables III.19 and III.20 bring into light the fact that the errors in the MVMs cases are significantly lower than the corresponding errors in the IVMs cases. This would suggest that when trying to define a general model, the MVM principle should be preserved. Moreover, as each individual video sequence featured a very nice inner stationarity (see all the validation procedures reported in Section. III.3.2.2.2) we shall go deeper into coefficient analysis by defining a new type of video sequence to be modelled. This new type is referred to as MSVM – Multiple Shuffled Video Model and is obtained by concatenating four quarters from four different video sequences previously shuffled.

In order to illustrate the MSVM computation, Tables III.21 and III.23 present the comparison among the DCT applied on blocks and on the whole frames, for both types of quality rates.

Tables III.22 and III.24 show a comparison among the models corresponding to the whole frames for four ranks. The results for all investigated ranks are detailed in Appendix I and on the accompanying CD-ROM.

Table III.21. MSVM for the DCT coefficient hierarchy and low quality video.

Rank $r = 1$	**Model parameters**											**Error**
4x4 blocks DCT	$P(k)$	0.029	0.006	0.135	0.079	0.009	0.074	0.014	0.021	0.007	0.625	0.011
	$\mu(k)$	0.029	0.076	0.004	0.009	0.086	0.010	0.037	0.053	0.024	0.001	
	$\sigma(k)$	0.001	0.009	0.002	0.004	0.010	0.004	0.002	0.003	0.001	0.001	
8x8 blocks DCT	$P(k)$	0.088	0.008	0.265	0.014	0.198	0.037	0.027	0.017	0.254	0.093	0.014
	$\mu(k)$	0.080	0.180	0.003	0.298	0.028	0.184	0.145	0.283	0.010	0.069	
	$\sigma(k)$	0.023	0.052	0.002	0.071	0.010	0.028	0.002	0.072	0.005	0.025	
whole frames DCT	$P(k)$	0.047	0.384	0.280	0.013	0.055	0.082	0.001	0.053	0.029	0.056	0.004
	$\mu(k)$	3.336	0.141	0.141	2.348	4.533	5.063	0.149	7.526	5.011	5.745	
	$\sigma(k)$	3.524	0.058	0.058	1.469	1.930	3.968	0.058	9.989	8.723	7.853	

Table III.22. MSVM for the DCT (applied on the whole frames) coefficient hierarchy and low quality video.

Rank	**Model parameters**											**Error**
1	$P(k)$	0.047	0.384	0.280	0.013	0.055	0.082	0.001	0.053	0.029	0.056	0.004
	$\mu(k)$	3.336	0.141	0.141	2.348	4.533	5.063	0.149	7.526	5.011	5.745	
	$\sigma(k)$	3.524	0.058	0.058	1.469	1.930	3.968	0.058	9.989	8.723	7.853	
100	$P(k)$	0.027	0.074	0.047	0.214	0.115	0.067	0.054	0.161	0.025	0.216	0.058
	$\mu(k)$	0.782	0.205	0.434	0.187	0.195	0.079	0.568	0.186	0.255	0.596	
	$\sigma(k)$	0.033	0.057	0.108	0.057	0.058	0.007	0.075	0.056	0.078	0.060	
200	$P(k)$	0.148	0.047	0.081	0.138	0.090	0.115	0.089	0.070	0.106	0.116	0.057
	$\mu(k)$	0.154	0.302	0.278	0.372	0.396	0.332	0.268	0.351	0.265	0.313	
	$\sigma(k)$	0.040	0.087	0.066	0.021	0.076	0.092	0.044	0.089	0.042	0.091	
300	$P(k)$	0.009	0.063	0.080	0.119	0.121	0.126	0.142	0.105	0.139	0.095	0.056
	$\mu(k)$	0.055	0.517	0.446	0.218	0.670	0.281	0.280	0.471	0.643	0.345	
	$\sigma(k)$	0.001	0.126	0.087	0.045	0.118	0.018	0.063	0.058	0.045	0.054	

Table III.23. MSVM for the DCT coefficient hierarchy and high quality video.

Rank $r = 1$	**Model parameters**											**Error**
4x4 blocks DCT	$P(k)$	0.042	0.055	0.038	0.359	0.016	0.059	0.025	0.199	0.155	0.052	0.005
	$\mu(k)$	0.008	0.022	0.045	0.001	0.002	0.008	0.007	0.000	0.004	0.006	
	$\sigma(k)$	0.004	0.005	0.013	0.001	0.011	0.004	0.012	0.001	0.001	0.004	
8x8 blocks DCT	$P(k)$	0.183	0.047	0.420	0.034	0.037	0.055	0.007	0.007	0.192	0.017	0.014
	$\mu(k)$	0.019	0.066	0.005	0.107	0.135	0.161	0.051	0.302	0.034	0.081	
	$\sigma(k)$	0.009	0.001	0.004	0.004	0.029	0.034	0.018	0.001	0.011	0.007	
whole frames DCT	$P(k)$	0.206	0.105	0.122	0.040	0.076	0.085	0.108	0.092	0.096	0.069	0.057
	$\mu(k)$	0.818	0.437	0.538	0.842	0.295	0.615	0.408	0.638	0.911	0.614	
	$\sigma(k)$	0.078	0.153	0.098	0.236	0.033	0.141	0.147	0.172	0.223	0.165	

Table III.24. MSVM for the DCT (applied on the whole frames) coefficient hierarchy and high quality video.

Rank	**Model parameters**											**Error**
1	$P(k)$	0.206	0.105	0.122	0.040	0.076	0.085	0.108	0.092	0.096	0.069	0.057
	$\mu(k)$	0.818	0.437	0.538	0.842	0.295	0.615	0.408	0.638	0.911	0.614	
	$\sigma(k)$	0.078	0.153	0.098	0.236	0.033	0.141	0.147	0.172	0.223	0.165	
100	$P(k)$	0.081	0.074	0.114	0.100	0.123	0.110	0.164	0.065	0.064	0.100	0.058
	$\mu(k)$	0.669	0.688	0.915	0.615	0.905	0.584	1.094	0.530	0.323	0.636	
	$\sigma(k)$	0.220	0.215	0.152	0.235	0.156	0.237	0.128	0.232	0.057	0.109	
200	$P(k)$	0.105	0.099	0.124	0.080	0.184	0.039	0.055	0.095	0.069	0.149	0.058
	$\mu(k)$	0.873	1.018	0.973	1.079	1.482	1.550	1.569	1.127	0.493	0.821	
	$\sigma(k)$	0.244	0.267	0.265	0.257	0.1409	0.427	0.423	0.239	0.046	0.228	
300	$P(k)$	0.146	0.084	0.096	0.091	0.047	0.112	0.091	0.097	0.113	0.122	0.046
	$\mu(k)$	1.901	1.936	2.077	2.339	2.835	1.823	2.124	2.722	2.119	1.079	
	$\sigma(k)$	0.277	0.839	0.808	0.724	0.256	0.432	0.794	0.257	0.416	0.244	

III.3.2.2.4. General Model Validation

This section concludes the DCT modelling investigation, presenting the result of the validation methods (*steps C* of the *ART.MOD-V* algorithm) applied to the MSVM. The experiments were repeated three times (for three random combinations of four quarters from four different video sequences previously shuffled).

Step C-(1): Is the model dependent on the similarity measure?

The values of *Error'* are presented in Table III.25 for all investigated DCT (whole frames and 4×4 or 8×8 blocks) for two types of video compression. In the MSVM case, the error is lower than 15% with a singular exceptions ($r=300$ for the whole frames; $r=1$ and $r=5$ for 4×4 blocks; $r=1$, $r=4$ and $r=5$ for 8×8 blocks). Hence, the values *Error'* can be considered as a support for the independency of the model with respect to the similarity measure.

Table III.25. The average errors according to *step-C (1),* in the MSVM case, low and high quality video.

Rank	Low quality video			High quality video		
	Error'			*Error'*		
	whole frame	**4x4 blocks**	**8x8 blocks**	**whole frame**	**4x4 blocks**	**8x8 blocks**
$r = 1$	0.028	0.290	0.169	0.101	0.200	0.146
$r = 2$	0.026	0.079	0.151	0.102	0.087	0.148
$r = 3$	0.127	0.144	0.139	0.198	0.146	0.147
$r = 4$	0.037	0.153	0.283	0.096	0.135	0.189
$r = 5$	0.138	0.242	0.191	0.098	0.235	0.200
$r = 25$	0.062			0.161		
$r = 50$	0.135			0.098		
$r = 75$	0.115			0.109		
$r = 100$	0.100			0.135		
$r = 125$	0.108			0.109		
$r = 150$	0.104			0.103		
$r = 175$	0.099			0.120		
$r = 200$	0.096			0.117		
$r = 225$	0.092			0.126		
$r = 250$	0.093			0.103		
$r = 275$	0.116			0.093		
$r = 300$	0.281			0.092		
$r = 325$	0.108			0.071		
$r = 350$	0.080			0.067		

Step C-(2) Is the model general or does it depends on the original video sequence?

The values for the validation method - *step-C(2)* between any 2 models out of the 3 MSVM models thus obtained are computed according to eq. (III.7.a), eq. (III.8.a) – eq. (III.8.b) and then the values thus obtained are averaged (*Error''*, E_{HL} and E_{KL}), see Tables III.26 and III.27.

A general conclusion that can be drawn from the values reported in Tables III.26 and III.27 is that the shuffling of the original content significantly improves the precision in *pdf* estimation:

- for 4×4 or 8×8 blocks: the *Error''* in MSVM case is lower than 5% while in the MVM case it was lower than 10%;
- whole frames: the *Error''* upper limit in the MSVM case is lower than 40%, with singular exceptions ($r = 125$ and $r = 275$ – for video coded at low quality rate and $r = 75$, $r = 275$ and $r = 300$ – for video coded at high rate); in the MVM case this upper limit was as high as 90%.

However, we may speak about a general model, independent with respect to the video sequence on which it was computed, only for the DCT coefficients corresponding to square blocks. For the whole frames DCT, no accurate theoretical model can be defined for each and every rank in the hierarchy.

Table III.26. The average errors according to *step-C (2),* in the MSVM case, low and high quality video.

Rank	Low quality video					
	4x4 blocks			**8x8 blocks**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.013	0.010	0.107	0.005	0.004	0.040
$r = 2$	0.019	0.010	0.109	0.033	0.017	0.232
$r = 3$	0.004	0.005	0.086	0.005	0.014	0.085
$r = 4$	0.007	0.012	0.172	0.009	0.006	0.027
$r = 5$	0.001	0.014	0.396	0.005	0.015	0.135
Rank	High quality video					
	4x4 blocks			**8x8 blocks**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.047	0.243	0.011	0.005	0.001	0.015
$r = 2$	0.004	0.050	0.007	0.015	0.032	0.005
$r = 3$	0.015	0.016	0.007	0.009	0.072	0.011
$r = 4$	0.008	0.025	0.010	0.008	0.214	0.019
$r = 5$	0.002	0.105	0.012	0.007	0.034	0.006

Table III.27. The average errors according to *step-C (2),* in the MSVM case, low and high quality video.

Rank	Low quality video			High quality video		
	whole frames			**whole frames**		
	Error''	*Error''*	E_{HL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.301	0.122	0.378	0.347	0.030	0.312
$r = 2$	0.282	0.119	0.390	0.338	0.037	0.314
$r = 3$	0.284	0.057	0.172	0.394	0.035	0.279
$r = 4$	0.286	0.020	0.069	0.421	0.045	0.304
$r = 5$	0.318	0.003	0.009	0.286	0.034	0.307
$r = 25$	0.450	0.036	0.142	0.393	0.028	0.308
$r = 50$	0.398	0.067	0.477	0.368	0.017	0.267
$r = 75$	0.413	0.138	0.448	0.503	0.019	0.239
$r = 100$	0.308	0.293	0.589	0.458	0.022	0.197
$r = 125$	0.507	0.352	0.449	0.096	0.004	0.016
$r = 150$	0.432	0.216	0.310	0.137	0.003	0.009
$r = 175$	0.182	0.129	0.615	0.194	0.019	0.035
$r = 200$	0.154	0.106	0.449	0.179	0.028	0.001
$r = 225$	0.168	0.103	0.396	0.312	0.063	0.011
$r = 250$	0.322	0.205	0.598	0.423	0.020	0.073
$r = 275$	0.528	0.225	0.409	0.551	0.080	0.281
$r = 300$	0.283	0.212	0.580	0.536	0.142	0.631
$r = 325$	0.198	0.010	0.041	0.159	0.487	0.834
$r = 350$	0.203	0.320	0.432	0.143	0.362	0.926

Step C-(3) Is the model dependent on the EM Gaussian mixtures estimation?

The confidence limit estimation procedure was applied for $S_{conf} = 10$ evenly distributed subintervals of the $[x_{\min}; x_{\max}]$ interval and for $\alpha = 0.05$.

In almost all cases (Figures III.19 and III.20), these probabilities computed using the MSVM belong to the corresponding 95% confidence limits, with no exceptions, thus validating the generality of the model.

For the interpretation of the errors in confidence limit estimation, see the Note in the corresponding section for DWT video modelling, *i.e.* Section III.3.2.1.2 – *Step C-(3)*.

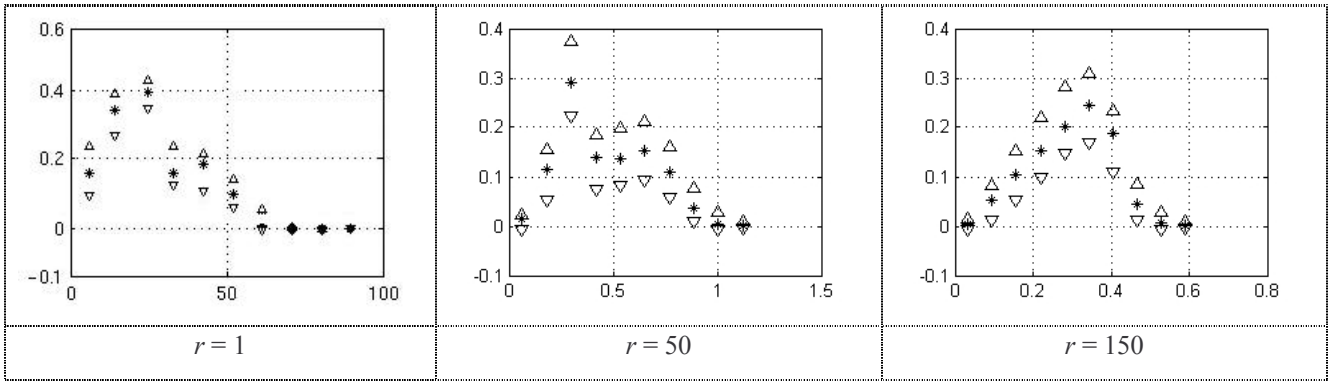


Figure III.19. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the MSVM (represented in *). The results correspond to the whole frames and low quality video.

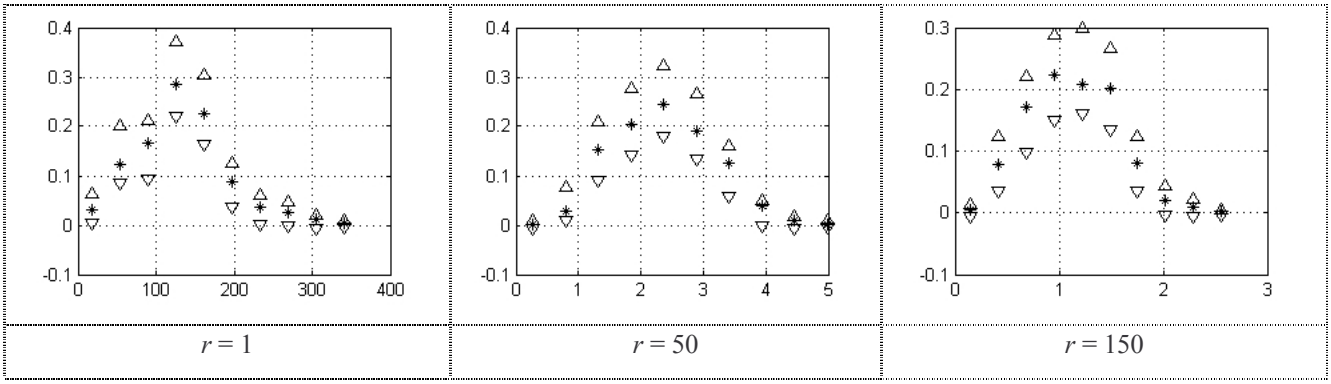


Figure III.20. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the MSVM (represented in *). The results correspond to the whole frames and high quality video.

III.4. Informational Description

This Section is dedicated to the informational description of the noise sources represented by natural video. In this respect the accurate models previously obtained are exploited for entropy computation.

The entropy of an X continuous random variable was defined by [Sha 80], see eq. (II.39). The main problem when evaluating the entropy of a signal is the fact that its value is known only for few, analytically defined *pdf*s. Even for Gaussian mixture models the problem still impossible, but some simplifications (to be further presented) are made by [Lei 04] in order to compute the entropy in some particular cases.

In case of a Gaussian mixture, the entropy computation lead to [Lei 04]:

$$H(X) = -\int_{-\infty}^{+\infty} \hat{p}(x)\log_2 \hat{p}(x)dx = -\int_{-\infty}^{+\infty} \sum_{k=1}^{K} P(k)p_k(x)\log_2\left(\sum_{k=1}^{K} P(k)p_k(x)\right)dx = \\ = -\sum_{k=1}^{K} P(k)\int_{-\infty}^{\infty} p_k(x)\log_2\left(\sum_{l=1}^{K} P(l)p_l(x)\right)dx . \tag{III.10}$$

Rewriting the argument inside the logarithm as:

$$\sum_{l=1}^{K} P(l)p_l(x) = P(k)p_k(x) + \sum_{l\neq k} P(l)p_l(x) = P(k)p_k(x)\left(1+\frac{\sum_{l\neq k} P(l)p_l(x)}{P(k)p_k(x)}\right) = \\ = P(k)p_k(x)\left(1+\varepsilon_k(x)\right). \tag{III.11}$$

[Lei 04] assumes $\varepsilon_k(x) << 1$, which is equivalent with the assumption *"the component of the model not to be very overlapped"*.

In this case $\log_2(1+\varepsilon_k(x)) \approx \varepsilon_k(x)$ the equation can be unfolded as:

$$H(X) \cong -\sum_k P(k)\int_{-\infty}^{+\infty} p_k(x)\log_2\left(P(k)p_k(x)(1+\varepsilon_k(x))\right)dx = \\ = -\sum_k P(k)\int_{-\infty}^{\infty} p_k(x)\log_2(P(k)p_k(x))dx - \sum_k P(k)\int_{-\infty}^{\infty} p_k(x)\varepsilon_k(x)dx . \tag{III.12}$$

Evaluating each term of the entropy the following equation is obtained:

$$H(X) \cong \sum_k P(k)\log_2\frac{\sigma_k\sqrt{2\pi e}}{P(k)} - \sum_k\sum_{l\neq k} P(l), \tag{III.13}$$

where σ_k is the variance of a Gaussian distribution $N(\mu_k, \sigma_k)$ [Tra 02].

After regrouping the terms in the equation the entropy has the form:

$$H(X) \cong \sum_{k=1}^{K} P(k)\log_2\frac{\sigma_k\sqrt{2\pi e}}{P(k)} - (K-1). \tag{III.14}$$

Unfortunately, such a procedure does not hold in natural video modelling: the 10 Gaussian distributions involved in the mixture are generally overlapping, as illustrated in Figure III.21.

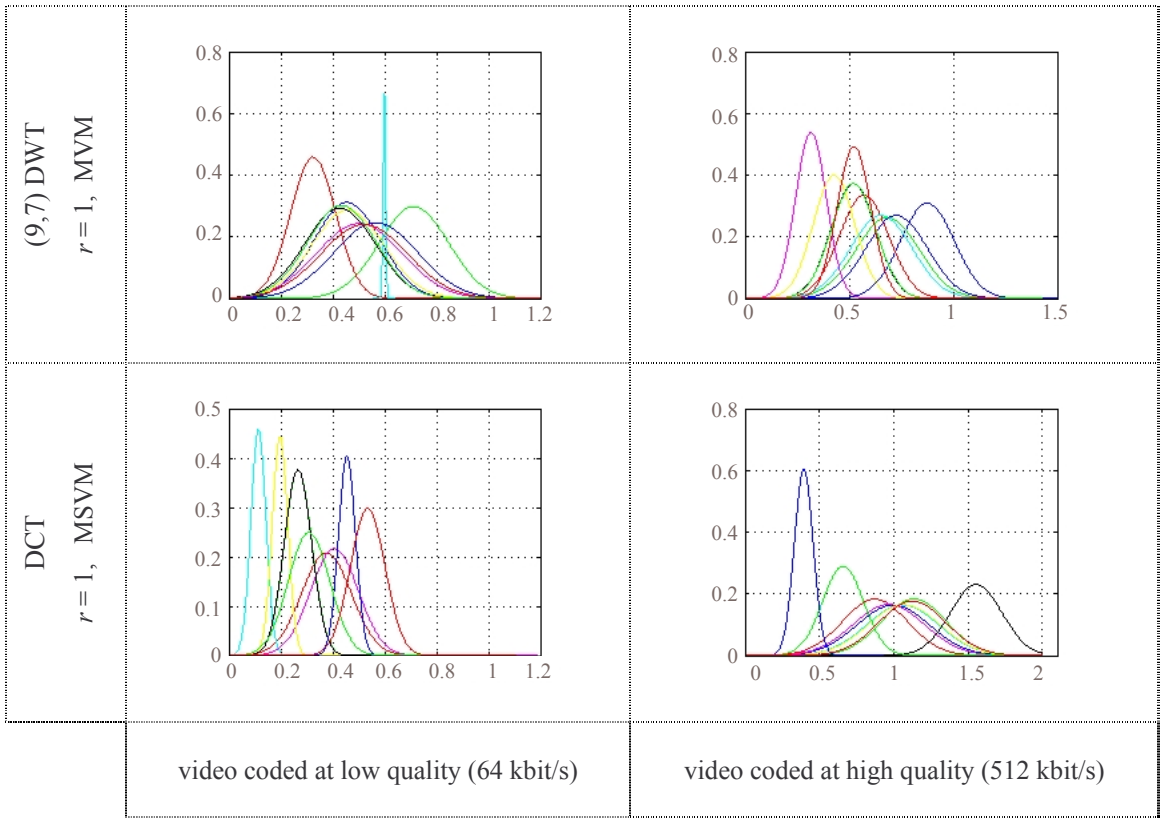


Figure III.21. Example of $K = 10$ Gaussian mixture in the DWT / DCT applied on the whole frames.

Consequently, in our approach, we had to consider the general case and to compute the entropy by considering the corresponding model.

For comparison, the entropies for the Gaussian distributions of the same mean value and variance as the corresponding model are also computed. Note that in the continuous case, the maximum entropy among all the sources of a given variance is provided by the Gaussian distribution (see Section II.4.1, entropy properties) [Sha 80].

Also note that the continuous entropy is not necessarily positive (actually, it is relative to the coordinate system).

The Table III.28 shows the entropies values for MVM in the (9,7) DWT case is structured as follows: the left columns correspond to Gaussian distribution and the right columns correspond to the model.

The Table III.29 and III.30 provide information about the entropy in the DCT domain computed from the MSVM.

Table III.28. Entropies for multiple video sequences in the (9,7) DWT domain.

Rank	MVM			
	Low quality video		High quality video	
	Gaussian	DWT model	Gaussian	DWT model
$r = 1$	-0.143	-0.547	-0.224	-0.259
$r = 25$	-0.852	-3.544	-0.351	-1.797
$r = 50$	-1.308	-4.774	-0.585	-2.586
$r = 75$	-1.662	-5.335	-0.703	-2.955
$r = 100$	-2.007	-6.162	-0.850	-3.453
$r = 125$	-2.354	-7.169	-0.947	-3.815
$r = 150$	-2.654	-8.239	-1.049	-4.045
$r = 175$	-2.917	-8.859	-1.157	-4.300
$r = 200$	-3.299	-8.852	-1.249	-4.171
$r = 225$	-3.642	-10.235	-1.354	-4.455
$r = 250$	-4.034	-8.573	-1.385	-4.813
$r = 275$	-4.491	-8.975	-1.482	-4.830
$r = 300$	-4.472	-5.050	-1.573	-4.680
$r = 325$	-2.503	-5.830	-1.587	-5.274
$r = 350$	-0.883	-1.353	-1.681	-5.131

Table III.29. Entropies for multiple shuffled video models in the DCT domain and low quality video.

Rank	MSVM					
	whole frames		4x4 block		8x8 block	
	Gaussian	DCT model	Gaussian	DCT model	Gaussian	DCT model
$r = 1$	11.918	3.510	-0.560	-2.3221	-0.219	-1.075
$r = 2$	6.698	2.109	-0.887	-2.863	-0.539	-1.302
$r = 3$	5.209	0.907	-0.782	-3.055	-0.647	-1.392
$r = 4$	4.601	0.707	-0.918	-2.609	-0.165	-1.853
$r = 5$	4.140	0.478	-2.727	-5.229	-0.807	-2.700
$r = 25$	-0.154	-0.634				
$r = 50$	-0.237	-1.258				
$r = 75$	-0.580	-0.653				
$r = 100$	-0.488	-0.804				
$r = 125$	-0.437	-1.091				
$r = 150$	-0.361	-1.412				
$r = 175$	-0.344	-1.921				
$r = 200$	-0.279	-2.590				
$r = 225$	-0.249	-2.610				
$r = 250$	-0.207	-2.179				
$r = 275$	-0.165	-1.534				
$r = 300$	-0.188	-1.915				
$r = 325$	-0.050	-0.607				
$r = 350$	1.293	1.270				

Table III.30. Entropies for multiple shuffled video models in the DCT domain and high quality video.

Rank	MSVM					
	whole frames		4x4 block		8x8 block	
	Gaussian	DCT model	Gaussian	DCT model	Gaussian	DCT model
$r = 1$	11.027	7.229	-0.774	-2.361	-0.781	-2.887
$r = 2$	10.521	4.687	-1.047	-2.979	-0.229	-1.557
$r = 3$	4.906	4.405	-0.653	-2.997	-0.565	-1.402
$r = 4$	2.842	2.654	-0.294	-2.745	-0.732	-1.394
$r = 5$	1.921	1.712	-0.541	-5.538	-0.262	-1.091
$r = 25$	0.904	0.901				
$r = 50$	0.352	0.011				
$r = 75$	0.133	-1.189				
$r = 100$	0.026	-3.188				
$r = 125$	0.030	-2.657				
$r = 150$	0.059	-1.159				
$r = 175$	0.028	-0.993				
$r = 200$	0.005	-0.844				
$r = 225$	-0.023	-0.560				
$r = 250$	-0.057	-0.471				
$r = 275$	-0.083	-0.458				
$r = 300$	-0.065	-0.453				
$r = 325$	-0.077	-0.449				
$r = 350$	-0.074	-0.434				

III.5. Conclusion

The present chapter focuses on a hot research topic: is there any mathematical model able to represent the natural video? By presenting an advanced statistical approach and by applying it to a large and heterogeneous video corpus (about five hours of video), the probability density functions able to model with precision the 2D-DWT and 2D-DCT coefficient hierarchies have been for the first time estimated.

This successful *pdf* estimation comes across with another fundamental result: it exhibits that the stationarity of the video sequences can be accepted. Each considered sequence (be it obtained from an individual video, from several videos or several shuffled videos) allowed for a very fine statistical model. However, for individual video sequences, the regularity stops here: different mathematical models are obtained for different video sequences. When considering mixed video sequences (simple in the DWT domain or shuffled in the DCT domain), the obtained mathematical model is independent with respect to the data on which it was computed. Hence, when talking about ergodicity for video sequences we are obliged to consider mixed video sequences (MVM) or mixed shuffled video sequences (MSVM) rather than individual video sequence (IVM).

These general models (MVM – DWT case or MSVM – DCT case) are validated by a large variety of tools: similarity measures & errors, confidence limit estimation, Hellinger distance and Kullback-Leibler divergence.

Such a general model can be exploited in many applications [Dum 08c]:

- *Compression/denoising*

  When considering these applications, the challenge is to dispose of a unique model which can be exploited on large video sequences. Consequently, our estimation procedure should be applied on a mixed video sequence (for DWT) and a mixed shuffled video sequence (for DCT).

- *Indexation/classification*:

  For indexation/classification applications, the models capable to identify data peculiarities are required. Hence, the estimation procedure should be applied on individual video sequences (for both DWT and DCT).

- *Watermarking*:

  In this case, both mixed and individual models may be useful for DWT and DCT.

  For Spread Spectrum methods, the insertion/detection techniques are independent with respect to the video content; hence, a mixed model would prove its efficiency.

  For Side Information methods, the embedding technique should exploit the original video salient characteristics; hence, a model estimated on this particular video would be useful.

III.6. References

[Ald 97] J. Aldrich, *"R.A. Fisher and the Making of Maximum Likelihood 1912-1922"*, Statistical Science, Vol. 12, No. 3, 1997, pp. 162 – 176.

[Ant 92] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, *"Image Coding Using Wavelet Transform"*, IEEE Trans. on Image Processing, Vol.1, No.2, 1992, pp. 205 – 220.

[Arc 03a] C. Archambeau, J. Lee, M. Verleysen, *"Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures"*, Proc. 11th European Symposium on Artificial Neural Networks, April 2003, Bruges, Belgium, pp. 99 – 106.

[Arc 06b] C. Archambeau, M; Valle, A. Assenza, M. Verleysen, *"Assessment of Probability Density Estimation Methods: Parzen Window and Finite Gaussian Mixtures"*, Proc. IEEE International Symposium on Circuits and Systems, May 2006, Kos, Greece.

[Bas 96] M. Basseville, *Information: Entropies,Ddivergences et Moyennes*, Publication Interne –INRIA, N°1020, 1996.

[Bor 04] S. Borman, *"The Expectation Maximization Algorithm – A Short Tutorial",* Tutorial Notes July 2004.

[Buc 99] R. Buccigrossi, and E. Simoncelli, *"Image Compression via Joint Statistical Characterization in the Wavelet Domain"*, IEEE Trans. on Image Processing, Vol.8, No.12, 1999, pp. 1688 – 1700.

[Das 02] A. Das, U. B. Desai, P.P. Vaidya, *"A New Scale Adaptive Watermarking Thresholding Method for Denoising using Chi-Square Test Statisitic",* Proc. of the 9th International Conference on Electronics, Circuits and Systems, Vol. 3, 2002, pp. 859 – 862.

[Dau 92] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[Dem 77] A.P. Dempster, N.M. Laird, D.B. Rubin, *"Maximum Likelihood from Incomplete Data via the EM Algorithm"*, Journal of the Royal Statistical Society, Series B, Vol. 39, No.1, 1977, pp.1 – 38.

[Dui 07] A. Duijster, S. De Backer, P. Scheunders, *"Wavelet-based Multicomponent Image Restoration"*, Proc. SPIE, Vol. 6763, 2007, pp. 67630J-1 – 67630J-10.

[Dum 07a] O. Dumitru, M. Mitrea, F. Prêteux, *"Accurate Watermarking Capacity Evaluation",* Proc. SPIE, Vol. 6763, September 2007, Boston–USA, pp. 6763 03: 1 – 12.

[Dum 07b] O. Dumitru, S. Duţă, M. Mitrea, F. Prêteux, *"Gaussian Hypothesis for Video Watermarking Attacks: Drawbacks and Limitations"*, EUROCON 2007, Warsaw–Poland, September 2007, pp.849 – 855

[Dum 08a] O. Dumitru, M. Mitrea, F. Prêteux, A. Pathak, *"Probability Density Function Estimation for Video in the DCT Domain"* Proc. SPIE, Vol. 6812, January 2008, San Jose–USA, pp. 6812 0L: 1 – 9.

[Dum 08b] O. Dumitru, M. Mitrea, F. Prêteux, *"Video Modelling in the DWT Domain"*, Proc. SPIE, Vol. 7000, April 2008, Strasbourg–France, pp. 7000 0P: 1 – 12.

[Dum 08c] O. Dumitru, M. Mitrea, F. Prêteux, *"Wavelet-based Video Modelling"*, Proc. ELMAR, September 2008, Zadar – Croatia, pp. 105 – 108.

[Do 02] M. N. Do and M. Vetterli, *"Wavelet-Based Texture Retrieval using Generalized Gaussian Density and Kullback-Leibler Distance"*, IEEE Transaction on Image Processing, Vol. 11, No. 2, February 2002, pp. 146 – 158.

[Ege 02] O. Egecioglu and A. Srinivasan, *"Efficient Non-parametric Estimation of Probability Density Functions"*, Computational Statistics & Data Analysis, Vol. 39, No. 1, pp. 75 – 95.

[Fri 83] B.R. Frieden, *Probability, Statistical Optics and Data Testing,* Springer-Verlag, N.Y, 1983.

[Fug 03] B. Fuglede and F; Topsøe, *"Jensen-Shannon Divergence and Hilbert Space Embedding"*, Manuscripts submitted to ISIT2004, December 2003.

[Gir 03] M. Girolami and C. He, *"Probability Density Estimation from Optimally Condensed Data Samples"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, October 2003, pp. 1253 – 1264.

[Hie 04] T. D. Hien, Z. Nakao, Y.-W. Chen *"An RDWT Based Logo Watermark Embedding Scheme with Independent Component Analysis Detection"*, Lecture Notes in Computer Science, Vol. 3214, October 2004, pp. 359-365.

[Hol 89] M. Holschneider, R. Kronland-Martinet, J. Morlet and P. Tchamitcqian, *A real-time algorithm for signal analysis with the help of the wavelet transform*, in Wavelets, Time-Frequency Methods and Phase Space**,** A.G. J. M. Combes and P. Tchamitchian, Eds. Berlin, Springer, JFTI, 1989, pp. 286-297.

[Jos 95] R. Joshi and T. Fischer, *"Comparison of Generalized Gaussian and Laplacian Modeling in DCT Image Coding"*, IEEE Signal Processing Letters, Vol. 2, No. 5, 1995, pp. 81 – 82.

[Jpg 00] Official JPEG 2000 web page http://www.jpeg.org/JPEG2000.htm

[Kwa 02] W.-G. Kwak and R.-H. Park, "*Data Hiding using Bit Repetition in the Wavelet Significant Tree Structure"*, Signal and Image Processing 2002, Vol. 359.

[Kiv 99] M. Kivanc, I. Kozintsev, K. Ramchandran, *"Spatially Adaptive Statistical Modeling of Wavelet Image Coefficients and its Application to Denoising"*, Proc. of the Acoustics, Speech, and Signal Processing, Vol. 6, 1999, pp. 3253 – 3256.

[Kul 51] S. Kullback and R.A. Leibler, *"On Information and Sufficiency"*, Annals of Mathematical Statistics, Vol. 22, No; 1, March 1951, pp. 79 – 86.

[Lam 00] E. Lam and J.A. Goodman, *"A Mathematical Analysis of the DCT Coefficient Distributions for Images"*, IEEE Trans. on Image Processing, Vol. 9, No. 10, 2000, pp. 1661 – 1666.

[Lei 04] J.M. Leiva-Murillo, A. Artés-Rodriguez, *"A Gaussian Mixture Based Maximization of Mutual Information for Supervised Feature Extraction"*, Lecture Notes in Computer Science, Vol. 3195, October 2004, pp. 271 – 278.

[Li 00] J. Q. Li and A.R. Barron, *"Mixture Density Estimation"*, In Advance in Neural Information Processing System 12, 2000, pp.279 – 285.

[Liu 07] J.-C. Liu, C.-H. Lin, L.-C. Kuo, J.-C. Chang, "*Robust Multi-scale Full-Band Image Watermarking for Copyright Protection"*, Lecture Notes in Computer Science, Vol. 4570, July 2007, pp. 176-184.

[Mal 89] S. Mallat, *A Theory of Multiresolution Signal Decomposition: The Wavelet Representation,* IEEE Tans. on Pattern Analysis and Machine Intelligence, Vol.11, No.7, 1989, pp. 674 – 693.

[Mil 98] G. Miller and D; Horn, *"Probability Density Estimation using Entropy Maximization"*, Neural Computation, Vol. 10, 1998, pp. 1925 – 1938.

[Mit 04a] M. Mitrea, F. Prêteux, A. Vlad, C. Fetița, *"The 2D-DCT Coefficient Statistical Behaviour: A Comparative Analysis on Different Types of Image Sequences"*, Journal of Optoelectronics and Advanced Materials, Vol. 6, 2004, pp. 95 – 102.

[Mit 04b] M. Mitrea, F. Prêteux, A. Vlad, *"Watermarking-oriented Video Modelling in the Wavelet Domain"*, WSEAS Transactions on Mathematics, Vol. 3, No. 1, January 2004, pp. 282 – 287.

[Mit 04c] M. Mitrea, T. Zaharia, F. Prêteux, A. Vlad, *"Accurate Data Modelling for Watermarking Applications"* Proceedings IMA International Conference on Mathematics in Signal Processing VI, Cirencester, UK, December 2004, pp. 167 – 170.

[Mit 05] M. Mitrea, F. Prêteux, S. Duță, M. Petrescu, *"The StirMark Watermarking Attack in the DWT Domain"*, IWSSIP 2005, Vol. 2, Halkida – Greece, September 2006, pp. 5-9.

[Mit 06] M.Mitrea, S.Duță, F. Prêteux, *"A Unified Approach to Multimedia Content Watermarking"*, TFIT 2006, Nancy, France, March 2006, pp. 275 – 289.

[Mit 07a] M. Mitrea, O. Dumitru, F. Prêteux, A. Vlad, *"Zero Memory Information Sources Approximating to Video Watermarking Attacks",* Lecture Notes in Computer Science (LNCS), Vol. 4707, Part III, 2007, pp. 445 – 459.

[Mit 07b] M. Mitrea, O. Dumitru, F. Prêteux, "*Video Watermarking Capacity in the DWT Hierarchy",* Proc. SPIE, Vol. 6576, Orlando–USA, April 2007, pp.65760E: 1 – 10.

[Mon 09] M. Monemizadeh, S.A. Seyedin, *"Optimal DWT-SVD Domain Image Watermarking Using Multi-objective Evolutionary Algorithms"*, World Congress on Computer Science and Information Engineering, Los Angeles, California USA, March 2009.

[Mul 93] F. Müller, *"Distribution Shape of Two–Dimensional DCT Coefficients of Natural Images",* Electronics Letters, Vol. 29, No. 22, 1993, pp. 1935 – 1936.

[Nik 04] B. Niklaus, F. Nicolas, *"La Divergence de Kullback-Leibler"*, Research Report, June 2004.

[Par 07] J.S. Park, B.H. Nam, *"A Blind Watermarking Using Data Matrix and Changing Coefficients in Wavelet Domain"*, Proc. SPIE - ICMIT 2007, Vol. 6794 (2), Gifu, JAPON, December 2007, pp. 679444.1 – 679444.6.

[Pat 03] US Patent US6556689, Watermarking methods for digital images and videos.

[Pod 01] C.I. Podilchuk, E.J. Delp, *"Digital Watermarking: Algorithms and Applications",* IEEE Signal Processing Magazine, Vol. 18, No. 4, July 2001, pp. 33 – 46.

[Red 09] R. Reddy, M. Prasad, S. Rao, *"Robust Digital Watermarking of Images using Wavelets"*, International Journal of Computer and Electrical Engineering, Vol. 1, No. 2, June 2009, pp. 1793 – 8163.

[Red 84] R. A. Redner and H.F. Walker, *"Mixture Densities, Maximum Likelihood and the EM Algorithm"*, Society for Industrial and Applied Mathematics, Vol. 26, No.2, April 1984, pp. 195 – 239.

[Rei 83] R. Reininger and J. Gibson, *"Distributions of the Two–Dimensional DCT Coefficients for Images"*, IEEE Trans. on Communications, Vol. COM–31, No. 6, 1983, pp. 835 – 839.

[Ros 78] J.P. Rossi, *"Digital Techniques for reducing television Noise"*, JSMPTE, Vol. 87, March 1978, pp. 134-140.

[Sch 06] P. Scheunders, S. De Backer, *"Wavelet Denoising of Multicomponent Images, using a Gaussian Scale Mixture Model",* The 18th International Conference on Pattern Recognition (ICPR 2006), 2006.

[Sco 80] D. W. Scott, R.A. tapia, J.R. Thompson, *"Nonparametric Probability Density Estimation by Discrete Maximum Penalized-Likelihood Criteria"*, Annals of Mathematical Statistics, Vol.8, No. 4, 1980, pp. 820 – 832.

[Sha 48] C.E. Shannon, *"A Mathematical Theory of Communication"*, The Bell System Technical Journal, Vol. 27, October 1948, pp. 379 – 423, 623 – 656.

[Sha 58] C. E. Shannon, *"Channels with Side Information at the Transmitter"*, IBM Journal, 1958, pp. 289 – 293.

[Sha 80] C.E. Shannon, W. Weaver, *The Mathematical Theory of Communication*, 8th ed., University of Illinois Press, 1980.

[She 92] M. J. Shensa, *"Affine wavelets: wedding the A trous and Mallat algorithms"* IEEE Transaction Signal Processing, Vol. 40, No. 10, October 1992, pp. 2464-2482.

[Sim 97] E. Simoncelli, *"Statistical Models for Images: Compression, Restoration and Synthesis"*, 31st Asilomar Conference on Signals, Systems, and Computers, 1997, pp. 673 – 678.

[Sri 03] A. Srivastava, A. Lee, E. Simoncelli, S. Zhu, *"On Advances in Statistical Modeling of Natural Images"*, Journal of Mathematical Imaging and Vision, Vol.18, No.1, 2003, pp. 17 – 33.

[Sul 03] G. Sullivan *"Approximate theoretical analysis of RGB to YCbCr to RGB conversion error"*, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), JVT PExt Ad Hoc Group Meeting, Trondheim, July 2003.

[Sun 04] S. Sun, P. Qiu, Y. Wang, L.F. Yao, *"Blind 'Watermarking Algorithm Based on General Gaussian Model"*, Proc. ICSP 2004, pp. 2302 – 2305.

[Tay 07] J. Shawe-Taylor and A. Dolia, *"A Framework for Probability Density Estimation",* Artificial Intelligence and Statistics, Porto-Rico, March 2007.

[Tin 08] C.-W. Ting, B.-M. Goi, S.-H. Heng, *"Attacks on a robust watermarking scheme based on self-reference image"*, Computer Standards & Interfaces vol. 30, 2008, pp. 32–35.

[Tka 08] M. Tkalcic, J.F. Tasic, *"Colour Spaces-Perceptual, Historical and Applicational Background"*, EUROCON 2003, September 2003, Vol. 1, pp. 304- 308.

[Tra 02] L. Trailovic and L.Y. Pao, *"Variance Estimation and Ranking of Gaussian Mixture Distributions in Target Tracking Applications"*, Proc. of the 41st IEEE Conference on Decision and Control, Las Vegas, USA, December 2002, pp. 2195 – 2201.

[Usc 08] The USC – SIPI Image Database
http://sipi.usc.edu/database

[Wal 89] R. E. Walpole, R. H. Myres, *Probability and Statistics for Engineers and Scientists*, 4th ed., MacMillan Publishing, 1989.

[Xia 98] X. G. Xia, C. Boncelet, G. Arce, *"Wavelet transform based watermark for digital images"*, Opt. Express Vol. 3, 1998, pp. 497-511.

"If we can get to the moon, or get to Mars … why can't we put a little watermark on our content?"

Michael Eisner (1942 - ___)

(CEO, Walt Disney, Fortune, 2002)

Chapter IV

Watermarking Attack Modelling

In the absence of any theoretical or experimental support, the video watermarking attacks are by default assumed to be Gaussian distributed. After bringing into evidence that such a model cannot be accepted (Section IV.1), an algorithm for attack modelling, able to compute the model with good accuracy is presented (Section IV.2). This algorithm, ART.MOD-A is further deployed in order to obtain a general model for attack modelling in the 2D-DWT and 2D-DCT domains (Section IV.3). Section IV.4 throws more light on the continuous capacity estimation for watermarking purposes. The validation of the attack model effectiveness under the framework of the watermarking method is presented in Section IV.5. The conclusions (Section IV.6) end the chapter.

Contents

IV.1. Previous Results

As already discussed in Chapter 1, watermarking can be considered as a communication problem: the owner attempts to communicate his/her mark over a hostile channel where non – intentional and intentional attacks may occur. The theoretical challenge is to communicate as much watermark information as possible, while maintaining the quality for the marked data and a prescribed degree of robustness. Consequently, designing high performance watermarking techniques and identifying their theoretical limits is a problem which cannot be solved unless an accurate model for attacks is provided. A succinct state-of-the-art of the studies devoted to watermarking attack modelling is further presented.

In literature, many authors [Bar01], [Bau 02], [Kun 01], [Mou 02] suppose that the model for attacks is the AWGN (additive white Gaussian noise).

For example, [Vol 99] and [Kov 04] make some improvement in a known-host-state watermarking methods by exploiting this AWGN model for attacks and by assuming *i.i.d.* Laplacian distribution for the host data.

The same model is considered by [Bau 02] in order to compute the capacity of the scalar Costa scheme for attacks modelled by AWGN. Both DCT and DWT transforms are considered.

[Cox 08] explains the utility of the AWGN model for different transforms. Although acknowledged as simplistic, such a model should be considered at least as a minimal requirement on robustness.

The same AWGN model is considered for capacity evaluation for compressed and uncompressed watermarking [Mou 02]. The tests are realized on DCT (8×8 blocks) and DWT coefficients extracted from 3 images of 512×512 pixels (*Lena, Baboon,* and *Papers*). The original coefficients follow a Gaussian distribution.

The results reported in [Bou 03] consider for the attack modelling the following assumptions:

- the JPEG compression at high quality level (low compression) is modelled by a Gaussian distribution. When the quality level decreases, the Laplacian distribution model becomes more suitable.
- the SPIHT [1] compression [Spi 09] is better modelled by a Laplacian distribution.

In order to establish the validity of the Gaussian hypothesis, [Mit 05] proposed a method based on an original statistical approach combining four tests, namely the Chi-square test, the Ro test, the Fisher test, and the Student test. Note that this approach is paired designed with the method devoted to natural video investigation *cf.* [Mit 07a], [Dum 08c] and Section III.3.2.

A brief presentation of the statistical investigation method is [Mit 06c]:

(1) Consider a sequence of L frames represented in the HSV space.

[1] Set Partition in Hierarchical Trees is a wavelet-based image compression method. In 1995 it has become the benchmark state-of-the-art algorithm for compression image, video and medical and signals [Spi 09].

(2) Apply the DWT or DCT to the V component; record the largest R coefficients; the corresponding vector is denoted by the *original*.

(3) Apply the attack to each frame ($L = 35000$ frames) and repeat the previous step; obtained the *attacked* vector.

(4) Compute the difference between *attacked* and *original* coefficient vectors; in this way, a *noise* vector with L components is obtained for each rank.

(5) For each rank (*i.e.* for each *noise* vector) partition the L values into D classes using a sampling period of $D = 250$, thus obtaining L / D independent elements.

(6) Repeat the previous step *D-1* times by shifting the sampling origin; obtain D classes with independent elements.

(6) Apply for each class the Chi-square test on concordance, the Ro test on correlation, the Fisher test on equality between two variances, and the Student test on equality between two means; all these tests are applied at an $\alpha = 0.05$ significance level.

The statistical investigation method was run for 6 types of attacks for (9,7) DWT and DCT. For all attacks the StirMark benchmarking tool was used by the authors. First, the filtering operations, be they linear or not: median, Gaussian, sharpening, FMLR (Frequency Model Laplacian Removal) [Bar 98]. Secondly, small rotations (+2°, -2°, +5°), compressions (JPEG compression at a quality factor of 70, individually applied to each frame) and row & column removal (5 rows and 1 column). Finally, the StirMark attack is individually applied to each frame, at its standard parameters.

Figure IV.1.a and Figure IV.1.b present the results obtained when applying the Chi-square test (the ratio of the number of test which are not passed to the D value can be considered as a measure of the overall Gaussian behaviour). The abscissa corresponds to the investigated rank ($R = 360$ ranks) and the ordinate corresponds to the ratio of the tests which are not passed. As it can be seen, in most cases the Gaussian behaviour has been refuted. In the DWT domain, only the FMLR attack effects can be considered Gaussian distributed up to the 200^{th} rank. In the DCT domain, only the effects of the median filter, the FMLR filter, the JPEG compression (up to the 200^{th} rank), and the row & column removal (between the 120 and 250 ranks), can be considered Gaussian distributed.

In order to verify whether the sampling period is large enough so as to ensure the data independency in each class, the Ro tests were applied and the results are displayed in Figure IV.2.a and Figure IV.2.b. The axes in Figure IV.2.a-b are the relative number of Ro tests which are not passed *vs*. the rank in the hierarchy. It can be accepted that the Ro tests are successfully passed (just about 5% of them are refuted).

Note that as the Chi-square tests were not passed, the Ro, the Fisher and the Student tests cannot properly run (such test are mathematically proved only for Gaussian data). Consequently, the corresponding results represent just hints and not proofs.

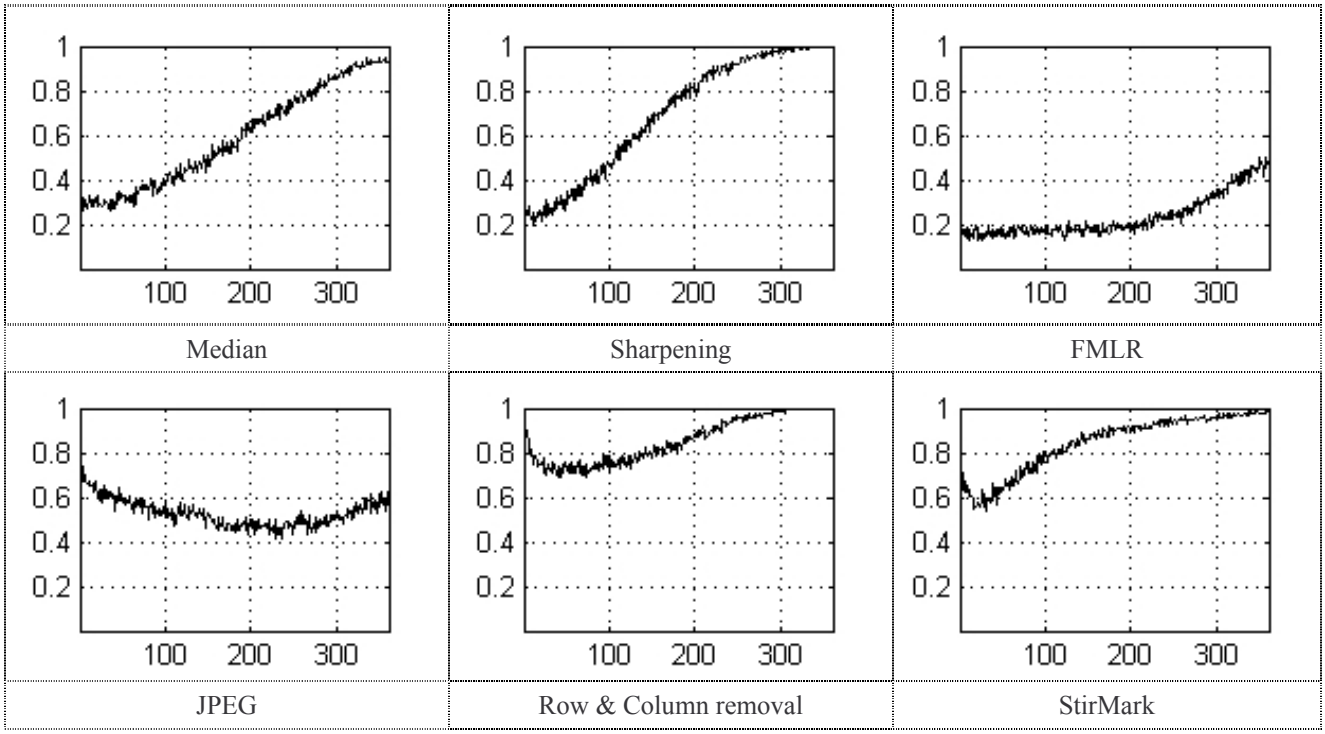


Figure IV.1.a. The relative number of Chi-square tests on concordance with the Gaussian law which are not passed – (9,7) DWT (L = 35000 frames, R = 360 coefficients, and D=250 sampling period).

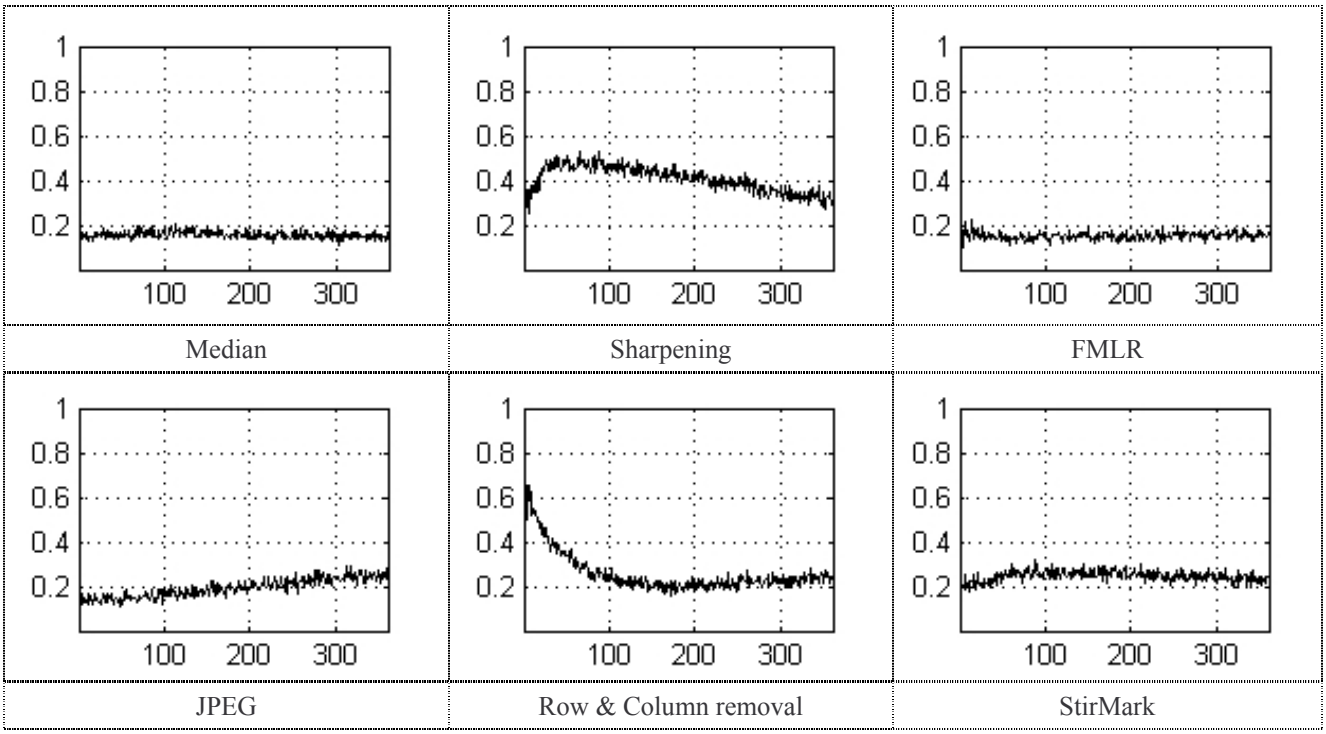


Figure IV.1.b. The relative number of Chi-square tests on concordance with the Gaussian law which are not passed – whole frame DCT (L = 35000 frames, R = 360 coefficients, and D=250 sampling period).

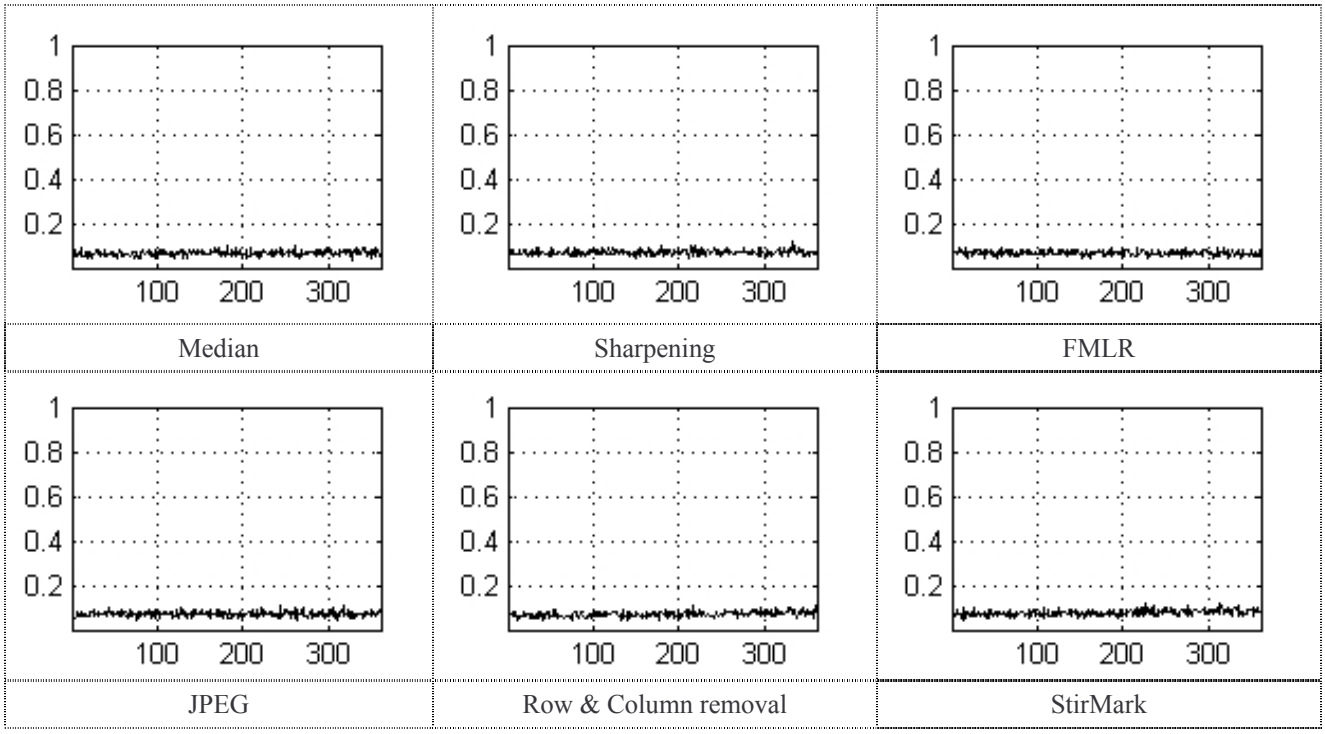


Figure IV.2.a. The results of the Ro test in the case of (9,7) DWT applied on whole frames ($L = 35000$ frames, $R = 360$ coefficients, and D=250 sampling period).

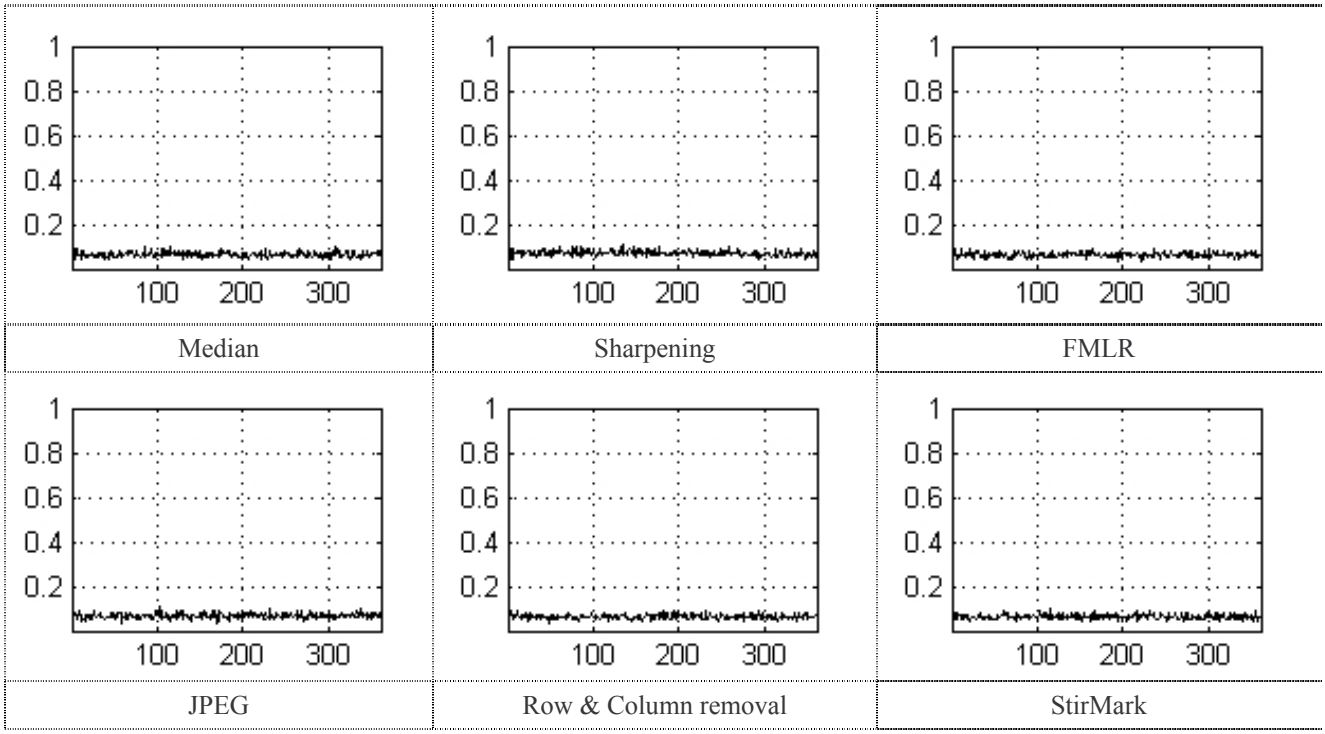


Figure IV.2.b. The results of the Ro test in the case of DCT applied on whole frames ($L = 35000$ frames, $R = 360$ coefficients, and D=250 sampling period).

Additional experimental results are presented in [Dum 07b] and in Appendix II which also contain a comparative study realised for different attacks and for sampling period $D = 600$.

IV.2. New Method for Attack Modelling (ART.MOD – A)

As it can be noticed, no general procedure, able to establish with mathematical rigour and error control the statistical model of the watermarking attacks can be found in the literature.

This section presents an original method [Mit 07a] which not only allows the true *pdf* modelling the watermarking attacks to be accurately estimated but also makes it possible for a reference investigation on the attack stationarity to be carried out.

Note that attack modelling is not a trivial task. Actually, any mathematical approach should properly answer at least the following three questions:

- *Does a general statistical model for the considered attack effects, independent with respect to the video sequence, really exist?*
- *When considering an individual video sequence, does a reliable model exist for any (intra)frame content and any (inter)frame dependency?*
- *In case such a model exists, which is its probability density function?*

In order to answer these questions, an algorithm further referred to as *ART.MOD-A* paired designed with *ART.MOD-V* (see Section III.3.2) is considered. The algorithm is structured into two parts: the model computation (*step A* and *step B*) and model validation (*step C*). Actually, the only difference with respect to *ART.MOD-V* is in the *step-A*.

Note: all the observations concerning the algorithm parameter choice can be found in Chapter III.

IV.2.1. Step A: Attack Effect Representation

Be there a colour video sequence consisting in L frames. Each frame is represented in the HSV (hue-saturation-value) space; the V component is normalised to the $[0,1]$ interval. The following 5 steps should be gone through for each frame in the video sequence:

Step A.1: Apply the considered transform (2D-DWT or 2D–DCT, Figure IV.3) to the V (luminance component) of the considered frame and record in a vector the coefficients to be investigated (see the corresponding flowchart in Section III.2.1).

Step A.2: Compute the coefficient hierarchy by sorting in a decreasing order the vector obtained in the previous step; record the largest R rank values alongside with their original positions ($R = 360$); the resulting vectors are denoted *original_coefficients* and *original_positions*, respectively.

Step A.3: Apply the chosen attack to the considered frame.

Step A.4: Resume the *Step A.1* and *Step A.2* on the attacked video sequence; record the coefficients corresponding to the *original_positions* in the attacked video, thus resulting the *attacked_coefficients* vector.

Step A.5: Compute the difference between the two vectors:

difference = attacked_coefficients - original_coefficients.

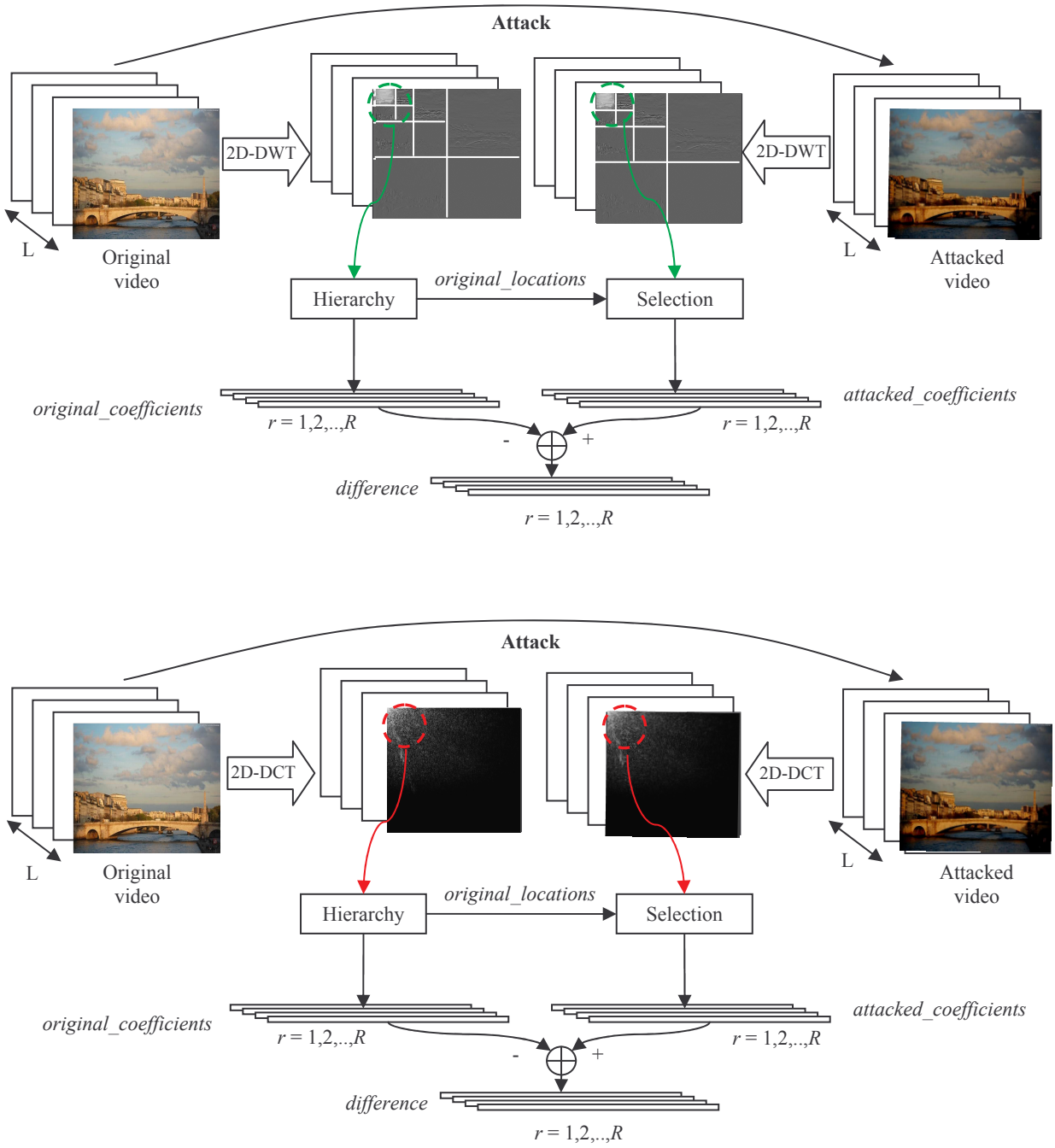


Figure IV.3. Obtaining a sample from the information source modelling the watermarking attacks in the DWT or DCT domain.

After applying the *Step A.1* - *Step A.5* to each frame in the video sequence, a set of L vectors (each of them with $R = 360$ components) of the same type as *difference* is obtained. Should we

consider only the values corresponding to an arbitrarily chosen rank r, a new vector of L components is obtained: $noise = [n_1, n_2, ..., n_L]$, see the upper part of Figure IV.4. An n_j component ($j = 1,2,...,L$) in a *noise* vector is the value corresponding to the rank r in the *difference* vector computed on the frame j. The *noise* vector components are random: they depend at least on the original frame content and on the random mechanism (if any) in the attack.

IV.2.2. Step B: Pdf Estimation for Attacks

This step is meant to elucidate the following aspects:

- The stationarity of the video sequence.
  Does a reliable model exist for any (intra)frame content and any (inter)frame dependency?
- The mathematical expression for a general model.
  In case such a model exists, which is its pdf (probability density function)?

In order to model the attack effects on a rank (arbitrarily chosen) in the DWT or DCT coefficient hierarchy, the following 6 steps should be gone through:

Step B.1: Periodically sample the *noise* vector, with a large enough period D. Shift the sampling origin and obtain D *iid* data sets $[x^i_1, x^i_2, ..., x^i_N] = [n_i, n_{D+i}, ..., n_{(N-1)D+i}]$ ($i = 1,2,...,D$ and $N = L/D$), see Figure IV.4.

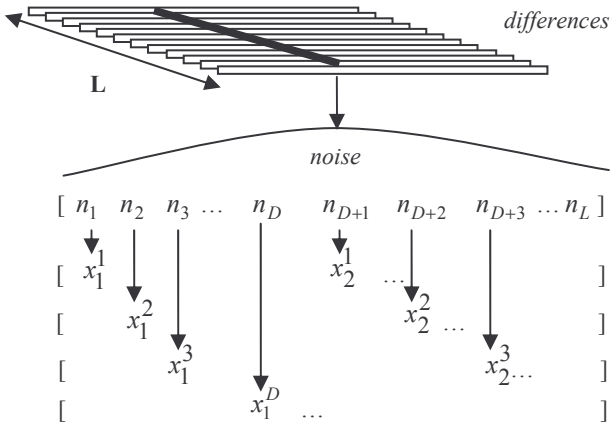


Figure IV.4. By shifting the sampling period, several *i.i.d.* data sets can be obtained form the same *noise* vector.

Step B.2: Apply the EM Gaussian mixture algorithm to each of the D *iid* data sets $[x^i_1, x^i_2, ..., x^i_N]$ and obtain the D *pdf* estimates $\hat{p}^i(x)$, $i = 1,2,...,D$.

Step B.3: Apply the EM Gaussian mixture algorithm to the whole *noise* vector and obtain the estimate $\hat{p}_{av}(x)$. Note that although $\hat{p}_{av}(x)$ has no mathematical support (it is

computed on the whole noise vector, hence on dependent data), it intuitively expresses an average value for the $\hat{p}^i(x)$, $i = 1,2,...,D$ estimates.

Step B.4: Define a similarity measure, eq. (III.3), denoted by $m(.)$, between two arbitrarily chosen $\hat{u}$ and $\hat{v}$ *pdf* estimates.

Step B.5: The model for the investigated attack and rank will be that $\hat{p}^i(x)$ estimate to which $\hat{p}_{av}(x)$ is closest in the $m(.)$ sense, see eq. (III.4).

Step B.6: In order to check up the pertinence of the model defined in the previous step, average the measures between each of the D estimates and the model, see eq. (III.5).

The *pdf* defined in *Step B.5* can be considered as a true model only if the *Error* computed by eq. (III.5) is lower than a prescribed upper limit, *e.g.* lower than 0.10.

IV.2.3. Step C: Model Validation

This step was designed so as to answer the following question:

- *Does a general statistical model for the considered attack effects, independent with respect to the video sequence, really exist?*

After the computation of the model, the second part is the model validation which is ment to establish the ability of the model to work in frameworks completely different from those which it was computed. To verify whether the model still preserves its usefulness or not, to the following questions should be answered:

(1) Is the model dependent on the similarity measure?

(2) Is the model general or does it depends on the original video sequence?

(3) Is the model dependent on the EM Gaussian mixtures estimation?

(1) Is the model dependent on the similarity measure?

The way in which the model was defined is intimately based on the similarity measure and error definitions, eq. (III.3) and eq. (III.4). However, a good model should be able to describe all the D *pdf* estimates even when the similarity measure changes.

A new average error is computed according to eq. (III.6.b) and all the errors should be lower than 0.15 in orders to consider them as a proof for our model.

(2) Is the model general or does it depends on the video sequence?

To verify whether the model is independent with respect to the video sequence on which it was computed, the rest of sequences from the video corpus are considered in order to build their models, according to the *Step B*. The models thus obtained are compared to the model obtained in *Step B* by there criteria: the similarity measure computed by eq. (III.7.b), the Hellinger distance

computed by eq. (III.9.a), and the Kullback-Leibler divergence computed by eq. (III.9.b). Should the *Error''* be lower then 0.15 and E_{HL}, E_{KL} be in the proximity of zero than the model is validated.

(3) Is the model dependent on the EM Gaussian mixtures estimation?

The procedure follows the flowchart in Section III.2.1 and the explanations in Section III.2.4 (*Step C-(3)*).

If the procedure is successfully applied (see Section IV.3.2.1.1 and Section IV.3.2.2.1), then positive answers to the last two questions (in the introduction of the Section IV.2) are obtained. A positive answer to the first question is obtained if and only if the same model is obtained for different video sequences (see Section IV.3.2.1 and Section IV.3.2.2).

IV.3. Experimental Results

The *algorithm ART.MOD-A* presented in Section IV.2 was applied for different attacks and the statistical models are illustrated in Section IV.3.2.1 for the (9,7) DWT and then in Section IV.3.3.1 for the DCT coefficients modelling. After the model computation, the second step was to validate the reference model based on the methods described in Section IV.2.3.

Additional experiment results are presented in Appendix III. The complete set of experimental results is available on the accompanying CD-ROM.

As in original video modelling case, the first difficulty in the experimental work is to establish the data on which the investigations procedure should be applied. Hence, we started the investigation by considering models computed on individual video sequences (of the type IVM – Individual Video Model). Should this modelling be successful, there is no need to also consider models of the MVM types (MVM – Multiple Video Model).

IV.3.1. Video Corpus

The experiments processed the same corpus as in Section III.3.1: 10 video sequences, belonging to different movies, each of them having about 25 minutes ($L = 35000$ frames for each video sequence). The videos are DivX coded at low quality rate (frame sizes 192x80 pixels, 64 kbit/s) and high quality rate (frame size 640x480 pixels and 512 kbit/s).

As in the previous chapter, in order not to bias the statistical relevance of the results, the black letterboxing bars were not considered in the investigation.

IV.3.2. Statistical models

In order to build the statistical models, the $R = 360$ largest coefficients are considered for 2D–DWT and 2D–DCT transforms applied on whole frame.

When applying the *ART.MOD-A*, the following numerical values are considered:

- $D = 250$ frame sampling period (*i.e.* 10 seconds);
- $K = 10$ Gaussian distributions (see Section IV.2.2);
- $N_{iter} = 200$ iterations in the EM algorithm;
- $S = 20$ distributed intervals (see *step B4*).

For confidence limits computation $S_{conf} = 10$ equal length subintervals were considered and $\alpha = 0.05$ ($z_{\alpha/2} = 1.96$).

The model computation is illustrated for the largest rank ($r = 1$) for the (9,7) DWT domain, in Table IV.1 (low quality video) and Table IV.2 (high quality video) and for the DCT domain in Table IV.9 (low quality video) and Table IV.10 (high quality video).

For each investigated attack the numerical values for the $P(k), \mu(k)$, and $\sigma(k)$ parameters ($k = 1, 2, ..., K = 10$) are displayed. These values are to be considered in order to obtain the $\hat{p}(x)$ model. Moreover, the average relative error *Error* eq. (III.5), in model estimation is calculated and presented in the right column of these tables.

The considered attacks are: FMLR, JPEG, Gaussian filtering, rotation with different degrees, median, sharpening, Stir Mark and were obtained by applying individually the StirMark benchmarking tool (see Chapter I – Section I.2.3) to each frame extracted from the video sequence.

IV.3.2.1. The 2D-DWT coefficients modelling

IV.3.2.1.1. Model Computation

In order to illustrate the model computation for the (9,7) DWT applied to an individual video sequence, coded at low quality rate, Table IV.1 presents the parameters and the associated *Error* corresponding to rank $r = 1$ for nine attacks, namely: FMLR, Gaussian filtering, JPEG compression with a quality factor of 70%, rotation +/- 2°, +5°, median, sharpening, and StirMark.

The models are illustrated in Figure IV.6 in continuous line. For comparison, the Gaussian *pdf* with the same mean values and variances are plotted in Figure IV.6 in dashed line. The illustration is realized only for the rank $r = 1$ and for three attacks, namely Gaussian filtering, sharpening and StirMark, when the video was coded at low quality.

Table IV.2 present the same type of information as Table IV.1, this time obtained by processing high quality video sequences.

Similar results (for all considered attacks) can be reported for others nine models (one model obtained for each video from the video corpus). Each and every time the modelling procedure (*step – A* and *step –B*) was applied for each rank, each attack and for both video quality rates, see Appendix III.

As a general observation, it ca be considered that the model computation was successful, *i.e.* that each type of video sequence and attack allowed an accurate *pdf* estimation to be obtained: for all

IVM models (each rank and attack) these *Error* values are lower then 5%, with one exception, for rotations with -2° and $r = 1$, the *Error* was 5.4%.

Note that Section IV.3.2.1.3 is devoted to a comparison among the DWT types: three transforms are considered namely, the (2,2) DWT, the (4,4) DWT, and the (9,7) DWT. In this respect Tables IV.7 and IV.8 IVM illustrates the results obtained for video sequence coded at low quality and $r = 1$ and $r = 150$, respectively.

Table IV.1. IVM for (9,7) DWT coefficients hierarchy, $r = 1$, low quality video and different attacks.

Attacks	Model parameters											Error
FMLR	*P(k)*	0.099	0.071	0.086	0.026	0.077	0.052	0.151	0.068	0.150	0.218	0.031
	μ(k)	-0.076	-0.073	-0.069	-0.193	-0.111	-0.097	-0.067	-0.108	-0.087	-0.057	
	σ(k)	0.030	0.030	0.030	0.011	0.035	0.038	0.029	0.036	0.023	0.026	
Gaussian filtering	*P(k)*	0.075	0.024	0.270	0.063	0.069	0.056	0.072	0.118	0.20	0.022	0.035
	μ(k)	-0.087	-0.157	-0.048	-0.078	-0.069	-0.115	-0.088	-0.051	-0.057	-0.078	
	σ(k)	0.019	0.021	0.014	0.018	0.018	0.003	0.019	0.015	0.016	0.001	
JPEG	*P(k)*	0.103	0.007	0.183	0.060	0.111	0.180	0.069	0.080	0.114	0.092	0.011
	μ(k)	-0.002	-0.207	0.002	-0.058	-0.004	-0.008	-0.009	0.003	-0.001	-0.028	
	σ(k)	0.021	0.001	0.003	0.027	0.005	0.007	0.022	0.020	0.021	0.016	
Rot +2°	*P(k)*	0.086	0.065	0.068	0.089	0.094	0.105	0.111	0.105	0.136	0.140	0.045
	μ(k)	-0.686	-0.007	-0.541	-0.541	-0.260	-0.969	-0.404	-0.789	-0.234	0.583	
	σ(k)	0.334	0.076	0.338	0.337	0.269	0.268	0.248	0.315	0.134	0.169	
Rot -2°	*P(k)*	0.102	0.143	0.122	0.142	0.091	0.112	0.120	0.007	0.114	0.048	0.054
	μ(k)	-0.699	-0.629	-0.583	-0.443	-0.247	-0.020	-0.639	-2.067	-0.681	-0.140	
	σ(k)	0.297	0.296	0.285	0.181	0.182	0.141	0.297	0.001	0.298	0.009	
Rot +5°	*P(k)*	0.096	0.061	0.127	0.086	0.096	0.098	0.097	0.101	0.092	0.147	0.040
	μ(k)	-0.571	-0.023	-0.527	-0.835	-0.628	-0.754	-0.584	-0.519	-0.802	-0.462	
	σ(k)	0.217	0.104	0.175	0.254	0.254	0.268	0.115	0.194	0.261	0.150	
Median	*P(k)*	0.064	0.078	0.075	0.122	0.170	0.072	0.091	0.205	0.071	0.050	0.024
	μ(k)	0.014	-0.032	-0.028	-0.011	-0.026	-0.012	-0.101	-0.017	-0.055	0.010	
	σ(k)	0.056	0.044	0.044	0.025	0.024	0.053	0.032	0.021	0.042	0.006	
Sharpening	*P(k)*	0.112	0.041	0.074	0.135	0.128	0.128	0.104	0.103	0.098	0.077	0.032
	μ(k)	0.184	0.245	0.020	0.113	0.045	0.088	0.054	0.051	0.019	0.053	
	σ(k)	0.017	0.012	0.072	0.022	0.038	0.048	0.060	0.063	0.072	0.062	
StirMark	*P(k)*	0.070	0.377	0.084	0.079	0.087	0.026	0.084	0.085	0.078	0.027	0.018
	μ(k)	-0.108	-0.028	-0.037	-0.033	-0.024	-0.509	-0.338	-0.114	-0.049	-0.172	
	σ(k)	0.073	0.047	0.053	0.119	0.118	0.255	0.127	0.073	0.122	0.002	

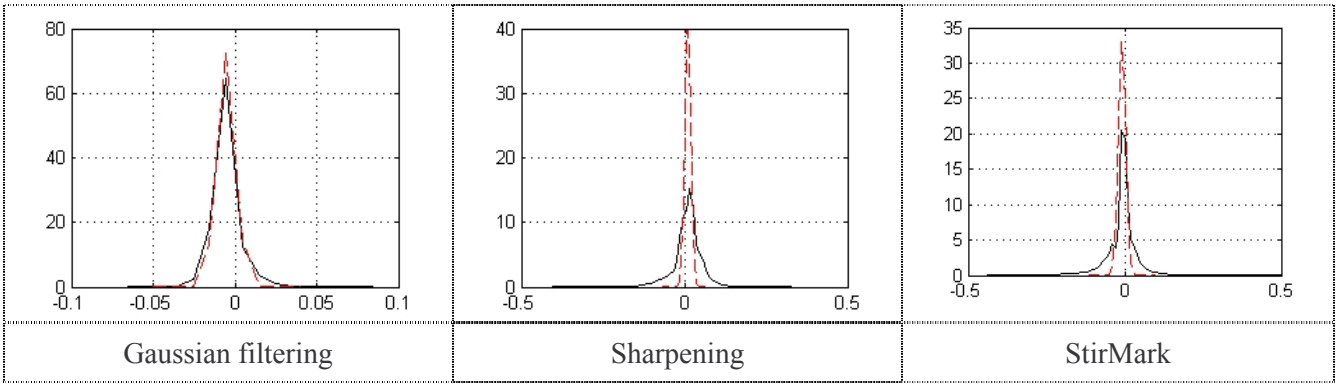


Figure IV.6. The attack models (continuous line) and the corresponding Gaussian distributions (dashed lines), for rank $r = 1$, low quality video, and for Gaussian filtering, sharpening, and StirMark attack.

Table IV.2. IVM for (9,7) DWT coefficients hierarchy, $r = 1$, high quality video and different attacks.

Attacks	Model parameters											Error
FMLR	*P(k)*	0.092	0.127	0.060	0.199	0.077	0.080	0.075	0.080	0.138	0.072	0.033
	μ(k)	0.116	-0.108	-0.116	-0.085	-0.110	-0.140	-0.039	-0.084	-0.110	-0.132	
	σ(k)	0.036	0.034	0.060	0.024	0.036	0.056	0.059	0.024	0.035	0.056	
Gaussian filtering	*P(k)*	0.022	0.029	0.155	0.099	0.039	0.068	0.077	0.104	0.177	0.227	0.030
	μ(k)	-0.033	-0.184	-0.077	-0.131	-0.107	-0.080	-0.089	-0.094	-0.050	-0.067	
	σ(k)	0.001	0.007	0.017	0.027	0.033	0.014	0.026	0.030	0.014	0.020	
JPEG	*P(k)*	0.031	0.032	0.040	0.419	0.202	0.007	0.142	0.028	0.065	0.034	0.003
	μ(k)	-0.026	-0.028	-0.033	-0.004	0.008	-0.099	-0.012	-0.024	-0.033	-0.026	
	σ(k)	0.015	0.015	0.014	0.004	0.006	0.001	0.007	0.015	0.014	0.015	
Rot +2°	*P(k)*	0.119	0.096	0.077	0.107	0.110	0.095	0.132	0.098	0.080	0.093	0.027
	μ(k)	-0.851	-0.541	-0.158	-0.754	-0.682	-0.747	-0.509	-0.702	-1.149	-0.943	
	σ(k)	0.263	0.032	0.100	0.259	0.239	0.258	0.175	0.152	0.372	0.123	
Rot -2°	*P(k)*	0.144	0.134	0.161	0.034	0.035	0.101	0.046	0.144	0.079	0.123	0.024
	μ(k)	-0.673	-0.750	-0.845	-1.101	-0.055	-0.670	-1.649	-0.627	-0.701	-0.507	
	σ(k)	0.255	0.258	0.231	0.339	0.088	0.255	0.220	0.244	0.259	0.174	
Rot +5°	*P(k)*	0.087	0.082	0.087	0.090	0.174	0.101	0.129	0.155	0.075	0.020	0.019
	μ(k)	-1.080	-0.812	-0.646	-0.678	-0.625	-0.651	-0.698	-0.678	-0.643	-0.047	
	σ(k)	0.235	0.279	0.190	0.191	0.186	0.189	0.178	0.182	0.192	0.117	
Median	*P(k)*	0.121	0.042	0.009	0.103	0.427	0.013	0.118	0.099	0.029	0.038	0.016
	μ(k)	-0.003	-0.829	-0.528	-0.071	-0.011	-0.517	-0.043	-0.070	0.007	-0.158	
	σ(k)	0.004	0.129	0.077	0.045	0.028	0.078	0.041	0.045	0.109	0.070	
Sharpening	*P(k)*	0.069	0.129	0.136	0.078	0.159	0.020	0.133	0.116	0.076	0.084	0.026
	μ(k)	0.204	0.085	0.080	0.201	0.101	0.005	0.093	0.081	0.071	0.104	
	σ(k)	0.110	0.069	0.067	0.052	0.045	0.003	0.068	0.069	0.067	0.072	
StirMark	*P(k)*	0.100	0.085	0.084	0.089	0.226	0.084	0.062	0.092	0.092	0.086	0.040
	μ(k)	-0.956	-0.382	-0.574	-0.730	-0.047	-0.713	-0.222	-0.335	-0.354	-0.865	
	σ(k)	0.355	0.219	0.334	0.377	0.064	0.375	0.106	0.292	0.216	0.369	

IV.3.2.1.2. Model Validation

The models computed in Table IV.1 and IV.2 are now investigated according to the three procedures presented in the *step C* of the *ART.MOD-A* algorithm.

Step C-(1): Is the model dependent on the similarity measure?

The values of the *Error'* computed based on eq. (III.6.a) are displayed in Table IV.3 and Table IV.4 for some investigated ranks namely $r = 1$, $r = 50$, $r = 100$, $r = 150$, $r = 200$, $r = 250$, $r = 300$, $r = 350$, all investigated attacks, and low and high quality video, respectively. It can be seen that each and every time the *Error'* values for the IVM are lower then 15% with singular exceptions:

- for median filtering when $r = 1, 50, 300$ in the low quality video;
- for median filtering when $r = 250$ and for rotation with 5° when $r = 50$ in the high quality video.

Table IV.3. The average errors according to *step –C (1)* for the (9,7) DWT and for low quality video.

Rank / Attack	*Error'*							
	$r = 1$	$r = 50$	$r = 100$	$r = 150$	$r = 200$	$r = 250$	$r = 300$	$r = 350$
FMLR	0.062	0.056	0.055	0.054	0.055	0.045	0.046	0.081
Gaussian filtering	0.063	0.057	0.049	0.078	0.035	0.058	0.072	0.033
JPEG	0.061	0.088	0.048	0.055	0.108	0.039	0.068	0.054
Rot +2°	0.119	0.132	0.103	0.087	0.112	0.015	0.097	0.095
Rot -2°	0.114	0.089	0.100	0.097	0.111	0.061	0.084	0.104
Rot +5°	0.085	0.065	0.103	0.133	0.094	0.072	0.133	0.098
Median	0.185	0.155	0.089	0.074	0.084	0.067	0.441	0.080
Sharpening	0.097	0.063	0.107	0.137	0.077	0.066	0.104	0.123
StirMark	0.088	0.109	0.117	0.073	0.087	0.086	0.054	0.066

Table IV.4. The average errors according to *step –C (1)* for the (9,7) DWT and for high quality video.

Rank / Attack	*Error'*							
	$r = 1$	$r = 50$	$r = 100$	$r = 150$	$r = 200$	$r = 250$	$r = 300$	$r = 350$
FMLR	0.065	0.068	0.058	0.068	0.062	0.066	0.064	0.059
Gaussian filtering	0.077	0.078	0.069	0.052	0.061	0.051	0.054	0.136
JPEG	0.082	0.057	0.125	0.052	0.047	0.061	0.049	0.057
Rot +2°	0.126	0.095	0.091	0.103	0.085	0.082	0.084	0.072
Rot -2°	0.082	0.086	0.088	0.151	0.079	0.119	0.141	0.083
Rot +5°	0.076	0.289	0.0225	0.083	0.096	0.080	0.155	0.077
Median	0.135	0.064	0.080	0.127	0.129	0.313	0.062	0.105
Sharpening	0.089	0.081	0.113	0.173	0.076	0.066	0.080	0.070
StirMark	0.107	0.111	0.146	0.117	0.097	0.135	0.100	0.124

Step C-(2): Is the model general or does it depends on the video sequence?

In order to verify if the model computed on a particular video can represent all the video sequences, the *algorithm ART.MOD-A* was repeated for the rest of nine video sequences. The errors between the reference model and these new models are calculated for each attack and each rank based on the three considered measures (see algorithm *step C*) according to eq.(III.7.b), eq.(III.9.a), and eq.(III.9.b) respectively.

The average errors (*Error''*, E_{HL}, E_{KL}) thus obtained are illustrated in Tables IV.5 and IV.6, corresponding to all investigated attacks and to low and high quality video, respectively.

As a general conclusion, it can be said that the *Error''* is acceptably low, lower then 15% for most of the attacks and ranks with some exceptions:

- for low quality rate: for rank $r = 1$ namely, the FMLR *Error''* = 41.4%, the Gaussian filtering *Error''* = 16.8% , the median *Error''* = 15.6%. However, for JPEG all the values are higher then 15% with one exception for $r = 300$.
- for high quality rate : the rotation with 2° when the value is *Error''* = 15.9% for the rank $r = 1$; and rotation with 5° when the value is *Error''* = 19.6% for rank $r = 1$, *Error''* = 20.9% for rank $r = 50$ and *Error''* = 16.9% for rank $r = 100$.

The Kullback-Leibler divergence and the Hellinger distance lead to acceptably small values for all the attacks, with the same presented exceptions when the values become slightly large.

Table IV.5. Average errors according to *step-C(2)* for different ranks and attacks in the (9,7) DWT (low quality video).

Rank	FMLR			Gaussian filtering		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.414	0.191	0.306	0.168	0.089	0.549
$r = 50$	0.151	0.075	0.182	0.132	0.026	0.262
$r = 100$	0.034	0.013	0.082	0.064	0.006	0.031
$r = 150$	0.014	0.007	0.039	0.111	0.008	0.054
$r = 200$	0.011	0.005	0.036	0.017	0.001	0.018
$r = 250$	0.010	0.004	0.020	0.070	0.001	0.007
$r = 300$	0.017	0.003	0.006	0.011	0.001	0.002
$r = 350$	0.014	0.001	0.020	0.011	0.001	0.026
Rank	**JPEG**			**Rot +2**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.992	0.078	0.202	0.083	0.093	0.599
$r = 50$	0.961	0.029	0.037	0.064	0.019	0.167
$r = 100$	0.957	0.015	0.021	0.092	0.017	0.120
$r = 150$	0.929	0.008	0.008	0.108	0.010	0.081
$r = 200$	0.542	0.003	0.003	0.085	0.005	0.072
$r = 250$	0.533	0.007	0.004	0.029	0.006	0.056
$r = 300$	0.038	0.001	0.002	0.035	0.002	0.007
$r = 350$	0.257	0.001	0.001	0.012	0.002	0.013
Rank	**Rot - 2**			**Rot + 5**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.089	0.076	0.465	0.126	0.153	0.353
$r = 50$	0.061	0.020	0.169	0.090	0.023	0.222
$r = 100$	0.058	0.008	0.098	0.092	0.020	0.146
$r = 150$	0.069	0.008	0.108	0.059	0.008	0.098
$r = 200$	0.041	0.004	0.051	0.050	0.004	0.052
$r = 250$	0.018	0.003	0.048	0.039	0.002	0.024
$r = 300$	0.020	0.001	0.002	0.026	0.002	0.018
$r = 350$	0.011	0.001	0.001	0.021	0.001	0.002
Rank	**Median**			**Sharpening**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.156	0.028	0.093	0.095	0.023	0.131
$r = 50$	0.143	0.014	0.057	0.065	0.010	0.125
$r = 100$	0.062	0.002	0.013	0.073	0.006	0.068
$r = 150$	0.030	0.003	0.018	0.058	0.003	0.042
$r = 200$	0.018	0.001	0.016	0.038	0.001	0.012
$r = 250$	0.006	0.001	0.006	0.038	0.001	0.024
$r = 300$	0.008	0.001	0.002	0.034	0.001	0.013
$r = 350$	0.005	0.002	0.021	0.023	0.001	0.003
Rank	**StirMark**					
	Error''	E_{HL}	E_{KL}			
$r = 1$	0.037	0.008	0.093			
$r = 50$	0.037	0.003	0.048			
$r = 100$	0.036	0.002	0.019			
$r = 150$	0.022	0.002	0.023			
$r = 200$	0.018	0.001	0.005			
$r = 250$	0.014	0.001	0.006			
$r = 300$	0.031	0.001	0.006			
$r = 350$	0.014	0.001	0.002			

Table IV.6. Average errors according to *step-C(2)* for different ranks and attacks in the (9,7) DWT (high quality video).

Rank	FMLR			Gaussian filtering		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.011	0.019	0.105	0.148	0.099	0.549
$r = 50$	0.023	0.010	0.159	0.078	0.044	0.218
$r = 100$	0.009	0.007	0.101	0.094	0.031	0.283
$r = 150$	0.010	0.006	0.092	0.092	0.028	0.232
$r = 200$	0.018	0.013	0.106	0.077	0.004	0.023
$r = 250$	0.017	0.014	0.092	0.063	0.008	0.083
$r = 300$	0.024	0.010	0.116	0.058	0.014	0.189
$r = 350$	0.025	0.012	0.091	0.041	0.011	0.099
Rank	**JPEG**			**Rot +2**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.011	0.001	0.004	0.159	0.119	0.673
$r = 50$	0.009	0.001	0.002	0.103	0.031	0.229
$r = 100$	0.027	0.001	0.010	0.118	0.031	0.230
$r = 150$	0.010	0.001	0.002	0.084	0.022	0.139
$r = 200$	0.021	0.001	0.001	0.084	0.024	0.177
$r = 250$	0.009	0.001	0.001	0.099	0.018	0.141
$r = 300$	0.015	0.001	0.003	0.083	0.015	0.133
$r = 350$	0.013	0.001	0.005	0.074	0.015	0.0138
Rank	**Rot - 2**			**Rot + 5**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.136	0.096	0.551	0.196	0.142	0.683
$r = 50$	0.093	0.039	0.252	0.209	0.081	0.336
$r = 100$	0.124	0.028	0.192	0.168	0.060	0.295
$r = 150$	0.120	0.030	0.223	0.119	0.034	0.204
$r = 200$	0.143	0.041	0.422	0.129	0.026	0.177
$r = 250$	0.084	0.019	0.119	0.111	0.019	0.137
$r = 300$	0.108	0.024	0.135	0.140	0.022	0.149
$r = 350$	0.079	0.009	0.101	0.125	0.028	0.258
Rank	**Median**			**Sharpening**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.015	0.006	0.034	0.053	0.017	0.064
$r = 50$	0.004	0.002	0.017	0.023	0.021	0.329
$r = 100$	0.018	0.002	0.018	0.022	0.011	0.110
$r = 150$	0.011	0.003	0.024	0.029	0.013	0.116
$r = 200$	0.011	0.003	0.013	0.036	0.010	0.141
$r = 250$	0.019	0.003	0.032	0.020	0.008	0.125
$r = 300$	0.015	0.001	0.002	0.030	0.007	0.098
$r = 350$	0.011	0.002	0.004	0.027	0.013	0.191
Rank	**StirMark**					
	Error'	E_{HL}	E_{KL}			
$r = 1$	0.037	0.035	0.271			
$r = 50$	0.032	0.012	0.104			
$r = 100$	0.042	0.012	0.096			
$r = 150$	0.040	0.009	0.108			
$r = 200$	0.051	0.008	0.122			
$r = 250$	0.049	0.005	0.053			
$r = 300$	0.043	0.007	0.088			
$r = 350$	0.028	0.008	0.108			

Step C-(3): Is the model dependent on the EM Gaussian mixtures estimation?

In order to check out if the model is independent with respect of the estimation procedure, the confidence limits for probability were considered.

The integrals of the *pdf* on 10 subintervals where the corresponding *pdf*s take non-zero values were computed and each and every time it was verify that they belong to the corresponding confidence limits (see Figures IV.7 and IV.8 for rank $r = 1$ and for low and high quality video respectively).

This is a strong support for the generality of the IVM model: the *pdf* estimation obtained by a maximum likelihood procedure proves to be representative also from the confidence estimation meaning.

For the interpretation of the errors in confidence limit estimation, see the Note in the corresponding section for DWT video modelling, *i.e.* Section III.3.2.1.2 – *Step C-(3)*.

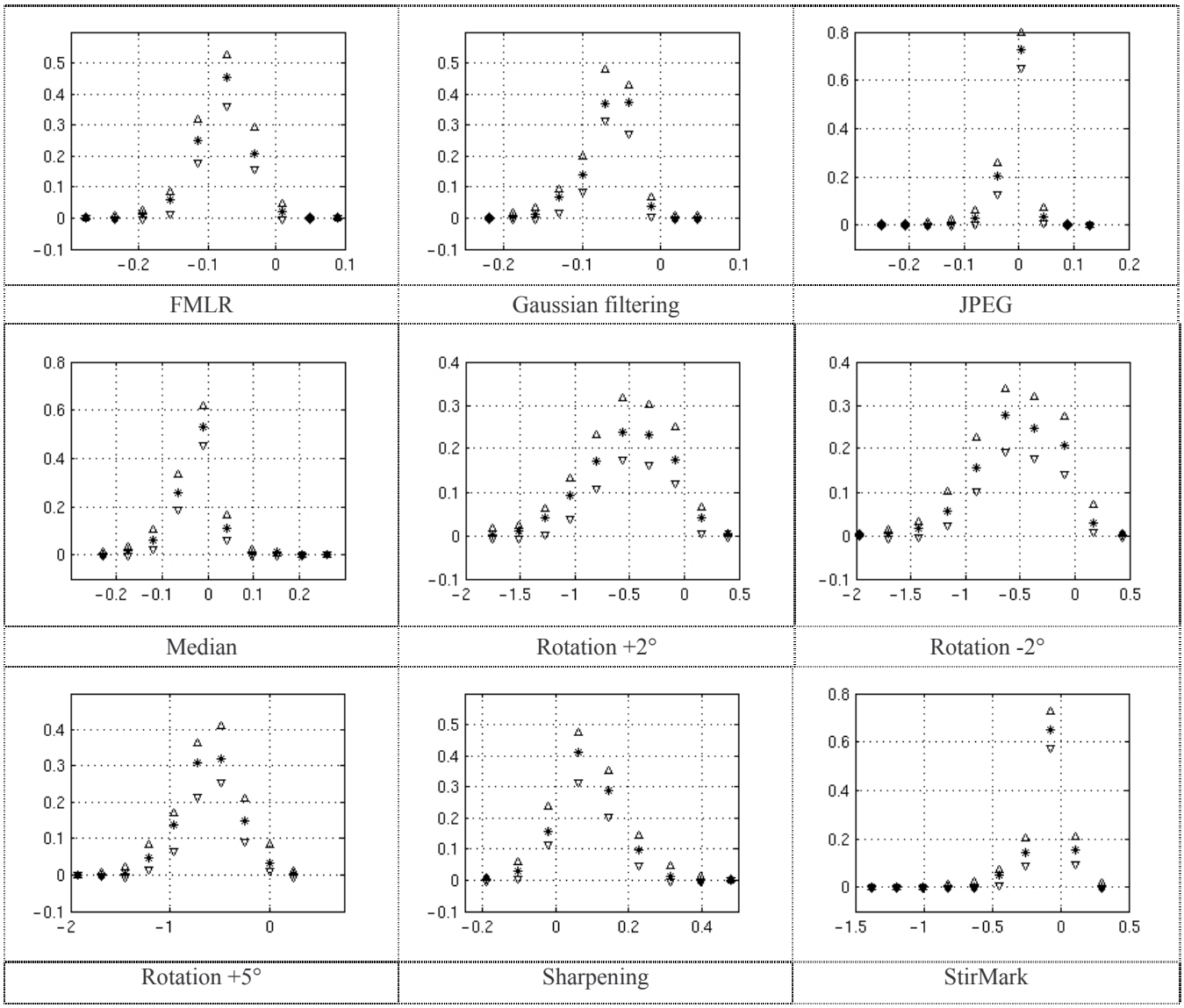


Figure IV.7. IVM model validation for (9,7) DWT ($r = 1$, video coded at low quality rate and nine different attacks): the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

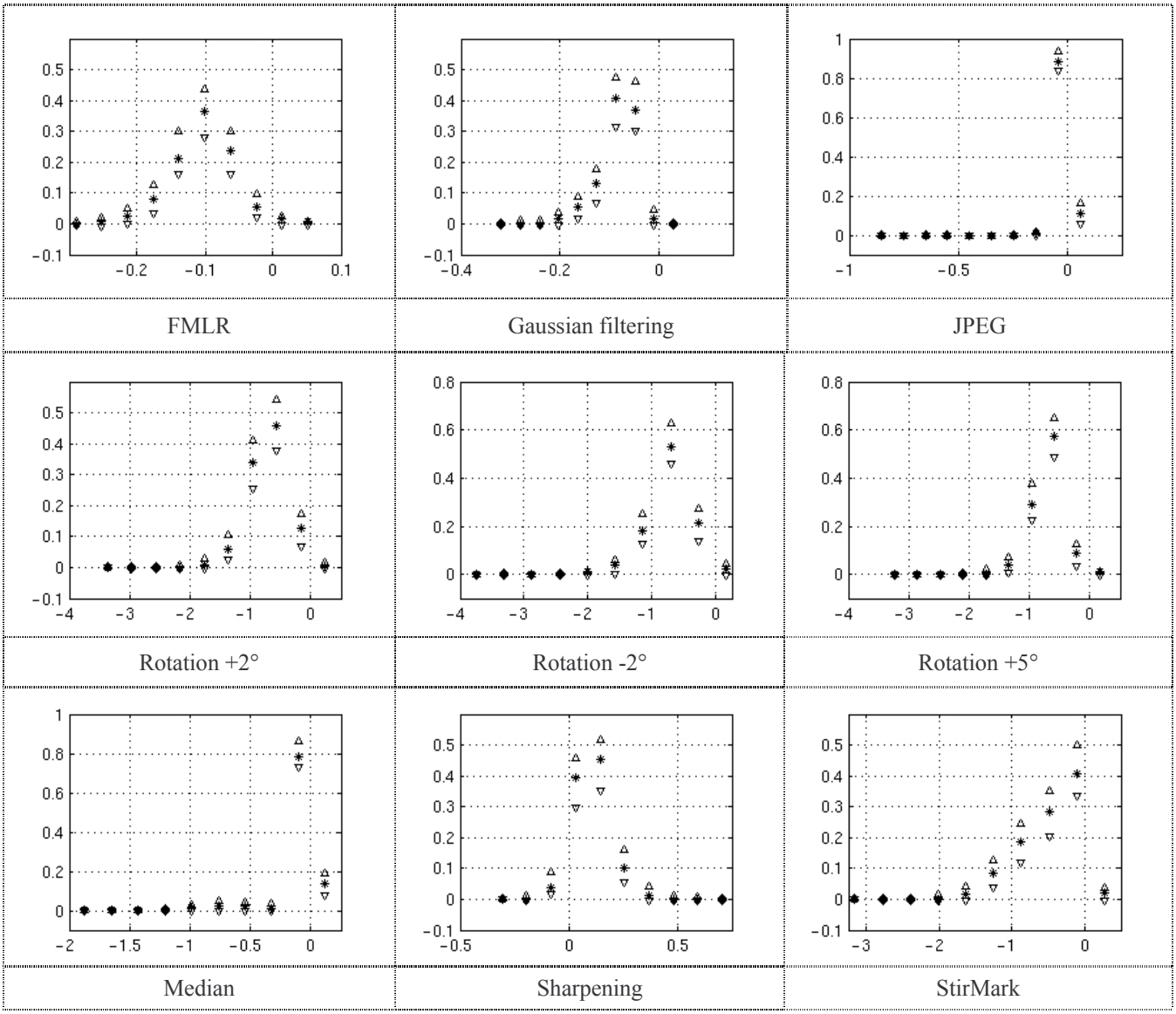


Figure IV.8. IVM model validation for (9,7) DWT, (for $r = 1$, video coded at high quality rate and different attacks -): the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

To conclude with, we proved that in the (9,7) DWT domain an individual model (IVM) computed on a particular video sequence, is valuable for all the video sequences which are involved in the experiments, and moreover that it does not depend either on the estimation procedure and or on the similarity measure with which the model was computed.

IV.3.2.1.3. Comparison among (2,2), (4,4) and (9,7) DWT

Tables IV.7 and IV.8 display the parameters of the IVM model and the *Error*s and for low quality video for two selected ranks ($r = 1$ and $r = 150$, respectively) for three DWTs (namely (2,2), (4,4), and (9,7)). Similar results were obtained for the rest of the ranks in the video coded at low quality rate case.

The *Error*s reported in these tables are lower then 6% for the $r = 1$ and lower then 2% for the rank $r = 150$, with no exception.

The models (*i.e.* the Gaussian mixtures) obtained for rank $r = 150$ and for six attacks (Gaussian filtering, sharpening, rotations $\pm 2°$, $+5°$, and StirMark), are plotted in Figure IV.9: very small differences (with one exception, rotation $+5°$ in the case of (4,4) DWT transform) among the three types of transforms can be noticed.

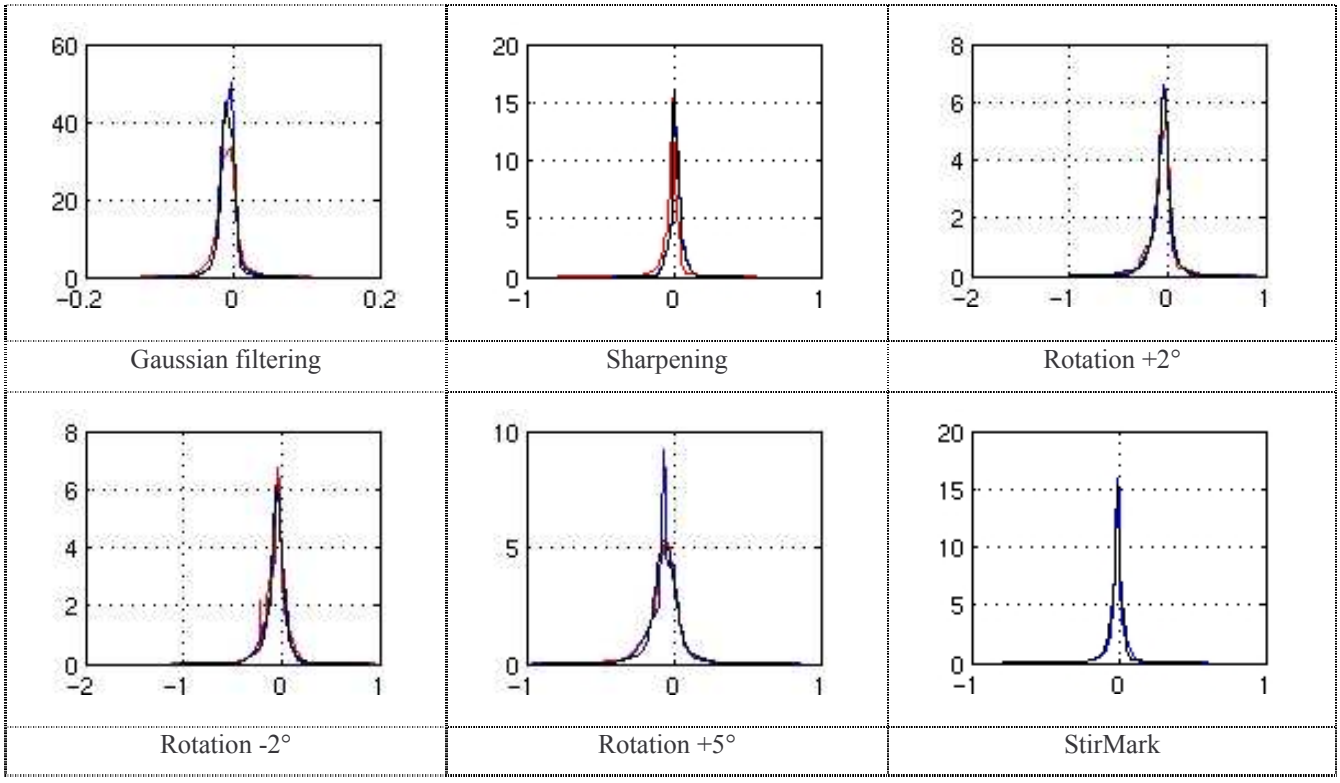


Figure IV.9. Each plot contains three graphics, corresponding to the (2,2) DWT– in red , (4,4) DWT – in bleu and (9,7) DWT – in black, computed for the rank r = 150, for six attacks, and for low quality video.

Table IV.7. IVM model for (2,2), (4,4) and (9,7) DWT coefficient hierarchy, $r = 1$ and low quality video.

		Model parameters											Error
Gaussian filtering	(2,2)	$P(k)$	0.047	0.038	0.073	0.221	0.056	0.052	0.188	0.039	0.174	0.113	0.032
		$\mu(k)$	-0.181	-0.131	-0.129	-0.071	-0.195	-0.127	-0.053	-0.190	-0.079	-0.107	
		$\sigma(k)$	0.047	0.022	0.024	0.022	0.045	0.026	0.019	0.045	0.018	0.032	
	(4,4)	$P(k)$	0.066	0.090	0.149	0.192	0.147	0.139	0.053	0.057	0.055	0.051	0.034
		$\mu(k)$	-0.107	-0.146	-0.082	-0.067	-0.046	-0.075	-0.110	-0.106	-0.078	-0.109	
		$\sigma(k)$	0.041	0.034	0.023	0.016	0.010	0.031	0.042	0.041	0.033	0.041	
	(9,7)	$P(k)$	0.015	0.199	0.119	0.076	0.027	0.046	0.078	0.338	0.060	0.042	0.036
		$\mu(k)$	-0.021	-0.061	-0.067	-0.082	-0.153	-0.082	-0.078	-0.046	-0.115	-0.082	
		$\sigma(k)$	0.001	0.015	0.015	0.020	0.023	0.019	0.019	0.012	0.003	0.020	
Sharpening	(2,2)	$P(k)$	0.067	0.147	0.066	0.182	0.057	0.067	0.172	0.042	0.135	0.064	0.035
		$\mu(k)$	0.221	0.052	0.101	0.094	0.317	0.349	0.066	0.186	0.158	0.113	
		$\sigma(k)$	0.120	0.082	0.083	0.082	0.098	0.084	0.085	0.120	0.063	0.085	
	(4,4)	$P(k)$	0.096	0.131	0.058	0.058	0.088	0.129	0.123	0.129	0.131	0.057	0.037
		$\mu(k)$	0.158	0.115	0.104	0.146	0.082	0.025	0.284	0.107	0.083	0.048	
		$\sigma(k)$	0.092	0.075	0.115	0.096	0.119	0.052	0.071	0.072	0.061	0.056	
	(9,7)	$P(k)$	0.022	0.024	0.107	0.067	0.194	0.014	0.110	0.087	0.100	0.274	0.036
		$\mu(k)$	0.101	-0.029	-0.078	0.071	0.040	0.189	0.148	0.158	0.142	0.059	
		$\sigma(k)$	0.001	0.001	0.104	0.104	0.056	0.002	0.092	0.090	0.015	0.052	
Rotation +2°	(2,2)	$P(k)$	0.148	0.068	0.081	0.170	0.046	0.078	0.080	0.169	0.113	0.046	0.047
		$\mu(k)$	-0.287	-0.931	-0.887	-0.053	-0.765	-1.074	-0.970	-0.481	-0.656	-0.991	
		$\sigma(k)$	0.079	0.294	0.301	0.087	0.010	0.256	0.286	0.057	0.083	0.015	
	(4,4)	$P(k)$	0.014	0.194	0.167	0.084	0.028	0.110	0.124	0.132	0.105	0.043	0.047
		$\mu(k)$	0.118	-0.875	-0.232	-0.756	-0.477	-0.517	-0.030	-0.493	-0.574	-1.007	
		$\sigma(k)$	0.003	0.212	0.102	0.239	0.006	0.169	0.059	0.172	0.182	0.420	
	(9,7)	$P(k)$	0.105	0.102	0.078	0.109	0.115	0.098	0.108	0.100	0.062	0.122	0.041
		$\mu(k)$	-0.446	-0.516	-0.517	-0.798	-0.388	-0.176	-0.930	-0.548	-0.671	-0.039	
		$\sigma(k)$	0.186	0.202	0.202	0.279	0.160	0.112	0.267	0.206	0.231	0.082	
Rotation -2°	(2,2)	$P(k)$	0.099	0.038	0.171	0.075	0.007	0.089	0.106	0.224	0.095	0.096	0.048
		$\mu(k)$	-0.811	-0.625	-0.463	-1.096	-0.759	-0.682	-0.771	-0.113	-0.707	-0.851	
		$\sigma(k)$	0.264	0.014	0.102	0.402	0.002	0.255	0.262	0.108	0.256	0.262	
	(4,4)	$P(k)$	0.128	0.046	0.152	0.045	0.134	0.105	0.043	0.115	0.135	0.098	0.049
		$\mu(k)$	-0.361	-0.938	-0.458	-1.113	-0.079	-0.657	-0.986	-0.553	-0.642	-0.680	
		$\sigma(k)$	0.171	0.369	0.222	0.351	0.094	0.249	0.371	0.246	0.251	0.243	
	(9,7)	$P(k)$	0.142	0.049	0.192	0.083	0.080	0.098	0.116	0.073	0.161	0.007	0.052
		$\mu(k)$	-0.402	-0.014	-0.620	-0.601	-0.335	-0.822	-0.433	-0.999	-0.112	-2.068	
		$\sigma(k)$	0.244	0.009	0.182	0.268	0.234	0.247	0.249	0.209	0.186	0.001	
Rotation +5°	(2,2)	$P(k)$	0.079	0.111	0.156	0.119	0.118	0.080	0.077	0.117	0.045	0.098	0.041
		$\mu(k)$	-1.155	-0.667	-0.581	-0.682	-0.645	-0.781	-0.658	-0.554	-0.095	-0.594	
		$\sigma(k)$	0.142	0.232	0.205	0.193	0.231	0.217	0.234	0.221	0.110	0.227	
	(4,4)	$P(k)$	0.163	0.133	0.096	0.101	0.098	0.046	0.144	0.094	0.112	0.013	0.041
		$\mu(k)$	-0.804	-0.538	-0.529	-0.719	-0.397	-1.213	-0.513	-0.450	-0.449	-0.158	
		$\sigma(k)$	0.154	0.241	0.063	0.210	0.195	0.090	0.237	0.232	0.231	0.037	
	(9,7)	$P(k)$	0.010	0.014	0.175	0.088	0.194	0.090	0.087	0.105	0.108	0.128	0.039
		$\mu(k)$	-0.882	-0.710	-0.551	-0.383	-0.542	-0.875	-0.793	-0.649	-0.209	-0.709	
		$\sigma(k)$	0.001	0.001	0.152	0.127	0.155	0.213	0.232	0.234	0.130	0.242	
StirMark	(2,2)	$P(k)$	0.085	0.076	0.054	0.335	0.142	0.141	0.024	0.120	0.014	0.010	0.021
		$\mu(k)$	-0.074	-0.215	-0.544	-0.044	-0.117	-0.070	0.065	-0.125	-1.056	-0.341	
		$\sigma(k)$	0.171	0.051	0.123	0.053	0.181	0.055	0.001	0.182	0.068	0.003	
	(4,4)	$P(k)$	0.021	0.023	0.113	0.089	0.281	0.053	0.096	0.146	0.097	0.082	0.019
		$\mu(k)$	-0.611	-0.219	-0.072	-0.251	-0.010	-0.285	-0.133	-0.091	-0.109	-0.058	
		$\sigma(k)$	0.292	0.002	0.120	0.192	0.035	0.188	0.113	0.037	0.123	0.134	
	(9,7)	$P(k)$	0.045	0.097	0.120	0.016	0.097	0.102	0.092	0.122	0.185	0.124	0.019
		$\mu(k)$	-0.039	-0.277	-0.155	-0.586	-0.012	-0.101	-0.019	-0.047	0.005	-0.065	
		$\sigma(k)$	0.122	0.146	0.084	0.253	0.013	0.140	0.123	0.062	0.047	0.060	

Table IV.8. IVM model for (2,2), (4,4) and (9,7) DWT coefficient hierarchy, r = 150 and low quality video.

		Model parameters											Error
Gaussian filtering	(2,2)	$P(k)$	0.028	0.092	0.286	0.038	0.032	0.014	0.035	0.180	0.163	0.133	0.015
		$\mu(k)$	-0.008	-0.028	-0.005	-0.011	-0.002	-0.001	-0.022	-0.011	-0.009	-0.008	
		$\sigma(k)$	0.019	0.016	0.007	0.019	0.017	0.017	0.027	0.008	0.008	0.016	
	(4,4)	$P(k)$	0.014	0.158	0.119	0.224	0.156	0.016	0.078	0.168	0.026	0.041	0.014
		$\mu(k)$	0.028	-0.004	-0.005	-0.010	-0.012	0.027	-0.004	-0.007	-0.021	-0.025	
		$\sigma(k)$	0.018	0.001	0.010	0.004	0.007	0.018	0.010	0.010	0.017	0.017	
	(9,7)	$P(k)$	0.016	0.199	0.119	0.075	0.027	0.046	0.078	0.338	0.060	0.042	0.017
		$\mu(k)$	-0.021	-0.061	-0.067	-0.082	-0.153	-0.081	-0.078	-0.046	0.115	-0.082	
		$\sigma(k)$	0.001	0.015	0.015	0.020	0023	0.020	0.019	0.012	0.003	0.020	
Sharpening	(2,2)	$P(k)$	0.099	0.085	0.035	0.251	0.042	0.065	0.093	0.202	0.068	0.060	0.016
		$\mu(k)$	0.085	0.060	-0.002	0.002	-0.001	-0.004	0.013	0.022	-0.002	-0.003	
		$\sigma(k)$	0.067	0.005	0.083	0.017	0.083	0.082	0.083	0.023	0.082	0.082	
	(4,4)	$P(k)$	0.014	0.099	0.111	0.011	0.247	0.147	0.073	0.134	0.078	0.118	0.017
		$\mu(k)$	-0.204	0.047	-0.008	-0.009	0.024	0.017	0.028	0.004	0.030	-0.003	
		$\sigma(k)$	0.032	0.036	0.048	0.047	0.013	0.05	0.057	0.008	0.057	0.049	
	(9,7)	$P(k)$	0.047	0.133	0.117	0.041	0.262	0.061	0.062	0.101	0.143	0.015	0.012
		$\mu(k)$	0.044	0.031	0.032	0.045	0.004	0.034	0.042	0.009	-0.034	0.045	
		$\sigma(k)$	0.051	0.006	0.025	0.051	0.008	0.050	0.051	0.039	0.034	0.051	
Rotation +2°	(2,2)	$P(k)$	0.119	0.031	0.077	0.232	0.024	0.060	0.051	0.289	0.100	0.017	0.016
		$\mu(k)$	-0.054	0.053	-0.069	-0.077	-0.412	-0.101	0.155	-0.001	-0.222	-0.318	
		$\sigma(k)$	0.048	0.141	0.066	0.043	0.165	0.079	0.122	0.036	0.050	0.196	
	(4,4)	$P(k)$	0.044	0.054	0.058	0.118	0.077	0.065	0.055	0.378	0.123	0.030	0.019
		$\mu(k)$	-0.031	0.095	-0.257	-0.106	-0.095	-0.091	-0.052	-0.031	-0.051	-0.409	
		$\sigma(k)$	0.108	0.080	0.146	0.066	0.070	0.079	0.113	0.034	0.067	0.124	
	(9,7)	$P(k)$	0.102	0.037	0.038	0.056	0.106	0.044	0.045	0.353	0.032	0.187	0.015
		$\mu(k)$	-0.128	-0.074	-0.033	0.007	-0.106	-0.065	-0.066	-0.042	-0.070	-0.002	
		$\sigma(k)$	0.080	0.173	0.162	0.150	0.084	0.172	0.172	0.035	0.172	0.051	
Rotation -2°	(2,2)	$P(k)$	0.054	0.085	0.053	0.074	0.060	0.016	0.305	0.180	0.083	0.090	0.014
		$\mu(k)$	-0.176	-0.003	-0.164	0.023	0.004	-0.210	-0.035	-0.119	-0.054	-0.005	
		$\sigma(k)$	0.172	0.085	0.171	0.077	0.151	0.002	0.024	0.037	0.162	0.082	
	(4,4)	$P(k)$	0.245	0.146	0.047	0.109	0.139	0.077	0.104	0.007	0.062	0.064	0.017
		$\mu(k)$	-0.049	-0.077	-0.121	0.019	-0.030	-0.140	-0.016	0.401	-0.127	-0.128	
		$\sigma(k)$	0.029	0.083	0.123	0.040	0.101	0.122	0.055	0.001	0.123	0.123	
	(9,7)	$P(k)$	0.070	0.146	0.046	0.048	0.031	0.018	0.202	0.272	0.156	0.011	0.014
		$\mu(k)$	-0.192	-0.058	-0.171	-0.127	0.181	-0.034	-0.010	-0.086	0.025	-0.023	
		$\sigma(k)$	0.114	0.023	0.119	0.124	0.140	0.160	0.032	0.050	0.067	0.001	
Rotation +5°	(2,2)	$P(k)$	0.040	0.064	0.099	0.189	0.022	0.158	0.047	0.084	0.260	0.036	0.018
		$\mu(k)$	0.010	0.040	-0.062	-0.034	-0.172	-0.165	-0.009	-0.072	-0.046	-0.388	
		$\sigma(k)$	0.121	0.113	0.049	0.051	0.177	0.076	0.124	0.119	0.052	0.150	
	(4,4)	$P(k)$	0.035	0.257	0.102	0.066	0.085	0.060	0.040	0.105	0.107	0.145	0.017
		$\mu(k)$	0.028	-0.015	-0.053	-0.183	-0.085	-0.147	0.107	-0.080	-0.101	-0.068	
		$\sigma(k)$	0.155	0.038	0.090	0.119	0.090	0.125	0.143	0.085	0.077	0.009	
	(9,7)	$P(k)$	0.038	0.139	0.138	0.158	0.036	0.132	0.040	0.061	0.055	0.202	0.017
		$\mu(k)$	-0.149	-0.038	-0.033	-0.056	-0.118	-0.030	-0.135	-0.138	-0.135	-0.089	
		$\sigma(k)$	0.140	0.062	0.062	0.059	0.146	0.062	0.143	0.091	0.143	0.047	
StirMark	(2,2)	$P(k)$	0.027	0.048	0.296	0.100	0.063	0.230	0.158	0.029	0.020	0.030	0.009
		$\mu(k)$	0.170	-0.121	-0.010	-0.039	-0.069	0.021	-0.035	0.052	-0.036	0.051	
		$\sigma(k)$	0.135	0.070	0.009	0.030	0.050	0.019	0.031	0.099	0.001	0.010	
	(4,4)	$P(k)$	0.253	0.080	0.138	0.086	0.104	0.036	0.046	0.042	0.096	0.119	0.010
		$\mu(k)$	-0.004	-0.023	-0.024	-0.022	0.020	-0.127	0.032	-0.084	0.010	0.010	
		$\sigma(k)$	0.010	0.054	0.017	0.020	0.045	0.090	0.125	0.098	0.054	0.050	
	(9,7)	$P(k)$	0.215	0.081	0.022	0.043	0.287	0.046	0.102	0.138	0.038	0.028	0.008
		$\mu(k)$	0.001	0.003	-0.092	-0.032	-0.017	-0.002	-0.003	0.003	-0.020	-0.055	
		$\sigma(k)$	0.024	0.068	0.092	0.053	0.020	0.067	0.067	0.002	0.060	0.090	

IV.3.2.2. The 2D-DCT coefficients modelling

IV.3.2.2.1. Model Computation

Table IV.9 present the model parameters and the *Error*s corresponding to the DCT applied on an individual video sequence (IVM) coded at low quality.

FigureIV.10 depicts with continuous line the attack models and with dashed line the Gaussian *pdf* (with the same mean values and variances as the computed models). The illustration is realized for the rank $r = 1$ for the Gaussian filtering, the sharpening and the StirMark attack.

Table IV.10 presents the same type of information as the Table IV.9, this time obtained for the video of high quality.

Each and every time the procedure was repeated nine times (one model for each video available in the video corpus was obtained) for each attack, rank and for low and high quality rates (see Appendix III and the accompanying CD-ROM).

As a general observation, in each case (each attack, rank and quality) the *Error* values are lower then 5%, with no exception.

Table IV.9. IVM for the DCT coefficient hierarchy, $r = 1$, low quality video and different attacks.

Attacks	Model parameters											Error
Gaussian filtering	*P(k)*	0.118	0.062	0.091	0.102	0.105	0.101	0.047	0.054	0.212	0.109	0.041
	μ(k)	0.465	0.502	0.284	0.255	0.316	0.151	0.426	0.349	0.216	0.121	
	σ(k)	0.089	0.174	0.120	0.118	0.119	0.099	0.170	0.075	0.039	0.092	
JPEG	*P(k)*	0.195	0.139	0.094	0.077	0.089	0.071	0.137	0.109	0.044	0.044	0.028
	μ(k)	-0.016	-0.067	-0.005	0.008	-0.068	-0.006	-0.098	-0.048	-0.219	-0.130	
	σ(k)	0.044	0.059	0.104	0.102	0.092	0.105	0.063	0.053	0.056	0.004	
Rot +2°	*P(k)*	0.101	0.112	0.099	0.078	0.099	0.104	0.093	0.105	0.105	0.104	0.029
	μ(k)	-0.408	-1.888	1.216	-0.802	-0.541	-1.070	-1.234	-0.481	-0.953	1.505	
	σ(k)	0.956	1.098	0.645	1.087	0.994	1.163	0.648	0.971	1.111	0.824	
Rot -2°	*P(k)*	0.131	0.084	0.101	0.085	0.108	0.098	0.101	0.091	0.084	0.115	0.029
	μ(k)	-0.814	-1.813	-1.554	0.013	1.151	-0.494	-0.894	-0.436	0.385	-1.705	
	σ(k)	0.556	0.999	1.016	0.883	0.667	0.976	1.054	0.965	0.828	0.977	
Rot +5°	*P(k)*	0.078	0.088	0.095	0.084	0.071	0.122	0.162	0.081	0.084	0.136	0.032
	μ(k)	1.379	-2.651	-3.688	-2.059	1.267	-3.004	-0.675	-0.404	-1.082	-1.191	
	σ(k)	2.437	2.868	2.667	2.954	2.462	1.117	1.569	1.708	2.013	1.720	
Median	*P(k)*	0.106	0.156	0.142	0.079	0.106	0.029	0.146	0.056	0.087	0.092	0.020
	μ(k)	-60.51	-59.59	-89.95	-55.61	-60.34	-39.42	-60.02	-36.29	-76.15	-67.05	
	σ(k)	10.19	6.797	6.792	9.509	1.102	5.806	7.097	5.662	1.325	1.287	
Sharpening	*P(k)*	0.101	0.037	0.157	0.088	0.079	0.078	0.072	0.167	0121	0.099	0.021
	μ(k)	-1.169	-3.381	-1.937	-1.246	-1.436	-1.201	-1.400	-1/809	-2.707	-1.965	
	σ(k)	0.870	0.537	0.503	0.883	0.090	0.876	0.329	0.475	0.299	0.521	
StirMark	*P(k)*	0.144	0.169	0.121	0.078	0.114	0.089	0.108	0.068	0.056	0.054	0.039
	μ(k)	-0.201	-0.278	-0.276	-0.384	-0.196	-0.510	-0.163	-0.334	-0.667	0.300	
	σ(k)	0.293	0.264	0.300	0.296	0.294	0.162	0.284	0.350	0.423	0.554	

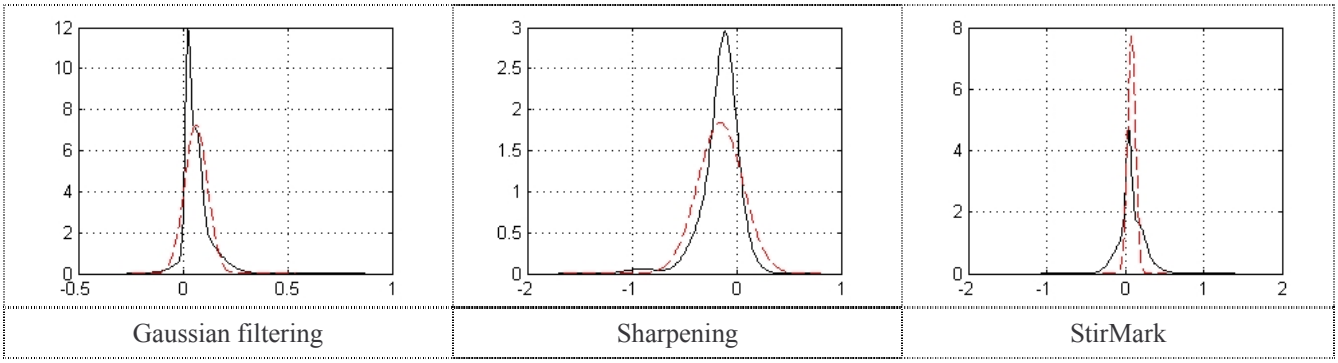


Figure IV.10. The attack models (continuous line) and the corresponding Gaussian distributions (dashed line), for rank $r = 1$ low quality video, and for Gaussian filtering, sharpening, and StirMark attack.

Table IV.10. IVM for the DCT coefficient hierarchy, $r = 1$, high quality video, and different attacks.

Attacks	Model parameters											Error
Gaussian filtering	*P(k)*	0.125	0.096	0.098	0.057	0.157	0.118	0.109	0.044	0.061	0.134	0.043
	μ(k)	-0.444	-0.250	-0.249	-0.595	-0.315	-0.277	-0.256	0.066	-0.203	-0.076	
	σ(k)	0.063	0.107	0.107	0.015	0.032	0.109	0.108	0.153	0.093	0.071	
JPEG	*P(k)*	0.079	0.147	0.091	0.074	0.167	0.016	0.117	0.082	0.106	0.122	0.001
	μ(k)	0.106	-0.078	0.018	-0.013	0.029	0.080	0.097	-0.276	-0.050	0.008	
	σ(k)	0.219	0.040	0.155	0.172	0.131	0.007	0.061	0.270	0.165	0.155	
Rot +2°	*P(k)*	0.097	0.121	0.056	0.077	0.140	0.087	0.057	0.122	0.099	0.142	0.029
	μ(k)	-2.206	-3.308	-9.665	-4.512	-0.812	3.916	-5.936	-2.826	-0.276	-2.166	
	σ(k)	2.991	1.741	3.861	2.477	2.167	2.652	4.546	2.463	3.296	2.452	
Rot -2°	*P(k)*	0.088	0.115	0.109	0.080	0.106	0.112	0.109	0.094	0.089	0.099	0.045
	μ(k)	-3.301	0.533	-2.203	-5.328	-0.287	0.275	-2.476	-2.842	-3.848	-2.310	
	σ(k)	3.832	3.879	3.158	1.747	3.918	3.897	3.312	3.769	2.806	3.595	
Rot +5°	*P(k)*	0.165	0.089	0.128	0.102	0.056	0.099	0.108	0.065	0.133	0.071	0.024
	μ(k)	-8.653	-2.287	-3.337	-1.522	8.324	-1.705	-2.523	3.544	-6.088	-7.996	
	σ(k)	4.108	6.203	5.589	5.711	5.333	7.882	5.655	6.012	4.966	9.399	
Median	*P(k)*	0.118	0.083	0.017	0.075	0.095	0.103	0.138	0.036	0.109	0.224	0.002
	μ(k)	2.257	3.491	6.754	3.441	3.514	3.414	3.293	-8.559	3.345	2.198	
	σ(k)	0.613	0.890	0.634	0.904	0.884	0.910	0.925	6.021	0.921	0.632	
Sharpening	*P(k)*	0.081	0.081	0.180	0.029	0.131	0.156	0.077	0.091	0.088	0.085	0.018
	μ(k)	-1.921	-2.014	-2.045	-0.221	-1.433	-2.842	-1.846	-2.809	-2.245	-1.657	
	σ(k)	1.083	0.821	0.169	1.900	0.205	0.210	0.930	1.178	0.812	0.999	
StirMark	*P(k)*	0.146	0.088	0.099	0.102	0.099	0.098	0.094	0.069	0.113	0.091	0.035
	μ(k)	-1.706	-0.316	-1.773	-1.323	-1.053	-1.532	0.137	-2.167	-1.341	-2.525	
	σ(k)	0.726	0.836	0.850	0.916	0.927	0.981	0.733	1.118	0.790	1.059	

IV.3.2.2.2. Model Validation

Up to this moment, the experimental results point to the existence of a model for the attack effects on a particular video sequence and estimate this model.

The IVM models previously obtained for the DCT are now investigated according to the three methods presented in *step C* of the proposed *ART.MOD-A* algorithm.

Step C-(1): Is the model dependent on the similarity measure?

In order to verify if the model is independent with respect to the similarity measure, the average error is computed with eq. (III.6.b) for different attacks and the results are displayed in Table IV.11 for low quality video and in Table IV.12 for high quality video. It should be notice that the average errors (*Error'*) for the IVM are lower then the considered threshold ($\varepsilon' = 15\%$) with the following exceptions:

- the rotation with +2° and -2°for rank $r = 300$, the median for $r = 350$ and sharpening for $r = 200$ (see Table IV.11 – low quality rate);
- the Stir Mark attack for rank $r = 100$ (see Table IV.12 – high quality rate).

Table IV.11. The average errors according to *step –C (1)* for the DCT and for low quality video.

Rank / Attack	*Error'*							
	$r = 1$	$r = 50$	$r = 100$	$r = 150$	$r = 200$	$r = 250$	$r = 300$	$r = 350$
Gaussian filtering	0.067	0.062	0.088	0.112	0.089	0.131	0.094	0.076
JPEG	0.069	0.081	0.059	0.071	0.073	0.067	0.093	0.076
Rot +2°	0.053	0.102	0.091	0.076	0.145	0.093	0.253	0.112
Rot -2°	0.047	0.099	0.082	0.093	0.137	0.087	0.194	0.109
Rot +5°	0.044	0.070	0.087	0.119	0.107	0.144	0.134	0.075
Median	0.033	0.090	0.076	0.087	0.077	0.074	0.086	0.181
Sharpening	0.060	0.102	0.102	0.097	0.199	0.074	0.105	0.090
StirMark	0.079	0.099	0.093	0.074	0.137	0.089	0.081	0.082

Table IV.12. The average errors according to *step –C (1)* for the DCT and for high quality video.

Rank / Attack	*Error'*							
	$r = 1$	$r = 50$	$r = 100$	$r = 150$	$r = 200$	$r = 250$	$r = 300$	$r = 350$
Gaussian filtering	0.094	0.165	0.100	0.076	0.051	0.062	0.082	0.100
JPEG	0.100	0.081	0.139	0.093	0.151	0.104	0.104	0.103
Rot +2°	0.042	0.053	0.054	0.102	0.061	0.075	0.112	0.065
Rot -2°	0.058	0.053	0.055	0.057	0.075	0.072	0.116	0.074
Rot +5°	0.030	0.073	0.071	0.070	0.071	0.137	0.062	0.083
Median	0.019	0.112	0.078	0.088	0.065	0.112	0.062	0.088
Sharpening	0.076	0.094	0.117	0.095	0.095	0.098	0.072	0.088
StirMark	0.062	0.082	0.184	0.134	0.077	0.145	0.185	0.097

Step C-(2): Is the model general or does it depends on the video sequence?

Here, it is necessary to precise if the model computed on a particular video sequence can be representative for the whole corpus or not. With this aim, the investigation *algorithm ART.MOD-A* (*step-A* and *step-B*) is resumed on the rest of nine video sequences from the video corpus and the corresponding models are computed. The errors between the reference model and these new models are evaluated according to the validation criteria (the similarity measure, the Hellinger distance, and the Kullback-Leibler divergence) and after that are averaged (*Error''* – eq. (III.7.b), E_{HL} – eq. (III.9.a), and E_{KL} – eq. (III.9.b)). For each criterion, attacks, and ranks the average errors are reported in Tables IV.13 and IV.14, for low and high quality video, respectively.

Table IV.13. Average errors according to *step-C (2)* for different ranks and attacks (on low quality video).

Rank	Gaussian filtering			JPEG		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.365	0.072	0.247	0.092	0.002	0.010
$r = 50$	0.025	0.005	0.058	0.044	0.003	0.019
$r = 100$	0.072	0.013	0.114	0.063	0.002	0.012
$r = 150$	0.032	0.009	0.058	0.077	0.003	0.023
$r = 200$	0.047	0.011	0.095	0.069	0.005	0.034
$r = 250$	0.037	0.012	0.095	0.086	0.002	0.020
$r = 300$	0.034	0.013	0.096	0.064	0.002	0.015
$r = 350$	0.083	0.008	0.053	0.089	0.001	0.007
Rank	**Rot +2**			**Rot -2**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.065	0.029	0.204	0.074	0.042	0.174
$r = 50$	0.067	0.001	0.011	0.104	0.001	0.012
$r = 100$	0.087	0.003	0.023	0.114	0.001	0.007
$r = 150$	0.065	0.002	0.015	0.070	0.003	0.018
$r = 200$	0.103	0.003	0.022	0.069	0.001	0.009
$r = 250$	0.036	0.004	0.029	0.045	0.006	0.033
$r = 300$	0.055	0.006	0.040	0.054	0.007	0.045
$r = 350$	0.065	0.004	0.035	0.078	0.005	0.045
Rank	**Rot +5**			**Median**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.074	0.056	0.263	0.801	0.010	0.742
$r = 50$	0.117	0.002	0.019	0.696	0.023	0.485
$r = 100$	0.046	0.001	0.006	0.865	0.036	0.749
$r = 150$	0.059	0.002	0.019	0.921	0.062	0.802
$r = 200$	0.112	0.003	0.018	0.907	0.063	0.670
$r = 250$	0.104	0.003	0.020	0.916	0.071	0.808
$r = 300$	0.008	0.002	0.020	0.951	0.096	0.831
$r = 350$	0.064	0.002	0.017	0.953	0.092	0.799
Rank	**Sharpening**			**StirMark**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.073	0.221	0.477	0.040	0.028	0.199
$r = 50$	0.021	0.008	0.080	0.026	0.001	0.011
$r = 100$	0.020	0.016	0.131	0.035	0.002	0.023
$r = 150$	0.011	0.006	0.054	0.045	0.002	0.037
$r = 200$	0.022	0.038	0.259	0.048	0.004	0.043
$r = 250$	0.012	0.011	0.112	0.031	0.003	0.033
$r = 300$	0.018	0.010	0.100	0.059	0.005	0.059
$r = 350$	0.031	0.011	0.131	0.052	0.004	0.042

The numerical values reported in Tables IV.13 and IV.14 ascertain a quite good accuracy and generality for the model provided in Table IV.9 and IV.10.

- The values found in the case of low quality video (see Table IV.13):

  - the average *Error''* are lower than the chosen threshold ($\varepsilon'' = 15\%$), with the following exceptions: the Gaussian filtering when the *Error''* is 36.5% for rank $r = 1$, and median when all values (for all investigated ranks) are higher than 15%.

  - the Kullback-Leibler divergence and the Hellinger distance lead to acceptably small values for all the attacks, with one exception for median when the E_{KL} divergences are not so small.

- The values find in the case of video coded at high quality rate (see Table IV.14):

  - for the first similarity measure, one exception can be reported for Gaussian filtering, and rank $r = 1$, when the *Error''* is 35.1%.

  - the E_{HL} distance lead to acceptably small values for all the attacks with one exception namely, Gaussian filtering when $r = 1$. However, the E_{KL} divergences are not so small for rank $r = 1$ and for all attacks.

Table IV.14. Average errors according to *step-C (2)* for different ranks and attacks (on high quality video).

Rank	**Gaussian filtering**			**JPEG**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.351	0.470	0.680	0.010	0.001	0.020
$r = 50$	0.048	0.005	0.046	0.010	0.003	0.019
$r = 100$	0.005	0.003	0.010	0.010	0.003	0.028
$r = 150$	0.003	0.004	0.035	0.020	0.005	0.036
$r = 200$	0.011	0.004	0.048	0.010	0.004	0.031
$r = 250$	0.002	0.003	0.028	0.010	0.004	0.074
$r = 300$	0.075	0.005	0.052	0.010	0.001	0.017
$r = 350$	0.005	0.002	0.027	0.020	0.002	0.012
Rank	**Rot +2**			**Rot -2**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.075	0.090	0.432	0.079	0.108	0.678
$r = 50$	0.057	0.024	0.196	0.075	0.024	0.214
$r = 100$	0.072	0.017	0.198	0.074	0.014	0.105
$r = 150$	0.059	0.013	0.108	0.076	0.011	0.076
$r = 200$	0.081	0.007	0.076	0.085	0.008	0.060
$r = 250$	0.047	0.005	0.034	0.100	0.004	0.028
$r = 300$	0.041	0.003	0.025	0.047	0.003	0.029
$r = 350$	0.004	0.001	0.011	0.002	0.003	0.020
Rank	**Rot +5**			**Median**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.065	0.170	0.613	0.100	0.129	0.354
$r = 50$	0.045	0.026	0.211	0.010	0.001	0.002
$r = 100$	0.108	0.019	0.106	0.010	0.001	0.004
$r = 150$	0.077	0.012	0.081	0.050	0.001	0.001
$r = 200$	0.028	0.005	0.027	0.010	0.001	0.001
$r = 250$	0.054	0.004	0.025	0.040	0.001	0.001
$r = 300$	0.012	0.003	0.036	0.020	0.001	0.001
$r = 350$	0.001	0.002	0.014	0.020	0.001	0.002
Rank	**Sharpening**			**StirMark**		
	Error''	E_{HL}	E_{KL}	*Error''*	E_{HL}	E_{KL}
$r = 1$	0.137	0.234	0.556	0.109	0.076	0.437
$r = 50$	0.046	0.011	0.082	0.039	0.005	0.056
$r = 100$	0.017	0.005	0.061	0.014	0.001	0.010
$r = 150$	0.023	0.008	0.051	0.024	0.001	0.011
$r = 200$	0.026	0.005	0.051	0.052	0.001	0.009
$r = 250$	0.009	0.005	0.055	0.055	0.001	0.006
$r = 300$	0.028	0.010	0.065	0.023	0.001	0.005
$r = 350$	0.002	0.005	0.066	0.003	0.001	0.004

Step C-(3): Is the model dependent on the EM Gaussian mixtures estimation?

In the last investigation, the concordance between the maximum likelihood estimation and the popular confidence limit estimation is checked.

The interval where the Gaussian mixture IVM takes non-zero values is evenly divided into 10 sub-intervals. On the one hand, confidence limits for the probability that the noise effects would take values in these sub-intervals are computed. On the other hand, the integral of the Gaussian mixture model on the same sub-intervals are evaluated. The values obtained in the experiments bring into evidence that each and every time (*i.e.* for each type of investigated attack and rank) the integral on the EM Gaussian model belongs to the corresponding confidence limits, thus validating the generality of the model.

For the interpretation of the errors in confidence limit estimation, see the Note in the corresponding section for DWT video modelling, *i.e.* Section III.3.2.1.2 – *Step C-(3)*.

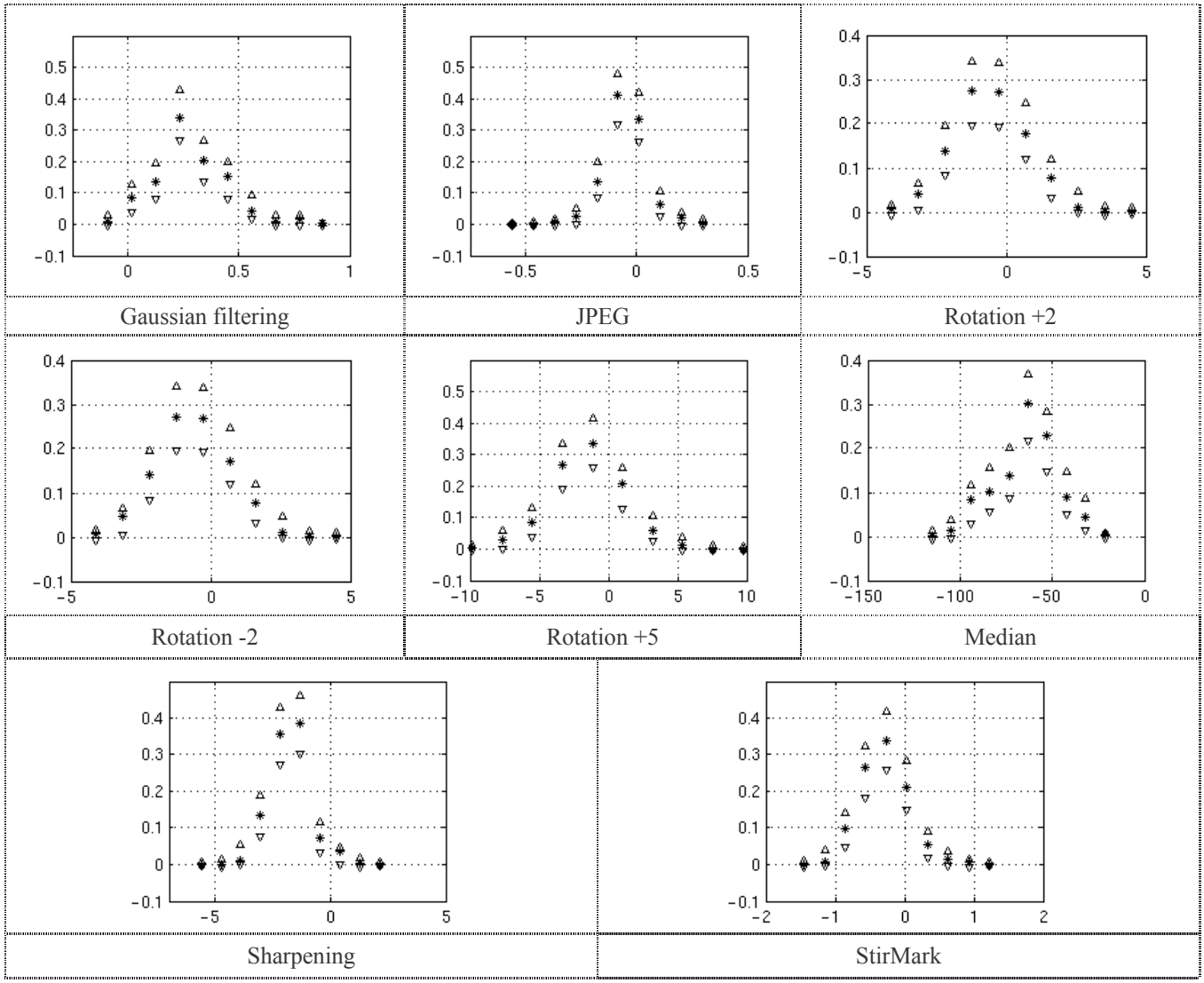


Figure IV.11. IVM model validation for DCT applied on whole frame (for $r = 1$, video coded at low quality rate, and for eight attacks): the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

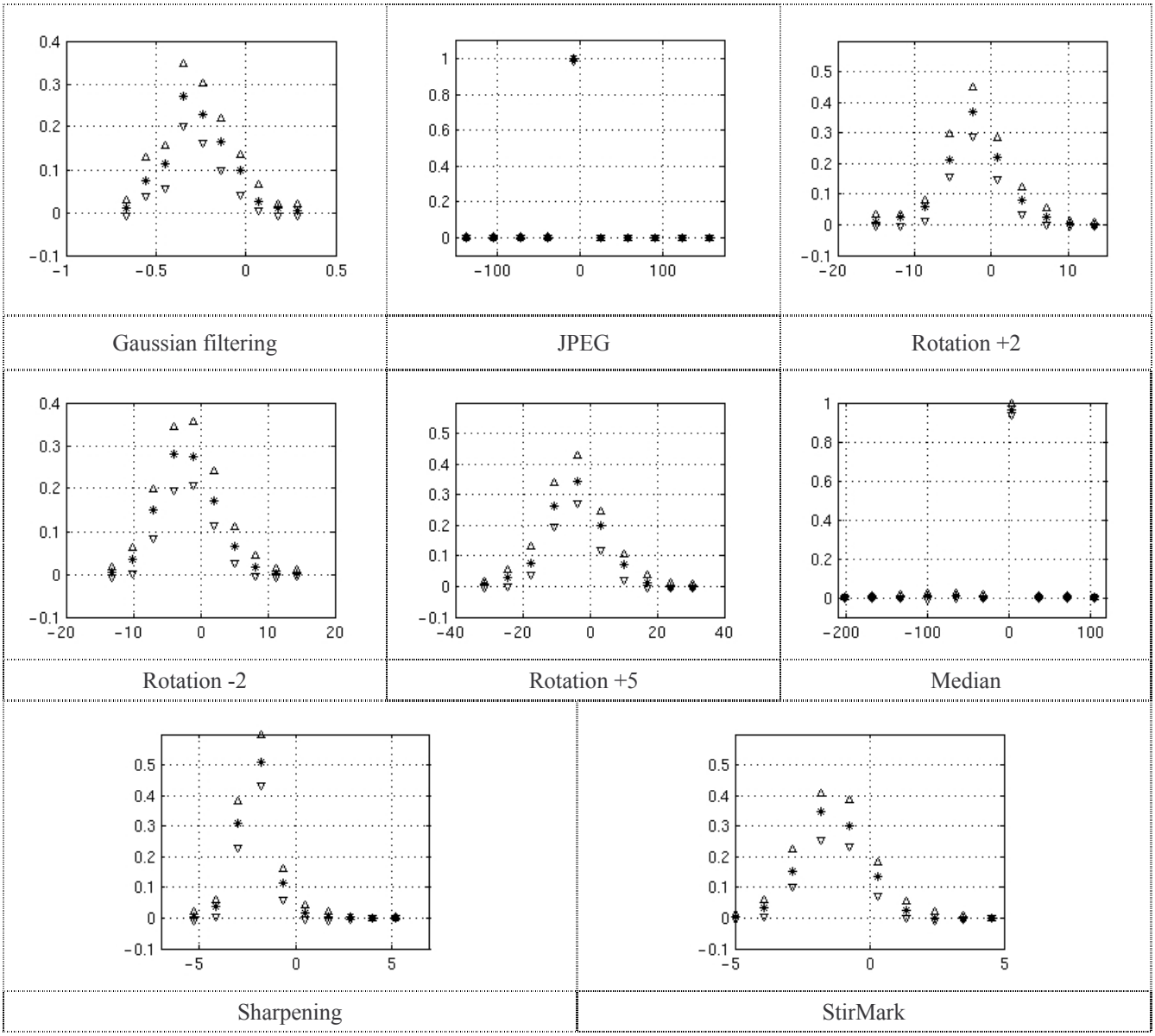


Figure IV.12. IVM model validation for DCT applied on whole frame (for $r = 1$, video coded at high quality rate, and for eight attacks): the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

In conclusion, the IVM have passed all the three procedures chosen in the *step C* of the *ART.MOD-A* algorithm. This means that also in the DCT domain the model is independent with respect to the similarity measure, to the video sequence and to the estimation procedure.

IV.4. Watermarking Capacity Assessment

The *watermarking capacity* is the maximal theoretical limit for the *data payload*, corresponding to prescribed *transparency* and *robustness* constraints.

From the theoretical point of view, Figure IV.13, the *watermarking capacity* is the capacity of the channel modelling the watermarking application. Consequently, any watermarking capacity evaluation relies on the statistical model of the noise on the corresponding channel: the host data and attacks.

Our interest is to evaluate the watermarking capacity for informed watermarking [Cox 02]. This approach is founded on the Shannon's [Sha 58] and Costa's [Cos 83] results according to which a noise source known at the embedded and unknown at the detector will not decrease the channel capacity. In the watermarking case, this means that the noise source corresponding to the original content may be neglected, at least as a first hand approximation. Consequently the watermarking capacity solely depends on the model of the watermarking attacks.

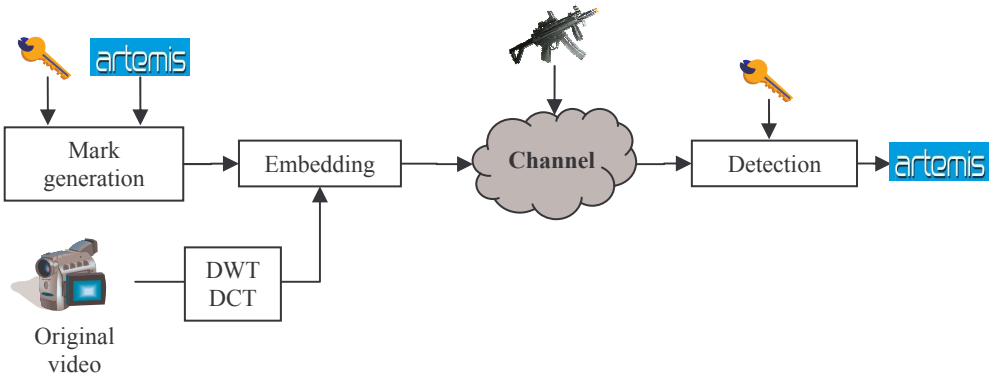


Figure IV.13. The watermarking within the information theory.

The capacity formula for a continuous channel, with input information source X and output information source Y is defined as the maximum value of the mutual information $I(X,Y)$ and is expressed by eq. (II.46). Note that the integrals in eq. (II.46) are evaluated on the intervals on which the input and output *pdf*s have non-zero values.

In literature, several studies already addressed the challenging issue of watermarking capacity [Bar 05], [Cos 83], [Gel 80], [Gam 80], [Hee 83], [Lin 01a], [Lin 01b], [Ser 98], [Sha 58], [Zha 04a], [Zha 04b], [Mou 02], most of them by directly applying the Shannon's formula and considering a Gaussian model for noise. Our studies on capacity [Mit 07b], [Dum 07a], [Dum 07b], [Dum 07c], [Mit 08], [Dum 08a], [Dum 08b] try to offer a solution to the problem of capacity evaluation for non-Gaussian attacks.

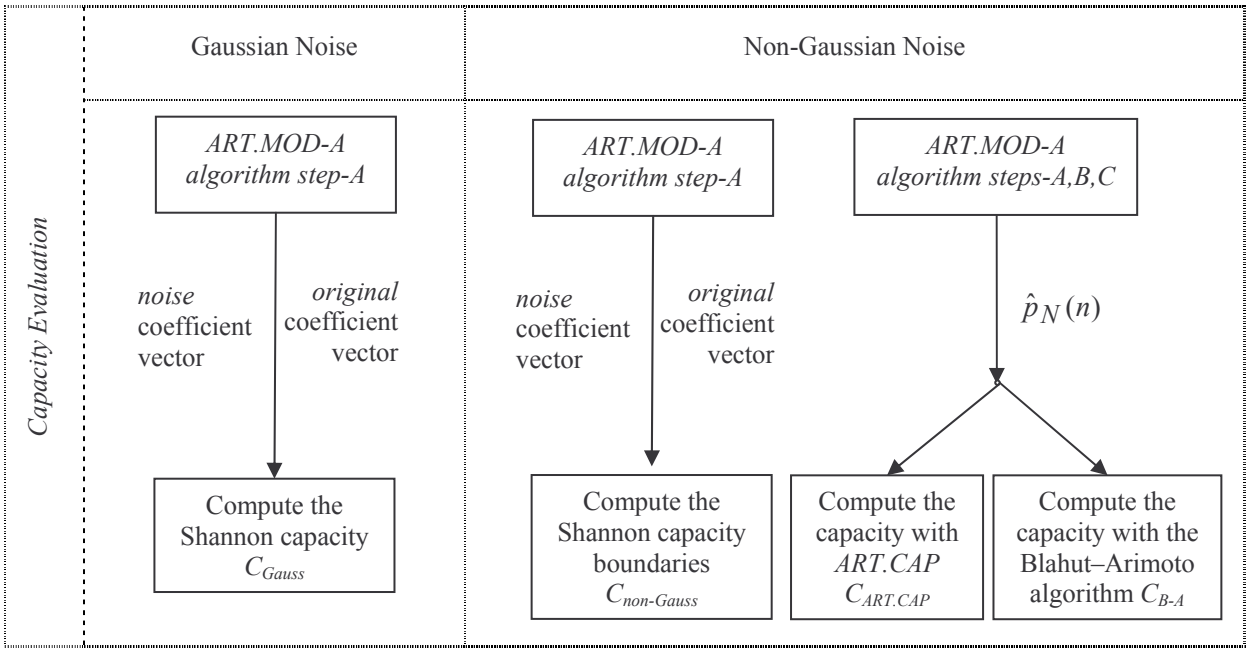
In order to be able to evaluate the capacity, four methods are described in the following sub-Sections: three methods already existed in the literature and a fourth one which was developed in this study.

IV.4.1. Capacity Computation Methods

As most of the authors in the literature consider the Gaussian hypothesis for the attacks, the capacity evaluation starts with the computation of the capacity of a Gaussian channel: we emphasise that such a value is useful only for discussion and comparison. In our studies devoted to watermarking attacks, we demonstrated (Section IV.1.1, [Mit 05b], [Dum 07b]) that this Gaussian behaviour is refuted for most of the attacks. Hence, the most important part of Section IV.4 deals with evaluating the watermarking capacity for non-Gaussian noise.

The main contribution was to develop an accurate method (denoted by *ART.CAP*) to compute the capacity of an additive non-Gaussian channel assuming that the noise *pdf* is expressed by Gaussian mixtures (see Section IV.3.2). These capacity values are further confronted to the Shannon boundaries (the capacity for a non-Gaussian noise) and to the values obtained by the Blahut–Arimoto algorithm extended for continuous channels.

This section summarises as a flowchart the methods involved in capacity evaluation (see Section IV.4.5); thir full description will be presented in Sections IV.4.2 ÷ IV.4.4.



IV.4.2. Shannon's Classical Formulae

The basic formula for the capacity of a channel having white Gaussian noise was established by Shannon [Sha 48], see eq. (IV.1). In the case of a noise source with a non-Gaussian $p_N(n)$ *pdf* (*i.e.* for an arbitrarily input-output dependency) the capacity cannot be theoretically computed in the general case and numerical algorithms should be considered [Bla 72]. However, for the case of an additive noise, independent with respect to the input information source, some general upper and lower limits can be derived (eq. (IV.2)):

$$C_{Gauss} = W \log \frac{P+N}{N}, \tag{IV.1}$$

$$W \log \frac{P+N_1}{N_1} \le C_{non-Gauss} \le W \log \frac{P+N}{N_1}, \tag{IV.2}$$

where W is the channel bandwidth (computed – in our experiments by considering the sample rate as being twice the bandwidth), P is the signal power, N is the noise power, and N_1 is the noise entropy power (*i.e.* the power of a white type noise, which has the same bandwidth and entropy as the analysed noise).

$$F_{rate} = 2W = 25s^{-1},\ W = 12.5Hz \text{ or } W = 0.5\,frames^{-1} \tag{IV.3}$$

$$N_1 = \frac{\exp(2H(X))}{2\pi e}, \tag{IV.4}$$

where *H(X)* is the entropy of the continuous information source modelling the noise.

IV.4.3. Capacity Method based on Gaussian Mixtures – ART.CAP

As the Shannon formula may lead to a quite large interval, this section presents an original method for computing the capacity of a channel with stationary additive non-Gaussian noise, whose *pdf* is expressed by a mixture of Gaussian distributions.

Under these hypotheses, eq. (II.46) can be written in the following form:

$$C = \max_{p_X(x)} \int_{x_1}^{x_2} \int_{x_1+n_1}^{x_2+n_2} p_X(x) p_N(y-x) \log_2 \frac{p_N(y-x)}{\int_{n_1}^{n_2} p_N(n) p_X(y-n) dn} dy dx, \tag{IV.5}$$

where the noise limits are $n_1 = y_1 - x_1$ and $n_2 = y_2 - x_2$ and $p_X(x)$ and $p_N(n)$ are the input and noise *pdf*s.

In can be seen that eq. (IV.5) no longer depends on the output *pdf*, which was expressed as a convolution between the input $p_X(x)$ and noise $p_N(n)$. Hence, the only unknown entity remains the input *pdf* $p_X(x)$.

In order to compute the capacity according to eq. (IV.5), the $p_X(x)$ input *pdf* was also represented by a Gaussian mixture (with $K = 10$) whose mean value was set to 0 (generally, in watermarking applications, the mark can be represented as a 0 mean signal). The variance of the $p_X(x)$ distribution was set so as to correspond to a transparent watermarking application. In other words, the variance of the original video was first estimated and then a $SNR = 30\text{dB}$ (SNR–Signal to Noise Ratio) condition was imposed, where the signal is the original video and the noise is the mark.

The relationship [Tra 02] among the parameters of the Gaussian laws composing the mixture and the mixture parameters, as well as the Chebyshev inequality made it possible for the capacity to be computed by a searching in the corresponding finite space (see Figure IV.14), with a resolution of 0.01 with respect to any parameters ($P(k)$ –weights , $\mu(k)$ –means and $\sigma(k)$ –variances).

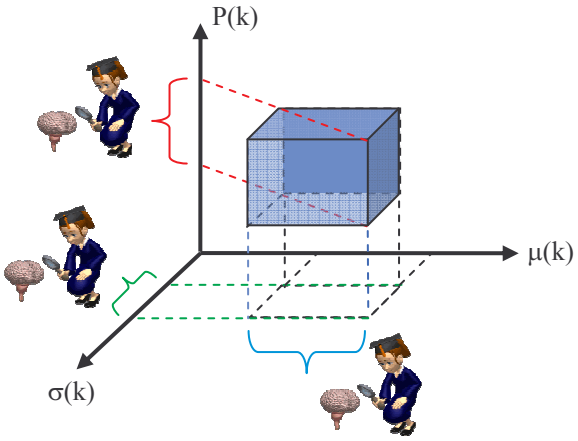


Figure IV.14. Example: searching in the space of the input Gaussian mixture parameters of the $p_X(x)$.

IV.4.4. The Blahut–Arimoto Algorithm

The first Blahut–Arimoto algorithm was devoted to the discrete and finite case [Bla 72], [Ari 72]. This algorithm was extended by Chang [Cha 04], and Vontobel [Von 08] for discrete and infinite case and by Dauwels [Dau 04], [Dau 05], [Dau 06] and Metz [Met 04] for continuous case.

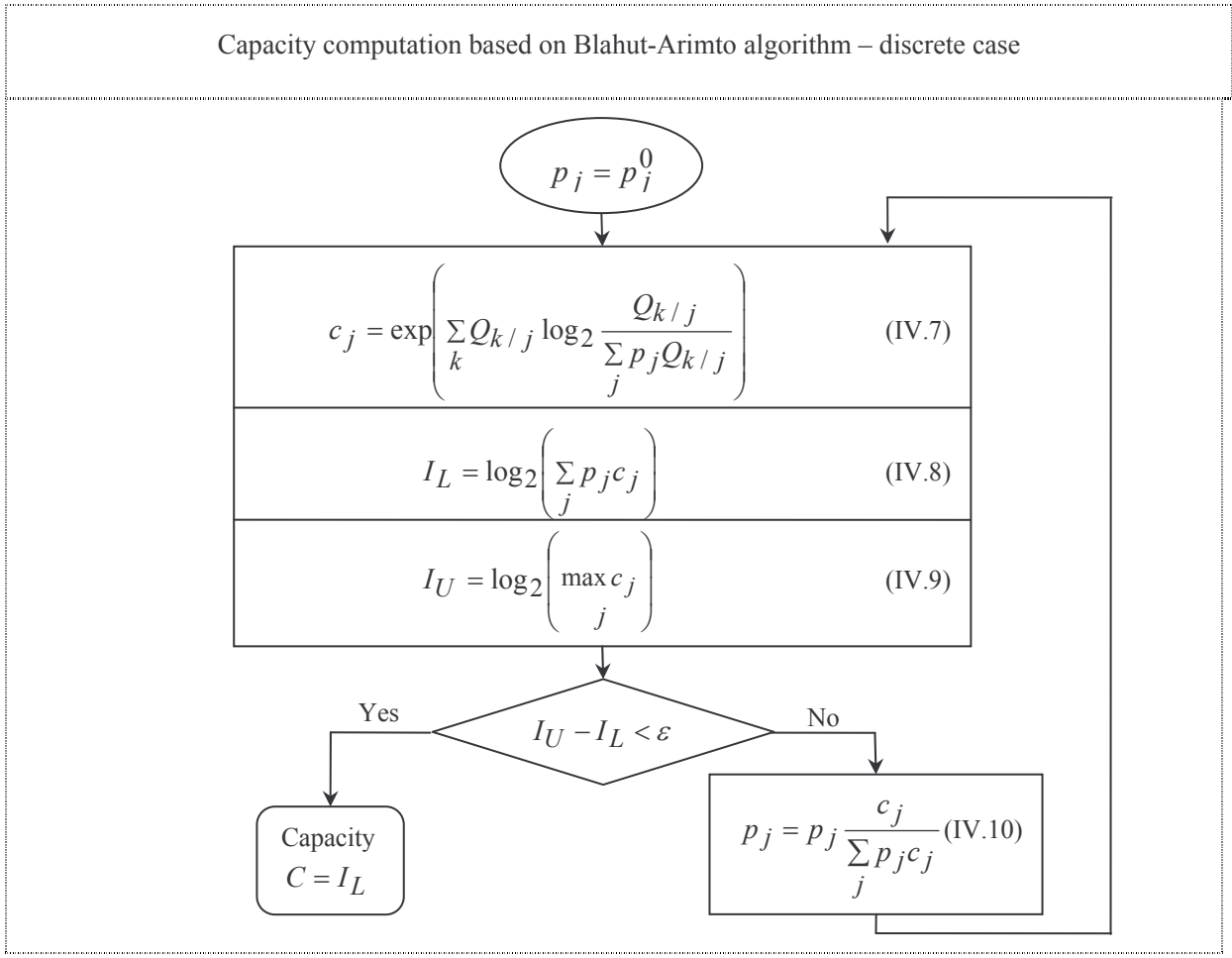
IV.4.4.1. The Discrete Blahut–Arimoto Algorithm

The basic discrete and finite algorithm is presented like a flowchart, where: $Q = [Q_{k/j}]$ is the probability of receiving the k output letter given that the j input letter was transmitted, p_j is a probability distribution of the input channel, p_j^0 is an arbitrary probability and ε is computation precision.

The capacity of the channel is defined as:

$$C = \max_{p \in P^n} I(p,Q) = \max_{p \in P^n} \sum_j \sum_k p_j Q_{k/j} \log_2 \frac{Q_{k/j}}{\sum_j p_j Q_{k/j}}, \qquad \text{(IV.6)}$$

where $I(p,Q)$ is the mutual information defined between the channel input and output and $P^n = \left\{ p \in R^n \middle| p_j \geq 0, \forall j, \sum_j p_j = 1 \right\}$ is the set of all probability density on the channel input.



IV.4.4.2. The Continuous Blahut–Arimoto Algorithm (J. Dauwels)

J. Dauwels presented [Dau 06] an extension of the discrete Blahut–Arimoto algorithm [Bla 72], [Ari 72] to a continuous Blahut–Arimoto algorithm (see the next flowchart). The algorithm proposed by Dauwels outperforms its many existing competitors [Cha 04], [Von 08], [Met 04] because it is a quick solution for capacity evaluation.

In the first step, the searched input $p^{(k=0)}(x)$ is randomly chosen. At each iteration, the *pdf* $p^{(k)}(x)$ is computed by using eq. (IV.16). When the difference between the two bounds ($M^{(k)}$ and $I^{(k)}$) is lower then ε (after *n* iterations, $n \le N$), the algorithm STOPs and the estimated capacity is obtain $C = I^{(n)}$.

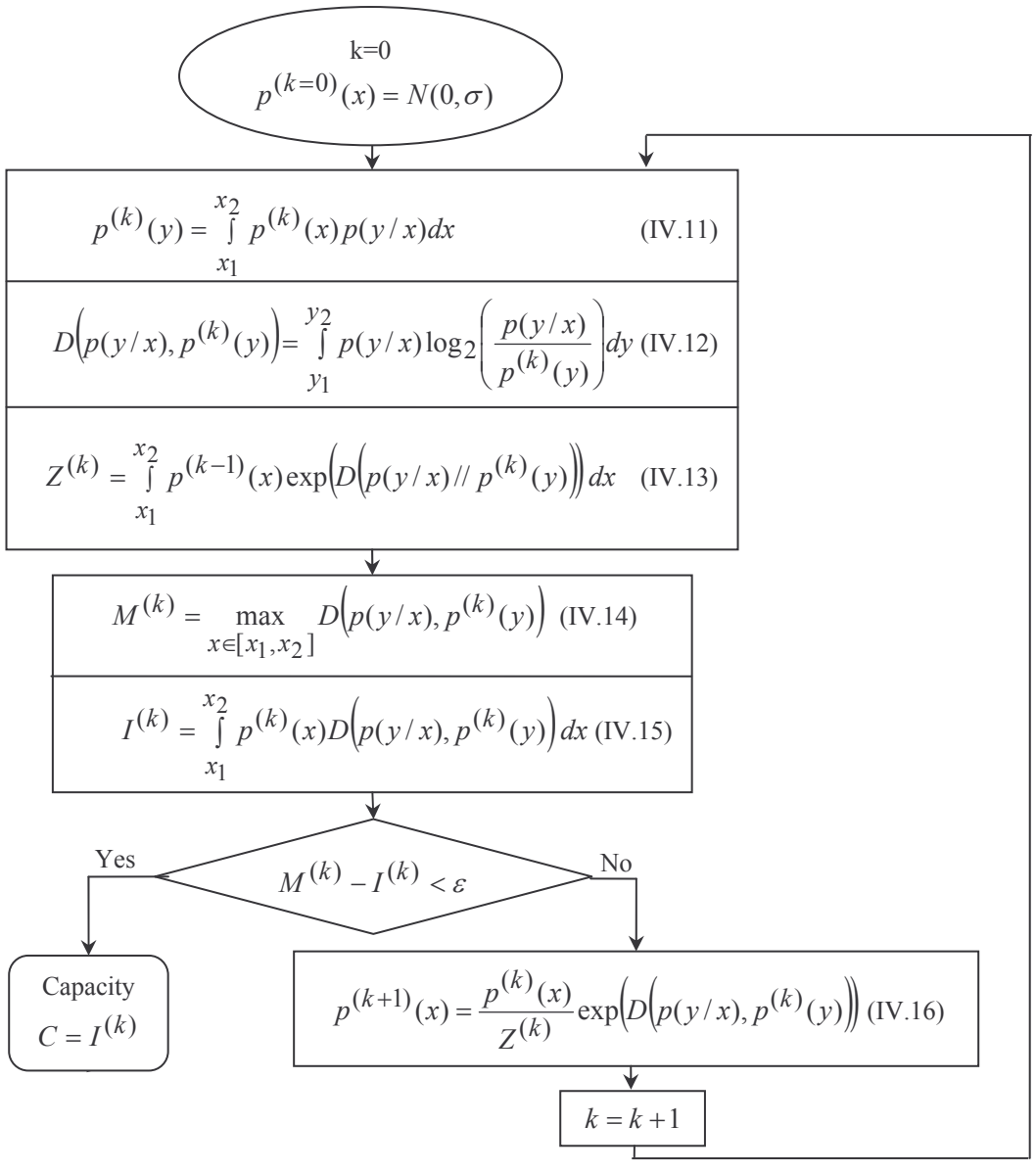
The following parameters were considered when applying the Blahut-Arimoto algorithm:

- the number of iterations is $k = \overline{1, N = 10^5}$;
- the limits for input are $x \in [-3\sigma; 3\sigma]$ while the output belongs to $y \in [x_1 + n_1; x_2 + n_2]$; the value σ^2 is the same as above (Section IV.4.2);
- the probability $p^{(0)}(x)$ is chosen as a Gaussian distribution with zero mean and σ^2 variance;

- the conditional probability of the output knowing the input is $p_{Y/X}(y/x) = p_N(y-x)$;
- the relative entropy or Kullback-Leibler divergence is $D\big(p(.),q(.)\big)$, eq. (IV.12);
- the normalising factor is Z, eq. (IV.13);
- the $p^{(k)}(x)$, eq. (IV.16).

The STOP condition is $M^{(k)} - I^{(k)} < \varepsilon = 10^{-4}$.

Capacity computation based on Blahut-Arimto algorithm – continuous case



IV.4.4.3. Others Extensions of the Blahut–Arimoto Algorithm

In order to compute the capacity for the continuous case, some others extensions [Cha 04], [Dau 04], [Dau 05], [Mat 04], [Von 08] of the discrete case have been reported in the literature.

Matz proposed two new variants of the Blahut–Arimoto algorithm for computing the capacity of the channel namely, *natural-gradient-based algorithm* and *accelerated Blahut–Arimoto algorithm* [Mat 04]. Both algorithms often converge significantly faster then the classical Blahut–Arimoto.

In *natural-gradient-based algorithm*, the *pdf* $p^{(k)}(x)$ is recursively computed using the following eq. (IV.17):

$$p^{(k)}(x) = p^{(k-1)}(x)\left(1 + \mu^{(k)}\left(D\left(p(y/x), p^{(k-1)}(y)\right) - I^{(k-1)}\right)\right). \tag{IV.17}$$

In the second case, *accelerated Blahut–Arimoto algorithm* the update rule for the *pdf* $p^{(k)}(x)$ is realized with the eq. (IV.18):

$$p^{(k)}(x) = \frac{p^{(k-1)}(x)}{Z^{(k)}} \exp\left(\mu^{(k)} D\left(p(y/x), p^{(k-1)}(y)\right)\right), \tag{IV.18}$$

where in the last two equations $\mu^{(k)}$ is a parameter for the step-size, and the parameters ($p^{(k)}(y)$, D, $Z^{(k)}$ and $I^{(k-1)}$) were defined by the eq. (IV.11), eq. (IV.12), eq. (IV.13) and eq. (IV.15) and refers to the output probability, the relative entropy, the normalized factor and the mutual information, respectively.

After each update, the classical Blahut–Arimoto algorithm is applied to compute the capacity. The STOP condition is the one defined in Section IV.4.4.2. The difference between the two algorithms (the one proposed by Dauwels and that proposed by Metz) consist in the equation used to update the probability $p^{(k)}(x)$.

In practice, there are many channels with constrains. A channel with constrains is a channel with the requirement that the average expense of $e(x)$ should be less than or equal to some specified number E. In this case the capacity is

$$C = \max_{p(x)\in P_E} \int_x \int_y p(x)p(y/x)\log_2 \frac{p(y/x)}{\int_x p(x)p(y/x)dx} dxdy = \max_{p(x)\in P_E} I(X,Y), \tag{IV.19.a}$$

where $P_E = \left\{ p : \mathrm{K}^m \to R \;\middle|\; \int_x p(x)dx = 1,\ p(x) \geq 0,\ E \overset{\Delta}{=} \int_x p(x)e(x)dx \leq E^{\max} \right\}$, $\mathrm{K} = R$ or $\mathrm{K} = C$. The Blahut–Arimoto is extended to constrained channels by replacing eq. (IV.16) with the following one:

$$p^{(k)}(x) = \frac{1}{Z^{(k)}} p^{(k-1)}(x)\left(D\left(p(y/x), p^{(k-1)}(y)\right) - se(x)\right), \tag{IV.20}$$

where s is a positive number (real number) and is adjustable after each iteration of the Blahut–Arimoto algorithm update in order to keep the average smaller or equal with E.

After n iterations of eq. (IV.20) the estimated capacity is:

$$C = I^{(n)} = \max_{x \in [x_1, x_2]} \left(D\left(p(y/x), p^{(n)}(y)\right) - se(x) \right) + sE^{(n)}. \quad \text{(IV.19.b)}$$

In 2008, Vontobel [Von 08] extended the classical Blahut–Arimoto algorithm to channels with memory and finite-state source of the input. The proposed algorithm for discrete case solves and finds the numerical value of the capacity.

For the case of memoryless channels with continuous alphabet (continuous for the input and continuous or discrete for the output), the classical Blahut–Arimoto algorithm can not be applied because this algorithm evaluate integrals to over the whole input space. In order to solve this problem, Dauwels offers one more time a solution [Dau 05]. The main idea is to use the sequential Monte–Carlo integration methods.

Another two solutions to the previous problem was proposed by Chang [Cha 98]. Instead of computing integrals, both proposed algorithms are iterative and require only the computations of a number of finite sums, which reduces the computational complexity. The first algorithm in an algorithm with infinite input space (discretization of the channel inputs) and finite output space, whereas for the second algorithm both the input and the output spaces are infinite. These algorithms may be an option in capacity evaluation, other than the algorithm proposed by Dauwels.

As it can be seen, although several solutions for capacity evaluation exists it is difficult to a priori asses their effectiveness for watermarking application. Consequently, in our study we considered two ways of compute the watermarking capacity, namely the *ART.CAP* and the continuous Blahut–Arimoto (J. Dauwels) algorithm. Note that the latter is fast and simple to implement and all others are starting from this one.

IV.4.5. Accurate Capacity Evaluation

Sections IV.3.2 proved that there is a *pdf* which model the noise sources corresponding to the video watermarking attacks do exist. All the experiments carried out brought into evidence that this *pdf* is independent with respect to the similarity measure, on the video content or on the estimation procedure.

In this moment, we may consider such accurate models for the watermarking attacks as the starting point in evaluating the watermarking capacity.

In the following two sub-Sections the IVM *pdf*s (which were computed and validated in Section IV.3.2.1 and IV.3.2.2) afford the capacity evaluation in different cases (transforms, attacks, ranks, *etc*). All the capacity values, *C* are expressed in bit/frame/rank.

The quantitative results are illustrated in Tables IV.16 and IV.17 for the DWT and in Tables IV.18 and IV.19 for the DCT, both applied on the whole frame (video coded at low quality rate, 64kbits/s). The results presented in these tables correspond to all capacity evaluation methods presented in the flowchart (see Section IV.4.1) for different attacks and ranks.

IV.4.5.1. The Capacity for the 2D-DWT

Table IV.16 presents capacity values for four ranks ($r = 1$, $r = 100$, $r = 200$, and $r = 300$), six types of attacks (Gaussian filtering, sharpening, rotation +2°, rotation -2°, rotation +5°, and StirMark), and the (9,7) DWT.

Table IV.16. Gaussian, Shannon limits, *ART.CAP*, and Blahut–Arimoto values for watermarking capacity in the (9,7) DWT.

r	Attack / C	Gaussian filtering	Sharpening	Rotation +2°	Rotation -2°	Rotation +5°	StirMark
$r = 1$	C_{Gauss}	0.05	0.02	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.31 ; 1.51)	(0.04 ; 0.57)	(0.01 ; 0.90)	(0.01 ; 0.90)	(0.01 ; 1.26)	(0.03 ; 0.55)
	$C_{ART.CAP}$	**0.58**	**0.44**	**0.36**	**0.37**	**0.36**	**0.38**
	$C_{B\text{-}A}$	**0.41**	**0.37**	**0.31**	**0.31**	**0.30**	**0.32**
$r = 100$	C_{Gauss}	0.03	0.01	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.06 ; 0.61)	(0.01 ; 0.26)	(0.01 ; 0.40)	(0.01 ; 0.42)	(0.01 ; 0.52)	(0.01 ; 0.35)
	$C_{ART.CAP}$	**0.46**	**0.26**	**0.25**	**0.25**	**0.25**	**0.28**
	$C_{B\text{-}A}$	**0.39**	**0.21**	**0.20**	**0.20**	**0.20**	**0.21**
$r = 200$	C_{Gauss}	0.01	0.01	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.02 ; 0.53)	(0.01 ; 0.25)	(0.01 ; 0.38)	(0.01 ; 0.35)	(0.01 ; 0.39)	(0.01 ; 0.47)
	$C_{ART.CAP}$	**0.40**	**0.21**	**0.19**	**0.19**	**0.19**	**0.24**
	$C_{B\text{-}A}$	**0.34**	**0.17**	**0.15**	**0.15**	**0.15**	**0.19**
$r = 300$	C_{Gauss}	0.01	0.01	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.01 ; 0.92)	(0.01 ; 0.49)	(0.01 ; 0.60)	(0.01 ; 0.61)	(0.01 ; 0.56)	(0.01 ; 0.73)
	$C_{ART.CAP}$	**0.33**	**0.18**	**0.12**	**0.12**	**0.12**	**0.21**
	$C_{B\text{-}A}$	**0.28**	**0.13**	**0.08**	**0.08**	**0.08**	**0.16**

All capacity values obtained by applying the *ART.CAP* and Blahut–Arimoto algorithms are between the lower and upper Shannon limits with no exceptions. Note that in Table IV.16 the capacity value obtained for the Shannon limits and the largest rank, $r = 1$ are identical for rotation +2° and -2°.

Table IV.17 is a comparison for the largest rank among various DWTs ((2,2), (4,4), and (9,7)), attacks (Gaussian filtering, sharpening, and StirMark) and levels of transparency (SNRs values of 25dB, 30dB, and 35dB). The following observations can be reported:

- The capacity values in the case of Gaussian noise are always lower than the Shannon lower limit in case of non-Gaussian noise;
- The estimated capacity (both the *ART.CAP* and Blahut–Arimoto algorithms), falls within the Shannon limits, except for the Gaussian filtering attack when :
  - for (2,2) DWT, and *SNR* = 30dB the capacity value obtained with the Blahut–Arimoto algorithm is lower then the Shannon lower limit;
  - for (4,4) DWT, and *SNR* = 25dB the same observation can be reported for the value evaluated with Blahut–Arimoto algorithm; when *SNR* = 30dB, both $C_{ART.CAP}$ and $C_{B\text{-}A}$ are lower then the Shannon limits.

Table IV.17. Gaussian, Shannon limits, *ART.CAP*, and Blahut–Arimoto values for watermarking capacity (for rank r = 1, for three DWTs, for three SNR levels, and for low quality video).

SNR	DWT	Attack / C	Gaussian filtering	Sharpening	StirMark
25dB	(2,2)	C_{Gauss}	0.09	0.03	0.02
		$C_{non\text{-}Gauss}$	(0.42 ; 1.40)	(0.07 ; 0.64)	(0.05 ; 0.61)
		$C_{ART.CAP}$	**0.53**	**0.52**	**0.53**
		$C_{B\text{-}A}$	**0.44**	**0.46**	**0.43**
	(4,4)	C_{Gauss}	0.09	0.03	0.02
		$C_{non\text{-}Gauss}$	(0.50 ; 1.48)	(0.08 ; 0.70)	(0.05 ; 0.60)
		$C_{ART.CAP}$	**0.53**	**0.51**	**0.52**
		$C_{B\text{-}A}$	**0.43**	**0.45**	**0.41**
	(9,7)	C_{Gauss}	0.15	0.05	0.02
		$C_{non\text{-}Gauss}$	(0.72 ; 1.60)	(0.19 ; 0.61)	(0.07 ; 0.60)
		$C_{ART.CAP}$	**0.85**	**0.56**	**0.57**
		$C_{B\text{-}A}$	**0.74**	**0.48**	**0.47**

Table IV.17. *(continued)*.

SNR	DWT	Attack / C	Gaussian filtering	Sharpening	StirMark
30dB	(2,2)	C_{Gauss}	0.09	0.03	0.02
		$C_{non\text{-}Gauss}$	(0.42 ; 1.39)	(0.07 ; 0.63)	(0.05 ; 0.60)
		$C_{ART.CAP}$	**0.47**	**0.38**	**0.42**
		$C_{B\text{-}A}$	**0.39**	**0.33**	**0.37**
	(4,4)	C_{Gauss}	0.09	0.01	0.02
		$C_{non\text{-}Gauss}$	(0.48 ; 1.48)	(0.06 ; 0.68)	(0.05 ; 0.60)
		$C_{ART.CAP}$	**0.47**	**0.37**	**0.38**
		$C_{B\text{-}A}$	**0.39**	**0.33**	**0.32**
	(9,7)	C_{Gauss}	0.05	0.02	0.01
		$C_{non\text{-}Gauss}$	(0.31 ; 1.51)	(0.04 ; 0.57)	(0.03 ; 0.55)
		$C_{ART.CAP}$	**0.58**	**0.44**	**0.38**
		$C_{B\text{-}A}$	**0.41**	**0.37**	**0.32**
35dB	(2,2)	C_{Gauss}	0.09	0.03	0.02
		$C_{non\text{-}Gauss}$	(0.05 ; 1.31)	(0.01 ; 0.60)	(0.01 ; 0.58)
		$C_{ART.CAP}$	**0.07**	**0.07**	**0.07**
		$C_{B\text{-}A}$	**0.06**	**0.06**	**0.06**
	(4,4)	C_{Gauss}	0.09	0.01	0.02
		$C_{non\text{-}Gauss}$	(0.06 ; 1.40)	(0.01 ; 0.64)	(0.01 ; 0.58)
		$C_{ART.CAP}$	**0.10**	**0.09**	**0.10**
		$C_{B\text{-}A}$	**0.08**	**0.07**	**0.08**
	(9,7)	C_{Gauss}	0.14	0.06	0.03
		$C_{non\text{-}Gauss}$	(0.12 ; 1.47)	(0.01 ; 0.56)	(0.01 ; 0.54)
		$C_{ART.CAP}$	**0.22**	**0.10**	**0.15**
		$C_{B\text{-}A}$	**0.18**	**0.08**	**0.14**

IV.4.5.2. Capacity for the 2D-DCT

The same types of experiments have been resumed for computing the capacity in the DCT domain. Table IV.18 illustrates the capacity computed for four ranks ($r = 1$, $r = 100$, $r = 200$, and $r = 300$), and six attacks (Gaussian filtering, sharpening, rotation +2°, rotation -2°, rotation +5°, and StirMark), while Table IV.19 compares the capacity values corresponding to three levels of transparency (expressed by *SNR* = 25dB, *SNR* = 30dB, and *SNR* = 35dB).

Table IV.18. Gaussian, Shannon limits, *ART.CAP*, and Blahut–Arimoto values for watermarking capacity in the DCT for low quality video.

r	Attack / C	Gaussian filtering	Sharpening	Rotation +2°	Rotation -2°	Rotation +5°	StirMark
$r = 1$	C_{Gauss}	2.49	0.42	0.69	0.69	0.21	1.99
	$C_{non\text{-}Gauss}$	(3.63 ; 3.65)	(1.26 ; 1.72)	(0.29 ; 0.46)	(0.29 ; 0.46)	(0.28 ; 0.48)	(2.39 ; 2.42)
	$C_{ART.CAP}$	**2.57**	**1.33**	**0.30**	**0.30**	**0.28**	**2.31**
	$C_{B\text{-}A}$	**2.42**	**1.27**	**0.23**	**0.23**	**0.26**	**2.18**
$r = 100$	C_{Gauss}	0.02	0.01	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.06 ; 1.05)	(0.01 ; 0.56)	(0.01 ; 0.48)	(0.01 ; 0.48)	(0.01 ; 0.68)	(0.01 ; 0.22)
	$C_{ART.CAP}$	**0.68**	**0.39**	**0.20**	**0.20**	**0.19**	**0.55**
	$C_{B\text{-}A}$	**0.61**	**0.34**	**0.18**	**0.18**	**0.17**	**0.49**
$r = 200$	C_{Gauss}	0.01	0.01	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.03 ; 0.94)	(0.01 ; 0.54)	(0.01 ; 0.58)	(0.01 ; 0.58)	(0.01 ; 0.70)	(0.01 ; 0.27)
	$C_{ART.CAP}$	**0.50**	**0.31**	**0.10**	**0.10**	**0.10**	**0.42**
	$C_{B\text{-}A}$	**0.44**	**0.27**	**0.08**	**0.08**	**0.08**	**0.36**
$r = 300$	C_{Gauss}	0.01	0.01	0.01	0.01	0.01	0.01
	$C_{non\text{-}Gauss}$	(0.02 ; 0.94)	(0.01 ; 0.62)	(0.01 ; 0.62)	(0.01 ; 0.62)	(0.01 ; 0.74)	(0.01 ; 0.27)
	$C_{ART.CAP}$	**0.37**	**0.28**	**0.09**	**0.09**	**0.08**	**0.33**
	$C_{B\text{-}A}$	**0.31**	**0.25**	**0.06**	**0.06**	**0.05**	**0.28**

Note that the capacity values are identically for the rotation +2° and -2° and for the ranks ($r=1$, $r=100$, $r=200$, and $r=300$) displayed in Table IV.18. For the rank $r=200$ the capacity values computed with *ART.CAP* and Blahut–Arimoto are identically for all rotations (+2°, -2°, and +5°).

In Table IV.18 the $C_{ART.CAP}$ and $C_{B\text{-}A}$ values belong to the corresponding limits, with some exceptions:

- for the Gaussian filtering, the StirMark attacks, and the rank $r=1$, the capacity values are lower than the Shannon lower limit;
- for the StirMark attack and the ranks $r=100$, $r=200$ and $r=300$ the capacity value is higher than the Shannon upper limit.

The results reported in Table IV.19 follow a similar general behaviour.

One explanation for the large differences (see Table IV.18 and IV.19) between the lower Shannon limit and *ART.CAP* and Blahut–Arimoto algorithms is the high error in the model computation: for the Gaussian filtering and $r=1$, the error in model validation (*step-C(2)* of the *ART.MOD-A*) was *Error''* = 36.5% (see Section IV.3.2.2).

Table IV.19. Gaussian, Shannon limits, *ART.CAP*, and Blahut–Arimoto values for watermarking capacity (for rank *r* = 1, for DCT, for three SNR levels, and for low quality video).

SNR	Attack / *C*	Gaussian filtering	Sharpening	StirMark
25dB	C_{Gauss}	3.31	0.90	2.78
	$C_{non\text{-}Gauss}$	(4.46 ; 4.47)	(2.00 ; 2.21)	(3.20 ; 3.22)
	$C_{ART.CAP}$	**2.97**	**2.10**	**3.11**
	$C_{B\text{-}A}$	**2.78**	**1.93**	**3.01**
30dB	C_{Gauss}	2.49	0.42	1.99
	$C_{non\text{-}Gauss}$	(3.63 ; 3.65)	(1.26 ; 1.72)	(2.39 ; 2.42)
	$C_{ART.CAP}$	**2.57**	**1.33**	**2.31**
	$C_{B\text{-}A}$	**2.42**	**1.27**	**2.18**
35dB	C_{Gauss}	1.71	0.16	1.25
	$C_{non\text{-}Gauss}$	(2.81 ; 2.87)	(0.67 ; 1.46)	(1.62 ; 1.68)
	$C_{ART.CAP}$	**2.05**	**0.73**	**1.54**
	$C_{B\text{-}A}$	**1.96**	**0.64**	**1.38**

IV.4.5.3. Relative Error in Capacity Evaluation

In order to allow a general assessment of the accuracy in the capacity evaluation, two types of errors have been computed (see Table IV.20 – IV.21):

- the relative errors between Shannon lower C_{low} and upper C_{upper} limits:

$$E_{low/upper} = \frac{\left|C_{low} - C_{upper}\right|}{C_{upper}}, \tag{IV.21}$$

- the relative errors between $C_{B\text{-}A}$ and $C_{ART.CAP}$ values:

$$E_{B-A/ART.CAP} = \frac{\left|C_{B-A} - C_{ART.CAP}\right|}{C_{ART.CAP}}. \tag{IV.22}$$

Table IV.20. Capacity and relative errors for SNR =30dB – (9,7) DWT applied on the whole frames of low quality video.

DWT	Attack / r = 1	Gaussian filtering	Sharpening	Rotation +2°	Rotation -2°	Rotation +5°	StirMark
Capacity	(C_{low} ; C_{upper})	(0.31 ; 1.51)	(0.04 ; 0.57)	(0.01 ; 0.90)	(0.01 ; 0.90)	(0.01 ; 1.26)	(0.03 ; 0.55)
	$C_{ART.CAP}$	**0.58**	**0.44**	**0.36**	**0.37**	**0.36**	**0.38**
	$C_{B\text{-}A}$	**0.41**	**0.37**	**0.31**	**0.31**	**0.30**	**0.32**
E relative	$E_{low/upper}$	**0.79**	**0.93**	**0.99**	**0.99**	**0.97**	**0.94**
	$E_{B\text{-}A/ART.CAP}$	**0.29**	**0.16**	**0.14**	**0.16**	**0.17**	**0.16**

Table IV.21. Capacity and relative errors for SNR =30dB – DCT applied on the whole frames of low quality video.

DCT	Attack / r = 1	Gaussian filtering	Sharpening	Rotation +2°	Rotation -2°	Rotation +5°	StirMark
Capacity	(C_{low} ; C_{upper})	(3.63 ; 3.65)	(1.26 ; 1.72)	(0.29 ; 0.46)	(0.29 ; 0.46)	(0.28 ; 0.48)	(2.39 ; 2.42)
	$C_{ART.CAP}$	**2.57**	**1.33**	**0.30**	**0.30**	**0.28**	**2.31**
	$C_{B\text{-}A}$	**2.42**	**1.27**	**0.23**	**0.23**	**0.26**	**2.18**
E relative	$E_{low/upper}$	**0.01**	**0.31**	**0.37**	**0.37**	**0.42**	**0.01**
	$E_{B\text{-}A/ART.CAP}$	**0.06**	**0.05**	**0.23**	**0.23**	**0.07**	**0.06**

As a general observation, the relative errors obtained for the largest rank ($r = 1$) and six attacks (Gaussian filtering, sharpening, rotations with +2°, -2°, +5°, and StirMark) are lower for the $E_{B-A/\,ART.CAP}$. Two exceptions can be reported in Tables IV.20 and IV.21: in the DCT domain, for Gaussian filtering and StirMark attacks the $E_{B-A/\,ART.CAP}$ is higher then the $E_{low/upper}$; this can be explained by the fact that in these cases the Shannon boundaries (lower and upper limits) are very closer and the relative error for $E_{low/upper}$ is very small ($E_{low/upper} = 0.01$).

The capacity (*ART.CAP* and Blahut–Arimoto algorithm) for the whole frame of the video can be obtained by summing up the values corresponding to the ranks in which the mark is inserted. Note that here are presented only the capacity values for a few ranks.

As a general observation, in all cases the capacities obtained for the equivalent Gaussian distributed attack noise (the first line of each table – presented in Section IV.4.5.1 and IV.4.5.2) are smaller than the lowest Shannon limit result for the non-Gaussian noise (the second line of each table). This means, if the non-Gaussian distribution of the attack is supposed, then the number of the bits inserted into a video sequence is higher.

IV.4.5.4. Watermarking Capacity vs. Multimedia Applications

Beyond the property right protection, capacity evaluation opens the door for a potential use of the watermarking techniques in various *enriched multimedia* applications.

In such scenarios, the watermarking techniques have several inherent advantages; for instance, they allow the *in-band enrichment* of the multimedia stream [Mit 06b] thus alleviating the backward compatibility problem.

Concerning the viability of such emerging applications, four scenarios are brought into discussion (see Figure IV.16), namely the *indexing* [Jan 98], [Lsm 09], *subtitles* [Mpg 09], [Gpa 09], *hyper video* or *clickable video* [Mpg 09], [Cli 08] and *interactive digital television* [Bou 00], [Fre 01]. Should the capacity values be larger than the amount of enrichment data required by these applications, a proof of viability is obtained.

Each of these applications is considered in three cases: when the video content is subject to filtering (Gaussian / sharpening), rotations (+2°, -2°, and +5°), and StirMark attacks. A comparison between the capacity values obtained for the equivalent Gaussian-distributed noise (see eq. (IV.1)) and the non-Gaussian noise (Shannon limits computed with eq. (IV.2)) was made.

All the numerical results (in bits/frame) are illustrated in Appendix IV (see Tables A.IV.1 ÷ A.IV.3 and Figures A.IV.1 ÷ IV.3) by processing a video sequence from the video corpus [Dum 07b]. However, similar results are obtained for all the video sequences which are available in our video corpus and for different transforms.

Considering the applications and the capacity values reported in Appendix IV, it can be noticed that the first two applications may be turned into practice (at least with some restrictions – see

Tables A.IV.4 and A.IV.5 in Appendix IV). However, suspicions about the possibility of implementing *hyper video* and *interactive digital television* by means of watermarking arise (at least when inserting the mark in realised on the DWT/DCT coefficients hierarchy) [Dum 07b], [Dum 07c].

Finally, the results obtained with *ART.CAP* need to be applied for *enriched multimedia* application in order to see which application can be considered for practice. This should be the subject of further research.

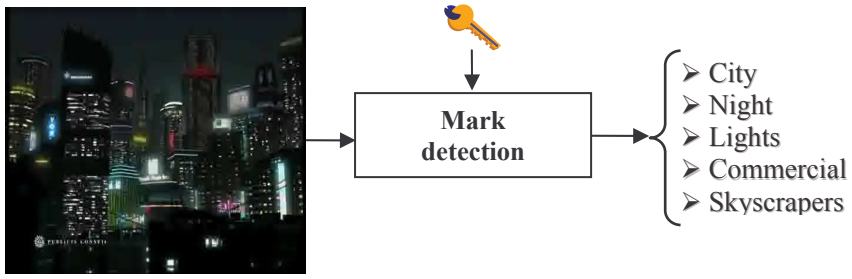


Figure IV.16.a. Video indexation application.

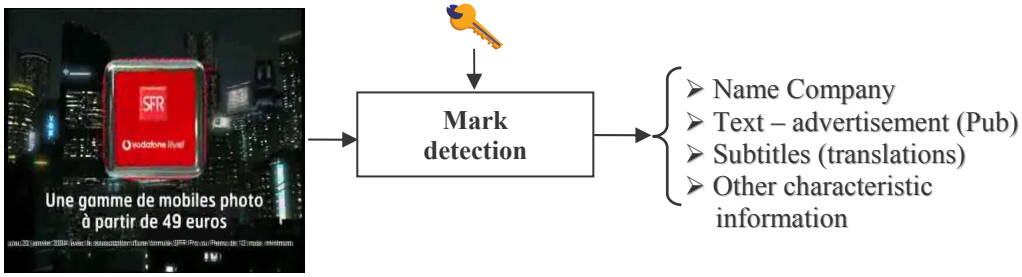


Figure IV.16.b. Video subtitles application.

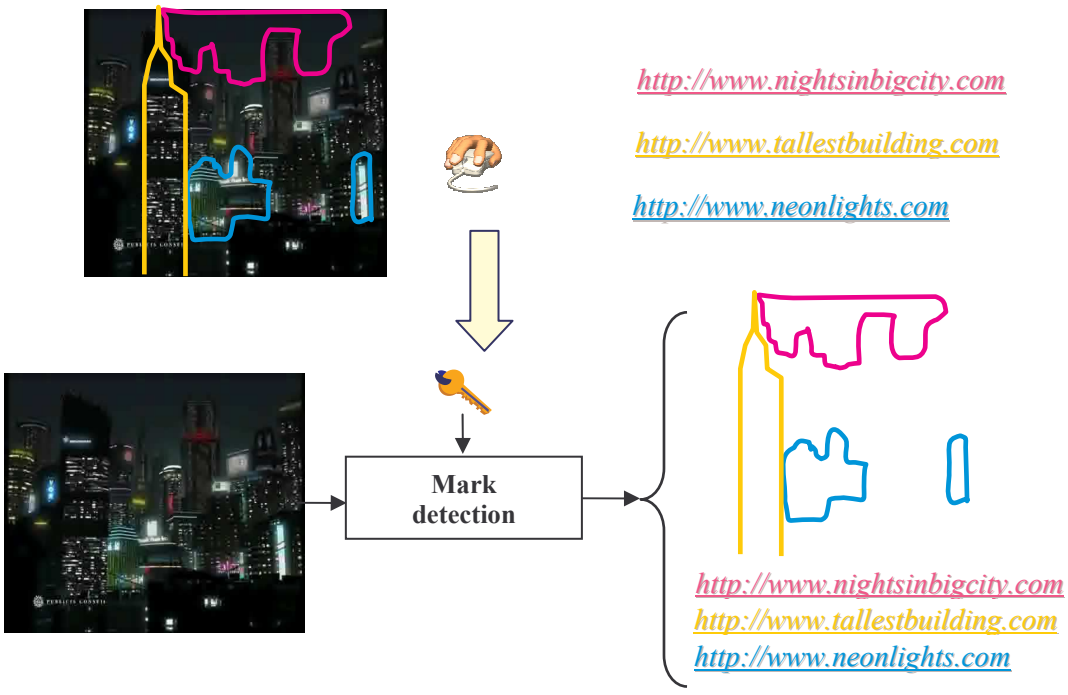


Figure IV.16.c. Hyper video or clickable video application.

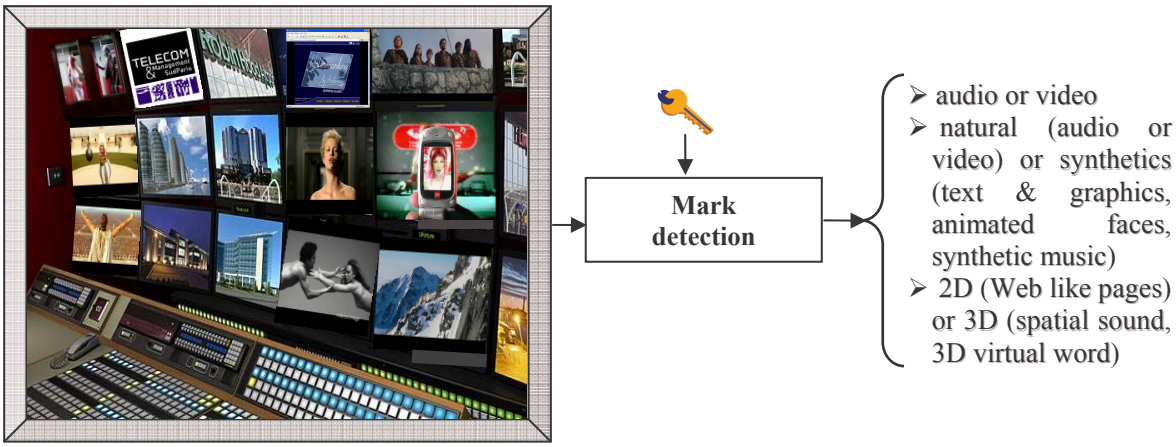


Figure IV.16.d. Interactive digital television application.

IV.5. The Attack Modelling in Real–life

The method developed at the ARTEMIS department [Mit 06a] in collaboration with the SFR (Vodafone group) combines the Spread Spectrum (SS) techniques (where the *transparency* and *robustness* are very good) and the Side Information (SI) techniques (where the *data payload* is large enough and the *transparency* is good) in order to reach the trade off between *transparency*, *robustness* and *data payload*. With this aim, a new insertion technique taking into account the particularities of each attack (and no longer their Gaussian approximation) has been patented [Mit 07c]. However, the side effect of this method is the time of execution which is prohibitive for a large class of applications. Consequently, the study in [Cha 09] was devoted to the speed-up of the method.

The average time of execution, between different blocks of the watermarking scheme (*example:* pre-processing, insertion, mark generation, post-processing, and detection) shows that the insertion part represents 90% (see Figure IV.17), so that more than 99% are considered for the real behaviour of the attacks. Note that for the experiments a PC with an Intel Centrino processor with 1GB RAM was used; for a video sequence of 100 frames the mark procedure take around one hour.

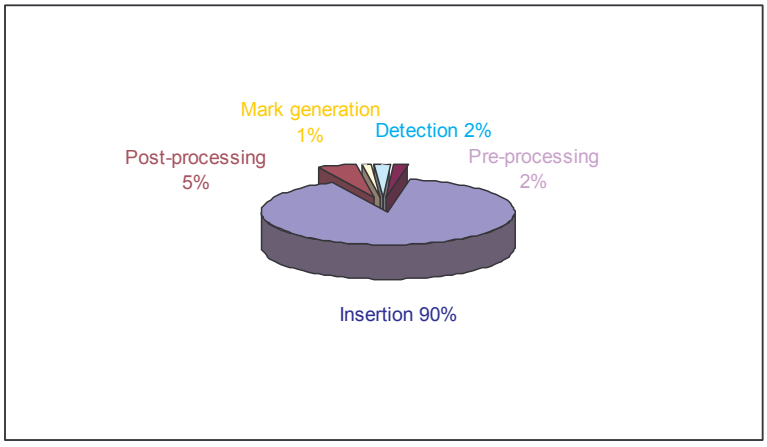


Figure IV.17. Execution time for different operations for the original method – Figure IV.18.a.

Hence, it was proposed to replace the real instantiations of the attacks by some values sampled from the Monte Carlo generator implemented according to the attack models, see Figure IV.18.

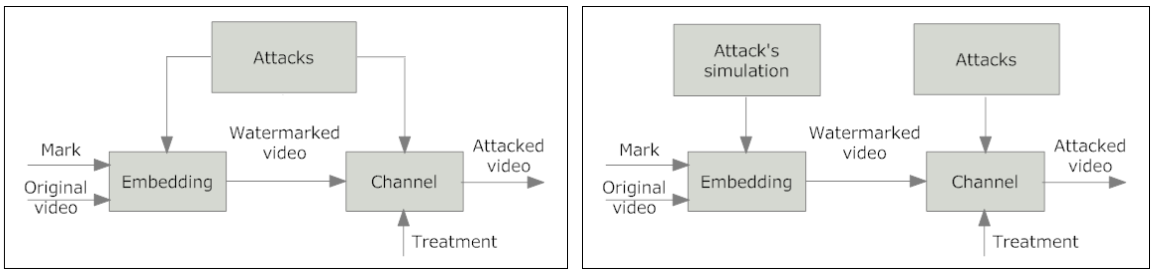


Figure IV.18. Hybrid watermarking method: (a) original method and (b) optimized method by inserting a simulator attacks.

The Monte Carlo simulation algorithm was realised in two steps:

Step 1 – Select the index k of the Gaussian random variable

- generate a value $\alpha \in (0,1)$ using a uniform random numbers generator;
- if $\alpha \le P_1$, choose $k = 1$ (the first variable in the mixture);
- else, choose the index k that satisfied the following condition: $\sum_{j=1}^{k-1} P_j < \alpha \le \sum_{j=1}^{k} P_j$.

Step 2 – Sampling the Gaussian distribution corresponding to index k

- generate x_1 and x_2, two values uniformly distributed between 0 and 1;
- compute $y = \sigma_k \sqrt{-2 \ln x_1} \cos(2\pi x_2) + \mu_k$.

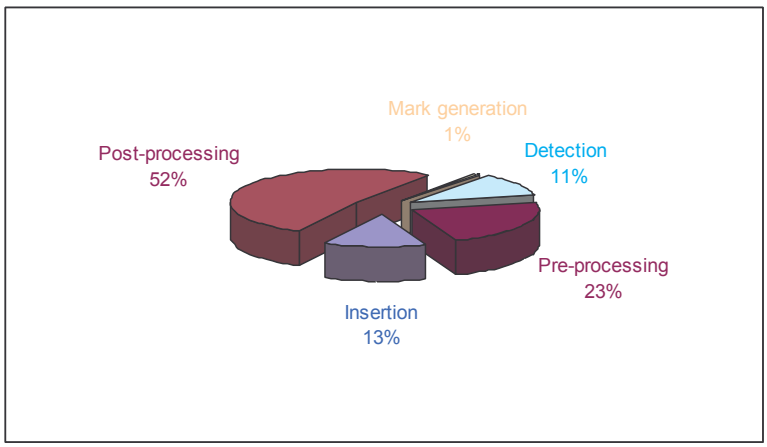


Figure IV.19. Execution time for different operations for the hybrid watermarking method–Figure IV.18.b.

From the practical point of view, it was experimentally shown that a factor of 100 was obtained during the execution of the hybrid watermarking method, preserving the same performance of *transparency* and *robustness* of the initial method [Cha 09], see Figure IV.19.

From the theoretical point of view, these results proved the effectiveness of the model obtained from the watermarking attacks.

IV.6. Conclusion

The study presented in the first part of the chapter computes for the first time accurate statistical models for several types of watermarking attacks in the DWT or DCT hierarchy.

Alongside with the *pdf* estimation for the attacks, this study brings evidence in the favour of their stationarity hypothesis: it was checked out that the computed models are independent with respect to the similarity measure, to the considered video sequence, and to the estimation procedure. These models are further involved in the capacity evaluation.

Instead of some upper and lower limits for capacity, now it is possible to compute accurate values with the new method *ART.CAP*. As these values were obtained by a numerical procedure, one idea was to evaluate their precision and for this purposes the Blahut–Arimoto algorithm extended by J. Dauwels was considered for its simplicity and speed. The relevance of the quantitative results was discussed beyond the watermarking fields for *enriched multimedia* applications.

Such general attack models can be exploited in many real-life applications, like the hybrid watermarking method presented in Section IV.5. There, the attack simulation was realised with a Monte Carlo generator taking into account the particularities of the each attack; for this the models obtained in Section IV.3.2 were involved in the experiments.

The accurate models for attacks can be also the starting point in optimising the detection rule for watermarking applications. Note that a large class of the methods reported in the literature achieves mark detection by means of correlation (matched filters), whose optimality is proved when the noise is Gaussian distributed. For non–Gaussian noise, new rules should be devised.

IV.7. References

[Ari 72] S. Arimoto, *"An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels"*, IEEE Trans. on Information Theory, Vol. 18, No. 1, July 1972, pp. 14 - 20.

[Bar 98] R. Barnett, D.E. Pearson, *"Frequency mode LR attack operator for digitally watermarked images"*, Electronics Letters, Vol. 34, September 1998, pp. 1837 – 1839.

[Bar 00] M. Barni, F. Bartolini, A. De Rosa, A. Piva, *"Capacity of Full Frame DCT Image Watermarks",* IEEE Transactions on Image Processing, Vol. 9, No.8, August 2000, pp. 1450 – 1455.

[Bar 01] M. Barni, C. Podilchuk, F. Bartolini, E.J. Delp, *"Watermarking Embedding: Hiding a Signal within a Cover Image"*, IEEE Comm. Magazine, August 2001, pp. 102 – 108.

[Bau 02] R. Bauml, J.J. Eggers, R. Tzschoppe, J. Huber, *"A Channel Model for Watermarking Subject to Desynchronization Attacks"*, Proc. SPIE, Vol. 4675, San Jose, CA, January 2002, pp. 281-292.

[Bla 72] R. Blahut, *"Computation of Channel Capacity and Rate-distortion Functions"*, IEEE Trans. on Information Theory, Vol. 18, No. 4, July 1972, pp. 460 - 473.

[Bou 00] F. Bouilhaguet, J-C. Dufourd, S. Boughoufalah, C. Havet, *"Interactive Broadcast Digital Television – The OpenTV Platform versus the MPEG-4 Standard Framework"*, ISCAS 2000, Geneva, Switzerland, May 2000, pp. III-626 – III.629.

[Bou 03] S. Bounkong, B. Toch, D. Saad, D. Lowe, *"ICA for Watermarking Digital Images"*, Journal of Machine Learning Research, Vol. 4, 2003, pp. 1471 – 1498.

[Cha 04] C. Chang, L.D. Davisson, *"On Calculating the Capacity of an Infinite-Input Finite (Infinet)-Output Channel",* IEEE Transaction on Information Theory, Vol. 34, September 2004, pp. 1004 – 1010.

[Cha 09] A. Chammen, O. Dumitru, M. Mitrea, F. Prêteux, *"Simulations Monte Carlo pour le tatouge robuste des séquence vidéo",* TAIMA 2009, Hammamet–Tunisia, May 2009 pp. 501 – 507.

[Cli 08] Clickable Video – example software VideoClix https://secure.rtraction.com/yahoo-bsls/presantations/Brent_Stafford_Yahoo-VideoClix.pdf

[Cos 83] M.H.M. Costa, *"Writing on Dirty Paper"*, IEEE Transactions on Information Theory, IT-29, Vol.3, May 1983, pp. 439-441.

[Cox 02] I.J. Cox, M.L. Miller, J.A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, 2002.

[Cox 08] I. J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, T. Kalker, *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers, 2008.

[Cve 04] N. Cvejic, T. Seppanen, *"Channel Capacity of High Bit Rate Audio Data Hiding Algorithms in Diverse Transform Domains"*, International Syposium on Communications and Information Technologies 2004, October 2004, Sapporo, Japan, pp. 84 - 88.

[Dau 04] J. Dauwels, H-A. Loeliger, *"Computation of Information Rates by Particle Methods"*, Proc. 2004 IEEE International Symposium Information Theory, Chicago, IL, June 2004, p.178.

[Dau 05] J. Dauwels, *"Numerical Computation of the Capacity of Continuous Memoryless Channels"*, 26th Symposium on Information Theory, Benelux 2005.

[Dau 06] J. Dauwels, *On Graphical Models for Communications and Machine Learning: Algorithms, Bounds and Analog Implementation*, Ph.D. thesis-ETH Zürich, May 2006.

[Du 02] J. Du, C.H. Lee, H.K. Lee, Y. Suh, *"BSS : A New Approach for Watermarking Attack"*, Proc. IEEE Fourth International Symposium on Multimedia Software Engineering (MSE 2002), pp. 182 – 187.

[Dum 07a] O. Dumitru, M. Mitrea, F. Prêteux, *"Accurate Watermarking Capacity Evaluation",* Proc. SPIE, Vol. 6763, September 2007, Boston–USA, pp. 6763 03: 1–12.

[Dum 07b] O. Dumitru, S. Duță, M. Mitrea, F. Prêteux, *"Gaussian Hypothesis for Video Watermarking Attacks: Drawbacks and Limitations"*, EUROCON 2007, Warsaw–Poland, September 2007, pp.849 – 855.

[Dum 07c] C.O. Dumitru, M. Mitrea, F. Prêteux, *"Capacité du Tatouage Vidéo dans le Domaine des Ondelettes: Analyse Comparée",* TAIMA 2007, Hammamet–Tunisia, May 2007, pp. 355 – 360.

[Dum 08a] O. Dumitru**,** M. Mitrea, F. Prêteux, *"DCT Domain Video Watermarking: Attack Estimation and Capacity Evaluation"*, ICINCO 2008, Funchal–Portugal, May 2008, pp. 239 – 244.

[Dum 08b] O. Dumitru, M. Mitrea, F. Prêteux, *"Theoretical Limits in DWT Video Watermarking"*, Proc. SPIE, Vol. 7075, San Diego–USA, August 2008, pp. 7075 0C: 1 – 9.

[Dum 08c] O. Dumitru, M. Mitrea, F. Prêteux, A. Pathak, *"Probability Density Function Estimation for Video in the DCT Domain"* Proc. SPIE, Vol. 6812, January 2008, San Jose–USA, pp. 6812 0L: 1 – 9.

[Fre 01] France Telecom & Envivio *MPEG-4: A Solution for Interactive Digital Television*, http://www.mp4if.org/resources/sampte143/SMPTEEnvivio.pdf

[Gam 80] A. EL Gamal, T.M. Cover, *"Multiple user Information Theory"*, Proc. of the IEEE, Vol. 68, No. 12, December 1980, pp. 1466 – 1483.

[Gel 80] S.I. Gel'fand, M.S. Pinsker, *"Coding for Channel with Random Parameters"*, Problems of Control and Information Theory, Vol. 9, No. 1, 1980, pp. 19 – 31.

[Gho 05] L. Ghouti, A. Bouridane, S. Boussakta, *"High Capacity Watermarking using Balanced Multiwavelet Transforms"*, Proc. International Conference on Image Processing (ICIP 2005), Vol. 1, September 2005, pp. 997 – 980.

[Gpa 09] GPAC Project – encode subtitles http://gpac.souceforge.net/doc_ttxt.php

[Hee 83] C. Heegard, A.E. Gamal, *"On the Capacity of Computer Memory with Defects"*, IEEE Transactions on Information Theory, Vol. 29, No. 5, September 1983, pp. 731 – 739.

[Jan 98] H. Jane, R. Iannella, *The Application of Metadata Standards to Video Indexing,* http://www.itee.uq.edu.au/~eresearch/papers/1998/ECDL2.pdf

[Kov 04] O. Koval, S. Voloshynovskiy, F. Perez-Gonzalez, *"Quantization-based Watermarking Performance Improvement using Host Statistics: AWGN Attack Case"*, Proc. of the 2004 Workshop on Multimedia and Security, Magdeburg, Germany, 2004, pp. 35– 39.

[Kun 01] D. Kundur, D. Hatzinakos, *"Diversity and Attack Characterization for Improved Robust Watermarking"*, IEEE Transactions on Signal Processing, Vol. 49, No. 10, October 2001, pp. 2382 – 2396.

[Kut 99] M. Kutter, F.A.P. Petitcolas, *"A Fair Benchmark for Image Watermarking Systems"*, Electronic Imaging 1999, Vol. 3657, San Jose–USA, January 1999.

[Lin 01a] C.Y. Lin, S.F. Chang, *"Zero-error Information Hiding Capacity of Digital Images"*, Proc. of the 2001 IEEE International Conference on Image Processing, Vol. 3, Greece, October 2001, pp. 1007 - 1010.

[Lin 01b] C.Y. Lin, S.F. Chang, *"Watermarking Capacity of Digital Images based on Domain-Specific Masking Effects",* Proc. of International Conference on Information Technology: Coding and Computing, April 2001, pp. 90 – 94.

[Lsm 09] Live Staging of Media Events – Video Indexing http://www.ist-live.org/intranet/school-of-informatics-university -of-bradford001-6

[Mat 04] G. Matz, P. Duhamel, *"Information Geometric Formulation and Interpretation of Accelerated Blahut-Arimoto-type Algorithms"*, Proc. of IEEE Information Theory Workshop, San Antonio, TX, October 2004, pp. 66 – 70.

[Mit 04a] M. Mitrea, F. Prêteux, A. Vlad, C. Fetița, *"The 2D-DCT Coefficient Statistical Behaviour: A Comparative Analysis on Different Types of Image Sequences"*, Journal of Optoelectronics and Advanced Materials, Vol. 6, 2004, pp. 95 – 102.

[Mit 04b] M. Mitrea, F. Prêteux, A. Vlad, *"Watermarking-oriented Video Modelling in the Wavelet Domain"*, WSEAS Transactions on Mathematics, Vol. 3, No. 1, January 2004, pp. 282 – 287.

[Mit 05a] M. Mitrea, F. Prêteux, S. Duță, M. Petrescu, *"Wavelet-based Mobile Video Watermarking: Spread Spectrum vs. Informed Embedding"*, Proc. SPIE, Vol. 6001, Boston – USA, October 2005, pp. 60010J: 1 – 9.

[Mit 05b] M. Mitrea, F. Prêteux, S. Duță, M. Petrescu, *"The StirMark Watermarking Attack in the DWT Domain"*, IWSSIP 2005, Vol. 2, Halkida – Greece, September 2006, pp. 5-9.

[Mit 06a] M. Mitrea, S. Duță, F. Prêteux, *"A Unified Approach to Multimedia Content Watermarking"*, TFIT 2006, Nancy-France, March 2006, pp. 275 – 289.

[Mit 06b] M. Mitrea, S. Duță, T. Zaharia, F. Prêteux, "Ensuring multimedia content enrichment by means of data hiding techniques", *Pro. SPIE*, Vol. 6383, October 2006, Boston–USA, pp. 63830P61-63830P6.

[Mit 06c] M. Mitrea, S. Duță, F. Prêteux A. Vlad, *"Video Content Distribution: Modelling the Watermarking Attacks as Noise Sources"*, COMMUNICATION 2006, Bucharest–Romania, June 2006, pp. 165 –168.

[Mit 07a] M. Mitrea, O. Dumitru, F. Prêteux A. Vlad, *"Zero Memory Information Sources Approximating to Video Watermarking Attacks",* Lecture Notes in Computer Science (LNCS), Vol. 4707, Part III, 2007, pp. 445 – 459.

[Mit 07b] M. Mitrea, O. Dumitru, F. Prêteux, "*Video Watermarking Capacity in the DWT Hierarchy",* Proc. SPIE, Vol. 6576, Orlando–USA, April 2007, pp.65760E: 1 – 10.

[Mit 07c] M. Mitrea, F. Prêteux, J. Nunez, *Procédé de Tatouage d'une Séquence Video*, European Patent No. 1804213, 2007.

[Mit 08] M. Mitrea, O. Dumitru, S. Duță, F. Prêteux, A. Vlad, *"A Comparative Study on Video Watermarking Capacity",* COMMUNICATIONS 2008, Bucharest–Romania, June 2008, pp. 335 – 338.

[Moo 99] D.S. Moore, G. P. McCabe, *Introduction to the practice of Statistics*, W.H. Freeman, New York, 1999.

[Mou 01] P. Moulin, *"The Role of Information Theory in Watermarking and its Application to Image watermarking",* Signal Processing (Elsevier), Vol. 81, pp. 1121 – 1139.

[Mou 02] P. Moulin, M.K. Mihcak, *"A Framework for Evaluating the Data-hiding Capacity of Image Sources"*, IEEE Transactions on Image Processing, Vol. 11, No. 9, September 2002, pp. 1029 – 1042.

[Mou 03] P. Moulin, J.A. O'Sullivan, *"Information-Theoretic Analysis of Information Hiding",* IEEE Transaction on Information Theory, Vol. 49, No. 3, March 2003, pp. 563 – 593.

[Mou 04] P. Moulin, M.K. Mihcak, *"The Parallel-Gaussian Watermarking Game",* IEEE Transaction on Information Theory, Vol. 50, No. 2, February 2004, pp. 272 – 289.

[Mpg 09] MPEG home page http://www.chiariglione.org/mpeg

[Nen 03] Y. Nenghai, C. Liangliang, F. Wen, L; Xuelong, *"Practical Analysis of Watermarking Capacity"*, Proc. ICCT 2003, pp. 1872 – 1877.

[Per 01] S. Pereira, S. Voloshynovski, M. Madueño, S. Marchand-Maillet and T. Pun, *"Second Generation Benchmarking and Application Oriented Evaluation",* Information Hiding Workshop III, Pittsburgh, PA, April 2001.

[Pet 98] F. Petitcolas, R. Anderson, M. Kuhn, *"Attacks on Copyright Marking Systems"*, Lecture Notes in Computer Science (LNCS), Vol. 1525, 1998, pp. 219 – 239.

[Pet 00] F. Petitcolas, *"Watermarking Schemes Evaluation"* I.E.E.E. Signal Processing, Vol. 17, No. 5, September 2000, pp. 58 – 64.

[Ser 98] S. Servetto, Ch. Podilchuk, K. Ramchandran, *"Capacity Issues in Digital Image Watermarking"*, Proc. of IEEE International Conference on Image Processing, Vol. 1, October 1998, pp.445 – 449.

[Sha 48] C.E. Shannon, *"A Mathematical Theory of Communication"*, The Bell System Technical Journal, Vol. 27, October 1948, pp. 379 – 423, 623 – 656.

[Sha 58] C.E. Shannon, *"Channels with Side Information at the Transmitter"*, IBM Journal, October 1958, pp .289 – 293.

[Sha 80] C.E. Shannon, W. Weaver, *The Mathematical Theory of Communication,* 8th ed., University of Illinois Press, 1980.

[Spi 09] Image Compression with Set Partitioning in Hierarchical Trees http://www.cipr.rpi.edu/research/SPIHT

[Sto 04] P.Stoica, Y.Selen, *"Cyclic Minimizers, Majorization Techniques, and the Expectation-Maximitation Algorithm: a Refresher"*, IEEE Signal Processing Magazine, pp. 112–114. January 2004.

[Su 01] K. Su, D. Kundur, D. Hatzinakos, *"A Content-dependent Spatially Localized Video Watermark for Resistance to Collusion and Interpolation Attacks"*, Proc. IEEE Int. Conf. on Image Processing, 2001.

[Tra 02] L. Trailovic, L. Pao, *"Variance Estimation and Ranking of Gaussian Mixture Distribution in Target Tracking Applications"*, Proc. of the 41st IEEE Conf. on decision and Control, Las Vegas – Nevada, USA, December 2002, pp. 2195 - 2201.

[Tul 49] W.G. Tuller, *"Theoretical Limitation on the Rate Transmission of Information",* Proc. of the IRE, Vol. 37, No. 5, May 1949, pp. 468 – 478.

[Vol 99] S. Voloshynovskiy, A. Herrigel, F. Jordan, *"A Noise Removal Attack for Watermarked Images"*, Multimedia and Security Workshop at ACM Multimedia'99, Orlando, Florida, October 1999, pp. 71 – 80.

[Vol 01a] S. Voloshynovskiy, S. Pereira, V. Iquise, T. Pun, *"Attacks Modelling: Towards a Second Generation Watermarking Benchmark"*, Signal Processing Vol. 81, 2001, pp. 1177 – 1214.

[Vol 01b] S. Voloshynovskiy, S. Pereira, T. Pun, J.J. Eggers, J.K. Su, *"Attacks on Digital Watermarking: Classification, Estimation-based Attacks and Benchmarks"*, Communication Magazine, Vol. 39, No. 8, August 2001, pp. 118 – 126.

[Vol 06] S. Voloshynovskiy, O. Koval, M.K. Mihcak, T; Pun, *"The Edge Process Model and its Application to Information-Hiding Capacity Analysis"*, IEEE Transactions on Signal Processing, Vol. 54, No. 5, May 2006, pp. 1813 – 1825.

[Von 08] P.O. Vontobel, A. Kavcic, D.M. Arnold, H-A. Loeliger, *"A Generalization of the Blahut–Arimoto Algorithm to Finite-State Channels"*, IEEE Transactions on Information Theory, Vol.54, Issue 5, May 2008, pp. 1887 – 1918.

[Wie 48] N. Wiener, *Cybernetics*, MIT Press, Cambridge, 1948.

[Zha 04a] F. Zhang, H. Zhang, *"Digital Watermarking Capacity and Reliability"*, Proc. of the IEEE International Conference on E-Commerce Technology, 2004, pp. 295-298.

[Zha 04b] F. Zhang, H. Zhang, *"Digital Watermarking Capacity Research"*, Proc. of the International Conference on Communication, Circuits and Systems (ICCCAS 2004), Vol. 2, June 2004, pp. 796 – 799.

[Zha 05] F. Zhang, H. Zhang, *"Wavelet Domain Watermarking Capacity Analysis",* Proc. of the SPIE Electronic Imaging and Multimedia Technology IV, Vol. 5637, February 2005, pp. 657 – 664.

The best way to predict the future is to invent it."

Alan Kay (1940 - ___)

(American computer scientist, Palo Alto Research Center, 1971)

Conclusion

A watermarking system can be modelled as a noisy channel. The watermark is a sample from the information source and should be recovered at the detection side. The elements making the watermark detection difficult can be modelled as the channel noise, namely the original content and the attacks.

The thesis focuses on the modelling of natural video and watermarking attacks and establishes with a mathematical rigour their stationarity (*i.e.* it is proved for the first time how a mathematical model, independent with respect to the processed data and the computation procedure can be obtained) as well as the corresponding first order approximations. In this respect, accurate statistical models were computed for the first time in the DWT and DCT coefficients hierarchy for natural video and watermarking attacks. Finally, these new models are involved in the entropy and capacity evaluation as an information theory application.

As explained in the theoretical introduction, stationarity/ergodicity are very sophisticated concepts, which can only be approached but never reached. Actually, this thesis represents just some necessary but not sufficient mathematical supports for the natural video / watermarking attack stationarity. On the one hand, in the case the estimation procedure was unsuccessful, it can be stated that no general model exists. On the other hand, in the cases the procedure was successful, future investigations are to be done, on both fundamental and applicative levels. For instance, on the fundamental level, higher order stationarity investigation and resuming the experiments on a different corpus are first priorities. On the applicative level, effort should be done in validating these models under the watermarking and compression frameworks. Particularly, effort will be directed toward devising the optimal mark insertion/detection rule (*i.e.* the watermark method reaching the capacity) in the DWT/DCT domains. Resuming the statistical investigation in the spatial domain can also be considered as a potential future work.

The experiments are carried out in the framework of the ***TAMUSO*** and ***DWD projects*** (partnership between the ARTEMIS Department and SFR – Vodafone mobile service provider in France) and of the ***HD3D-IIO project*** (a CapDigital competitiveness cluster in Ile de France).

The thesis can be synoptically represented as in the following figure.

	Stationarity	Accurate Model
Natural Video	**State-of-the-art:** - Always supposed, never investigated.	**State-of-the-art:** - Different popular distributions (*e.g.* Gaussian, generalised Gaussian, Laplacian) are empirical selected according to a particular application. - Sometimes statistical tests are considered.
	Deadlock: - Very sophisticated mathematical concept, requiring huge statistical investigations. - Lack of a mathematical procedure for investigation.	**Deadlock:** - Lack of a sound support for the stationarity. - Huge statistical investigation.
	Thesis Contributions:	**Thesis Contributions:**
	ART.MOD-V algorithm	
	- Establishing for the first time whether and how the first order stationarity works.	- Refuting the Gaussian model on mathematical basis. - Establishing accurate and general models for natural video, and computing the corresponding continuous entropy.
	Perspectives: - Large order stationarity investigation.	**Perspectives:** - Exploiting the models in: compression, denoising, indexation, classification, watermarking, …

<table>
<tr><th></th><th>Stationarity</th><th>Accurate Model</th></tr>
<tr><td rowspan="6">Watermarking Attacks</td><td>State-of-the-art:
- Implicitly included in the AWGN (additive white Gaussian noise) hypothesis but never discussed and/or investigated.</td><td>State-of-the-art:
- AWGN (additive white Gaussian noise) hypothesis: always considered, although acknowledged as simplistic.</td></tr>
<tr><td>Deadlock:
- Lack of an investigation procedure.
- Huge experimental work.</td><td>Deadlock:
- Lack of a sound method for investigation.
- No reliable approach for computing the theoretical limits for watermarking.</td></tr>
<tr><td>Thesis Contributions:</td><td>Thesis Contributions:</td></tr>
<tr><td colspan="2">ART.MOD-A algorithm</td></tr>
<tr><td>- Proving for the first time the first order stationarity of the random processes representing the watermarking attacks.</td><td>- Refuting the Gaussian model on mathematical basis.
- Estimating for the first time accurate models for a large class of attacks (while keeping the additive hypothesis).
- Validating the models by speeding up by a factor of 100 an existing watermarking method.
ART.CAP
- Identifying the watermarking theoretical limits by computing accurate limits for watermarking capacity and discuss them for in-band enrichment purposes.</td></tr>
<tr><td>Perspectives:
- Large order stationarity investigation.</td><td>Perspectives:
- Optimising the watermarking detection rule under non-Gaussian noise constraints.</td></tr>
</table>

Appendix 0

Probability Distributions Involved in the Experiments

The most important *pdf*s and *cdf*s for the continuous case are presented further and a graphical representation is realized in Figure A.0.1:

1. **The uniform distribution**

A random variable X is said to be uniformly distributed on the interval (a,b), $a<b$, if its *pdf* is defined by the eq. (A.0.1) and the *cdf* of X is given by eq. (A.0.2):

$$f_X(x)=\begin{cases}\dfrac{1}{b-a}, & a\le x\le b\\ 0, & \text{otherwise}\end{cases}, \tag{A.0.1}$$

$$F_X(x)=P(X\le x)=\int_{-\infty}^{x} f_X(u)du=\begin{cases}0, & x<a\\ \dfrac{x-a}{b-a}, & a\le x<b\\ 1, & x\ge b\end{cases}. \tag{A.0.2}$$

2. **The normal distribution (Gaussian distribution)**

The Gaussian distribution is one of the most important continuous distributions and its *pdf* is given by the eq. (A.0.3) and the *cdf* by eq. (A.0.4):

$$f_X(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right],\ x\in R, \tag{A.0.3}$$

$$F_X(x)=P(X\le x)=\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{x}\exp\left[-\frac{(u-\mu)^2}{2\sigma^2}\right]du, \tag{A.0.4}$$

where it can be proved that μ and σ are the mean and the standard deviation of the X ($-\infty<\mu<\infty,\sigma>0$).

3. **The exponential distribution**

A random variable X has an exponential distribution with parameters α and β, $\beta>0$, if its probability density function is given by the equation:

$$f_X(x)=\begin{cases}\dfrac{1}{\beta}e^{-\frac{1}{\beta}(x-a)}, & x\ge a, a\in R\\ 0, & \text{otherwise}\end{cases}. \tag{A.0.5}$$

For a particular case, $a=0$ and $\alpha=\dfrac{1}{\beta}$ the *pdf* and *cdf* becomes:

$$f_X(x)=\begin{cases}\alpha e^{-\alpha x}, x\geq 0\\ 0\quad, \text{otherwise}\end{cases}, \tag{A.0.6}$$

$$F_X(x)=P(X\leq x)=\int_{-\infty}^{x} f_X(x)=1-e^{-\alpha x},\ x\geq 0\,. \tag{A.0.7}$$

4. **The Laplacian distribution**

The Laplace density function and *cdf* are defined as:

$$f_X(x)=\frac{1}{2\lambda}\begin{cases}\exp\left(-\dfrac{\mu-x}{\lambda}\right), & x<\mu\\ \exp\left(-\dfrac{x-\mu}{\lambda}\right), & x\geq\mu\end{cases},\ \mu\in R,\ \lambda>0\,, \tag{A.0.8}$$

$$F_X(x)=\begin{cases}\dfrac{1}{2}\exp\left(-\dfrac{\mu-x}{\lambda}\right), & x<\mu\\ 1-\dfrac{1}{2}\exp\left(-\dfrac{x-\mu}{\lambda}\right), & x\geq\mu\end{cases}. \tag{A.0.9}$$

5. **The Rayleigh distribution**

The *pdf* and the *cdf* for Rayleigh distribution are defined as:

$$f_X(x)=\begin{cases}\dfrac{x}{\sigma^2}\exp\left(-\dfrac{x^2}{2\sigma^2}\right), & y\geq 0\\ 0, & \text{otherwise}\end{cases}, \tag{A.0.10}$$

$$F_X(x)=\begin{cases}1-e^{-\dfrac{x^2}{2\sigma^2}}, x\geq 0\\ 0\quad, \text{otherwise}\end{cases}. \tag{A.0.11}$$

6. **The Chi-Square distribution**

The *pdf* of the Chi-Square distribution with *n* degrees of freedom is defined as:

$$f_X(x)=\begin{cases}\dfrac{1}{2^{n/2}\Gamma(n/2)}x^{(n/2)-1}e^{-x/2}, & x\geq 0\\ 0 & \text{, otherwise}\end{cases},\quad n\in N^*, \tag{A.0.12}$$

where $\Gamma(x)$ stands for the Gamma function: $\Gamma(x)=\int_0^{\infty} x^{\alpha-1}e^{-x}dx,\ \alpha>0$.

7. **The Student distribution**

The Student distribution with n degrees of freedom has its *pdf* defined as:

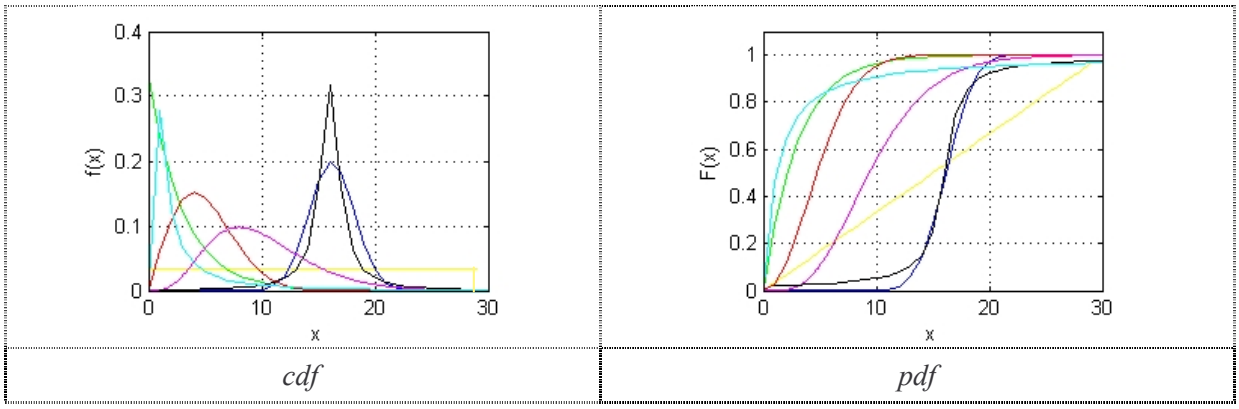
$$f_T(t)=\frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\Gamma(n/2)}\left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}}=\frac{1}{B(\frac{1}{2},\frac{n}{2})\sqrt{n}}\left(1+\frac{t^2}{n}\right)^{-\frac{n+1}{2}},\ t\in R,\ n\in N^*, \qquad \text{(A.0.13)}$$

where B is the beta function, $B(\alpha,\beta)=\Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)\quad \alpha,\beta>0$.

8. **The Fisher distribution**

For the Fisher distribution with n_1 and n_2 degrees of freedom has the *pdf* given by:

$$f_U(u)=\frac{\Gamma\left(\frac{n_1+n_2}{2}\right)}{\Gamma(n_1/2)\Gamma(n_2/2)}n_1^{n_1/2}n_2^{n_2/2}\frac{u^{\frac{n_1}{2}-1}}{(n_2+n_1u)^{(n_1+n_2)/2}},\ u>0,\ n_1,n_2>N^*. \qquad \text{(A.0.14)}$$



cdf *pdf*

Figure A.0.1. The *pdf* and *cdf* (continuous distributions) are displayed: uniform distribution (yellow), normal distribution with $\mu=0.15,\sigma=2$ (blue), exponential distribution with $\alpha=3$ (green), Rayleigh distribution with $\sigma=4$ (red), Chi-Square distribution with $n=10$ degrees of freedom (magenta), Student distribution with $n=1$ degrees of freedom (black), Fisher distribution with $n_1=3,\ n_2=2$ degrees of freedom (cyan).

Appendix I

Natural Video Modelling

A.I.1. The 2D-DWT coefficients modelling

A.I.1.1. Model Computation – IVM and MVM

Table A.I.1. IVM for (9,7) DWT coefficient hierarchy and low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.075	0.070	0.074	0.134	0.071	0.131	0.151	0.098	0.060	0.137	0.039
	$\mu(k)$	0.575	0.405	0.646	0.497	0.544	0.535	0.627	0.628	0.654	0.850	
	$\sigma(k)$	0.112	0.076	0.138	0.089	0.100	0.098	0.062	0.133	0.140	0.125	
25	$P(k)$	0.082	0.087	0.097	0.007	0.237	0.106	0.048	0.055	0.040	0.245	0.026
	$\mu(k)$	0.232	0.229	0.226	0.522	0.222	0.228	0.306	0.256	0.352	0.232	
	$\sigma(k)$	0.051	0.051	0.052	0.001	0.049	0.051	0.005	0.016	0.021	0.050	
50	$P(k)$	0.071	0.057	0.007	0.152	0.142	0.148	0.101	0.035	0.063	0.224	0.004
	$\mu(k)$	0.155	0.122	0.369	0.148	0.181	0.168	0.161	0.117	0.156	0.194	
	$\sigma(k)$	0.044	0.054	0.001	0.033	0.040	0.007	0.042	0.053	0.044	0.037	
75	$P(k)$	0.107	0.090	0.096	0.083	0.126	0.164	0.096	0.052	0.028	0.160	0.033
	$\mu(k)$	0.130	0.128	0.122	0.127	0.131	0.135	0.107	0.087	0.202	0.107	
	$\sigma(k)$	0.033	0.033	0.034	0.034	0.001	0.032	0.027	0.042	0.060	0.025	
100	$P(k)$	0.064	0.101	0.056	0.070	0.090	0.080	0.216	0.073	0.043	0.207	0.002
	$\mu(k)$	0.086	0.094	0.084	0.130	0.090	0.090	0.090	0.090	0.100	0.106	
	$\sigma(k)$	0.033	0.025	0.035	0.039	0.026	0.026	0.024	0.025	0.040	0.015	
125	$P(k)$	0.125	0.072	0.117	0.087	0.053	0.060	0.107	0.077	0.202	0.102	0.028
	$\mu(k)$	0.057	0.109	0.078	0.068	0.076	0.075	0.083	0.074	0.079	0.071	
	$\sigma(k)$	0.024	0.022	0.021	0.026	0.028	0.027	0.021	0.027	0.011	0.027	
150	$P(k)$	0.074	0.164	0.071	0.061	0.037	0.050	0.292	0.090	0.034	0.126	0.017
	$\mu(k)$	0.059	0.066	0.060	0.60	0.059	0.059	0.060	0.061	0.059	0.060	
	$\sigma(k)$	0.027	0.021	0.023	0.023	0.023	0.022	0.012	0.012	0.027	0.023	
175	$P(k)$	0.094	0.103	0.091	0.122	0.024	0.147	0.148	0.089	0.065	0.117	0.025
	$\mu(k)$	0.038	0.030	0.054	0.050	0.077	0.053	0.051	0.041	0.070	0.050	
	$\sigma(k)$	0.013	0.011	0.015	0.015	0.020	0.010	0.009	0.014	0.021	0.015	
200	$P(k)$	0.058	0.178	0.057	0.083	0.099	0.056	0.068	0.054	0.157	0.189	0.010
	$\mu(k)$	0.041	0.036	0.040	0.059	0.043	0.041	0.043	0.041	0.032	0.033	
	$\sigma(k)$	0.013	0.014	0.013	0.019	0.012	0.013	0.012	0.013	0.007	0.009	
225	$P(k)$	0.110	0.030	0.087	0.192	0.170	0.086	0.078	0.097	0.111	0.041	0.026
	$\mu(k)$	0.033	0.048	0.032	0.024	0.030	0.034	0.009	0.033	0.032	0.049	
	$\sigma(k)$	0.009	0.015	0.009	0.005	0.009	0.009	0.004	0.009	0.009	0.015	
250	$P(k)$	0.073	0.164	0.132	0.095	0.247	0.034	0.029	0.074	0.067	0.085	0.338
	$\mu(k)$	0.040	0.027	0.017	0.018	0.017	0.021	0.028	0.019	0.018	0.017	
	$\sigma(k)$	0.008	0.002	0.007	0.007	0.007	0.005	0.010	0.007	0.007	0.007	
275	$P(k)$	0.104	0.123	0.133	0.070	0.062	0.067	0.075	0.045	0.218	0.094	0.028
	$\mu(k)$	0.026	0.014	0.016	0.018	0.017	0.017	0.018	0.018	0.008	0.018	
	$\sigma(k)$	0.008	0.005	0.006	0.006	0.006	0.006	0.006	0.006	0.004	0.006	
300	$P(k)$	0.450	0.030	0.233	0.032	0.016	0.021	0.032	0.012	0.035	0.034	0.021
	$\mu(k)$	0.007	0.013	0.012	0.013	0.015	0.020	0.013	0.012	0.013	0.020	
	$\sigma(k)$	0.003	0.003	0.004	0.003	0.005	0.006	0.004	0.004	0.003	0.005	
325	$P(k)$	0.044	0.103	0.180	0.202	0.086	0.070	0.060	0.139	0.071	0.050	0.030
	$\mu(k)$	0.007	0.007	0.005	0.002	0.012	0.008	0.008	0.005	0.008	0.008	
	$\sigma(k)$	0.002	0.002	0.002	0.001	0.004	0.002	0.002	0.001	0.002	0.002	
350	$P(k)$	0.067	0.198	0.076	0.083	0.055	0.062	0.031	0.027	0.044	0.358	0.996
	$\mu(k)$	0.004	0.002	0.004	0.003	0.004	0.004	0.003	0.003	0.004	0.003	
	$\sigma(k)$	0.002	0.001	0.002	0.002	0.002	0.002	0.005	0.005	0.002	0.001	

Table A.I.2. MVM for (9,7) DWT coefficient hierarchy and low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.129	0.046	0.100	0.133	0.110	0.116	0.060	0.133	0.060	0.110	0.038
	$\mu(k)$	0.574	0.685	0.443	0.566	0.314	0.383	0.515	0.514	0.431	0.299	
	$\sigma(k)$	0.125	0.226	0.142	0.128	0.108	0.111	0.141	0.141	0.139	0.105	
25	$P(k)$	0.090	0.075	0.098	0.113	0.087	0.070	0.091	0.219	0.082	0.071	0.045
	$\mu(k)$	0.166	0.181	0.205	0.212	0.133	0.197	0.177	0.0867	0.233	0.197	
	$\sigma(k)$	0.060	0.061	0.055	0.053	0.012	0.058	0.061	0.040	0.049	0.058	
50	$P(k)$	0.070	0.102	0.054	0.046	0.150	0.165	0.052	0.104	0.142	0.115	0.037
	$\mu(k)$	0.140	0.150	0.134	0.138	0.043	0.089	0.127	0.088	0.145	0.014	
	$\sigma(k)$	0.041	0.045	0.037	0.040	0.016	0.024	0.033	0.023	0.043	0.043	
75	$P(k)$	0.150	0.109	0.163	0.103	0.075	0.104	0.094	0.024	0.075	0.102	0.045
	$\mu(k)$	0.027	0.073	0.069	0.098	0.097	0.115	0.117	0.047	0.112	0.118	
	$\sigma(k)$	0.009	0.021	0.016	0.028	0.028	0.032	0.032	0.001	0.031	0.031	
100	$P(k)$	0.114	0.110	0.129	0.127	0.069	0.020	0.077	0.129	0.119	0.105	0.036
	$\mu(k)$	0.070	0.053	0.025	0.091	0.091	0.013	0.079	0.049	0.057	0.107	
	$\sigma(k)$	0.018	0.018	0.011	0.023	0.023	0.001	0.024	0.016	0.018	0.019	
125	$P(k)$	0.072	0.116	0.046	0.139	0.145	0.040	0.105	0.112	0.122	0.102	0.040
	$\mu(k)$	0.055	0.029	0.061	0.092	0.031	0.062	0.012	0.057	0.059	0.059	
	$\sigma(k)$	0.013	0.009	0.017	0.012	0.010	0.018	0.004	0.015	0.016	0.017	
150	$P(k)$	0.007	0.223	0.043	0.107	0.074	0.122	0.121	0.092	0.152	0.058	0.043
	$\mu(k)$	0.047	0.015	0.048	0.047	0.051	0.032	0.032	0.053	0.068	0.048	
	$\sigma(k)$	0.001	0.006	0.015	0.015	0.014	0.006	0.009	0.014	0.012	0.015	
175	$P(k)$	0.047	0.103	0.082	0.104	0.075	0.177	0.065	0.095	0.122	0.130	0.040
	$\mu(k)$	0.048	0.038	0.036	0.048	0.008	0.015	0.028	0.038	0.025	0.058	
	$\sigma(k)$	0.013	0.009	0.009	0.013	0.001	0.004	0.007	0.009	0.006	0.011	
200	$P(k)$	0.026	0.061	0.066	0.154	0.143	0.086	0.251	0.072	0.127	0.013	0.043
	$\mu(k)$	0.030	0.032	0.030	0.022	0.045	0.032	0.009	0.033	0.025	0.014	
	$\sigma(k)$	0.009	0.010	0.009	0.006	0.007	0.010	0.004	0.010	0.006	0.001	
225	$P(k)$	0.059	0.078	0.129	0.020	0.092	0.024	0.105	0.079	0.287	0.125	0.042
	$\mu(k)$	0.030	0.021	0.019	0.027	0.024	0.027	0.005	0.032	0.011	0.036	
	$\sigma(k)$	0.008	0.005	0.004	0.008	0.006	0.008	0.002	0.008	0.003	0.007	
250	$P(k)$	0.039	0.045	0.196	0.032	0.031	0.081	0.199	0.263	0.067	0.045	0.040
	$\mu(k)$	0.025	0.025	0.013	0.024	0.025	0.024	0.014	0.006	0.024	0.024	
	$\sigma(k)$	0.007	0.007	0.004	0.007	0.007	0.007	0.005	0.002	0.007	0.007	
275	$P(k)$	0.101	0.180	0.184	0.121	0.035	0.025	0.022	0.105	0.0172	0.211	0.049
	$\mu(k)$	0.022	0.011	0.010	0.006	0.016	0.017	0.017	0.017	0.017	0.003	
	$\sigma(k)$	0.003	0.003	0.003	0.001	0.005	0.005	0.005	0.005	0.005	0.001	
300	$P(k)$	0.087	0.195	0.181	0.036	0.264	0.042	0.035	0.027	0.098	0.035	0.044
	$\mu(k)$	0.008	0.014	0.006	0.009	0.002	0.009	0.009	0.010	0.006	0.009	
	$\sigma(k)$	0.002	0.004	0.001	0.003	0.001	0.003	0.003	0.003	0.002	0.003	
325	$P(k)$	0.295	0.036	0.076	0.037	0.137	0.034	0.135	0.118	0.093	0.038	0.035
	$\mu(k)$	0.002	0.004	0.005	0.004	0.004	0.004	0.005	0.010	0.004	0.004	
	$\sigma(k)$	0.001	0.002	0.003	0.002	0.002	0.002	0.003	0.002	0.002	0.002	
350	$P(k)$	0.099	0.188	0.046	0.036	0.025	0.028	0.132	0.038	0.251	0.156	0.024
	$\mu(k)$	0.001	0.001	0.003	0.002	0.003	0.005	0.002	0.003	0.001	0.002	
	$\sigma(k)$	0.001	0.002	0.002	0.002	0.002	0.002	0.001	0.002	0.001	0.002	

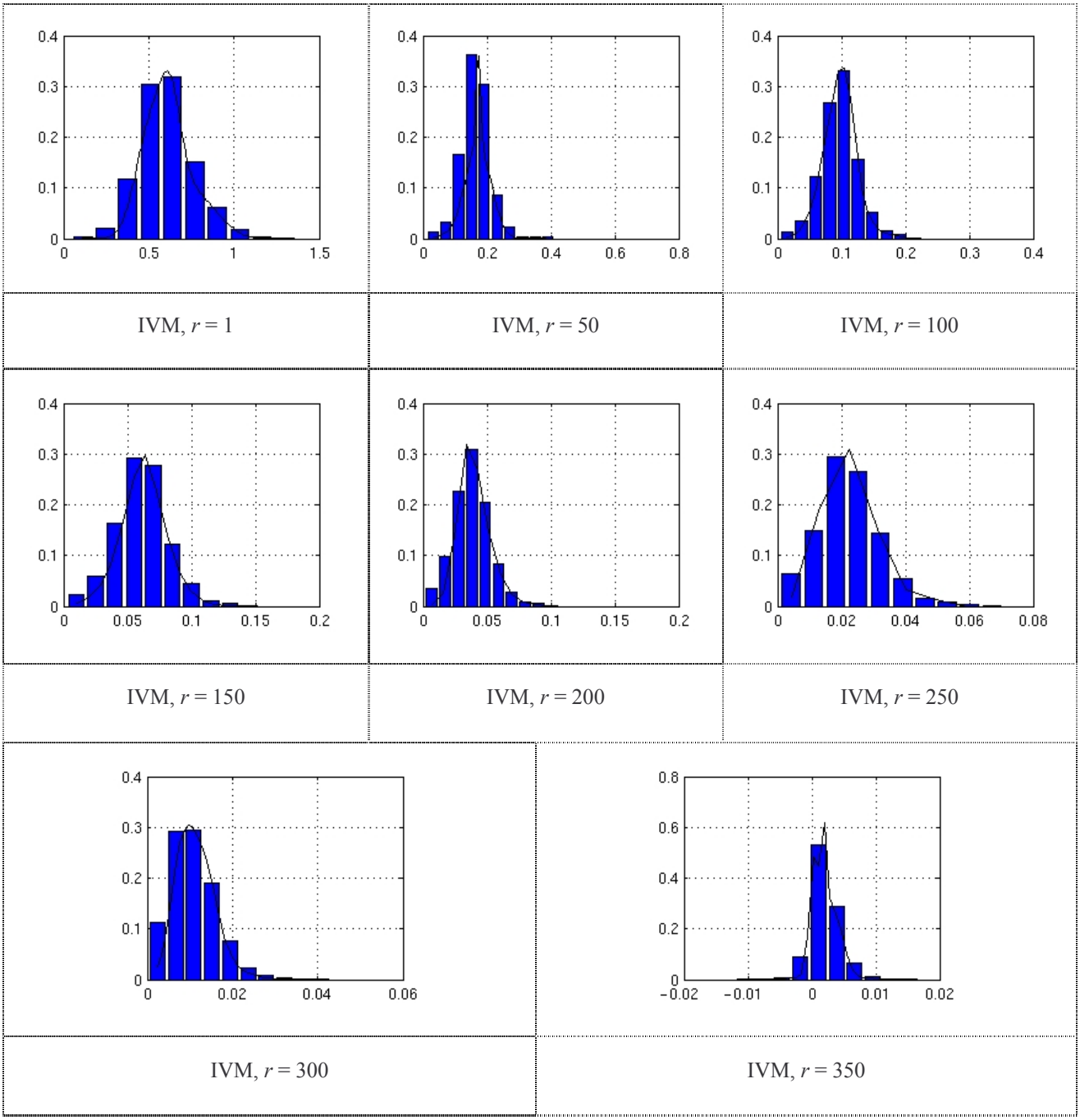


Figure A.I.1. IVM and histograms computed for (9,7) DWT for eight ranks and low quality video.

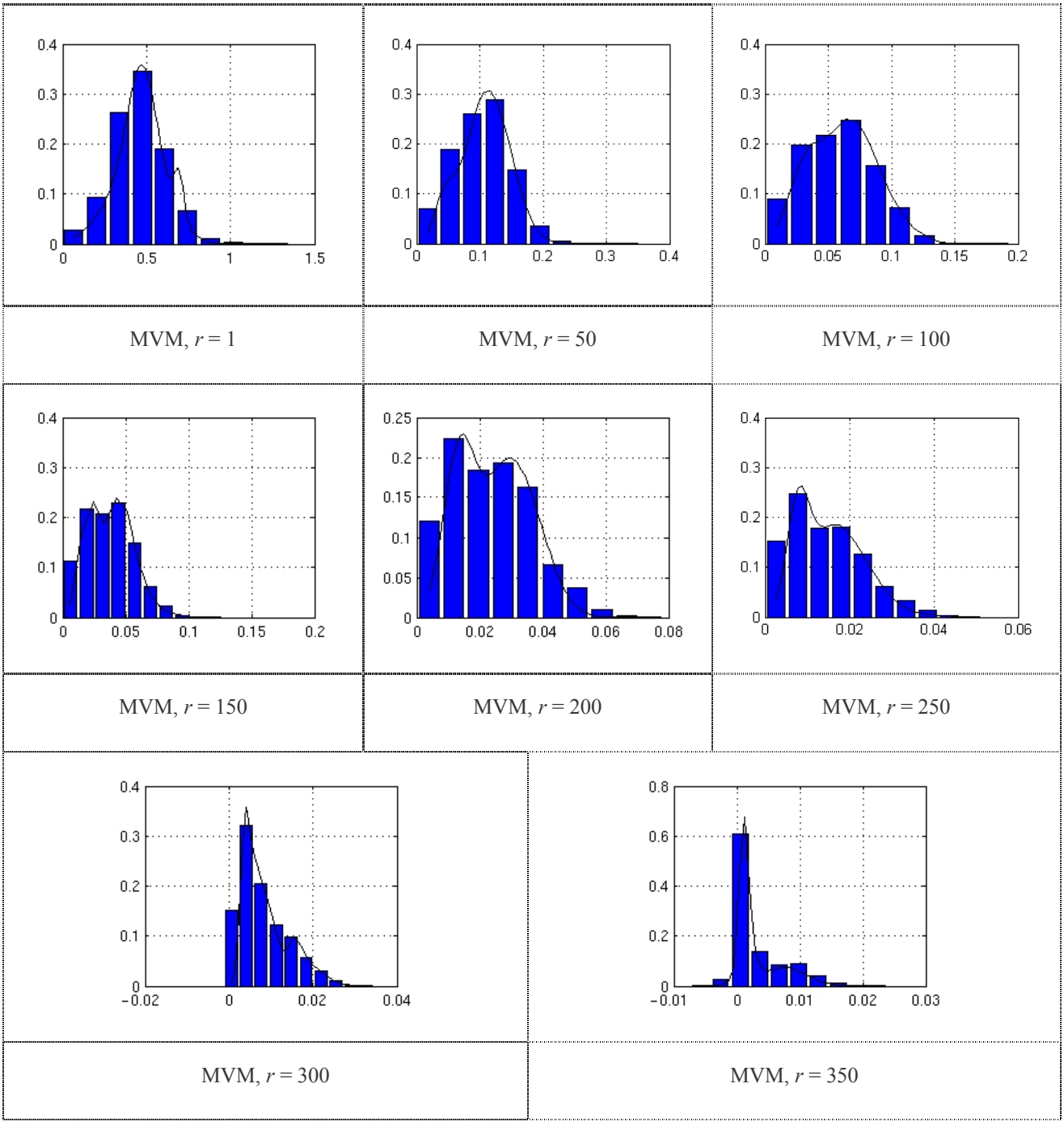


Figure A.I.2. IVM and histograms for (9,7) DWT computed for eight ranks and low quality video.

Table A.I.3. IVM for (9,7) DWT coefficient hierarchy and high quality video.

Rank	Model parameters											Error
1	$P(k)$	0.119	0.122	0.107	0.119	0.129	0.118	0.117	0.014	0.096	0.059	0.028
	$\mu(k)$	0.570	0.713	0.740	0.782	0.508	0.921	0.747	1.730	0.628	0.849	
	$\sigma(k)$	0.124	0.148	0.157	0.175	0.108	0.169	0.159	0.044	0.139	0.109	
25	$P(k)$	0.099	0.138	0.068	0.065	0.234	0.144	0.016	0.073	0.073	0.088	0.022
	$\mu(k)$	0.342	0.348	0.412	0.494	0.308	0.398	0.455	0.365	0.455	0.333	
	$\sigma(k)$	0.119	0.067	0.105	0.044	0.061	0.021	0.001	0.119	0.102	0.121	
50	$P(k)$	0.132	0.097	0.136	0.045	0.012	0.075	0.105	0.056	0.121	0.220	0.028
	$\mu(k)$	0.322	0.321	0.327	0.502	0.700	0.327	0.260	0.324	0.177	0.325	
	$\sigma(k)$	0.058	0.059	0.051	0.088	0.030	0.006	0.041	0.058	0.026	0.057	
75	$P(k)$	0.065	0.083	0.176	0.069	0.007	0.071	0.120	0.041	0.239	0.127	0.033
	$\mu(k)$	0.305	0.242	0.295	0.321	0.280	0.304	0.158	0.158	0.266	0.223	
	$\sigma(k)$	0.079	0.051	0.036	0.077	0.001	0.079	0.038	0.004	0.033	0.049	
100	$P(k)$	0.188	0.047	0.284	0.061	0.059	0.047	0.065	0.079	0.069	0.100	0.033
	$\mu(k)$	0.140	0.226	0.250	0.277	0.227	0.223	0.240	0.310	0.220	0.241	
	$\sigma(k)$	0.032	0.048	0.033	0.070	0.049	0.048	0.060	0.065	0.049	0.034	
125	$P(k)$	0.041	0.287	0.145	0.079	0.061	0.059	0.063	0.080	0.091	0.092	0.035
	$\mu(k)$	0.362	0.210	0.247	0.204	0.245	0.204	0.141	0.143	0.279	0.126	
	$\sigma(k)$	0.026	0.022	0.018	0.064	0.053	0.064	0.053	0.054	0.039	0.018	
150	$P(k)$	0.072	0.058	0.107	0.121	0.062	0.290	0.059	0.065	0.097	0.069	0.038
	$\mu(k)$	0.228	0.167	0.118	0.143	0.243	0.200	0.211	0.153	0.236	0.236	
	$\sigma(k)$	0.062	0.052	0.015	0.046	0.059	0.024	0.062	0.048	0.017	0.061	
175	$P(k)$	0.118	0.091	0.054	0.082	0.068	0.086	0.088	0.216	0.097	0.099	0.034
	$\mu(k)$	0.170	0.193	0.133	0.103	0.220	0.131	0.215	0.199	0.234	0.189	
	$\sigma(k)$	0.049	0.047	0.045	0.013	0.064	0.045	0.062	0.021	0.065	0.048	
200	$P(k)$	0.085	0.083	0.251	0.067	0.108	0.108	0.062	0.094	0.047	0.097	0.030
	$\mu(k)$	0.094	0.180	0.187	0.226	0.170	0.125	0.135	0.173	0.229	0.175	
	$\sigma(k)$	0.013	0.065	0.026	0.063	0.036	0.052	0.054	0.044	0.062	0.041	
225	$P(k)$	0.078	0.152	0.271	0.057	0.047	0.069	0.107	0.079	0.036	0.104	0.030
	$\mu(k)$	0.134	0.091	0.183	0.141	0.223	0.158	0.155	0.157	0.218	0.141	
	$\sigma(k)$	0.042	0.022	0.027	0.041	0.056	0.035	0.036	0.036	0.057	0.041	
250	$P(k)$	0.199	0.063	0.147	0.208	0.162	0.080	0.049	0.016	0.033	0.042	0.028
	$\mu(k)$	0.086	0.132	0.148	0.174	0.171	0.164	0.146	0.193	0.295	0.163	
	$\sigma(k)$	0.017	0.025	0.026	0.032	0.033	0.034	0.022	0.040	0.021	0.034	
275	$P(k)$	0.105	0.148	0.048	0.196	0.164	0.025	0.051	0.031	0.016	0.216	0.030
	$\mu(k)$	0.141	0.076	0.265	0.178	0.131	0.160	0.140	0.089	0.174	0.142	
	$\sigma(k)$	0.029	0.015	0.025	0.021	0.021	0.036	0.029	0.009	0.042	0.029	
300	$P(k)$	0.079	0.073	0.066	0.121	0.086	0.145	0.036	0.076	0.130	0.195	0.032
	$\mu(k)$	0.218	0.0143	0.071	0.113	0.099	0.118	0.133	0.129	0.129	0.159	
	$\sigma(k)$	0.048	0.039	0.001	0.031	0.024	0.047	0.004	0.036	0.035	0.016	
325	$P(k)$	0.110	0.071	0.184	0.035	0.173	0.056	0.032	0.064	0.101	0.174	0.032
	$\mu(k)$	0.136	0.131	0.127	0.264	0.141	0.141	0.197	0.131	0.128	0.069	
	$\sigma(k)$	0.031	0.031	0.030	0.021	0.031	0.032	0.036	0.031	0.030	0.014	
350	$P(k)$	0.070	0.169	0.120	0.074	0.037	0.125	0.082	0.101	0.028	0.195	0.033
	$\mu(k)$	0.123	0.067	0.108	0.120	0.174	0.134	0.128	0.123	0.216	0.134	
	$\sigma(k)$	0.029	0.017	0.028	0.029	0.047	0.028	0.029	0.029	0.037	0.028	

Table A.I.4. MVM for (9,7) DWT coefficient hierarchy and high quality video.

Rank	Model parameters											Error
1	$P(k)$	0.098	0.071	0.119	0.054	0.089	0.115	0.078	0.152	0.114	0.111	0.042
	$\mu(k)$	0.454	0.709	0.508	0.682	0.791	0.527	0.598	0.302	0.435	0.696	
	$\sigma(k)$	0.121	0.175	0.120	0.178	0.158	0.120	0.141	0.089	0.117	0.017	
25	$P(k)$	0.066	0.079	0.077	0.121	0.062	0.097	0.152	0.092	0.150	0.104	0.034
	$\mu(k)$	0.245	0.449	0.398	0.284	0.302	0.301	0.218	0.324	0.137	0.283	
	$\sigma(k)$	0.062	0.051	0.026	0.049	0.077	0.077	0.052	0.079	0.041	0.071	
50	$P(k)$	0.159	0.161	0.066	0.077	0.143	0.085	0.075	0.071	0.228	0.079	0.035
	$\mu(k)$	0.188	0.146	0.333	0.277	0.238	0.203	0.372	0.228	0.121	0.210	
	$\sigma(k)$	0.054	0.018	0.019	0.058	0.041	0.057	0.055	0.060	0.041	0.058	
75	$P(k)$	0.038	0.186	0.072	0.096	0.164	0.063	0.085	0.110	0.078	0.107	0.039
	$\mu(k)$	0.363	0.097	0.146	0.200	0.279	0.219	0.213	0.170	0.179	0.179	
	$\sigma(k)$	0.065	0.027	0.037	0.048	0.034	0.046	0.048	0.044	0.046	0.046	
100	$P(k)$	0.262	0.057	0.084	0.103	0.164	0.045	0.028	0.144	0.067	0.046	0.036
	$\mu(k)$	0.090	0.180	0.229	0.180	0.259	0.164	0.173	0.148	0.160	0.175	
	$\sigma(k)$	0.029	0.048	0.032	0.008	0.031	0.047	0.048	0.023	0.045	0.048	
125	$P(k)$	0.032	0.057	0.043	0.139	0.183	0.071	0.041	0.048	0.290	0.096	0.036
	$\mu(k)$	0.203	0.168	0.164	0.156	0.234	0.177	0.190	0.174	0.085	0.124	
	$\sigma(k)$	0.031	0.046	0.046	0.013	0.033	0.045	0.041	0.046	0.026	0.022	
150	$P(k)$	0.134	0.065	0.207	0.075	0.111	0.065	0.072	0.115	0.062	0.092	0.039
	$\mu(k)$	0.196	0.141	0.069	0.126	0.235	0.147	0.137	0.113	0.139	0.141	
	$\sigma(k)$	0.015	0.036	0.023	0.033	0.019	0.016	0.036	0.029	0.036	0.036	
175	$P(k)$	0.156	0.150	0.038	0.202	0.115	0.047	0.059	0.097	0.062	0.074	0.038
	$\mu(k)$	0.100	0.207	0.138	0.063	0.116	0.147	0.143	0.153	0.144	0.140	
	$\sigma(k)$	0.026	0.025	0.036	0.018	0.026	0.036	0.036	0.035	0.036	0.036	
200	$P(k)$	0.102	0.141	0.056	0.052	0.090	0.257	0.087	0.066	0.070	0.079	0.038
	$\mu(k)$	0.134	0.103	0.146	0.165	0.103	0.057	0.176	0.199	0.141	0.152	
	$\sigma(k)$	0.039	0.022	0.041	0.040	0.023	0.018	0.036	0.060	0.040	0.042	
225	$P(k)$	0.074	0.149	0.238	0.053	0.058	0.061	0.054	0.126	0.039	0.147	0.038
	$\mu(k)$	0.118	0.044	0.084	0.127	0.0129	0.136	0.125	0.096	0.132	0.183	
	$\sigma(k)$	0.032	0.016	0.024	0.033	0.033	0.032	0.033	0.027	0.033	0.024	
250	$P(k)$	0.170	0.095	0.093	0.050	0.063	0.081	0.119	0.141	0.087	0.100	0.037
	$\mu(k)$	0.067	0.127	0.188	0.129	0.102	0.100	0.039	0.136	0.134	0.086	
	$\sigma(k)$	0.014	0.031	0.019	0.031	0.022	0.017	0.011	0.029	0.030	0.022	
275	$P(k)$	0.001	0.204	0.244	0.067	0.096	0.099	0.105	0.076	0.043	0.065	0.035
	$\mu(k)$	0.137	0.044	0.079	0.099	0.176	0.120	0.118	0.122	0.121	0.119	
	$\sigma(k)$	0.013	0.014	0.019	0.027	0.019	0.029	0.029	0.029	0.029	0.029	
300	$P(k)$	0.071	0.124	0.079	0.068	0.014	0.125	0.065	0.174	0.148	0.132	0.037
	$\mu(k)$	0.160	0.105	0.088	0.104	0.081	0.149	0.101	0.062	0.034	0.083	
	$\sigma(k)$	0.022	0.027	0.022	0.027	0.001	0.024	0.026	0.015	0.011	0.020	
325	$P(k)$	0.074	0.088	0.060	0.175	0.138	0.065	0.067	0.132	0.071	0.130	0.037
	$\mu(k)$	0.111	0.109	0.119	0.058	0.154	0.114	0.108	0.033	0.087	0.067	
	$\sigma(k)$	0.025	0.019	0.026	0.012	0.020	0.026	0.020	0.008	0.004	0.013	
350	$P(k)$	0.161	0.077	0.117	0.171	0.138	0.053	0.058	0.111	0.076	0.038	0.035
	$\mu(k)$	0.058	0.088	0.060	0.032	0.102	0.134	0.133	0.136	0.088	0.132	
	$\sigma(k)$	0.015	0.024	0.015	0.009	0.022	0.034	0.034	0.033	0.024	0.034	

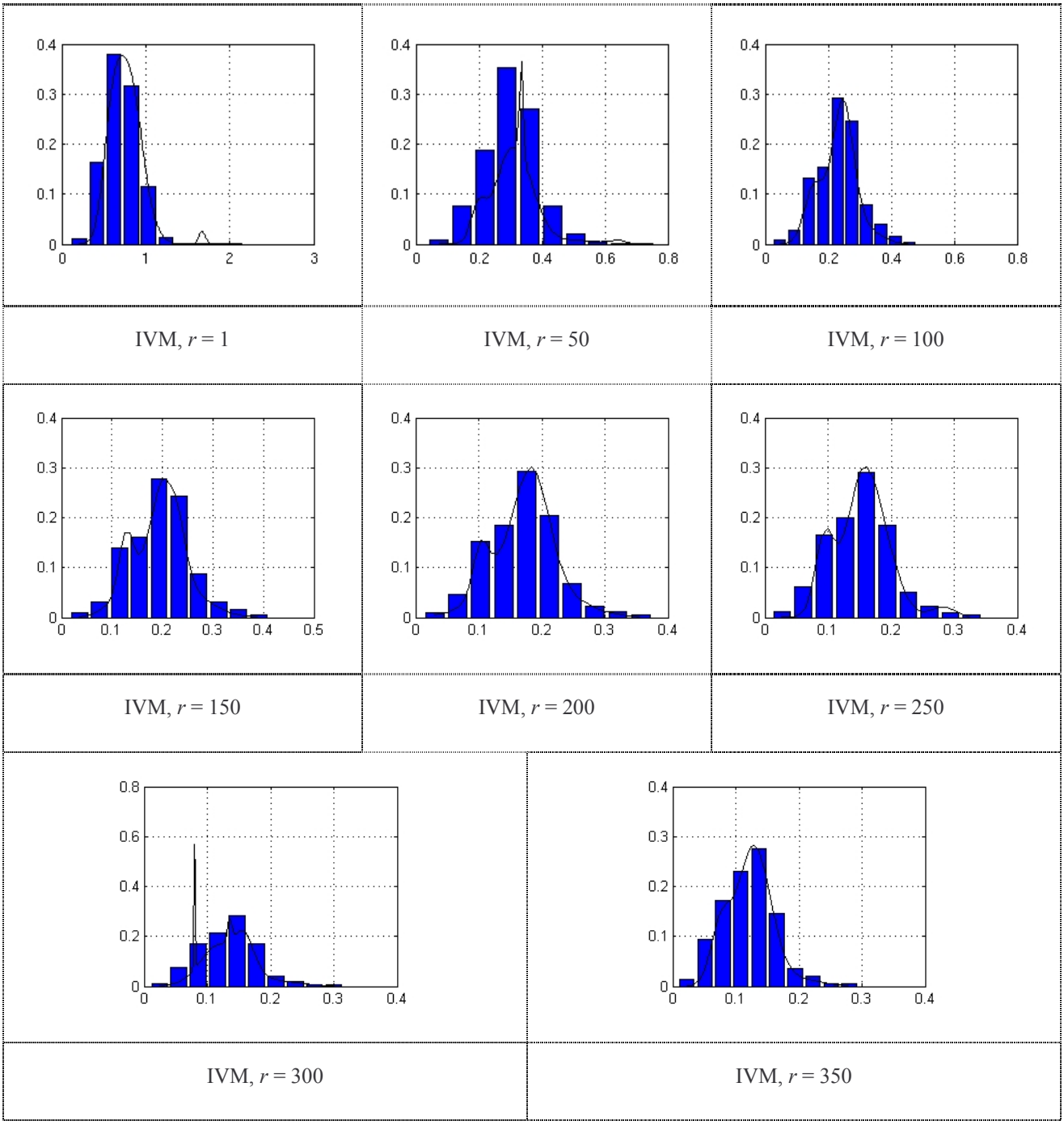


Figure A.I.3. IVM and histograms for (9,7) DWT computed for eight ranks and high quality video.

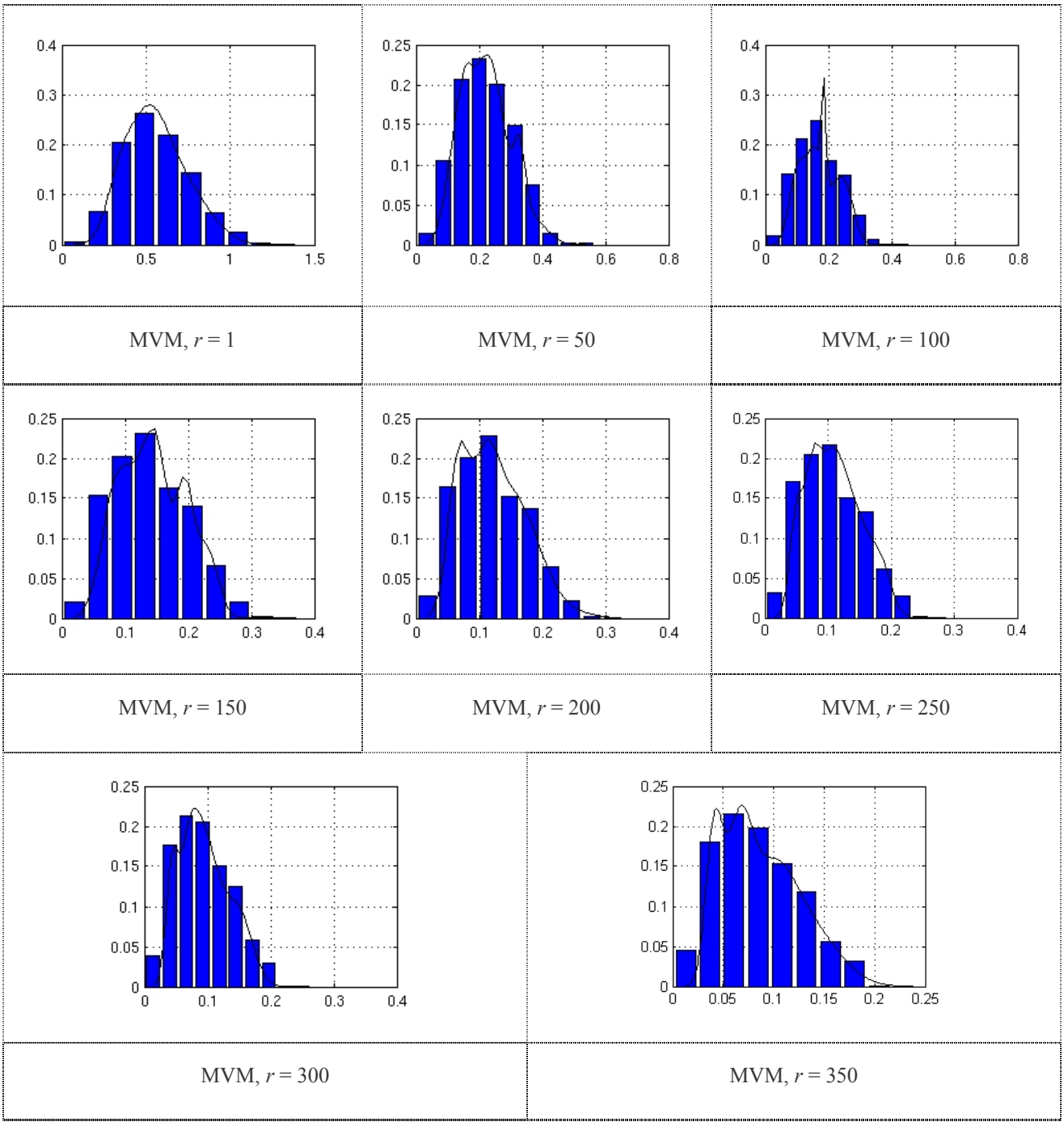


Figure A.I.4. MVM and histograms for (9,7) DWT computed for eight ranks and high quality video.

A.I.1.2. Model validation

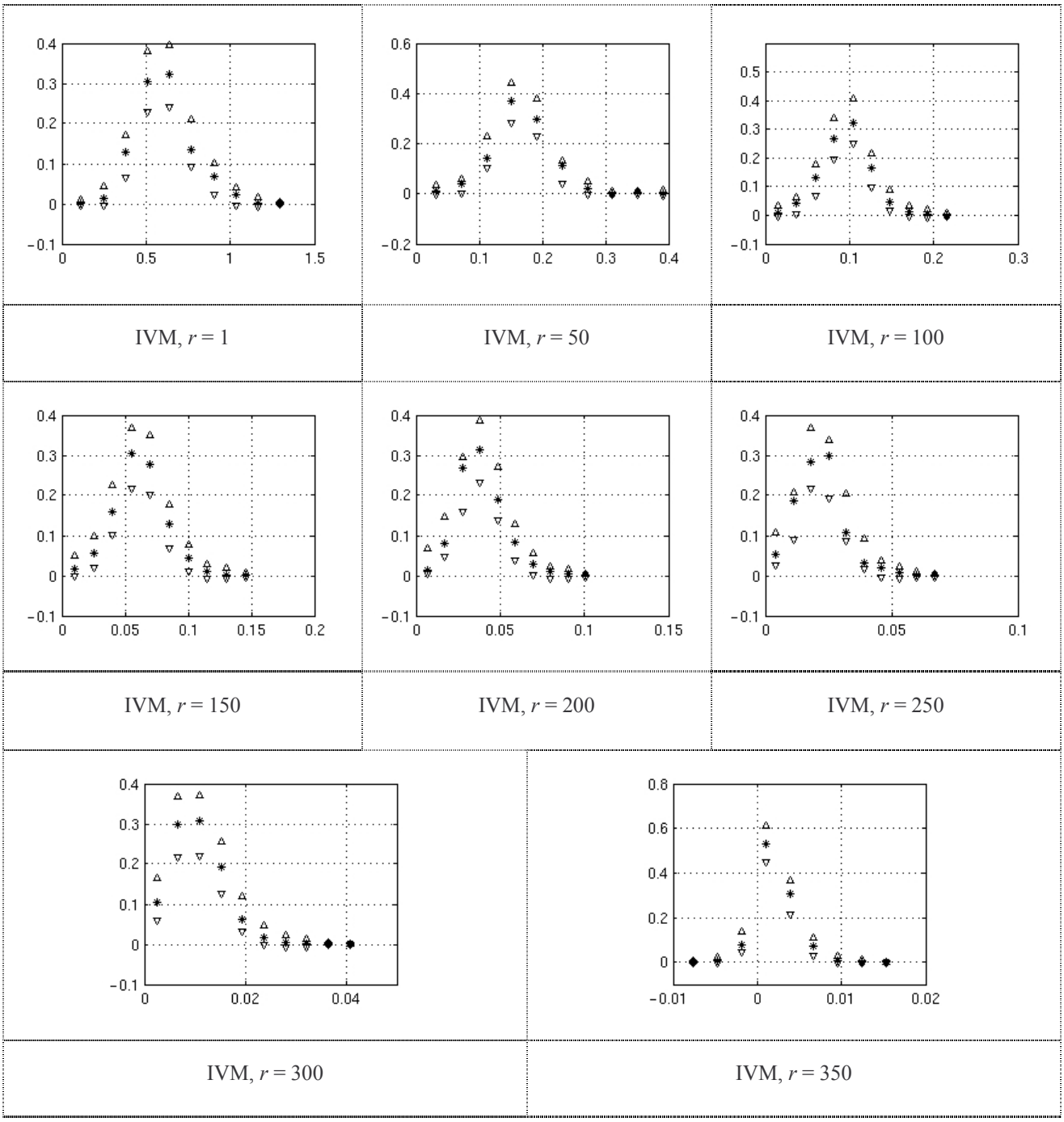


Figure A.I.5. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the IVM (represented in *). The results correspond to the (9,7) DWT, to eight ranks, and to low quality video.

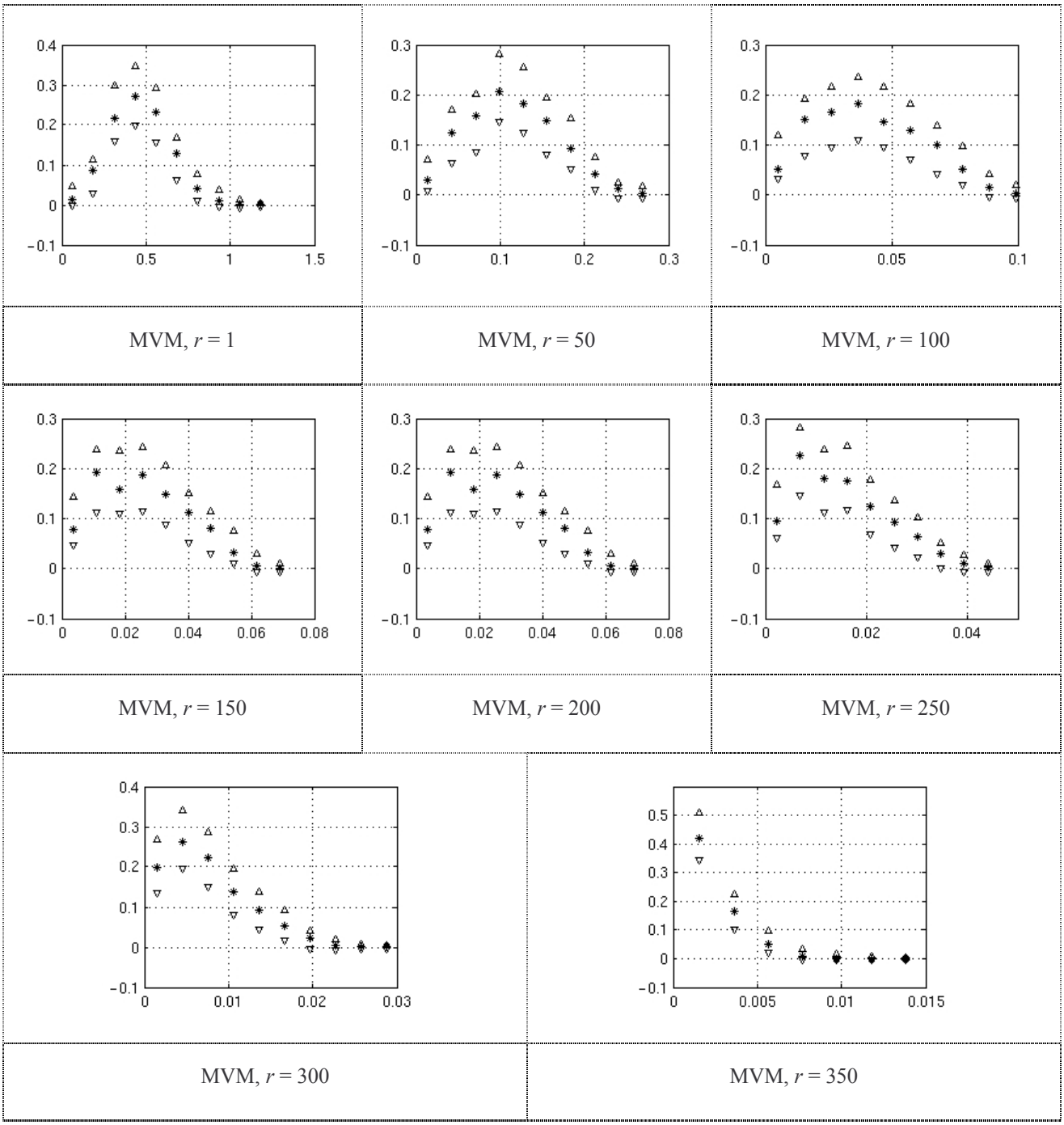
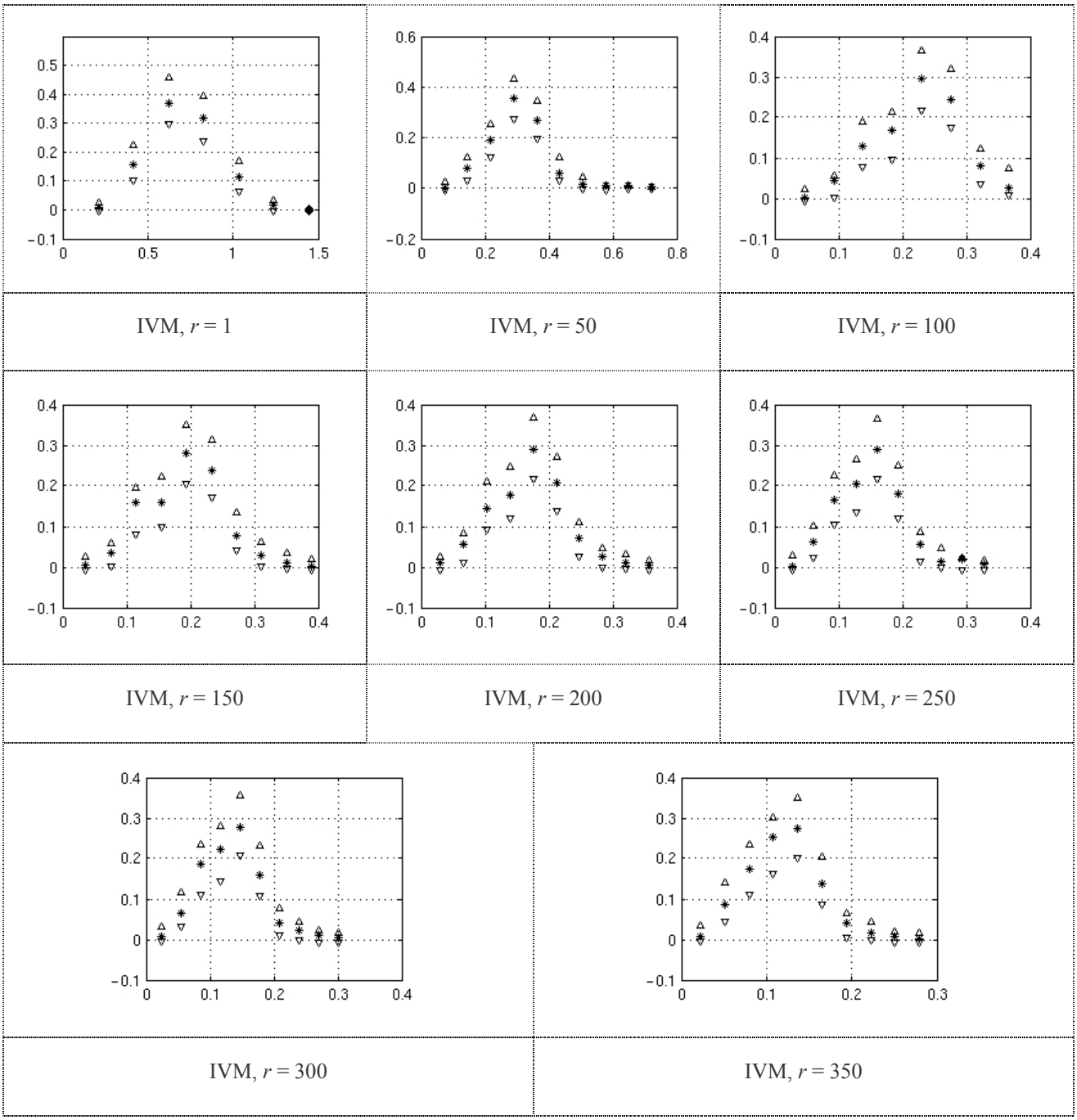


Figure A.I.6. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the MVM (represented in *). The results correspond to the (9,7) DWT, to eight ranks, and to low quality video.



IVM, $r = 1$ | IVM, $r = 50$ | IVM, $r = 100$

IVM, $r = 150$ | IVM, $r = 200$ | IVM, $r = 250$

IVM, $r = 300$ | IVM, $r = 350$

Figure A.I.7. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the IVM (represented in *). The results correspond to the (9,7) DWT, to eight ranks, and to high quality video.

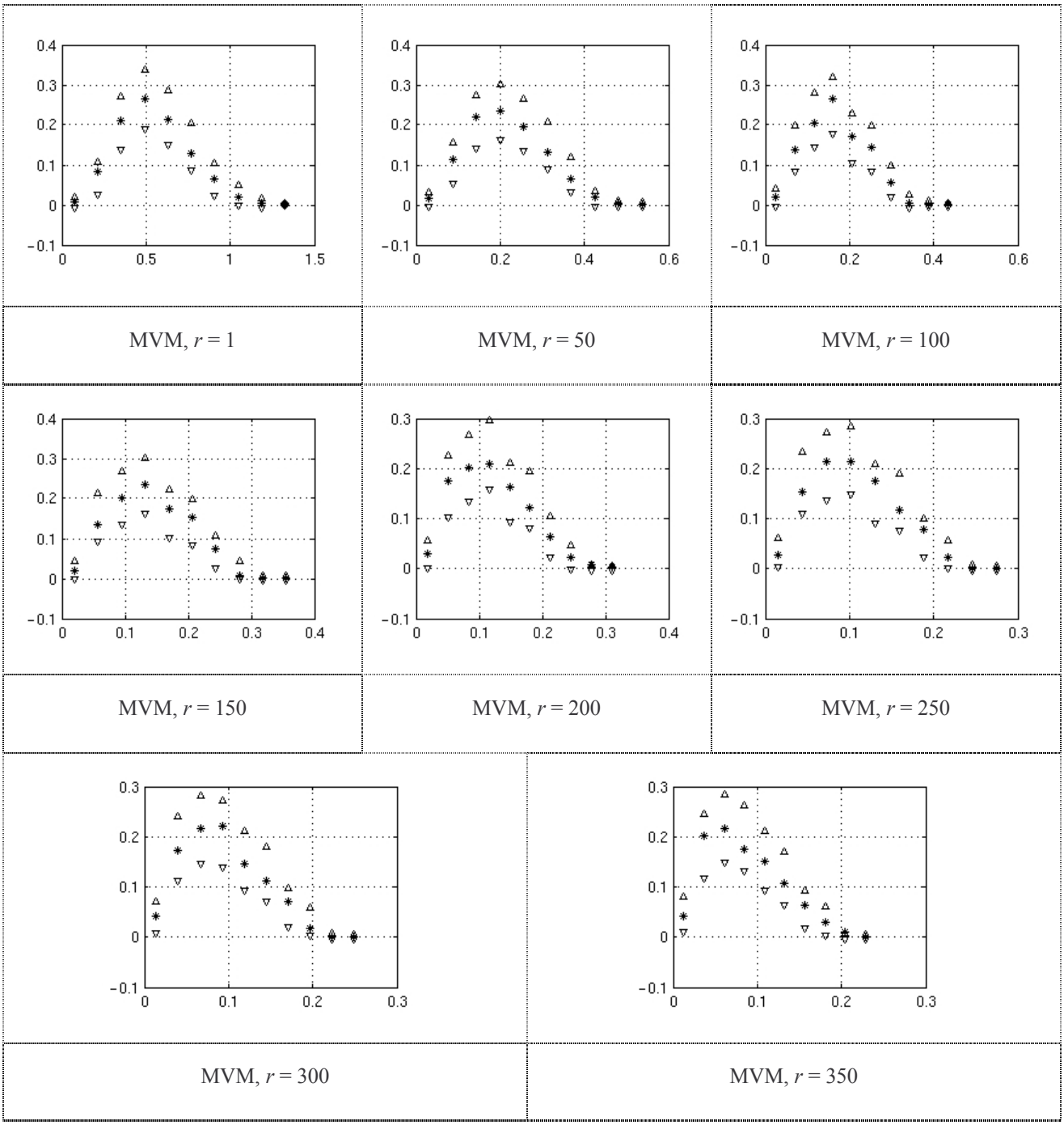


Figure A.I.8. The 95% confidence limits (represented in ∇ and Δ) include the probabilities computed from the MVM (represented in *). The results correspond to the (9,7) DWT, to eight ranks, and to high quality video.

A.I.2. The 2D-DCT coefficients modelling

A.I.2.1. Model Computation – IVM and MVM

Table A.I.5. IVM for the DCT coefficient hierarchy, low quality video. The DCT is successively applied to 4×4 blocks, 8×8 blocks and to whole frames.

Rank / DCT	Model parameters											Error
$r=1$ 4x4	$P(k)$	0.062	0.089	0.132	0.136	0.078	0.110	0.166	0.065	0.090	0.070	0.077
	$\mu(k)$	1.184	3.759	1.118	0.353	2.371	2.895	0.887	1.983	1.430	2.047	
	$\sigma(k)$	0.561	0.165	0.413	0.214	0.545	0.372	0.263	0.581	0.444	0.583	
$r=1$ 8x8	$P(k)$	0.106	0.089	0.113	0.123	0.099	0.071	0.083	0.073	0.135	0.108	0.065
	$\mu(k)$	4.828	4.152	4.998	6.722	2.926	2.479	2.642	3.758	1.019	1.865	
	$\sigma(k)$	1.189	1.158	1.189	0.959	0.955	0.877	0.872	1.201	0.634	0.699	
$r=1$ whole frames	$P(k)$	0.065	0.129	0.065	0.177	0.114	0.099	0.129	0.071	0.064	0.085	0.026
	$\mu(k)$	5.646	3.361	5.951	4.817	4.717	4.927	8.096	5.710	6.411	5.215	
	$\sigma(k)$	1.141	6.894	1.258	5.233	9.309	1.027	9.070	1.170	1.323	8.906	
$r=2$ 4x4	$P(k)$	0.039	0.048	0.119	0.054	0.456	0.103	0.022	0.074	0.074	0.010	0.017
	$\mu(k)$	-0.663	-0.077	0.176	-0.076	-0.002	0.004	-0.533	0.008	-0.178	-0.357	
	$\sigma(k)$	0.252	0.009	0.123	0.144	0.028	0.149	0.249	0.149	0.100	0.006	
$r=2$ 8x8	$P(k)$	0.068	0.288	0.075	0.074	0.061	0.082	0.117	0.080	0.079	0.075	0.023
	$\mu(k)$	-0.323	-0.006	0.442	0.070	-0.021	-0.377	-0.148	0.038	-0.197	0.341	
	$\sigma(k)$	0.371	0.052	0.238	0.306	0.185	0.328	1.023	0.170	0.369	0.266	
$r=2$ whole frames	$P(k)$	0.109	0.066	0.172	0.059	0.173	0.087	0.091	0.079	0.091	0.074	0.019
	$\mu(k)$	4.179	1.897	7.465	1.715	5.778	9.195	7.971	9.354	1.017	8.414	
	$\sigma(k)$	0.808	3.803	1.017	4.201	1.080	2.087	1.799	2.121	2.447	1.939	
$r=3$ 4x4	$P(k)$	0.174	0.015	0.091	0.174	0.079	0.129	0.058	0.068	0.125	0.087	0.010
	$\mu(k)$	-0.001	0.199	-0.004	0.001	-0.019	-0.006	-0.022	0.087	0.001	-0.034	
	$\sigma(k)$	0.015	0.006	0.078	0.002	0.081	0.078	0.216	0.226	0.155	0.033	
$r=3$ 8x8	$P(k)$	0.057	0.073	0.334	0.078	0.125	0.056	0.053	0.021	0.156	0.047	0.018
	$\mu(k)$	0.471	-0.203	0.003	-0.056	0.016	0.149	-0.465	-0.398	-0.002	0.261	
	$\sigma(k)$	0.167	0.061	0.089	0.254	0.167	0.229	0.454	0.160	0.020	0.624	
$r=3$ whole frames	$P(k)$	0.110	0.104	0.088	0.099	0.108	0.021	0.080	0.093	0.106	0.189	0.026
	$\mu(k)$	5.694	6.601	5.363	8.698	2.969	1.352	6.790	5.664	5.519	4.108	
	$\sigma(k)$	0.906	0.998	0.746	1.654	0.233	2.558	1.161	0.934	0.703	0.470	
$r=4$ 4x4	$P(k)$	0.098	0.026	0.073	0.150	0.262	0.036	0.161	0.026	0.104	0.062	0.006
	$\mu(k)$	0.044	0.242	0.015	-0.001	-0.007	-0.138	0.003	-0.099	-0.027	0.007	
	$\sigma(k)$	0.046	0.056	0.017	0.001	0.013	0.075	0.005	0.086	0.039	0.052	
$r=4$ 8x8	$P(k)$	0.067	0.132	0.065	0.073	0.339	0.065	0.088	0.053	0.044	0.073	0.013
	$\mu(k)$	-0.056	0.071	0.424	0.012	0.003	-0.202	-0.001	-0.181	-0.074	0.009	
	$\sigma(k)$	0.108	0.085	0.162	0.113	0.027	0.045	0.002	0.252	0.275	0.113	
$r=4$ whole frames	$P(k)$	0.150	0.132	0.085	0.060	0.112	0.119	0.060	0.067	0.098	0.115	0.027
	$\mu(k)$	4.437	5.586	4.768	4.299	5.285	6.320	3.997	3.025	2.829	4.232	
	$\sigma(k)$	0.604	0.907	0.932	0.716	0.992	1.222	0.599	0.494	0.406	0.687	
$r=5$ 4x4	$P(k)$	0.054	0.241	0.149	0.098	0.061	0.071	0.097	0.014	0.086	0.128	0.017
	$\mu(k)$	0.015	0.003	-0.051	-0.099	0.538	-0.103	-0.105	-0.734	0.071	0.067	
	$\sigma(k)$	0.070	0.016	0.040	0.168	0.271	0.167	0.167	0.021	0.072	0.135	
$r=5$ 8x8	$P(k)$	0.076	0.057	0.120	0.050	0.070	0.041	0.391	0.040	0.116	0.038	0.009
	$\mu(k)$	0.124	-0.003	0.057	-0.232	-0.025	-0.026	-0.001	-0.023	-0.064	0.272	
	$\sigma(k)$	0.096	0.073	0.034	0.100	0.070	0.143	0.008	0.144	0.040	0.200	
$r=5$ whole frames	$P(k)$	0.108	0.062	0.107	0.110	0.109	0.145	0.065	0.079	0.106	0.108	0.027
	$\mu(k)$	4.112	5.720	3.832	4.174	3.996	2.503	5.561	3.819	4.379	3.712	
	$\sigma(k)$	0.760	1.172	0.561	0.790	0.687	0.339	1.109	0.553	0.863	0.529	

Table A.I.6. IVM for the DCT (applied on the whole frames) coefficient hierarchy, low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.065	0.129	0.065	0.177	0.114	0.099	0.129	0.071	0.064	0.085	0.026
	$\mu(k)$	5.646	3.361	5.951	4.817	4.717	4.927	8.096	5.710	6.411	5.215	
	$\sigma(k)$	1.141	6.894	1.258	5.233	9.309	1.027	9.070	1.170	1.323	8.906	
25	$P(k)$	0.127	0.105	0.084	0.017	0.039	0.165	0.11	0.122	0.119	0.109	0.032
	$\mu(k)$	1.108	1.434	1.597	1.152	1.209	1.597	1.386	1.441	1.856	1.345	
	$\sigma(k)$	1.003	0.172	0.233	0.025	0.396	0.126	0.157	0.173	0.209	0.142	
50	$P(k)$	0.123	0.052	0.158	0.203	0.089	0.103	0.132	0.068	0.007	0.065	0.031
	$\mu(k)$	0.727	1.224	1.089	0.954	0.953	0.989	0.955	0.832	0.553	0.709	
	$\sigma(k)$	0.064	0.256	0.106	0.067	0.122	0.126	0.122	0.079	0.001	0.210	
75	$P(k)$	0.146	0.072	0.058	0.067	0.193	0.057	0.132	0.005	0.104	0.164	0.029
	$\mu(k)$	0.757	0.749	0.936	0.419	0.743	0.573	0.675	0.391	0.569	0.749	
	$\sigma(k)$	0.094	0.123	0.097	0.156	0.072	0.196	0.082	0.009	0.057	0.075	
100	$P(k)$	0.098	0.088	0.095	0.103	0.055	0.111	0.077	0.073	0.209	0.091	0.030
	$\mu(k)$	0.670	0.567	0.568	0.603	0.546	0.532	0.450	0.567	0.603	0.680	
	$\sigma(k)$	0.103	0.120	0.118	0.108	0.124	0.079	0.114	0.121	0.033	0.064	
125	$P(k)$	0.006	0.078	0.172	0.187	0.079	0.021	0.081	0.193	0.089	0.094	0.034
	$\mu(k)$	0.472	0.569	0.376	0.536	0.485	0.131	0.528	0.502	0.600	0.521	
	$\sigma(k)$	0.085	0.103	0.046	0.052	0.069	0.040	0.079	0.035	0.100	0.062	
150	$P(k)$	0.088	0.136	0.088	0.082	0.075	0.058	0.113	0.173	0.080	0.109	0.034
	$\mu(k)$	0.374	0.461	0.370	0.455	0.399	0.341	0.572	0.467	0.462	0.431	
	$\sigma(k)$	0.097	0.037	0.097	0.068	0.051	0.035	0.049	0.031	0.068	0.054	
175	$P(k)$	0.129	0.109	0.282	0.104	0.124	0.027	0.057	0.060	0.031	0.077	0.035
	$\mu(k)$	0.412	0.394	0.405	0.410	0.288	0.336	0.446	0.449	0.161	0.500	
	$\sigma(k)$	0.0497	0.055	0.036	0.054	0.033	0.113	0.082	0.081	0.076	0.072	
200	$P(k)$	0.093	0.171	0.121	0.119	0.126	0.021	0.066	0.102	0.160	0.034	0.036
	$\mu(k)$	0.382	0.369	0.386	0.418	0.366	0.089	0.376	0.386	0.271	0.353	
	$\sigma(k)$	0.069	0.031	0.034	0.062	0.048	0.022	0.069	0.069	0.040	0.068	
225	$P(k)$	0.137	0.133	0.238	0.065	0.077	0.029	0.035	0.028	0.126	0.132	0.034
	$\mu(k)$	0.240	0.316	0.349	0.378	0.444	0.092	0.326	0.322	0.297	0.385	
	$\sigma(k)$	0.022	0.035	0.019	0.047	0.031	0.026	0.039	0.001	0.028	0.016	
250	$P(k)$	0.079	0.169	0.021	0.062	0.102	0.063	0.333	0.055	0.001	0.114	0.033
	$\mu(k)$	0.309	0.227	0.101	0.307	0.326	0.312	0.323	0.319	0.274	0.411	
	$\sigma(k)$	0.042	0.028	0.017	0.044	0.048	0.045	0.028	0.048	0.040	0.030	
275	$P(k)$	0.100	0.132	0.080	0.061	0.077	0.056	0.087	0.016	0.146	0.245	0.037
	$\mu(k)$	0.366	0.286	0.279	0.212	0.332	0.251	0.197	0.244	0.296	0.302	
	$\sigma(k)$	0.053	0.035	0.028	0.075	0.061	0.077	0.011	0.002	0.034	0.033	
300	$P(k)$	0.121	0.122	0.106	0.059	0.038	0.035	0.110	0.150	0.222	0.037	0.033
	$\mu(k)$	0.296	0.259	0.315	0.308	0.286	0.082	0.265	0.188	0.274	0.288	
	$\sigma(k)$	0.016	0.035	0.060	0.061	0.055	0.030	0.038	0.021	0.034	0.055	
325	$P(k)$	0.042	0.098	0.052	0.091	0.165	0.057	0.109	0.095	0.202	0.088	0.034
	$\mu(k)$	0.179	0.265	0.189	0.229	0.214	0.276	0.299	0.270	0.274	0.295	
	$\sigma(k)$	0.069	0.053	0.071	0.036	0.034	0.052	0.047	0.053	0.014	0.048	
350	$P(k)$	0.091	0.095	0.029	0.049	0.039	0.141	0.147	0.065	0.122	0.221	0.030
	$\mu(k)$	0.296	0.246	0.161	0.119	0.195	0.229	0.209	0.291	0.227	0.265	
	$\sigma(k)$	0.051	0.041	0.003	0.060	0.078	0.038	0.033	0.052	0.037	0.018	

Table A.I.7. MVM for the DCT coefficient hierarchy, low quality video. The DCT is successively applied to 4×4 blocks, 8×8 blocks and to whole frames.

Rank / DCT	Model parameters											Error
$r=1$ 4x4	$P(k)$	0.042	0.381	0.116	0.102	0.123	0.014	0.194	0.007	0.007	0.014	0.005
	$\mu(k)$	0.330	0.020	0.070	0.080	0.003	1.054	0.067	0.776	0.654	2.632	
	$\sigma(k)$	0.083	0.023	0.113	0.113	0.002	0.108	0.046	0.001	0.001	0.389	
$r=1$ 8x8	$P(k)$	0.165	0.195	0.170	0.044	0.198	0.065	0.007	0.056	0.084	0.016	0.007
	$\mu(k)$	0.188	0.001	0.088	0.522	0.158	0.012	1.199	0.663	0.077	3.701	
	$\sigma(k)$	0.153	0.003	0.073	0.223	0.149	0.004	0.001	0.176	0.019	2.081	
$r=1$ whole frames	$P(k)$	0.106	0.096	0.112	0.033	0.177	0.081	0.119	0.092	0.031	0.152	0.030
	$\mu(k)$	4.364	4.761	4.490	2.333	1.516	4.704	4.437	5.011	2.727	3.475	
	$\sigma(k)$	8.410	9.726	8.300	1.439	2.999	9.528	8.387	1.021	1.546	5.113	
$r=2$ 4x4	$P(k)$	0.085	0.264	0.021	0.252	0.103	0.188	0.034	0.007	0.007	0.038	0.008
	$\mu(k)$	0.122	0.010	2.847	0.027	0.080	0.002	-0.008	-0.238	1.438	0.001	
	$\sigma(k)$	0.088	0.008	0.461	0.027	0.036	0.002	0.065	0.001	0.001	0.067	
$r=2$ 8x8	$P(k)$	0.074	0.138	0.221	0.087	0.069	0.091	0.230	0.007	0.075	0.007	0.005
	$\mu(k)$	0.180	0.105	0.024	0.258	0.210	-0.004	0.001	-0.016	0.131	-0.155	
	$\sigma(k)$	0.094	0.040	0.017	0.321	0.083	0.072	0.002	0.001	0.099	0.001	
$r=2$ whole frames	$P(k)$	0.075	0.094	0.082	0.115	0.062	0.114	0.098	0.183	0.059	0.118	0.028
	$\mu(k)$	5.774	8.184	4.194	7.246	1.326	7.355	3.258	6.252	1.236	1.006	
	$\sigma(k)$	1.593	2.444	2.029	2.612	3.554	2.599	0.702	1.334	3.803	1.782	
$r=3$ 4x4	$P(k)$	0.007	0.124	0.100	0.014	0.290	0.045	0.075	0.061	0.044	0.239	0.003
	$\mu(k)$	0.171	0.022	0.033	-0.194	0.008	1.687	0.021	0.062	0.027	0.003	
	$\sigma(k)$	0.001	0.037	0.038	0.068	0.011	1.160	0.037	0.036	0.038	0.003	
$r=3$ 8x8	$P(k)$	0.191	0.083	0.024	0.035	0.064	0.131	0.178	0.021	0.016	0.275	0.006
	$\mu(k)$	0.288	0.140	0.335	0.207	0.145	0.083	0.014	-0.193	0.887	0.002	
	$\sigma(k)$	0.042	0.061	0.056	0.086	0.059	0.052	0.008	0.032	1.929	0.003	
$r=3$ whole frames	$P(k)$	0.096	0.077	0.078	0.074	0.095	0.112	0.144	0.145	0.069	0.110	0.031
	$\mu(k)$	5.900	6.598	5.979	7.133	5.576	3.900	3.819	2.407	8.889	5.137	
	$\sigma(k)$	1.285	1.685	1.300	1.828	1.331	0.088	0.835	0.677	1.483	1.319	
$r=4$ 4x4	$P(k)$	0.034	0.398	0.138	0.072	0.117	0.029	0.007	0.128	0.064	0.014	0.007
	$\mu(k)$	0.082	0.002	0.009	0.073	0.012	2.087	0.297	0.011	-0.031	-0.197	
	$\sigma(k)$	0.046	0.002	0.021	0.019	0.003	0.375	0.001	0.021	0.034	0.003	
$r=4$ 8x8	$P(k)$	0.094	0.285	0.051	0.091	0.164	0.127	0.159	0.007	0.014	0.007	0.007
	$\mu(k)$	0.100	0.004	0.068	0.192	0.046	0.062	0.007	2.893	-0.149	3.207	
	$\sigma(k)$	0.045	0.005	0.026	0.063	0.030	0.028	0.018	2.214	0.008	2.221	
$r=4$ whole frames	$P(k)$	0.077	0.059	0.133	0.068	0.071	0.182	0.195	0.072	0.067	0.075	0.040
	$\mu(k)$	4.352	4.748	3.808	5.011	5.265	3.038	2.446	5.319	5.379	4.508	
	$\sigma(k)$	1.352	1.427	0.911	1.407	1.362	0.983	0.875	1.351	1.338	1.402	
$r=5$ 4x4	$P(k)$	0.016	0.009	0.015	0.027	0291	0.009	0.040	0.027	0.533	0.033	0.011
	$\mu(k)$	0.014	0.063	0.005	0.005	0.008	0.073	-0.078	0.016	0.001	-0.129	
	$\sigma(k)$	0.046	0.064	0.046	0.046	0.009	0.063	0.503	0.046	0.001	0.017	
$r=5$ 8x8	$P(k)$	0.018	0.012	0.530	0.052	0.295	0.015	0.007	0.028	0.033	0.009	0.006
	$\mu(k)$	0.111	0.111	0.011	0.062	0.003	0.829	-0.293	-0.098	0.168	0.111	
	$\sigma(k)$	0.022	0.018	0.014	0.011	0.003	1.693	0.001	0.025	0.025	0.018	
$r=5$ whole frames	$P(k)$	0.102	0.117	0.070	0.106	0.095	0.107	0.081	0.094	0.093	0.135	0.033
	$\mu(k)$	2.968	4.455	2.288	2.789	2.798	3.442	1.716	3.014	2.948	4.570	
	$\sigma(k)$	0.854	0.664	0.802	0.917	0.919	0.827	0.712	0.928	0.931	0.643	

Table A.I.8. MVM for the DCT (applied on the whole frames) coefficient hierarchy, low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.106	0.096	0.112	0.033	0.177	0.081	0.119	0.092	0.031	0.152	0.030
	$\mu(k)$	4.364	4.761	4.490	2.333	1.516	4.704	4.437	5.011	2.727	3.475	
	$\sigma(k)$	8.410	9.726	8.300	1.439	2.999	9.528	8.387	1.021	1.546	5.113	
25	$P(k)$	0.064	0.124	0.094	0.112	0.093	0.084	0.055	0.109	0.152	0.115	0.038
	$\mu(k)$	1.094	1.650	1.171	1.319	1.264	1.109	1.121	0.996	0.651	1.009	
	$\sigma(k)$	0.299	1.876	0.310	0.295	0.303	0.303	0.306	0.253	0.212	0.190	
50	$P(k)$	0.102	0.024	0.139	0.119	0.111	0.149	0.007	0.122	0.125	0.102	0.033
	$\mu(k)$	0.648	0.919	0.644	0.858	0.653	0.543	0.271	0.658	0.984	0.686	
	$\sigma(k)$	0.218	0.138	0.217	0.241	0.218	0.143	0.005	0.219	0.104	0.227	
75	$P(k)$	0.056	0.095	0.068	0.114	0.114	0.118	0.117	0.123	0.069	0.124	0.041
	$\mu(k)$	0.492	0.550	0.426	0.410	0.802	0.732	0.398	0.453	0.695	0.430	
	$\sigma(k)$	0.158	0.144	0.156	0.154	0.103	0.106	0.151	0.157	0.106	0.114	
100	$P(k)$	0.007	0.040	0.164	0.036	0.043	0.114	0.083	0.113	0.148	0.252	0.044
	$\mu(k)$	1.025	0.616	0.346	0.610	0.508	0.445	0.506	0.675	0.425	0.298	
	$\sigma(k)$	0.001	0.021	0.106	0.008	0.099	0.109	0.099	0.064	0.076	0.092	
125	$P(k)$	0.064	0.103	0.105	0.146	0.073	0.158	0.046	0.131	0.093	0.081	0.034
	$\mu(k)$	0.464	0.336	0.529	0.293	0.500	0.227	0.585	0.344	0.398	0.421	
	$\sigma(k)$	0.127	0.091	0.037	0.091	0.117	0.076	0.127	0.063	0.123	0.127	
150	$P(k)$	0.013	0.089	0.075	0.023	0.021	0.091	0.068	0.069	0.251	0.112	0.034
	$\mu(k)$	0.370	0.297	0.286	0.344	0.470	0.340	0.511	0.320	0.218	0.286	
	$\sigma(k)$	0.002	0.085	0.074	0.001	0.048	0.101	0.129	0.095	0.076	0.065	
175	$P(k)$	0.131	0.111	0.068	0.145	0.116	0.136	0.048	0.111	0.075	0.059	0.037
	$\mu(k)$	0.259	0.267	0.452	0.218	0.412	0.175	0.319	0.361	0.392	0.281	
	$\sigma(k)$	0.062	0.061	0.117	0.071	0.025	0.060	0.096	0.097	0.105	0.082	
200	$P(k)$	0.097	0.168	0.062	0.080	0.108	0.053	0.108	0.165	0.081	0.078	0.034
	$\mu(k)$	0.222	0.385	0.456	0.126	0.218	0.296	0.276	0.212	0.295	0.253	
	$\sigma(k)$	0.060	0.033	0.105	0.042	0.060	0.080	0.076	0.061	0.079	0.068	
225	$P(k)$	0.025	0.135	0.051	0.060	0.150	0.077	0.072	0.089	0.188	0.154	0.035
	$\mu(k)$	0.181	0.230	0.261	0.408	0.352	0.358	0.239	0.229	0.131	0.221	
	$\sigma(k)$	0.001	0.052	0.070	0.107	0.033	0.047	0.061	0.051	0.043	0.050	
250	$P(k)$	0.082	0.263	0.020	0.106	0.067	0.029	0.165	0.105	0.046	0.117	0.034
	$\mu(k)$	0.226	0.327	0.480	0.193	0.206	0.257	0.130	0.237	0.236	0.153	
	$\sigma(k)$	0.063	0.040	0.063	0.026	0.059	0.060	0.043	0.062	0.013	0.046	
275	$P(k)$	0.079	0.093	0.210	0.126	0.134	0.104	0.051	0.065	0.022	0.117	0.033
	$\mu(k)$	0.192	0.203	0.315	0.115	0.165	0.224	0.152	0.206	0.442	0.215	
	$\sigma(k)$	0.054	0.057	0.035	0.040	0.048	0.058	0.047	0.057	0.065	0.058	
300	$P(k)$	0.107	0.196	0.217	0.026	0.072	0.138	0.007	0.071	0.055	0.111	0.034
	$\mu(k)$	0.158	0.296	0.126	0.393	0.220	0.153	0.203	0.220	0.218	0.209	
	$\sigma(k)$	0.047	0.033	0.044	0.070	0.051	0.047	0.001	0.051	0.051	0.053	
325	$P(k)$	0.122	0.060	0.014	0.080	0.036	0.194	0.117	0.077	0.116	0.184	0.035
	$\mu(k)$	0.200	0.194	0.429	0.160	0.204	0.283	0.172	0.219	0.133	0.112	
	$\sigma(k)$	0.048	0.049	0.019	0.046	0.048	0.030	0.049	0.043	0.042	0.039	
350	$P(k)$	0.145	0.131	0.087	0.131	0.054	0.067	0.106	0.164	0.014	0.101	0.034
	$\mu(k)$	0.149	0.116	0.187	0.092	0.187	0.183	0.184	0.274	0.408	0.219	
	$\sigma(k)$	0.030	0.037	0.047	0.032	0.047	0.048	0.048	0.026	0.018	0.037	

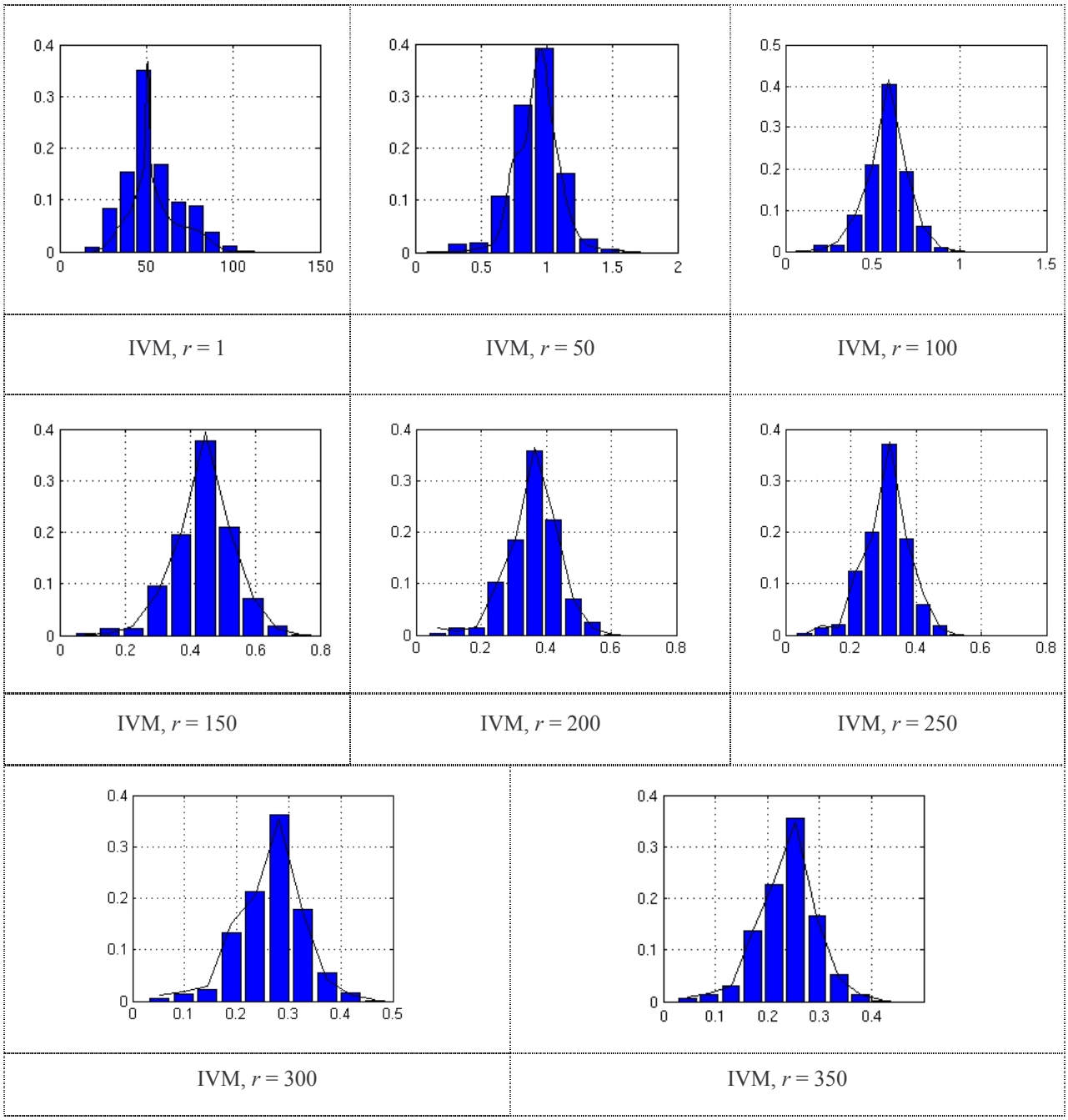


Figure A.I.9. IVM and histogram computed for eight ranks in the DCT (applied on whole frames), low quality video.

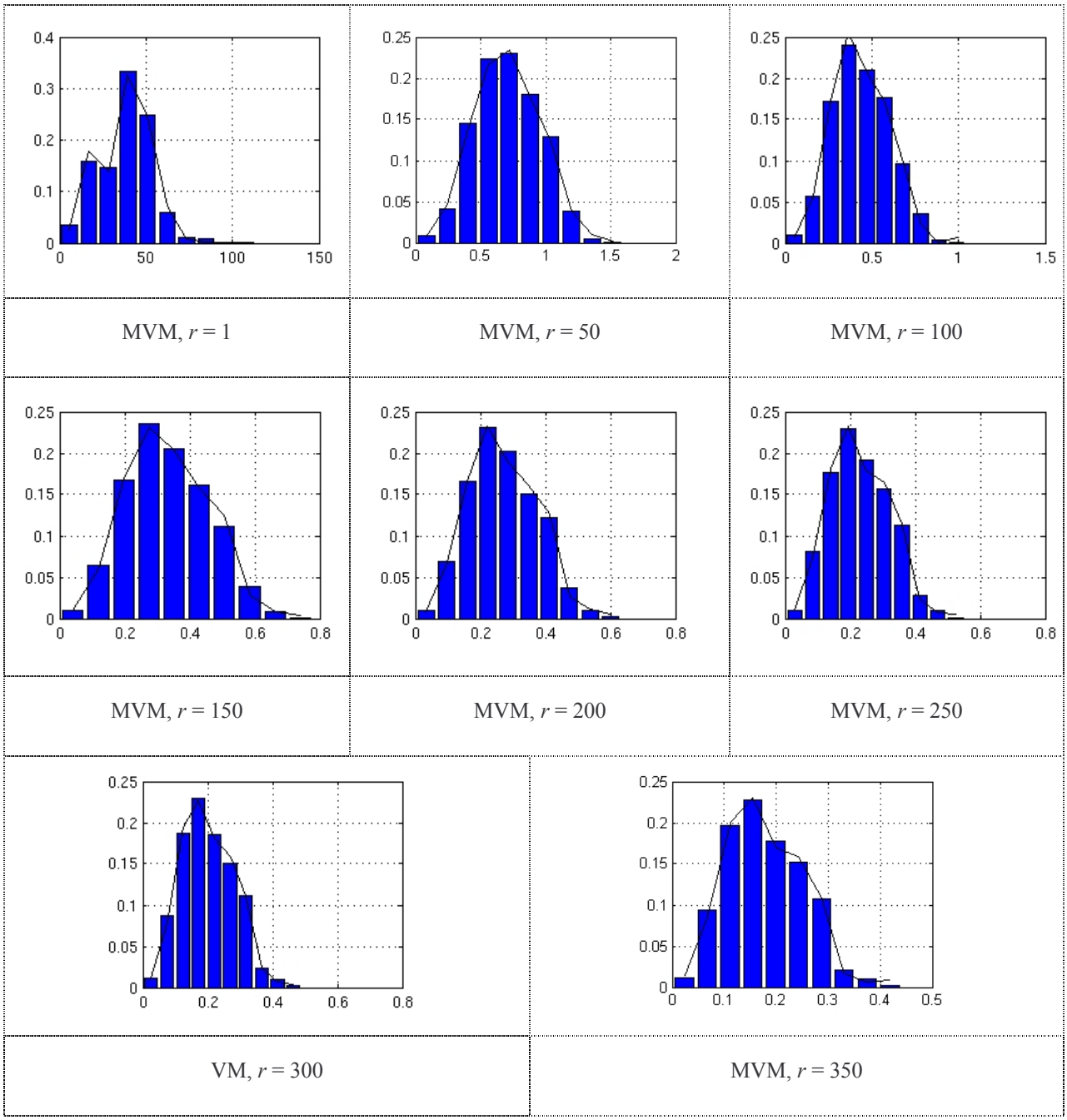


Figure A.I.10. MVM and histogram computed for eight ranks in the DCT (applied on whole frames), low quality video.

Table A.I.9. IVM for the DCT coefficient hierarchy, high quality video. The DCT is successively applied to 4×4 blocks, 8×8 blocks and to whole frames.

Rank / DCT	Model parameters										Error
$r=1$ 4x4	$P(k)$	0.169	0.030	0.021	0.054	0.379	0.020	0.146	0.069	0.036	0.075	0.018
	$\mu(k)$	0.038	0.301	0.241	0.188	0.012	0.417	0.072	0.122	0.712	0.115	
	$\sigma(k)$	0.013	0.026	0.062	0.019	0.007	0.036	0.015	0.037	0.063	0.030	
$r=1$ 8x8	$P(k)$	0.365	0.086	0.059	0.063	0.058	0.014	0.160	0.077	0.058	0.057	0.022
	$\mu(k)$	0.058	0.228	0.436	0.843	0.452	1.732	0.158	0.311	0.213	0.452	
	$\sigma(k)$	0.029	0.057	0.120	0.217	0.117	0.185	0.060	0.084	0.065	0.117	
$r=1$ whole frames	$P(k)$	0.111	0.047	0.093	0.073	0.107	0.111	0.081	0.059	0.112	0.207	0.031
	$\mu(k)$	9.197	127.23	154.61	145.82	158.72	169.18	165.88	143.27	169.84	243.29	
	$\sigma(k)$	11.277	21.180	25.140	20.254	26.491	29.053	38.436	19.121	29.188	36.449	
$r=2$ 4x4	$P(k)$	0.068	0.036	0.007	0.239	0.126	0.043	0.191	0.050	0.206	0.033	0.023
	$\mu(k)$	0.111	0.104	0.427	0.007	0.041	0.188	0.002	0.076	0.024	0.054	
	$\sigma(k)$	0.031	0.029	0.001	0.004	0.009	0.052	0.002	0.007	0.010	0.020	
$r=2$ 8x8	$P(k)$	0.048	0.058	0.137	0.054	0.046	0.132	0.152	0.255	0.079	0.038	0.023
	$\mu(k)$	0.312	0.220	0.008	0.319	0.589	0.103	0.151	0.039	0.101	0.304	
	$\sigma(k)$	0.079	0.021	0.006	0.082	0.162	0.031	0.015	0.016	0.031	0.071	
$r=2$ whole frames	$P(k)$	0.141	0.095	0.081	0.062	0.111	0.139	0.054	0.148	0.046	0.123	0.020
	$\mu(k)$	26.565	12.129	66.913	15.858	28.111	18.811	38.464	27.155	41.212	23.048	
	$\sigma(k)$	5.286	1.996	1.644	2.995	5.379	2.442	6.747	5.111	5.040	5.040	
$r=3$ 4x4	$P(k)$	0.083	0.081	0.022	0.128	0.021	0.129	0.007	0.404	0.040	0.085	0.021
	$\mu(k)$	0.038	0.025	0.093	0.012	0.122	0.013	0.055	0.003	0.127	0.050	
	$\sigma(k)$	0.009	0.006	0.024	0.005	0.034	0.005	0.001	0.002	0.033	0.013	
$r=3$ 8x8	$P(k)$	0.031	0.119	0.181	0.099	0.073	0.182	0.097	0.029	0.106	0.081	0.022
	$\mu(k)$	0.276	0.021	0.044	0.094	0.147	0.005	0.085	0.303	0.191	0.161	
	$\sigma(k)$	0.089	0.004	0.016	0.027	0.044	0.004	0.027	0.088	0.053	0.048	
$r=3$ whole frames	$P(k)$	0.089	0.081	0.093	0.118	0.094	0.105	0.135	0.096	0.083	0.105	0.026
	$\mu(k)$	9.262	14.392	28.328	17.920	24.852	13.913	19.384	19.927	13.630	16.726	
	$\sigma(k)$	1.193	2.358	5.642	3.201	4.348	2.272	2.475	4.085	2.252	3.170	
$r=4$ 4x4	$P(k)$	0.036	0.219	0.025	0.020	0.392	0.083	0.024	0.125	0.060	0.015	0.020
	$\mu(k)$	0.043	0.007	0.041	0.072	0.002	0.022	0.069	0.011	0.024	0.107	
	$\sigma(k)$	0.012	0.003	0.001	0.014	0.002	0.005	0.015	0.003	0.005	0.005	
$r=4$ 8x8	$P(k)$	0.054	0.037	0.060	0.065	0.342	0.010	0.197	0.094	0.070	0.070	0.024
	$\mu(k)$	0.121	0.155	0.231	0.062	0.028	0.119	0.005	0.083	0.086	0.183	
	$\sigma(k)$	0.029	0.003	0.063	0.021	0.013	0.001	0.004	0.024	0.014	0.040	
$r=4$ whole frames	$P(k)$	0.102	0.099	0.084	0.132	0.007	0.128	0.098	0.111	0.136	0.101	0.031
	$\mu(k)$	18.566	17.552	9.542	12.635	33.932	11.760	16.691	13.788	14.908	16.390	
	$\sigma(k)$	3.937	4.250	1.674	2.970	0.001	3.271	4.450	2.504	1.870	4.498	
$r=5$ 4x4	$P(k)$	0.011	0.132	0.387	0.245	0.002	0.026	0.007	0.027	0.129	0.034	0.010
	$\mu(k)$	0.022	0.009	0.001	0.003	0.013	0.020	-0.006	0.018	0.002	0.017	
	$\sigma(k)$	0.002	0.002	0.001	0.001	0.006	0.021	0.001	0.021	0.002	0.002	
$r=5$ 8x8	$P(k)$	0.049	0.050	0.053	0.029	0.067	0.024	0.212	0.065	0.209	0.239	0.021
	$\mu(k)$	0.039	0.061	0.047	0.097	0.057	0.098	0.007	0.121	0.002	0.018	
	$\sigma(k)$	0.011	0.018	0.015	0.030	0.017	0.030	0.003	0.028	0.002	0.006	
$r=5$ whole frames	$P(k)$	0.104	0.106	0.089	0.079	0.096	0.057	0.115	0.088	0.092	0.173	0.025
	$\mu(k)$	11.747	12.399	13.864	14.687	13.404	18.829	11.992	14.034	13.083	8.248	
	$\sigma(k)$	1.766	2.301	2.559	2.420	2.486	2.622	1.545	2.559	2.551	1.280	

Table A.I.10. IVM for the DCT (applied on the whole frames) coefficient hierarchy, high quality video.

Rank	Model parameters											Error
1	$P(k)$	0.111	0.047	0.093	0.073	0.107	0.111	0.081	0.059	0.112	0.207	0.031
	$\mu(k)$	9.197	127.23	154.61	145.82	158.72	169.18	165.88	143.27	169.84	243.29	
	$\sigma(k)$	11.277	21.180	25.140	20.254	26.491	29.053	38.436	19.121	29.188	36.449	
25	$P(k)$	0.103	0.136	0.108	0.104	0.063	0.109	0.134	0.072	0.087	0.085	0.036
	$\mu(k)$	4.971	3.746	5.029	4.737	2.998	4.546	4.056	5.939	5.482	5.264	
	$\sigma(k)$	0.439	0.452	0.444	0.713	0.922	0.667	0.573	0.909	0.902	0.911	
50	$P(k)$	0.104	0.101	0.108	0.088	0.091	0.124	0.114	0.119	0.058	0.092	0.028
	$\mu(k)$	2.568	2.954	3.252	3.192	3.217	3.139	3.474	2.436	1.967	3.229	
	$\sigma(k)$	0.330	0.420	0.284	0.492	0.494	0.225	0.475	0.282	0.631	0.494	
75	$P(k)$	0.063	0.103	0.233	0.128	0.105	0.051	0.094	0.111	0.063	0.049	0.037
	$\mu(k)$	2.260	2.604	2.411	1.960	2.519	2.279	2.994	2.286	1.589	1.529	
	$\sigma(k)$	0.322	0.335	0.157	0.203	0.346	0.355	0.239	0.306	0.117	0.526	
100	$P(k)$	0.075	0.093	0.118	0.221	0.065	0.090	0.113	0.078	0.067	0.080	0.029
	$\mu(k)$	2.420	2.112	1.657	2.006	2.327	1.864	1.599	1.962	1.673	1.988	
	$\sigma(k)$	0.532	0.111	0.277	0.143	0.303	0.398	0.280	0.414	0.332	0.413	
125	$P(k)$	0.100	0.107	0.096	0.124	0.169	0.018	0.105	0.165	0.068	0.046	0.031
	$\mu(k)$	1.205	1.497	1.778	1.673	1.729	2.194	1.591	1.902	2.360	0.977	
	$\sigma(k)$	0.076	0.177	0.253	0.218	0.107	0.004	0.195	0.121	0.171	0.379	
150	$P(k)$	0.176	0.161	0.118	0.055	0.096	0.059	0.053	0.125	0.036	0.121	0.032
	$\mu(k)$	1.532	1.552	1.385	1.310	1.069	1.032	1.942	1.436	1.981	1.614	
	$\sigma(k)$	0.165	0.168	0.187	0.355	0.054	0.329	0.219	0.191	0.208	0.157	
175	$P(k)$	0.234	0.087	0.159	0.014	0.105	0.012	0.126	0.041	0.111	0.110	0.037
	$\mu(k)$	1.467	1.359	0.958	0.485	1.389	1.297	1.329	1.387	1.201	1.788	
	$\sigma(k)$	0.148	0.179	0.088	0.031	0.068	0.181	0.178	0.189	0.138	0.206	
200	$P(k)$	0.096	0.090	0.100	0.084	0.178	0.009	0.183	0.095	0.110	0.054	0.032
	$\mu(k)$	1.224	1.505	1.257	1.137	1.307	2.089	1.318	0.869	1.164	1.690	
	$\sigma(k)$	0.197	0.255	0.184	0.237	0.090	0.219	0.163	0.101	0.226	0.086	
225	$P(k)$	0.122	0.060	0.042	0.039	0.166	0.225	0.089	0.118	0.126	0.009	0.036
	$\mu(k)$	1.059	1.547	0.785	1.585	0.920	1.245	1.133	1.053	1.178	1.137	
	$\sigma(k)$	0.221	0.089	0.016	0.254	0.185	0.084	0.229	0.220	0.103	0.001	
250	$P(k)$	0.072	0.072	0.073	0.081	0.099	0.033	0.132	0.128	0.127	0.180	0.036
	$\mu(k)$	1.031	1.418	0.864	1.195	1.0273	0.736	1.045	0.791	1.132	1.161	
	$\sigma(k)$	0.037	0.177	0.171	0.241	0.223	0.003	0.227	0.137	0.139	0.055	
275	$P(k)$	0.125	0.064	0.168	0.097	0.095	0.079	0.075	0.078	0.120	0.097	0.031
	$\mu(k)$	0.676	1.217	1.009	1.097	0.900	0.914	0.886	1.003	1.102	1.179	
	$\sigma(k)$	0.059	0.201	0.077	0.234	0.122	0.113	0.233	0.251	0.053	0.211	
300	$P(k)$	0.037	0.029	0.095	0.118	0.102	0.182	0.078	0.097	0.143	0.119	0.033
	$\mu(k)$	0.614	1.047	0.860	0.865	0.981	1.004	0.927	1.286	0.777	0.891	
	$\sigma(k)$	0.006	0.038	0.191	0.192	0.080	0.066	0.207	0.133	0.161	0.199	
325	$P(k)$	0.062	0.227	0.072	0.055	0.211	0.087	0.093	0.142	0.115	0.062	0.037
	$\mu(k)$	1.030	0.946	1.226	0.959	0.912	0.836	0.703	0.333	0.582	0.945	
	$\sigma(k)$	0.157	0.108	0.123	0.120	0.105	0.089	0.046	0.046	0.025	0.112	
350	$P(k)$	0.100	0.092	0.074	0.044	0.040	0.107	0.049	0.096	0.188	0.209	0.037
	$\mu(k)$	0.551	0.672	1.030	0.898	0.539	1.055	0.748	0.804	0.884	0.903	
	$\sigma(k)$	0.022	0.057	0.159	0.149	0.200	0.154	0.086	0.091	0.091	0.094	

Table A.I.11. MVM for the DCT coefficient hierarchy, high quality video. The DCT is successively applied to 4×4 blocks, 8×8 blocks and to whole frames.

Rank / DCT	Model parameters										Error
$r=1$ 4x4	$P(k)$	0.136	0.038	0.011	0.015	0.127	0.071	0.007	0.134	0.284	0.175	0.007
	$\mu(k)$	0.024	0.196	0.359	2.757	0.093	0.309	-0.192	0.049	0.024	0.003	
	$\sigma(k)$	0.079	0.020	0.025	1.149	0.044	0.143	0.001	0.086	0.019	0.005	
$r=1$ 8x8	$P(k)$	0.172	0.179	0.016	0.166	0.046	0.162	0.081	0.071	0.029	0.077	0.006
	$\mu(k)$	0.125	0.119	1.758	0.002	0.736	0.178	0.089	0.001	0.812	0.547	
	$\sigma(k)$	0.113	0.111	0.367	0.006	0.140	0.118	0.009	0.182	0.370	0.039	
$r=1$ whole frames	$P(k)$	0.084	0.098	0.021	0.112	0.086	0.129	0.096	0.126	0.142	0.106	0.029
	$\mu(k)$	56.025	156.37	264.42	145.93	169.81	130.69	131.01	130.99	127.06	50.152	
	$\sigma(k)$	4.466	30.703	8.931	32.519	28.145	25.662	34.621	25.528	25.151	14.989	
$r=2$ 4x4	$P(k)$	0.293	0.029	0.184	0.041	0.052	0.052	0.001	0.064	0.007	0.277	0.012
	$\mu(k)$	0.029	2.047	0.002	-0.102	0.136	0.108	0.040	0.122	0.487	0.011	
	$\sigma(k)$	0.023	0.392	0.002	0.055	0.050	0.057	0.078	0.055	0.001	0.011	
$r=2$ 8x8	$P(k)$	0.076	0.074	0.102	0.064	0.109	0.092	0.122	0.085	0.197	0.079	0.003
	$\mu(k)$	0.171	0.397	0.126	0.095	0.001	0.227	0.031	0.215	0.012	-0.043	
	$\sigma(k)$	0.091	0.262	0.057	0.012	0.002	0.088	0.018	0.091	0.024	0.157	
$r=2$ whole frames		0.103	0.097	0.137	0.025	0.154	0.129	0.115	0.021	0.162	0.054	0.028
	$\mu(k)$	29.756	28.765	17.165	59.266	11.120	22.523	27.161	60.163	20.777	32.090	
	$\sigma(k)$	7.706	8.152	6.287	1.108	3.710	8.502	8.662	11.011	3.427	6.295	
$r=3$ 4x4	$P(k)$	0.007	0.211	0.250	0.014	0.021	0.241	0.118	0.007	0.007	0.123	0.006
	$\mu(k)$	-0.169	0.034	0.009	0.245	2.570	0.002	0.076	-0.143	-0.271	-0.012	
	$\sigma(k)$	0.001	0.013	0.008	0.005	0.679	0.002	0.046	0.001	0.001	0.026	
$r=3$ 8x8	$P(k)$	0.135	0.046	0.197	0.141	0.027	0.106	0.036	0.047	0.257	0.007	0.003
	$\mu(k)$	0.142	0.089	0.001	0.014	0.460	0.142	0.243	0.147	0.032	-0.409	
	$\sigma(k)$	0.094	0.087	0.004	0.063	0.039	0.028	0.067	0.093	0.025	0.001	
$r=3$ whole frames	$P(k)$	0.093	0.158	0.159	0.099	0.007	0.095	0.108	0.094	0.135	0.051	0.025
	$\mu(k)$	20.396	13.139	8.059	23.705	49.601	18.190	16.688	19.638	13.908	24.254	
	$\sigma(k)$	5.519	2.854	1.931	5.330	0.001	4.394	3.968	5.257	3.305	5.2111	
$r=4$ 4x4	$P(k)$	0.092	0.007	0.048	0.061	0.018	0.304	0.076	0.038	0.299	0.056	0.012
	$\mu(k)$	0.026	0.032	0.006	0.022	0.098	0.009	0.001	0.065	0.002	0.069	
	$\sigma(k)$	0.012	0.022	0.026	0.012	0.004	0.006	0.025	0.013	0.002	0.269	
$r=4$ 8x8	$P(k)$	0.021	0.050	0.007	0.202	0.421	0.026	0.014	0.042	0.126	0.090	0.004
	$\mu(k)$	0.366	0.032	-0.162	0.091	0.007	-0.081	-0.342	0.063	0.039	0.174	
	$\sigma(k)$	0.044	0.051	0.001	0.019	0.010	0.017	0.012	0.044	0.006	0.013	
$r=4$ whole frames	$P(k)$	0.094	0.079	0.103	0.102	0.075	0.105	0.124	0.119	0.128	0.068	0.031
	$\mu(k)$	14.485	7.228	14.097	13.652	16.560	11.610	10.911	11.813	9.818	19.582	
	$\sigma(k)$	4.502	0.479	3.616	4.323	4.309	4.473	4.282	4.195	3.226	3.961	
$r=5$ 4x4	$P(k)$	0.063	0.029	0.158	0.001	0.007	0.413	0.007	0.056	0.051	0.214	0.015
	$\mu(k)$	-0.028	1.296	0.007	0.190	-0.289	0.003	0.190	0.015	0.108	0.000	
	$\sigma(k)$	0.062	0.630	0.016	0.001	0.001	0.003	0.001	0.011	0.031	0.001	
$r=5$ 8x8	$P(k)$	0.015	0.015	0.016	0.007	0.530	0.007	0.146	0.141	0.007	0.115	0.002
	$\mu(k)$	-0.065	0.765	0.081	0.215	0.004	-0.246	0.061	0.025	-0.365	0.103	
	$\sigma(k)$	0.030	2.317	0.039	0.001	0.007	0.001	0.018	0.015	0.001	0.036	
$r=5$ whole frames	$P(k)$	0.109	0.137	0.117	0.115	0.078	0.070	0.117	0.128	0.030	0.095	0.027
	$\mu(k)$	10.319	8.998	10.979	11.164	14.632	14.744	11.177	6.458	21.574	11.841	
	$\sigma(k)$	3.021	3.027	2.714	2.900	2.826	2.795	2.896	1.155	4.013	3.327	

Table A.I.12. MVM for the DCT (applied on the whole frames) coefficient hierarchy, high quality video.

Rank	Model parameters											Error
1	$P(k)$	0.084	0.098	0.021	0.112	0.086	0.129	0.096	0.126	0.142	0.106	0.029
	$\mu(k)$	56.025	156.37	264.42	145.93	169.81	130.69	131.01	130.99	127.06	50.152	
	$\sigma(k)$	4.466	30.703	8.931	32.519	28.145	25.662	34.621	25.528	25.151	14.989	
25	$P(k)$	0.099	0.089	0.125	0.087	0.089	0.045	0.102	0.111	0.100	0.150	0.038
	$\mu(k)$	3.194	3.160	4.078	3.392	3.275	2.495	2.009	3.747	5.267	4.342	
	$\sigma(k)$	1.044	1.048	0.903	1.071	0.248	0.113	0.759	1.034	0.395	0.853	
50	$P(k)$	0.090	0.100	0.131	0.105	0.075	0.081	0.112	0.114	0.096	0.093	0.039
	$\mu(k)$	2.785	2.098	2.511	2.163	1.441	2.086	2.669	3.280	1.738	2.299	
	$\sigma(k)$	0.835	0.365	0.540	0.628	0.493	0.638	0.792	0.357	0.572	0.630	
75	$P(k)$	0.145	0.079	0.188	0.137	0.072	0.071	0.085	0.078	0.068	0.076	0.036
	$\mu(k)$	2.490	2.516	1.283	1.399	1.800	1.826	1.717	1.791	2.057	1.816	
	$\sigma(k)$	0.202	0.575	0.374	0.392	0.441	0.449	0.414	0.437	0.562	0.445	
100	$P(k)$	0.057	0.067	0.101	0.076	0.121	0.157	0.132	0.104	0.090	0.094	0.037
	$\mu(k)$	2.171	2.060	1.843	1.587	1.189	1.057	1.157	1.746	1.729	1.444	
	$\sigma(k)$	0.366	0.055	0.392	0.359	0.324	0.295	0.317	0.397	0.401	0.363	
125	$P(k)$	0.108	0.148	0.098	0.081	0.077	0.022	0.153	0.108	0.114	0.089	0.038
	$\mu(k)$	1.267	1.085	1.372	1.317	1.371	2.225	0.9764	1.100	1.769	1.360	
	$\sigma(k)$	0.415	0.271	0.423	0.422	0.423	0.624	0.332	0.338	0.095	0.423	
150	$P(k)$	0.161	0.138	0.088	0.007	0.130	0.149	0.107	0.106	0.114	0.043	0.040
	$\mu(k)$	1.129	0.980	1.331	0.616	0.970	0.665	1.799	1.125	1.537	1.441	
	$\sigma(k)$	0.261	0.222	0.225	0.001	0.220	0.167	0.329	0.261	0.156	0.370	
175	$P(k)$	0.046	0.138	0.041	0.095	0.039	0.131	0.153	0.185	0.132	0.040	0.034
	$\mu(k)$	1.667	0.768	1.521	1.059	1.427	0.963	1.399	0.984	0.545	1.486	
	$\sigma(k)$	0.281	0.127	0.325	0.196	0.344	0.199	0.101	0.183	0.119	0.333	
200	$P(k)$	0.125	0.079	0.089	0.144	0.085	0.116	0.071	0.097	0.106	0.088	0.037
	$\mu(k)$	0.915	0.631	0.908	0.687	0.879	0.757	1.484	1.304	0.827	1.183	
	$\sigma(k)$	0.279	0.222	0.281	0.218	0.279	0.265	0.295	0.080	0.276	0.328	
225	$P(k)$	0.039	0.075	0.134	0.117	0.105	0.091	0.108	0.140	0.039	0.149	0.030
	$\mu(k)$	1.056	1.237	0.668	0.501	0.745	0.641	1.224	0.868	1.587	0.976	
	$\sigma(k)$	0.251	0.075	0.156	0.136	0.181	0.117	0.144	0.194	0.244	0.145	
250	$P(k)$	0.137	0.112	0.061	0.081	0.108	0.038	0.166	0.137	0.053	0.106	0.031
	$\mu(k)$	0.442	0.864	0.886	0.884	0.882	1.238	0.618	1.145	1.363	0.699	
	$\sigma(k)$	0.098	0.172	0.170	0.170	0.170	0.279	0.106	0.103	0.246	0.141	
275	$P(k)$	0.084	0.097	0.067	0.068	0.044	0.045	0.160	0.122	0.192	0.119	0.037
	$\mu(k)$	0.779	0.795	0.775	0.714	0.718	1.331	0.469	0.561	1.068	0.674	
	$\sigma(k)$	0.186	0.172	0.187	0.192	0.123	0.254	0.165	0.182	0.102	0.192	
300	$P(k)$	0.093	0.108	0.029	0.087	0.089	0.165	0.117	0.124	0.165	0.020	0.036
	$\mu(k)$	0.723	0.828	1.089	0.987	1.203	0.575	0.781	0.676	0.404	0.813	
	$\sigma(k)$	0.159	0.174	0.002	0.065	0.216	0.095	0.172	0.148	0.092	0.077	
325	$P(k)$	0.169	0.111	0.064	0.097	0.118	0.034	0.115	0.087	0.076	0.128	0.036
	$\mu(k)$	0.977	0.567	0.750	0.723	0.441	1.267	0.589	0.731	0.690	0.443	
	$\sigma(k)$	0.088	0.174	0.147	0.155	0.152	0.217	0.177	0.153	0.165	0.152	
350	$P(k)$	0.123	0.129	0.089	0.077	0.183	0.059	0.081	0.049	0.113	0.096	0.039
	$\mu(k)$	0.486	0.736	0.593	0.519	0.936	1.045	0.504	0.317	0.421	0.618	
	$\sigma(k)$	0.175	0.047	0.181	0.180	0.069	0.211	0.023	0.047	0.044	0.031	

A.I.2.2. Model computation – MSVM

Table A.I.13. MSVM for the DCT coefficient hierarchy, low quality video. The DCT is successively applied to 4×4 blocks, 8×8 blocks and to whole frames.

Rank / DCT	Model parameters											Error
$r=1$ 4x4	$P(k)$	0.029	0.006	0.135	0.079	0.009	0.074	0.014	0.021	0.007	0.625	0.011
	$\mu(k)$	0.029	0.076	0.004	0.009	0.086	0.010	0.037	0.053	0.024	0.001	
	$\sigma(k)$	0.001	0.009	0.002	0.004	0.010	0.004	0.002	0.003	0.001	0.001	
$r=1$ 8x8	$P(k)$	0.088	0.008	0.265	0.014	0.198	0.037	0.027	0.017	0.254	0.093	0.014
	$\mu(k)$	0.080	0.180	0.003	0.298	0.028	0.184	0.145	0.283	0.010	0.069	
	$\sigma(k)$	0.023	0.052	0.002	0.071	0.010	0.028	0.002	0.072	0.005	0.025	
$r=1$ whole frames	$P(k)$	0.047	0.384	0.280	0.013	0.055	0.082	0.001	0.053	0.029	0.056	0.004
	$\mu(k)$	3.336	0.141	0.141	2.348	4.533	5.063	0.149	7.526	5.011	5.745	
	$\sigma(k)$	3.524	0.058	0.058	1.469	1.930	3.968	0.058	9.989	8.723	7.853	
$r=2$ 4x4	$P(k)$	0.215	0.049	0.234	0.234	0.029	0.049	0.017	0.064	0.045	0.063	0.018
	$\mu(k)$	0.009	0.070	0.028	0.003	0.125	0.067	0.079	0.000	0.354	0.202	
	$\sigma(k)$	0.004	0.022	0.012	0.001	0.008	0.021	0.021	0.001	0.107	0.049	
$r=2$ 8x8	$P(k)$	0.090	0.205	0.106	0.202	0.218	0.028	0.238	0.218	0.251	0.050	0.011
	$\mu(k)$	0.292	0.097	0.157	0.007	0.418	0.768	0.219	1.521	0.038	0.472	
	$\sigma(k)$	0.032	0.021	0.021	0.005	0.085	0.022	0.047	0.361	0.017	0.077	
$r=2$ whole frames	$P(k)$	0.053	0.555	0.077	0.039	0.116	0.047	0.047	0.037	0.007	0.021	0.003
	$\mu(k)$	8.859	0.140	5.471	8.362	0.143	8.596	1.568	8.292	2.766	3.565	
	$\sigma(k)$	1.403	0.055	0.700	1.401	0.055	1.415	5.527	1.391	0.001	0.132	
$r=3$ 4x4	$P(k)$	0.037	0.092	0.053	0.014	0.053	0.031	0.373	0.049	0.250	0.048	0.018
	$\mu(k)$	0.001	0.074	0.062	0.101	0.032	0.006	0.053	0.023	0.178	0.005	
	$\sigma(k)$	0.001	0.021	0.001	0.021	0.011	0.003	0.002	0.009	0.050	0.049	
$r=3$ 8x8	$P(k)$	0.186	0.041	0.033	0.232	0.090	0.172	0.057	0.075	0.034	0.079	0.019
	$\mu(k)$	0.099	0.245	0.123	0.041	0.160	0.009	0.070	0.225	0.410	0.354	
	$\sigma(k)$	0.020	0.059	0.041	0.013	0.008	0.006	0.005	0.035	0.109	0.100	
$r=3$ whole frames	$P(k)$	0.070	0.341	0.042	0.293	0.014	0.031	0.016	0.052	0.072	0.069	0.004
	$\mu(k)$	7.632	0.144	2.809	0.145	0.155	4.115	0.155	6.303	5.523	5.413	
	$\sigma(k)$	1.453	0.058	0.155	0.059	0.058	0.378	0.058	1.131	0.813	0.870	
$r=4$ 4x4	$P(k)$	0.040	0.123	0.008	0.246	0.034	0.026	0.139	0.285	0.076	0.022	0.015
	$\mu(k)$	0.198	0.048	0.139	0.013	0.097	0.099	0.026	0.003	0.065	0.146	
	$\sigma(k)$	0.020	0.007	0.026	0.005	0.010	0.011	0.006	0.003	0.005	0.022	
$r=4$ 8x8	$P(k)$	0.062	0.115	0.027	0.068	0.108	0.415	0.026	0.040	0.063	0.077	0.023
	$\mu(k)$	0.223	0.098	0.389	0.159	0.148	0.045	0.530	0.003	0.188	0.294	
	$\sigma(k)$	0.043	0.029	0.087	0.059	0.015	0.027	0.052	0.001	0.058	0.065	
$r=4$ whole frames	$P(k)$	0.020	0.101	0.057	0.209	0.047	0.053	0.130	0.017	0.313	0.050	0.004
	$\mu(k)$	3.555	5.710	4.614	0.137	2.667	0.137	4.536	0.138	0.139	0.137	
	$\sigma(k)$	4.598	0.462	1.070	0.741	0.059	0.319	0.741	0.059	0.59	0.739	
$r=5$ 4x4	$P(k)$	0.062	0.042	0.070	0.285	0.100	0.201	0.116	0.038	0.057	0.028	0.015
	$\mu(k)$	0.082	0.164	0.090	0.014	0.050	0.002	0.036	0.359	0.106	0.261	
	$\sigma(k)$	0.012	0.011	0.023	0.006	0.009	0.001	0.014	0.096	0.025	0.041	
$r=5$ 8x8	$P(k)$	0.0129	0.039	0.012	0.049	0.237	0.048	0.173	0.079	0.042	0.192	0.017
	$\mu(k)$	0.361	0.584	0.005	0.008	0.043	0.867	0.190	0.206	0.316	0.096	
	$\sigma(k)$	0.100	0.154	0.001	0.004	0.016	0.197	0.051	0.061	0.084	0.022	
$r=5$ whole frames	$P(k)$	0.114	0.087	0.038	0.064	0.017	0.085	0.0778	0.039	0.051	0.427	0.004
	$\mu(k)$	0.144	3.818	0.143	4.516	3.613	0.143	2.517	4.043	5.011	0.144	
	$\sigma(k)$	0.052	0.455	0.052	0.685	0.577	0.052	0.376	0.699	1.032	0.052	

Table A.I.14. MSVM for the DCT (applied on the whole frames) coefficient hierarchy, low quality video.

Rank	Model parameters											Error
1	$P(k)$	0.047	0.384	0.280	0.013	0.055	0.082	0.001	0.053	0.029	0.056	0.004
	$\mu(k)$	3.336	0.141	0.141	2.348	4.533	5.063	0.149	7.526	5.011	5.745	
	$\sigma(k)$	3.524	0.058	0.058	1.469	1.930	3.968	0.058	9.989	8.723	7.853	
25	$P(k)$	0.069	0.021	0.137	0.341	0.099	0.016	0.130	0.024	0.023	0.138	0.018
	$\mu(k)$	0.155	1.374	1.287	0.141	0.155	0.157	1.559	1.395	1.826	0.161	
	$\sigma(k)$	0.055	0.219	0.221	0.056	0.055	0.055	0.171	0.216	0.311	0.054	
50	$P(k)$	0.218	0.015	0.307	0.185	0.008	0.022	0.021	0.025	0.165	0.035	0.035
	$\mu(k)$	0.991	0.645	0.149	0.226	0.649	0.752	0.368	0.779	0.099	0.829	
	$\sigma(k)$	0.104	0.073	0.040	0.032	0.074	0.123	0.040	0.131	0.033	0.138	
75	$P(k)$	0.042	0.040	0.067	0.073	0.030	0.447	0.029	0.018	0.192	0.064	0.042
	$\mu(k)$	0.657	0.070	0.825	0.730	0.664	0.179	0.495	0.603	0.162	0.745	
	$\sigma(k)$	0.755	0.005	0.087	0.048	0.077	0.057	0.032	0.002	0.061	0.103	
100	$P(k)$	0.027	0.074	0.047	0.214	0.115	0.067	0.054	0.161	0.025	0.216	0.058
	$\mu(k)$	0.782	0.205	0.434	0.187	0.195	0.079	0.568	0.186	0.255	0.596	
	$\sigma(k)$	0.033	0.057	0.108	0.057	0.058	0.007	0.075	0.056	0.078	0.060	
125	$P(k)$	0.036	0.025	0.022	0.176	0.263	0.092	0.079	0.256	0.008	0.044	0.055
	$\mu(k)$	0.326	0.335	0.642	0.516	0.149	0.246	0.251	0.215	0.237	0.611	
	$\sigma(k)$	0.086	0.090	0.058	0.046	0.060	0.071	0.072	0.065	0.001	0.069	
150	$P(k)$	0.109	0.126	0.060	0.126	0.107	0.076	0.066	0.007	0.226	0.095	0.062
	$\mu(k)$	0.257	0.244	0.555	0.477	0.374	0.302	0.092	0.071	0.161	0.338	
	$\sigma(k)$	0.045	0.042	0.054	0.032	0.077	0.075	0.008	0.001	0.035	0.082	
175	$P(k)$	0.068	0.137	0.058	0.116	0.039	0.131	0.117	0.164	0.099	0.071	0.055
	$\mu(k)$	0.256	0.328	0.269	0.269	0.117	0.217	0.432	0.235	0.255	0.439	
	$\sigma(k)$	0.096	0.063	0.096	0.089	0.023	0.093	0.039	0.060	0.095	0.136	
200	$P(k)$	0.148	0.047	0.081	0.138	0.090	0.115	0.089	0.070	0.106	0.116	0.057
	$\mu(k)$	0.154	0.302	0.278	0.372	0.396	0.332	0.268	0.351	0.265	0.313	
	$\sigma(k)$	0.040	0.087	0.066	0.021	0.076	0.092	0.044	0.089	0.042	0.091	
225	$P(k)$	0.088	0.103	0.063	0.120	0.059	0.094	0.090	0.214	0.087	0.083	0.058
	$\mu(k)$	0.298	0.265	0.337	0.150	0.296	0.443	0.303	0.371	0.296	0.298	
	$\sigma(k)$	0.039	0.032	0.060	0.045	0.070	0.048	0.069	0.035	0.070	0.070	
250	$P(k)$	0.131	0.083	0.108	0.150	0.046	0.063	0.129	0.099	0.102	0.088	0.050
	$\mu(k)$	0.385	0.300	0.423	0.164	0.378	0.371	0.318	0.320	0.357	0.294	
	$\sigma(k)$	0.089	0.067	0.081	0.046	0.091	0.092	0.037	0.080	0.092	0.063	
275	$P(k)$	0.040	0.013	0.065	0.087	0.136	0.044	0.281	0.154	0.107	0.072	0.059
	$\mu(k)$	0.385	0.497	0.351	0.446	0.364	0.378	0.230	0.541	0.312	0.418	
	$\sigma(k)$	0.026	0.001	0.088	0.103	0.095	0.081	0.067	0.082	0.018	0.106	
300	$P(k)$	0.009	0.063	0.080	0.119	0.121	0.126	0.142	0.105	0.139	0.095	0.056
	$\mu(k)$	0.055	0.517	0.446	0.218	0.670	0.281	0.280	0.471	0.643	0.345	
	$\sigma(k)$	0.001	0.126	0.087	0.045	0.118	0.018	0.063	0.058	0.045	0.054	
325	$P(k)$	0.132	0.085	0.079	0.061	0.089	0.091	0.090	0.087	0.113	0.174	0.061
	$\mu(k)$	0.922	0.635	0.763	0.806	0.623	0.576	0.354	0.596	0.216	0.288	
	$\sigma(k)$	0.150	0.214	0.194	0.184	0.214	0.210	0.181	0.213	0.021	0.026	
350	$P(k)$	0.055	0.317	0.046	0.071	0.042	0.047	0.070	0.074	0.162	0.114	0.036
	$\mu(k)$	1.380	0.238	1.474	1.669	2.548	0.803	1.587	1.704	1.455	2.170	
	$\sigma(k)$	0.532	0.042	0.527	0.548	0.503	0.449	0.528	0.492	0.274	0.273	

Table A.I.15. MSVM for the DCT coefficient hierarchy, high quality video. The DCT is successively applied to 4×4 blocks, 8×8 blocks and to whole frames.

Rank / DCT	Model parameters											Error
$r=1$ 4x4	$P(k)$	0.042	0.055	0.038	0.359	0.016	0.059	0.025	0.199	0.155	0.052	0.005
	$\mu(k)$	0.008	0.022	0.045	0.001	0.002	0.008	0.007	0.000	0.004	0.006	
	$\sigma(k)$	0.004	0.005	0.013	0.001	0.011	0.004	0.012	0.001	0.001	0.004	
$r=1$ 8x8	$P(k)$	0.183	0.047	0.420	0.034	0.037	0.055	0.007	0.007	0.192	0.017	0.014
	$\mu(k)$	0.019	0.066	0.005	0.107	0.135	0.161	0.051	0.302	0.034	0.081	
	$\sigma(k)$	0.009	0.001	0.004	0.004	0.029	0.034	0.018	0.001	0.011	0.007	
$r=1$ whole frames	$P(k)$	0.206	0.105	0.122	0.040	0.076	0.085	0.108	0.092	0.096	0.069	0.057
	$\mu(k)$	0.818	0.437	0.538	0.842	0.295	0.615	0.408	0.638	0.911	0.614	
	$\sigma(k)$	0.078	0.153	0.098	0.236	0.033	0.141	0.147	0.172	0.223	0.165	
$r=2$ 4x4	$P(k)$	0.092	0.014	0.048	0.239	0.020	0.042	0.032	0.219	0.039	0.255	0.014
	$\mu(k)$	0.050	0.317	0.105	0.001	0.233	0.174	0.561	0.019	0.077	0.005	
	$\sigma(k)$	0.007	0.006	0.009	0.001	0.012	0.015	0.022	0.007	0.006	0.002	
$r=2$ 8x8	$P(k)$	0.336	0.056	0.050	0.108	0.079	0.070	0.042	0.060	0.183	0.014	0.016
	$\mu(k)$	0.012	0.546	0.271	0.097	0.118	0.191	0.576	0.293	0.050	1.196	
	$\sigma(k)$	0.008	0.079	0.071	0.028	0.031	0.038	0.169	0.054	0.019	0.029	
$r=2$ whole frames	$P(k)$	0.089	0.066	0.065	0.049	0.109	0.075	0.184	0.221	0.215	0.118	0.055
	$\mu(k)$	0.656	0.259	0.552	0.737	0.755	0.474	0.846	0.462	1.223	0.659	
	$\sigma(k)$	0.189	0.033	0.165	0.187	0.1509	0.143	0.094	0.127	0.045	0.184	
$r=3$ 4x4	$P(k)$	0.343	0.027	0.044	0.082	0.059	0.024	0.033	0.113	0.023	0.252	0.013
	$\mu(k)$	0.011	0.227	0.091	0.056	0.027	0.099	0.129	0.034	0.115	0.003	
	$\sigma(k)$	0.005	0.036	0.010	0.009	0.009	0.034	0.033	0.009	0.035	0.002	
$r=3$ 8x8	$P(k)$	0.065	0.099	0.255	0.091	0.063	0.050	0.215	0.066	0.029	0.069	0.016
	$\mu(k)$	0.397	0.201	0.059	0.122	0.002	0.327	0.022	0.189	0.098	0.169	
	$\sigma(k)$	0.112	0.043	0.019	0.010	0.002	0.047	0.011	0.048	0.034	0.050	
$r=3$ whole frames	$P(k)$	0.094	0.068	0.078	0.161	0.117	0.057	0.078	0.196	0.066	0.085	0.057
	$\mu(k)$	0.672	0.636	0.683	0.462	0.265	1.085	0.612	0.841	0.603	0.602	
	$\sigma(k)$	0.163	0.157	0.165	0.081	0.053	0.133	0.153	0.082	0.150	0.150	
$r=4$ 4x4	$P(k)$	0.037	0.218	0.318	0.083	0.029	0.064	0.036	0.048	0.132	0.036	0.021
	$\mu(k)$	0.066	0.014	0.005	0.053	0.201	0.029	0.058	0.087	0.028	0.124	
	$\sigma(k)$	0.012	0.005	0.003	0.008	0.022	0.008	0.013	0.003	0.007	0.013	
$r=4$ 8x8	$P(k)$	0.055	0.068	0.078	0.053	0.107	0.085	0.224	0.049	0.037	0.244	0.029
	$\mu(k)$	0.266	0.188	0.168	0.255	0.082	0.151	0.054	0.265	0.529	0.011	
	$\sigma(k)$	0.072	0.036	0.036	0.075	0.012	0.019	0.026	0.072	0.122	0.007	
$r=4$ whole frames	$P(k)$	0.143	0.100	0.128	0.135	0.145	0.020	0.029	0.097	0.069	0.134	0.056
	$\mu(k)$	0.499	0.869	0.484	0.496	0.826	1.237	0.493	0.653	0.262	0.777	
	$\sigma(k)$	0.147	0.126	0.145	0.139	0.087	0.024	0.077	0.177	0.033	0.152	
$r=5$ 4x4	$P(k)$	0.032	0.021	0.197	0.175	0.176	0.067	0.155	0.029	0.041	0.108	0.019
	$\mu(k)$	0.276	0.153	0.028	0.009	0.030	0.153	0.003	0.690	0.103	0.076	
	$\sigma(k)$	0.035	0.048	0.012	0.004	0.013	0.022	0.002	0.121	0.005	0.016	
$r=5$ 8x8	$P(k)$	0.321	0.056	0.075	0.069	0.039	0.215	0.042	0.087	0.066	0.030	0.014
	$\mu(k)$	0.044	0.203	0.174	0.007	0.392	0.132	0.464	0.276	0.668	0.368	
	$\sigma(k)$	0.020	0.058	0.053	0.003	0.064	0.048	0.123	0.022	0.159	0.083	
$r=5$ whole frames	$P(k)$	0.126	0.137	0.080	0.090	0.084	0.007	0.089	0.234	0.074	0.078	0.067
	$\mu(k)$	0.822	0.615	0.649	0.738	0.843	0.164	0.857	0.430	0.244	0.759	
	$\sigma(k)$	0.184	0.129	0.142	0.172	0.168	0.001	0.179	0.086	0.031	0.179	

Table A.I.16. MSVM for the DCT (applied on the whole frames) coefficient hierarchy, high quality video.

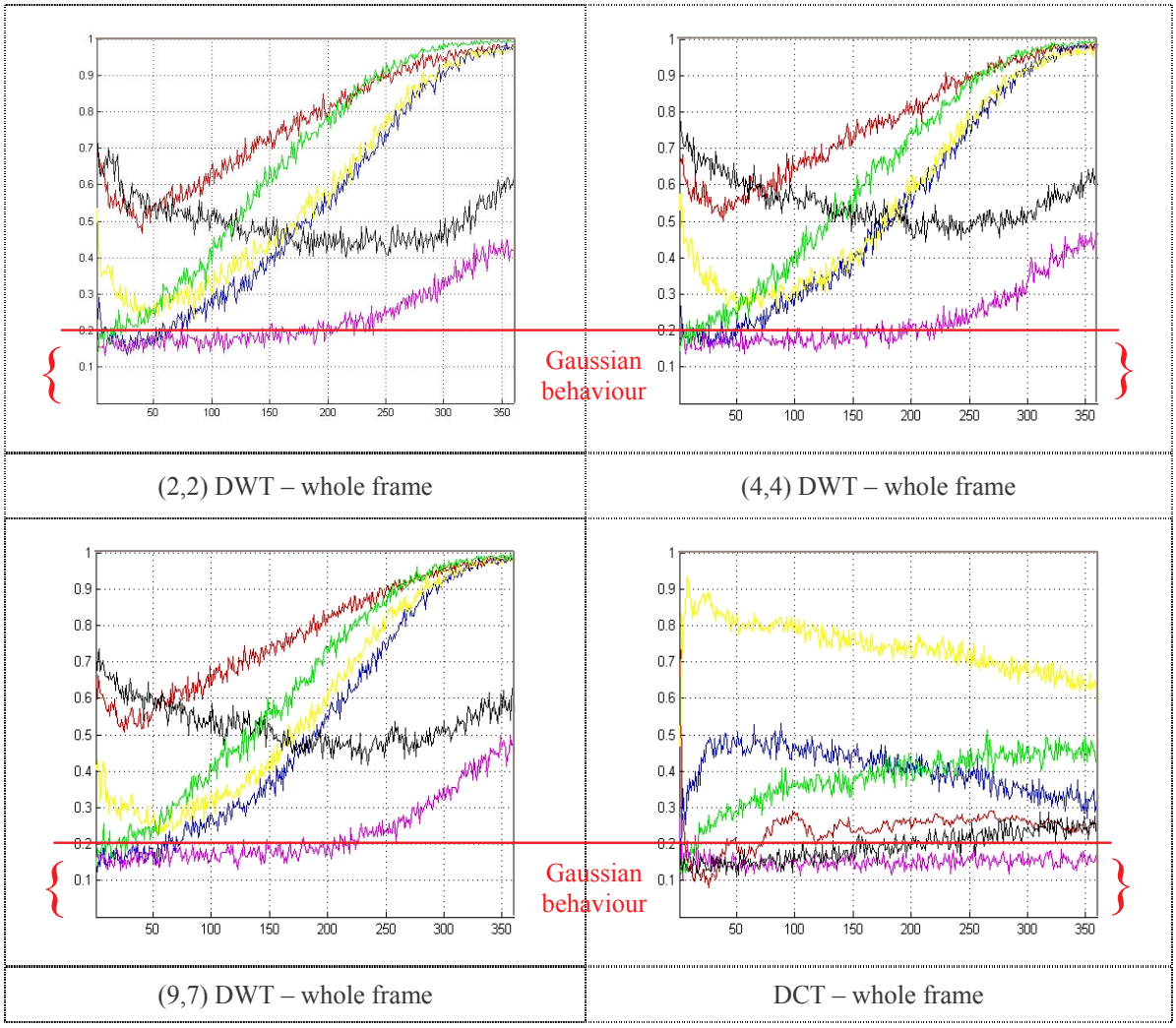
Rank	Model parameters											Error
1	$P(k)$	0.206	0.105	0.122	0.040	0.076	0.085	0.108	0.092	0.096	0.069	0.057
	$\mu(k)$	0.818	0.437	0.538	0.842	0.295	0.615	0.408	0.638	0.911	0.614	
	$\sigma(k)$	0.078	0.153	0.098	0.236	0.033	0.141	0.147	0.172	0.223	0.165	
25	$P(k)$	0.122	0.087	0.164	0.106	0.072	0.144	0.068	0.070	0.085	0.083	0.054
	$\mu(k)$	0.397	0.642	0.877	0.676	0.709	0.515	0.676	0.276	0.737	1.031	
	$\sigma(k)$	0.124	1.759	0.091	0.172	0.161	0.091	0.173	0.033	0.149	0.197	
50	$P(k)$	0.106	0.128	0.025	0.087	0.047	0.141	0.065	0.148	0.096	0.156	0.058
	$\mu(k)$	0.655	0.712	1.245	0.974	0.269	0.616	0.870	0.917	0.861	0.410	
	$\sigma(k)$	0.098	0.147	0.204	0.219	0.009	0.129	0.213	0.072	0.211	0.090	
75	$P(k)$	0.089	0.067	0.065	0.198	0.054	0.080	0.083	0.114	0.107	0.141	0.055
	$\mu(k)$	0.856	0.817	0.324	0.996	0.704	0.672	0.897	0.606	0.720	0.574	
	$\sigma(k)$	0.266	0.251	0.054	0.099	0.189	0.188	0.273	0.175	0.191	0.157	
100	$P(k)$	0.081	0.074	0.114	0.100	0.123	0.110	0.164	0.065	0.064	0.100	0.058
	$\mu(k)$	0.669	0.688	0.915	0.615	0.905	0.584	1.094	0.530	0.323	0.636	
	$\sigma(k)$	0.220	0.215	0.152	0.235	0.156	0.237	0.128	0.232	0.057	0.109	
125	$P(k)$	0.111	0.079	0.040	0.073	0.091	0.105	0.102	0.061	0.195	0.140	0.056
	$\mu(k)$	0.847	0.760	0.353	1.157	0.851	0.590	0.638	1.259	1.099	0.639	
	$\sigma(k)$	0.206	0.214	0.005	0.293	0.226	0.179	0.191	0.271	0.118	0.187	
150	$P(k)$	0.127	0.146	0.059	0.119	0.073	0.111	0.173	0.075	0.066	0.048	0.060
	$\mu(k)$	0.812	0.643	0.870	0.688	1.048	0.994	1.212	1.433	0.781	0.384	
	$\sigma(k)$	0.219	0.188	0.226	0.198	0.294	0.188	0.116	0.266	0.215	0.007	
175	$P(k)$	0.019	0.117	0.103	0.118	0.163	0.073	0.113	0.139	0.064	0.087	0.059
	$\mu(k)$	1.954	1.152	0.868	1.078	1.322	0.841	0.861	0.705	0.459	0.908	
	$\sigma(k)$	0.039	0.298	0.258	0.310	0.131	0.252	0.256	0.209	0.037	0.267	
200	$P(k)$	0.105	0.099	0.124	0.080	0.184	0.039	0.055	0.095	0.069	0.149	0.058
	$\mu(k)$	0.873	1.018	0.973	1.079	1.482	1.550	1.569	1.127	0.493	0.821	
	$\sigma(k)$	0.244	0.267	0.265	0.257	0.1409	0.427	0.423	0.239	0.046	0.228	
225	$P(k)$	0.087	0.051	0.156	0.119	0.079	0.114	0.129	0.096	0.088	0.081	0.053
	$\mu(k)$	1.443	1.601	1.166	1.676	1.389	0.593	0.830	1.132	1.593	1.318	
	$\sigma(k)$	0.294	0.407	0.154	0.157	0.307	0.101	0.136	0.200	0.405	0.301	
250	$P(k)$	0.080	0.076	0.123	0.134	0.019	0.100	0.098	0.136	0.080	0.152	0.050
	$\mu(k)$	2.032	1.74	1.393	1.776	0.652	1.517	1.474	0.789	1.387	1.167	
	$\sigma(k)$	0.412	0.439	0.315	0.182	0.007	0.317	0.321	0.131	0.314	0.192	
275	$P(k)$	0.075	0.103	0.118	0.128	0.068	0.105	0.094	0.097	0.095	0.118	0.055
	$\mu(k)$	1.948	1.639	1.979	2.280	2.457	1.559	0.755	1.150	1.813	1.438	
	$\sigma(k)$	0.430	0.377	0.382	0.244	0.609	0.349	0.146	0.226	0.413	0.299	
300	$P(k)$	0.146	0.084	0.096	0.091	0.047	0.112	0.091	0.097	0.113	0.122	0.046
	$\mu(k)$	1.901	1.936	2.077	2.339	2.835	1.823	2.124	2.722	2.119	1.079	
	$\sigma(k)$	0.277	0.839	0.808	0.724	0.256	0.432	0.794	0.257	0.416	0.244	
325	$P(k)$	0.065	0.120	0.088	0.116	0.144	0.086	0.080	0.095	0.125	0.080	0.047
	$\mu(k)$	1.351	3.492	1.757	4.053	2.575	3.545	2.267	3.034	3.250	2.499	
	$\sigma(k)$	0.115	0.579	0.159	0.491	0.361	0.565	0.325	0.572	0.610	0.371	
350	$P(k)$	0.072	0.139	0.051	0.107	0.101	0.121	0.080	0.123	0.128	0.078	0.044
	$\mu(k)$	4.815	3.928	8.625	6.825	6.375	7.347	7.416	7.393	5.523	5.129	
	$\sigma(k)$	2.532	0.850	1.508	1.634	1.518	1.622	1.689	1.269	1.197	1.155	

Appendix II

Gaussian Hypothesis

In the literature, in the absence of any theoretical or experimental support, the video watermarking attacks are by default assumed to be Gaussian distributed. In order to establish the validity of the Gaussian hypothesis, a statistical approach witch combines four types of tests (the *Chi - square*, the *Ro*, the *Fisher*, and the *Student*) was briefly presented in the state-of-the-art of the Section IV.1.1. This appendix presents additional experimental results.

Figure A.II.1 display the results obtained when the Chi – square test was applied to six attacks (the Gaussian filtering, the FMLR, the JPEG compression, sharpening, rotation +5°, and StirMark) for a sampling period of $D = 600$ frames. The investigation was realized in the hierarchy of the DWT and DCT coefficients, for video coded at low quality rate.



(2,2) DWT – whole frame | (4,4) DWT – whole frame

(9,7) DWT – whole frame | DCT – whole frame

Figure A.II.1. The results of the Chi-square tests for Gaussian filtering (yellow), FMLR (magenta), JPEG (black), sharpening (blue), rotation +5° (green), and StirMark (red) attacks. The tests were applied for $L = 35000$ frames, $R = 360$ coefficients, $D = 600$ frames, and $\alpha = 0.05$.

In the DWT domain, only the FMLR attack effects can be considered Gaussian distributed and only up to the 200th rank. However, in the DCT domain the effects of the FMLR attack and the JPEG compression up to the 200th rank can be considered Gaussian distributed.

Appendix III

Attack Modelling

A.III.1. The 2D-DWT coefficients modelling

A.III.1.1. Model Computation

Table A.III.1. IVM for (9,7) DWT coefficients hierarchy, low quality video and different attacks.

Attack/ Rank			Model parameters										Error
FMLR	1	*P(k)*	0.099	0.071	0.086	0.026	0.077	0.052	0.151	0.068	0.150	0.218	0.031
		μ(k)	-0.076	-0.073	-0.069	-0.193	-0.111	-0.097	-0.067	-0.108	-0.087	-0.057	
		σ(k)	0.030	0.030	0.030	0.011	0.035	0.038	0.029	0.036	0.023	0.026	
	50	*P(k)*	0.109	0.119	0.014	0.088	0.065	0.147	0.152	0.125	0.113	0.068	0.032
		μ(k)	-0.044	-0.015	-0.026	-0.043	-0.044	-0.028	-0.026	-0.026	-0.025	-0.043	
		σ(k)	0.023	0.018	0.001	0.023	0.023	0.019	0.020	0.020	0.020	0.023	
	100	*P(k)*	0.107	0.110	0.031	0.071	0.246	0.078	0.086	0.137	0.082	0.050	0.026
		μ(k)	-0.038	-0.013	-0.055	-0.011	-0.016	-0.050	-0.016	-0.019	-0.009	-0.011	
		σ(k)	0.005	0.015	0.019	0.015	0.012	0.021	0.026	0.010	0.016	0.024	
	150	*P(k)*	0.049	0.136	0.091	0.189	0.042	0.059	0.187	0.133	0.044	0.069	0.025
		μ(k)	-0.003	-0.015	-0.031	-0.004	-0.006	-0.017	-0.014	-0.027	-0.001	-0.020	
		σ(k)	0.023	0.011	0.017	0.010	0.023	0.020	0.011	0.017	0.023	0.019	
	200	*P(k)*	0.055	0.239	0.070	0.045	0.124	0.030	0.069	0.093	0.051	0.223	0.026
		μ(k)	-0.025	-0.009	-0.008	-0.020	0.005	-0.019	-0.009	-0.024	-0.015	-0.009	
		σ(k)	0.019	0.009	0.015	0.019	0.0013	0.018	0.015	0.019	0.017	0.009	
	250	*P(k)*	0.124	0.057	0.115	0.072	0.025	0.209	0.063	0.176	0.111	0.048	0.021
		μ(k)	-0.014	-0.009	-0.009	-0.009	0.031	0.005	-0.009	-0.004	-0.191	-0.011	
		σ(k)	0.013	0.013	0.003	0.013	0.016	0.008	0.013	0.005	0.012	0.013	
	300	*P(k)*	0.088	0.109	0.025	0.117	0.056	0.201	0.088	0.053	0.049	0.213	0.018
		μ(k)	0.002	-0.008	-0.036	-0.003	-0.013	-0.002	-0.002	-0.008	-0.009	-0.003	
		σ(k)	0.012	0.014	0.019	0.013	0.014	0.006	0.013	0.014	0.014	0.006	
	350	*P(k)*	0.060	0.078	0.164	0.058	0.067	0.070	0.135	0.082	0.124	0.162	0.014
		μ(k)	-0.006	-0.004	0.001	-0.011	0.011	0.001	0.001	-0.004	-0.001	-0.002	
		σ(k)	0.016	0.016	0.003	0.015	0.014	0.017	0.007	0.016	0.007	0.007	
Gaussian filtering	1	*P(k)*	0.075	0.024	0.270	0.063	0.069	0.056	0.072	0.118	0.20	0.022	0.035
		μ(k)	-0.087	-0.157	-0.048	-0.078	-0.069	-0.115	-0.088	-0.051	-0.057	-0.078	
		σ(k)	0.019	0.021	0.014	0.018	0.018	0.003	0.019	0.015	0.016	0.001	
	50	*P(k)*	0.034	0.257	0.083	0.022	0.239	0.049	0.049	0.117	0.013	0.126	0.022
		μ(k)	-0.005	-0.007	-0.030	0.004	-0.018	-0.029	-0.025	-0.023	0.008	-0.031	
		σ(k)	0.017	0.005	0.012	0.014	0.004	0.0012	0.0012	0.009	0.012	0.012	
	100	*P(k)*	0.142	0.118	0.097	0.014	0.065	0.062	0.110	0.017	0.172	0.202	0.016
		μ(k)	-0.013	0.001	-0.018	0.064	-0.014	-0.029	-0.016	-0.010	-0.009	-0.005	
		σ(k)	0.009	0.010	0.008	0.006	0.009	0.016	0.008	0.012	0.005	0.004	
	150	*P(k)*	0.192	0.042	0.098	0.032	0.152	0.018	0.020	0.262	0.007	0.176	0.017
		μ(k)	-0.002	-0.027	-0.007	-0.019	-0.008	0.009	0.030	-0.010	-0.064	0.001	
		σ(k)	0.002	0.005	0.008	0.001	0.003	0.010	0.004	0.004	0.001	0.007	
	200	*P(k)*	0.208	0.056	0.031	0.295	0.024	0.044	0.057	0.060	0.196	0.029	0.012
		μ(k)	-0.001	-0.010	0.0032	-0.005	-0.005	0.010	-0.010	-0.010	-0.003	0.021	
		σ(k)	0.005	0.009	0.018	0.005	0.010	0.009	0.009	0.009	0.002	0.010	
	250	*P(k)*	0.189	0.014	0.059	0.183	0.061	0.136	0.177	0.038	0.076	0.066	0.012
		μ(k)	-0.001	0.059	0.006	-0.001	-0.009	0.001	-0.003	0.023	-0.009	-0.009	
		σ(k)	0.003	0.004	0.005	0.003	0.006	0.003	0.002	0.008	0.007	0.006	
	300	*P(k)*	0.020	0.032	0.014	0.344	0.029	0.141	0.065	0.026	0.322	0.007	0.009
		μ(k)	0.002	-0.010	0.080	-0.003	-0.007	0.004	-0.012	0.024	-0.001	0.053	
		σ(k)	0.016	0.001	0.001	0.002	0.001	0.004	0.007	0.006	0.002	0.001	
	350	*P(k)*	0.085	0.147	0.032	0.095	0.569	0.015	0.008	0.011	0.018	0.019	0.011
		μ(k)	0.001	-0.001	-0.005	-0.001	-0.001	0.013	0.076	0.029	0.003	0.048	
		σ(k)	0.006	0.006	0.014	0.006	0.001	0.013	0.006	0.019	0.016	0.015	

Table A.III.1. *(continued)*

Attack/ Rank		Model parameters											Error
JPEG	1	$P(k)$	0.103	0.007	0.183	0.060	0.111	0.180	0.069	0.080	0.114	0.092	0.011
		$\mu(k)$	-0.002	-0.207	0.002	-0.058	-0.004	-0.008	-0.009	0.003	-0.001	-0.028	
		$\sigma(k)$	0.021	0.001	0.003	0.027	0.005	0.007	0.022	0.020	0.021	0.016	
	50	$P(k)$	0.164	0.059	0.355	0.058	0.066	0.069	0.062	0.090	0.007	0.069	0.010
		$\mu(k)$	0.002	-0.023	-0.002	-0.024	-0.001	0.021	-0.001	-0.029	0.019	-0.001	
		$\sigma(k)$	0.009	0.025	0.005	0.002	0.013	0.018	0.013	0.024	0.001	0.013	
	100	$P(k)$	0.004	0.049	0.258	0.029	0.072	0.002	0.073	0.050	0.117	0.345	0.009
		$\mu(k)$	-0.009	0.013	-0.004	-0.063	0.010	-0.005	0.001	-0.009	-0.021	0.004	
		$\sigma(k)$	0.021	0.015	0.004	0.011	0.015	0.018	0.014	0.001	0.007	0.005	
	150	$P(k)$	0.076	0.041	0.100	0.042	0.054	0.224	0.065	0.294	0.058	0.045	0.010
		$\mu(k)$	0.005	0.001	0.004	0.001	-0.011	-0.006	-0.007	0.002	-0.004	-0.001	
		$\sigma(k)$	0.010	0.027	0.010	0.027	0.014	0.006	0.014	0.004	0.013	0.027	
	200	$P(k)$	0.102	0.055	0.060	0.140	0.162	0.140	0.071	0.025	0.067	0.179	0.007
		$\mu(k)$	0.001	-0.014	-0.003	-0.004	0.006	-0.002	-0.006	0.037	-0.005	-0.001	
		$\sigma(k)$	0.009	0.020	0.009	0.009	0.006	0.010	0.009	0.028	0.022	0.002	
	250	$P(k)$	0.104	0.130	0.260	0.076	0.041	0.104	0.111	0.068	0.037	0.068	0.010
		$\mu(k)$	-0.003	-0.001	-0.001	-0.002	-0.004	0.001	-0.002	-0.001	-0.015	-0.002	
		$\sigma(k)$	0.010	0.004	0.004	0.010	0.033	0.009	0.010	0.010	0.032	0.010	
	300	$P(k)$	0.230	0.112	0.048	0.049	0.101	0.038	0.072	0.034	0.206	0.111	0.009
		$\mu(k)$	-0.004	-0.001	0.013	-0.020	-0.004	0.021	0.001	-0.002	-0.001	0.008	
		$\sigma(k)$	0.006	0.008	0.001	0.012	0.006	0.014	0.001	0.016	0.004	0.006	
	350	$P(k)$	0.100	0.017	0.130	0.073	0.039	0.050	0.007	0.068	0.460	0.056	0.006
		$\mu(k)$	0.007	0.018	-0.005	-0.001	-0.007	0.007	0.048	-0.003	-0.001	0.007	
		$\sigma(k)$	0.007	0.011	0.005	0.007	0.013	0.012	0.001	0.013	0.022	0.001	
Rotation +2°	1	$P(k)$	0.086	0.065	0.068	0.089	0.094	0.105	0.111	0.105	0.136	0.140	0.045
		$\mu(k)$	-0.686	-0.007	-0.541	-0.541	-0.260	-0.969	-0.404	-0.789	-0.234	0.583	
		$\sigma(k)$	0.334	0.076	0.338	0.337	0.269	0.268	0.248	0.315	0.134	0.169	
	50	$P(k)$	0.064	0.019	0.124	0.115	0.125	0.170	0.103	0.084	0.091	0.105	0.026
		$\mu(k)$	-0.168	0.337	-0.134	-0.111	-0.222	-0.046	-0.137	-0.116	-0.391	-0.035	
		$\sigma(k)$	0.131	0.253	0.090	0.093	0.063	0.068	0.093	0.153	0.093	0.163	
	100	$P(k)$	0.100	0.105	0.146	0.088	0.111	0.065	0.072	0.072	0.179	0.061	0.023
		$\mu(k)$	-0.012	0.067	-0.151	-0.109	-0.046	-0.200	-0.185	-0.282	-0.034	-0.083	
		$\sigma(k)$	0.008	0.038	0.025	0.088	0.048	0.090	0.092	0.121	0.054	0.046	
	150	$P(k)$	0.028	0.029	0.034	0.098	0.161	0.040	0.173	0.305	0.041	0.091	0.016
		$\mu(k)$	0.207	0.061	0.169	-0.130	-0.055	0.110	-0.065	-0.008	-0.244	-0.177	
		$\sigma(k)$	0.132	0.172	0.143	0.049	0.035	0.010	0.031	0.029	0.147	0.081	
	200	$P(k)$	0.232	0.064	0.146	0.135	0.053	0.147	0.057	0.064	0.045	0.057	0.013
		$\mu(k)$	-0028	-0.150	-0.025	-0.103	-0.047	-0.012	0.011	0.073	-0.077	0.009	
		$\sigma(k)$	0.043	0.102	0.044	0.024	0.121	0.012	0.062	0.087	0.120	0.060	
	250	$P(k)$	0.024	0.020	0.132	0.108	0.135	0.297	0.037	0.130	0.105	0.014	0.009
		$\mu(k)$	-0.266	-0.095	-0.075	-0.031	-0.034	-0.006	0.190	0.039	-0.084	0.087	
		$\sigma(k)$	0.068	0.001	0.052	0.004	0.022	0.007	0.060	0.017	0.051	0.001	
	300	$P(k)$	0.167	0.219	0.176	0.056	0.057	0.054	0.115	0.037	0.048	0.071	0.010
		$\mu(k)$	-0.022	-0.005	0.004	-0.015	0.001	-0.016	-0.016	-0.091	-0.103	0.059	
		$\sigma(k)$	0.021	0.003	0.024	0.108	0.108	0.108	0.008	0.006	0.084	0.046	
	350	$P(k)$	0.046	0.090	0.148	0.184	0.126	0.244	0.030	0.011	0.016	0.106	0.009
		$\mu(k)$	-0.122	-0.016	-0.022	0.005	0.013	-0.001	0.216	0.048	0.043	0.004	
		$\sigma(k)$	0.148	0.006	0.065	0.025	0.073	0.006	0.075	0.001	0.001	0.001	

Table A.III.1. *(continued)*

Attack/ Rank			Model parameters										Error
Rotation -2°	1	$P(k)$	0.102	0.143	0.122	0.142	0.091	0.112	0.120	0.007	0.114	0.048	0.054
		$\mu(k)$	-0.699	-0.629	-0.583	-0.443	-0.247	-0.020	-0.639	-2.067	-0.681	-0.140	
		$\sigma(k)$	0.297	0.296	0.285	0.181	0.182	0.141	0.297	0.001	0.298	0.009	
	50	$P(k)$	0.165	0.082	0.067	0.030	0.259	0.072	0.057	0.111	0.076	0.082	0.035
		$\mu(k)$	-0.156	-0.114	-0.274	-0.030	-0.049	-0.034	-0.269	-0.357	-0.134	-0.125	
		$\sigma(k)$	0.034	0.189	0.021	0.002	0.066	0.115	0.169	0.139	0.192	0.191	
	100	$P(k)$	0.061	0.195	0.140	0.122	0.096	0.133	0.007	0.08	0.057	0.109	0.023
		$\mu(k)$	-0.030	-0.063	-0.136	-0.176	-0.091	-0.041	0.092	-0.156	0.199	0.028	
		$\sigma(k)$	0.126	0.041	0.089	0.084	0.068	0.046	0.001	0.088	0.142	0.031	
	150	$P(k)$	0.075	0.103	0.111	0.080	0.308	0.020	0.055	0.016	0.148	0.086	0.015
		$\mu(k)$	-0.123	-0.095	-0.104	-0.109	-0.044	0.204	-0.255	-0.158	0.020	-0.014	
		$\sigma(k)$	0.111	0.067	0.062	0.105	0.023	0.012	0.148	0.001	0.030	0.090	
	200	$P(k)$	0.153	0.007	0.114	0.151	0.113	0.232	0.032	0.035	0.111	0.052	0.011
		$\mu(k)$	-0.023	0.328	-0.040	-0.059	-0.043	-0.017	0.002	0.034	-0.125	0.090	
		$\sigma(k)$	0.012	0.001	0.096	0.070	0.075	0.033	0.001	0.046	0.126	0.078	
	250	$P(k)$	0.007	0.159	0.094	0.101	0.070	0.174	0.205	0.141	0.028	0.021	0.009
		$\mu(k)$	0.415	0.020	-0.077	-0.028	-0.117	-0.009	-0.012	-0.031	0.160	-0.332	
		$\sigma(k)$	0.001	0.036	0.065	0.040	0.058	0.041	0.011	0.024	0.013	0.029	
	300	$P(k)$	0.040	0.155	0.043	0.148	0.104	0.128	0.034	0.289	0.020	0.039	0.010
		$\mu(k)$	-0.137	-0.009	0.120	-0.008	0.053	0.006	0.020	-0.016	-0.121	-0.323	
		$\sigma(k)$	0.050	0.030	0.028	0.005	0.006	0.024	0.0053	0.022	0.016	0.120	
	350	$P(k)$	0.067	0.007	0.034	0.228	0.239	0.067	0.093	0.007	0.064	0.194	0.007
		$\mu(k)$	0.064	0.011	-0.160	-0.008	-0.002	0.040	-0.588	0.471	0.022	-0.002	
		$\sigma(k)$	0.065	0.001	0.101	0.015	0.019	0.010	0.041	0.001	0.073	0.004	
Rotation +5°	1	$P(k)$	0.096	0.061	0.127	0.086	0.096	0.098	0.097	0.101	0.092	0.147	0.040
		$\mu(k)$	-0.571	-0.023	-0.527	-0.835	-0.628	-0.754	-0.584	-0.519	-0.802	-0.462	
		$\sigma(k)$	0.217	0.104	0.175	0.254	0.254	0.268	0.115	0.194	0.261	0.150	
	50	$P(k)$	0.062	0.101	0.100	0.217	0.165	0.082	0.072	0.044	0.069	0.086	0.027
		$\mu(k)$	-0.363	-0.175	-0.090	-0.104	-0.175	-0.105	-0.392	-0.222	-0.212	-0.089	
		$\sigma(k)$	0.174	0.096	0.134	0.061	0.077	0.136	0.168	0.005	0.083	0.134	
	100	$P(k)$	0.051	0.041	0.198	0.031	0.060	0.051	0.071	0.279	0.147	0.069	0.024
		$\mu(k)$	-0.010	-0.109	-0.040	-0.025	-0.359	-0.074	0.058	-0.133	-0.102	-0.054	
		$\sigma(k)$	0.139	0.169	0.028	0.145	0.082	0.163	0.136	0.055	0.066	0.087	
	150	$P(k)$	0.058	0.007	0.189	0.085	0.095	0.095	0.088	0.089	0.079	0.215	0.018
		$\mu(k)$	-0.086	-0.154	-0.063	-0.029	0.014	-0.040	-0.065	-0.077	0.008	-0.094	
		$\sigma(k)$	0.161	0.001	0.044	0.060	0.026	0.049	0.160	0.161	0.139	0.042	
	200	$P(k)$	0.183	0.040	0.091	0.216	0.201	0.095	0.052	0.029	0.079	0.014	0.016
		$\mu(k)$	-0.033	-0.149	-0.020	-0.031	-0.050	0.067	-0.055	-0.282	-0.029	-0.547	
		$\sigma(k)$	0.049	0.014	0.110	0.013	0.053	0.105	0.103	0.039	0.111	0.001	
	250	$P(k)$	0.081	0.029	0.114	0.094	0.068	0.065	0.024	0.064	0.437	0.024	0.014
		$\mu(k)$	-0.008	0.364	-0.104	-0.031	0.010	-0.004	-0.274	0.062	-0.016	0.049	
		$\sigma(k)$	0.097	0.178	0.057	0.022	0.037	0.098	0.023	0.099	0.020	0.034	
	300	$P(k)$	0.135	0.035	0.078	0.082	0.111	0.072	0.34	0.088	0.026	0.036	0.009
		$\mu(k)$	0.018	0.108	-0.076	-0.017	-0.018	-0.128	-0.016	0.014	0.132	0.054	
		$\sigma(k)$	0.021	0.150	0.047	0.080	0.023	0.099	0.014	0.071	0.145	0.156	
	350	$P(k)$	0.019	0.007	0.034	0.087	0.442	0.066	0.192	0.033	0.071	0.048	0.008
		$\mu(k)$	-0.047	0.402	-0.166	0.043	0.001	-0.068	-0.001	-0.191	0.083	0.068	
		$\sigma(k)$	0.001	0.001	0.049	0.037	0.017	0.010	0.017	0.127	0.061	0.061	

Table A.III.1. *(continued)*

Attack/ Rank		Model parameters										Error
Median	1	$P(k)$	0.064	0.078	0.075	0.122	0.170	0.072	0.091	0.205	0.071	0.050	0.024
		$\mu(k)$	0.014	-0.032	-0.028	-0.011	-0.026	-0.012	-0.101	-0.017	-0.055	0.010	
		$\sigma(k)$	0.056	0.044	0.044	0.025	0.024	0.053	0.032	0.021	0.042	0.006	
	50	$P(k)$	0.097	0.174	0.148	0.086	0.051	0.102	0.115	0.073	0.062	0.093	0.024
		$\mu(k)$	-0.014	-0.006	-0.011	-0.018	-0.009	-0.005	-0.008	-0.006	-0.014	-0.025	
		$\sigma(k)$	0.029	0.004	0.028	0.007	0.002	0.029	0.029	0.029	0.043	0.043	
	100	$P(k)$	0.022	0.157	0.043	0.123	0.068	0.086	0.075	0.061	0.039	0.324	0.013
		$\mu(k)$	0.016	-0.020	0.007	0.006	-0.015	-0.012	0.002	-0.022	0.008	-0.001	
		$\sigma(k)$	0.002	0.006	0.004	0.021	0.033	0.028	0.024	0.031	0.039	0.005	
	150	$P(k)$	0.171	0.127	0.057	0.122	0.094	0.160	0.094	0.024	0.073	0.079	0.016
		$\mu(k)$	0.001	-0.013	0.018	-0.004	-0.007	-0.004	-0.006	-0.081	-0.005	-0.005	
		$\sigma(k)$	0.002	0.007	0.013	0.013	0.023	0.012	0.022	0.014	0.022	0.022	
	200	$P(k)$	0.101	0.199	0.028	0.124	0.044	0.184	0.124	0.038	0.044	0.112	0.010
		$\mu(k)$	-0.001	-0.001	0.005	-0.011	0.002	-0.001	-0.001	-0.006	-0.004	0.001	
		$\sigma(k)$	0.018	0.003	0.033	0.015	0.033	0.007	0.018	0.031	0.0031	0.018	
	250	$P(k)$	0.129	0.059	0.341	0.054	0.044	0.050	0.117	0.034	0.039	0.134	0.005
		$\mu(k)$	-0.005	-0.021	0.005	0.029	0.004	-0.028	-0.005	-0.002	0.007	-0.006	
		$\sigma(k)$	0.007	0.049	0.004	0.006	0.017	0.008	0.007	0.016	0.045	0.007	
	300	$P(k)$	0.021	0.074	0.067	0.220	0.116	0.081	0.150	0.159	0.005	0.108	0.007
		$\mu(k)$	-0.055	0.027	0.006	0.002	-0.007	0.007	-0.004	-0.002	-0.009	-0.005	
		$\sigma(k)$	0.005	0.014	0.010	0.003	0.011	0.007	0.012	0.004	0.010	0.012	
	350	$P(k)$	0.091	0.097	0.039	0.184	0.038	0.365	0.041	0.054	0.062	0.030	0.008
		$\mu(k)$	-0.003	0.019	-0.013	0.004	-0.011	-0.001	-0.006	-0.006	0.011	0.016	
		$\sigma(k)$	0.007	0.05	0.014	0.003	0.015	0.002	0.016	0.016	0.018	0.076	
Sharpening	1	$P(k)$	0.112	0.041	0.074	0.135	0.128	0.128	0.104	0.103	0.098	0.077	0.032
		$\mu(k)$	0.184	0.245	0.020	0.113	0.045	0.088	0.054	0.051	0.019	0.053	
		$\sigma(k)$	0.017	0.012	0.072	0.022	0.038	0.048	0.060	0.063	0.072	0.062	
	50	$P(k)$	0.047	0.068	0.059	0.023	0.081	0.094	0.139	0.051	0.293	0.145	0.029
		$\mu(k)$	0.003	0.027	0.017	-0.027	0.057	0.028	0.096	-0.057	0.044	0.042	
		$\sigma(k)$	0.052	0.067	0.041	0.026	0.067	0.038	0.059	0.090	0.034	0.037	
	100	$P(k)$	0.048	0.071	0.173	0.193	0.054	0.085	0.083	0.187	0.042	0.064	0.021
		$\mu(k)$	0.016	-0.068	0.006	0.037	0.047	0.055	0.056	0.038	0.045	-0.020	
		$\sigma(k)$	0.055	0.040	0.007	0.019	0.069	0.031	0.031	0.020	0.069	0.003	
	150	$P(k)$	0.079	0.039	0.239	0.065	0.082	0.087	0.072	0.112	0.115	0.110	0.013
		$\mu(k)$	-0.054	0.037	0.011	0.029	0.09	0.066	0.030	0.005	0.009	0.005	
		$\sigma(k)$	0.046	0.002	0.016	0.054	0.034	0.014	0.054	0.032	0.035	0.032	
	200	$P(k)$	0.114	0.050	0.082	0.305	0.123	0.082	0.105	0.050	0.050	0.039	0.018
		$\mu(k)$	-0.021	0.079	0.038	0.005	0.028	0.033	0.019	-0.073	-0.047	-0.009	
		$\sigma(k)$	0.006	0.033	0.021	0.007	0.008	0.021	0.015	0.055	0.059	0.050	
	250	$P(k)$	0.040	0.223	0.050	0.153	0.245	0.136	0.072	0.038	0.009	0.033	0.012
		$\mu(k)$	-0.038	0.006	-0.026	0.020	-0.004	0.024	0.062	-0.074	-0.083	-0.001	
		$\sigma(k)$	0.022	0.006	0.027	0.017	0.010	0.016	0.025	0.051	0.002	0.003	
	300	$P(k)$	0.016	0.081	0.140	0.037	0.047	0.086	0.012	0.051	0.491	0.040	0.013
		$\mu(k)$	0.040	-0.060	-0.015	-0.017	-0.007	0.026	0.037	0.015	0.001	0.021	
		$\sigma(k)$	0.017	0.042	0.009	0.060	0.060	0.006	0.001	0.057	0.006	0.006	
	350	$P(k)$	0.022	0.030	0.103	0.083	0.197	0.076	0.032	0.124	0.194	0.150	0.011
		$\mu(k)$	-0.021	-0.096	0.004	0.012	0.001	-0.007	-0.034	0.011	-0.002	0.006	
		$\sigma(k)$	0.001	0.030	0.033	0.034	0.007	0.029	0.047	0.034	0.004	0.005	

Table A.III.1. *(continued)*

Attack/ Rank			Model parameters										Error
StirMark	1	*P(k)*	0.070	0.377	0.084	0.079	0.087	0.026	0.084	0.085	0.078	0.027	0.018
		μ(k)	-0.108	-0.028	-0.037	-0.033	-0.024	-0.509	-0.338	-0.114	-0.049	-0.172	
		σ(k)	0.073	0.047	0.053	0.119	0.118	0.255	0.127	0.073	0.122	0.002	
	50	*P(k)*	0.082	0.140	0.096	0.019	0.096	0.045	0.104	0.094	0.272	0.051	0.016
		μ(k)	-0.028	-0.048	-0.011	0.095	-0.040	-0.029	-0.033	-0.036	-0.009	0.035	
		σ(k)	0.069	0.070	0.021	0.001	0.062	0.180	0.070	0.070	0.021	0.178	
	100	*P(k)*	0.174	0.071	0.038	0.073	0.251	0.095	0.104	0.017	0.158	0.019	0.014
		μ(k)	-0.004	-0.008	-0.078	-0.038	-0.001	-0.060	-0.011	-0.056	-0.025	0.039	
		σ(k)	0.008	0.078	0.271	0.071	0.031	0.066	0.076	0.271	0.020	0.002	
	150	*P(k)*	0.123	0.119	0.036	0.189	0.033	0.011	0.176	0.211	0.007	0.095	0.009
		μ(k)	-0.010	-0.020	-0.100	0.001	0.092	-0.196	-0.017	0.003	0.010	-0.019	
		σ(k)	0.039	0.042	0.030	0.010	0.018	0.056	0.013	0.032	0.001	0.042	
	200	*P(k)*	0.299	0.040	0.039	0.056	0.127	0.075	0.053	0.205	0.039	0.065	0.005
		μ(k)	0.001	0.006	-0.008	0.051	-0.039	0.015	0.003	-0.011	-0.001	0.012	
		σ(k)	0.005	0.040	0.044	0.024	0.019	0.003	0.122	0.010	0.042	0.124	
	250	*P(k)*	0.226	0.100	0.129	0.099	0.107	0.098	0.037	0.052	0.052	0.098	0.011
		μ(k)	-0.004	-0.018	0.003	0.005	-0.006	0.004	0.039	-0.118	0.005	-0.002	
		σ(k)	0.003	0.015	0.014	0.032	0.028	0.032	0.079	0.096	0.033	0.030	
	300	*P(k)*	0.280	0.027	0.013	0.067	0.049	0.219	0.008	0.024	0.216	0.097	0.008
		μ(k)	0.001	0.011	-0.167	-0.061	0.046	-0.001	-0.096	0.116	-0.003	-0.006	
		σ(k)	0.004	0.022	0.024	0.022	0.011	0.011	0.058	0.038	0.012	0.013	
	350	*P(k)*	0.008	0.089	0.234	0.211	0.017	0.049	0.161	0.042	0.143	0.046	0.006
		μ(k)	0.145	-0.005	-0.001	0.002	-0.023	-0.054	0.013	0.067	-0.011	-0.042	
		σ(k)	0.020	0.014	0.001	0.006	0.024	0.018	0.009	0.015	0.013	0.022	

Table A.III.2. IVM for (9,7) DWT coefficients hierarchy, high quality video and different attacks.

Attack/ Rank			Model parameters										Error
FMLR	1	*P(k)*	0.092	0.127	0.060	0.199	0.077	0.080	0.075	0.080	0.138	0.072	0.033
		μ(k)	0.116	-0.108	-0.116	-0.085	-0.110	-0.140	-0.039	-0.084	-0.110	-0.132	
		σ(k)	0.036	0.034	0.060	0.024	0.036	0.056	0.059	0.024	0.035	0.056	
	50	*P(k)*	0.109	0.100	0.092	0.096	0.109	0.081	0.057	0.130	0.111	0.116	0.032
		μ(k)	-0.067	-0.013	-0.098	-0.075	-0.051	-0.061	-0.050	-0.055	-0.054	-0.039	
		σ(k)	0.025	0.031	0.041	0.022	0.026	0.042	0.037	0.027	0.027	0.017	
	100	*P(k)*	0.057	0.095	0.136	0.145	0.075	0.051	0.151	0.076	0.063	0.150	0.030
		μ(k)	-0.081	-0.056	-0.037	-0.052	-0.047	-0.079	-0.038	-0.050	-0.050	-0.033	
		σ(k)	0.045	0.029	0.028	0.028	0.032	0.045	0.026	0.032	0.034	0.023	
	150	*P(k)*	0.034	0.102	0.141	0.083	0.120	0.085	0.118	0.037	0.164	0.116	0.037
		μ(k)	-0.092	-0.51	-0.024	-0.062	-0.045	-0.040	-0.024	-0.119	-0.034	-0.045	
		σ(k)	0.036	0.019	0.024	0.019	0.025	0.027	0.024	0.028	0.026	0.025	
	200	*P(k)*	0.057	0.071	0.041	0.105	0.107	0.194	0.075	0.007	0.151	0.191	0.032
		μ(k)	-0.052	-0.065	-0.060	-0.069	-0.045	-0.031	-0.044	0.021	-0.025	-0.034	
		σ(k)	0.020	0.034	0.034	0.034	0.024	0.024	0.024	0.001	0.021	0.025	
	250	*P(k)*	0.214	0.086	0.187	0.087	0.021	0.253	0.069	0.022	0.028	0.035	0.026
		μ(k)	-0.019	-0.096	-0.034	-0.039	-0.065	-0.07	-0.037	-0.052	-0.052	-0.049	
		σ(k)	0.021	0.020	0.022	0.021	0.002	0.021	0.021	0.028	0.028	0.027	
	300	*P(k)*	0.133	0.099	0.068	0.062	0.129	0.152	0.044	0.078	0.189	0.046	0.034
		μ(k)	-0.038	-0.036	-0.034	-0.033	-0.033	-0.030	-0.035	0.001	-0.037	-0.105	
		σ(k)	0.023	0.024	0.026	0.026	0.024	0.023	0.028	0.031	0.022	0.017	
	350	*P(k)*	0.054	0.052	0.052	0.067	0.094	0.163	0.149	0.193	0.085	0.089	0.027
		μ(k)	-0.038	-0.021	-0.043	-0.019	-0.023	-0.020	-0.038	-0.033	-0.051	-0.049	
		σ(k)	0.033	0.028	0.033	0.028	0.029	0.014	0.018	0.018	0.032	0.033	

Table A.III.2. *(continued)*

Attack/ Rank		Model parameters											Error
Gaussian filtering	1	$P(k)$	0.022	0.029	0.155	0.099	0.039	0.068	0.077	0.104	0.177	0.227	0.030
		$\mu(k)$	-0.033	-0.184	-0.077	-0.131	-0.107	-0.080	-0.089	-0.094	-0.050	-0.067	
		$\sigma(k)$	0.001	0.007	0.017	0.027	0.033	0.014	0.026	0.030	0.014	0.020	
	50	$P(k)$	0.081	0.119	0.066	0.072	0.025	0.247	0.209	0.051	0.040	0.089	0.026
		$\mu(k)$	-0.049	-0.036	-0.037	-0.038	-0.009	-0.028	-0.021	-0.040	-0.010	-0.046	
		$\sigma(k)$	0.016	0.007	0.021	0.021	0.001	0.010	0.005	0.021	0.001	0.003	
	100	$P(k)$	0.103	0.049	0.052	0.046	0.047	0.041	0.049	0.172	0.081	0.359	0.025
		$\mu(k)$	-0.040	-0.027	-0.038	-0.029	-0.030	-0.032	-0.028	-0.009	-0.029	-0.021	
		$\sigma(k)$	0.016	0.020	0.017	0.020	0.020	0.019	0.020	0.004	0.008	0.007	
	150	$P(k)$	0.028	0.146	0.131	0.124	0.310	0.033	0.042	0.062	0.045	0.078	0.023
		$\mu(k)$	0.001	-0.017	-0.029	-0.024	-0.012	-0.008	-0.022	-0.050	-0.023	-0.025	
		$\sigma(k)$	0.016	0.010	0.008	0.011	0.006	0.017	0.011	0.005	0.011	0.010	
	200	$P(k)$	0.058	0.059	0.080	0.007	0.175	0.054	0.218	0.101	0.054	0.194	0.021
		$\mu(k)$	-0.027	-0.035	-0.009	.0063	-0.010	-0.020	-0.021	-0.009	-0.031	-0.011	
		$\sigma(k)$	0.013	0.010	0.014	0.001	0.008	0.014	0.005	0.002	0.012	0.008	
	250	$P(k)$	0.144	0.055	0.134	0.033	0.102	0.054	0.218	0.007	0.198	0.055	0.020
		$\mu(k)$	-0.030	-0.017	-0.014	-0.007	-0.014	-0.017	-0.006	0.068	-0.015	-0.016	
		$\sigma(k)$	0.009	0.012	0.012	0.011	0.006	0.012	0.004	0.001	0.006	0.012	
	300	$P(k)$	0.032	0.046	0.057	0.051	0.280	0.198	0.014	0.039	0.034	0.248	0.019
		$\mu(k)$	-0.032	-0.026	-0.019	-0.021	-0.010	-0.011	-0-49	-0.028	-0.026	-0.009	
		$\sigma(k)$	0.012	0.012	0.009	0.009	0.007	0.007	0.018	0.012	0.007	0.007	
	350	$P(k)$	0.279	0.055	0.021	0.047	0.064	0.256	0.0639	0.007	0.098	0.108	0.017
		$\mu(k)$	-0.005	-0.036	-0.019	-0.016	-0.016	-0.012	-0.009	0.048	-0.024	-0.009	
		$\sigma(k)$	0.002	0.011	0.001	0.008	0.008	0.004	0.010	0.001	0.005	0.010	
JPEG	1	$P(k)$	0.031	0.032	0.040	0.419	0.202	0.007	0.142	0.028	0.065	0.034	0.003
		$\mu(k)$	-0.026	-0.028	-0.033	-0.004	0.008	-0.099	-0.012	-0.024	-0.033	-0.026	
		$\sigma(k)$	0.015	0.015	0.014	0.004	0.006	0.001	0.007	0.015	0.014	0.015	
	50	$P(k)$	0.131	0.092	0.075	0.054	0.009	0.071	0.374	0.032	0.007	0.063	0.004
		$\mu(k)$	0.007	-0.009	-0.002	-0.009	-0.013	0.005	0.001	-0.015	-0.075	-0.016	
		$\sigma(k)$	0.008	0.003	0.019	0.017	0.016	0.017	0.004	0.001	0.001	0.015	
	100	$P(k)$	0.049	0.142	0.046	0.069	0.041	0.044	0.211	0.048	0.051	0.041	0.011
		$\mu(k)$	-0.009	-0.011	-0.014	-0.009	-0.010	-0.012	0.005	0.004	-0.001	-0.003	
		$\sigma(k)$	0.025	0.009	0.025	0.009	0.025	0.025	0.004	0.012	0.012	0.002	
	150	$P(k)$	0.147	0.139	0.044	0.029	0.069	0.019	0.022	0.058	0.366	0.105	0.003
		$\mu(k)$	-0.005	-0.004	0.015	-0.020	-0.003	-0.025	-0.051	-0.008	-0.001	-0.001	
		$\sigma(k)$	0.011	0.011	0.008	0.014	0.011	0.012	0.007	0.012	0.004	0.009	
	200	$P(k)$	0.024	0.022	0.079	0.197	0.129	0.198	0.015	0.262	0.016	0.058	0.003
		$\mu(k)$	-0.033	-0.013	-0.03	0.006	0.009	-0.001	-0.022	-0.009	-0.026	-0.001	
		$\sigma(k)$	0.011	0.018	0.008	0.005	0.011	0.0003	0.016	0.007	0.015	0.008	
	250	$P(k)$	0.214	0.224	0.114	0.098	0.027	0.078	0.038	0.047	0.077	0.082	0.005
		$\mu(k)$	-0.001	-0.002	0.001	-0.001	-0.041	-0.003	-0.008	-0.002	-0.002	-0.003	
		$\sigma(k)$	0.005	0.005	0.012	0.012	0.009	0.012	0.013	0.012	0.012	0.012	
	300	$P(k)$	0.014	0.024	0.075	0.129	0.093	0.182	0.139	0.028	0.178	0.138	0.014
		$\mu(k)$	-0.018	-0.044	-0.003	0.001	-0.003	-0.001	-0.001	-0.023	-0.001	-0.001	
		$\sigma(k)$	0.014	0.007	0.009	0.002	0.009	0.008	0.009	0.002	0.009	0.009	
	350	$P(k)$	0.049	0.195	0.099	0.029	0.080	0.138	0.040	0.088	0.238	0.044	0.006
		$\mu(k)$	-0.009	-0.004	0.011	-0.026	0.003	0.003	-0.022	-0.05	-0.002	-0.009	
		$\sigma(k)$	0.011	0.005	0.008	0.018	0.009	0.003	0.006	0.011	0.005	0.011	

Table A.III.2. *(continued)*

Attack/ Rank			Model parameters										Error
Rotation +2°	1	$P(k)$	0.119	0.096	0.077	0.107	0.110	0.095	0.132	0.098	0.080	0.093	0.027
		$\mu(k)$	-0.851	-0.541	-0.158	-0.754	-0.682	-0.747	-0.509	-0.702	-1.149	-0.943	
		$\sigma(k)$	0.263	0.032	0.100	0.259	0.239	0.258	0.175	0.152	0.372	0.123	
	50	$P(k)$	0.021	0.158	0.154	0.026	0.136	0.117	0.007	0.119	0.125	0.136	0.017
		$\mu(k)$	-0.727	-0.277	-0.186	-0.569	-0.183	-0.324	-0.741	-0.367	-0.508	-0.192	
		$\sigma(k)$	0.001	0.118	0.090	0.273	0.168	0.026	0.001	0.084	0.085	0.168	
	100	$P(k)$	0.108	0.099	0.098	0.039	0.026	0.147	0.163	0.080	0.192	0.046	0.019
		$\mu(k)$	-0.358	-0.341	-0.177	0.119	-0.645	-0.069	-0.024	-0.218	-0.269	-0.178	
		$\sigma(k)$	0.163	0.166	0.111	0.048	0.232	0.050	0.079	0.025	0.068	0.113	
	150	$P(k)$	0.033	0.047	0.032	0.108	0.139	0.111	0.142	0.186	0.105	0.095	0.016
		$\mu(k)$	-0.652	-0.241	-0.471	-0.059	-0.129	-0.165	-0.143	-0.192	-0.216	-0.324	
		$\sigma(k)$	0.248	0.167	0.049	0.120	0.086	0.095	0.090	0.035	0.077	0.022	
	200	$P(k)$	0.082	0.104	0.084	0.083	0.075	0.076	0.115	0.152	0.129	0.099	0.018
		$\mu(k)$	-0.196	-0.147	-0.016	-0.132	-0.284	-0.377	-0.216	-0.132	-0.137	-0.197	
		$\sigma(k)$	0.244	0.056	0.029	0.054	0.023	0.030	0.018	0.051	0.055	0.244	
	250	$P(k)$	0.103	0.128	0.221	0.007	0.013	0.117	0.124	0.088	0.050	0.149	0.017
		$\mu(k)$	0.141	-0.082	-0.218	-0.899	0.033	-0.082	-0.077	-0.080	-0.473	-0.187	
		$\sigma(k)$	0.023	0.066	0.065	0.001	0.001	0.069	0.056	0.123	0.070	0.069	
	300	$P(k)$	0.086	0.151	0.188	0.053	0.140	0.064	0.076	0.007	0.026	0.208	0.015
		$\mu(k)$	-0.286	-0.093	-0.069	-0.258	-0.157	-0.215	-0.75	-1.145	0.136	-0.149	
		$\sigma(k)$	0.138	0.078	0.068	0.139	0.030	0.126	0.014	0.001	0.034	0.082	
	350	$P(k)$	0.082	0.088	0.161	0.061	0.054	0.343	0.007	0.060	0.067	0.076	0.015
		$\mu(k)$	-0.108	-0.169	-0.115	-0.092	-0.039	-0.084	-1.006	-0.009	-0.104	-0.201	
		$\sigma(k)$	0.064	0.057	0.064	0.146	0.060	0.058	0.001	0.144	0.142	0.069	
Rotation -2°	1	$P(k)$	0.144	0.134	0.161	0.034	0.035	0.101	0.046	0.144	0.079	0.123	0.024
		$\mu(k)$	-0.673	-0.750	-0.845	-1.101	-0.055	-0.670	-1.649	-0.627	-0.701	-0.507	
		$\sigma(k)$	0.255	0.258	0.231	0.339	0.088	0.255	0.220	0.244	0.259	0.174	
	50	$P(k)$	0.156	0.043	0.084	0.029	0.193	0.071	0.155	0.007	0.165	0.097	0.019
		$\mu(k)$	-0.317	-0.728	-0.058	-0.531	-0.289	-0.326	-0.318	-0.641	-0.256	-0.306	
		$\sigma(k)$	0.133	0.162	0.093	0.215	0.134	0.133	0.133	0.001	0.130	0.031	
	100	$P(k)$	0.017	0.108	0.215	0.089	0.025	0.013	0.157	0.138	0.130	0.110	0.018
		$\mu(k)$	-0.157	-0.298	-0.242	-0.282	-0.599	-0.228	-0.148	-0.178	-0.115	-0.411	
		$\sigma(k)$	0.011	0.147	0.055	0.148	0.237	0.001	0.117	0.122	0.114	0.119	
	150	$P(k)$	0.096	0.022	0.199	0.154	0.053	0.166	0.061	0.079	0.027	0.143	0.016
		$\mu(k)$	-0.227	-0.571	-0.178	-0.141	-0.319	-0.153	-0.261	-0.315	-0.263	-0.119	
		$\sigma(k)$	0.142	0.092	0.091	0.119	0.155	0.029	0.154	0.155	0.003	0.119	
	200	$P(k)$	0.128	0.128	0.081	0.064	0.081	0.074	0.119	0.126	0.071	0.127	0.022
		$\mu(k)$	-0.131	-0.192	-0.095	-0.405	-0.111	-0.054	-0.147	-0.156	-0.130	-0.269	
		$\sigma(k)$	0.069	0.046	0.179	0.230	0.182	0.057	0.073	0.071	0.183	0.108	
	250	$P(k)$	0.106	0.103	0.093	0.052	0.011	0.115	0.069	0.240	0.126	0.082	0.013
		$\mu(k)$	-0.244	-0.133	-0.063	-0.086	0.077	-0.243	-0.093	-0.174	-0.114	-0.095	
		$\sigma(k)$	0.088	0.314	0.015	0.130	0.009	0.088	0.072	0.034	0.070	0.072	
	300	$P(k)$	0.078	0.351	0.080	0.053	0.026	0.093	0.091	0.157	0.025	0.046	0.017
		$\mu(k)$	-0.021	-0.107	-0.086	-0.188	-0.078	-0.068	-0.282	-0.144	0.006	-0.328	
		$\sigma(k)$	0.090	0.046	0.102	0.004	0.234	0.100	0.065	0.062	0.224	0.173	
	350	$P(k)$	0.079	0.101	0.060	0.116	0.171	0.059	0.082	0.064	0.077	0.192	0.011
		$\mu(k)$	-0.049	-0.133	-0.179	-0.086	-0.141	-0.160	-0.022	-0.251	-0.192	-0.073	
		$\sigma(k)$	0.122	0.072	0.022	0.050	0.074	0.149	0.117	0.139	0.149	0.045	

Table A.III.2. *(continued)*

Attack/ Rank			Model parameters										Error
Rotation +5°	1	$P(k)$	0.087	0.082	0.087	0.090	0.174	0.101	0.129	0.155	0.075	0.020	0.019
		$\mu(k)$	-1.080	-0.812	-0.646	-0.678	-0.625	-0.651	-0.698	-0.678	-0.643	-0.047	
		$\sigma(k)$	0.235	0.279	0.190	0.191	0.186	0.189	0.178	0.182	0.192	0.117	
	50	$P(k)$	0.302	0.094	0.081	0.092	0.139	0.098	0.018	0.030	0.046	0.099	0.018
		$\mu(k)$	-0.315	-0.279	-0.336	-0.196	-0.168	-0.429	-0.408	0.015	-0.068	-0.340	
		$\sigma(k)$	0.039	0.126	0.139	0.027	0.087	0.117	0.001	0.121	0.216	0.138	
	100	$P(k)$	0.167	0.139	0.117	0.153	0.121	0.040	0.059	0.089	0.091	0.023	0.018
		$\mu(k)$	-0.233	-0.134	-0.268	-0.308	-0.225	-0.207	-0.470	-0.124	-0.141	-0.675	
		$\sigma(k)$	0.031	0.034	0.080	0.065	0.086	0.002	0.049	0.138	0.141	0.151	
	150	$P(k)$	0.171	0.158	0.155	0.042	0.085	0.192	0.053	0.029	0.048	0.068	0.014
		$\mu(k)$	-0.129	-0.239	-0.206	-0.381	-0.239	-0.169	-0.211	-0.353	-0.403	-0.026	
		$\sigma(k)$	0.052	0.049	0.055	0.181	0.126	0.057	0.128	0.004	0.181	0.093	
	200	$P(k)$	0.007	0.045	0.142	0.116	0.094	0.187	0.131	0.069	0.096	0.113	0.012
		$\mu(k)$	-0.275	-0.282	-0.139	-0.139	-0.078	-0.273	-0.151	-0.136	-0.218	-0.207	
		$\sigma(k)$	0.001	0.135	0.055	0.129	0.149	0.057	0.054	0.156	0.070	0.063	
	250	$P(k)$	0.066	0.065	0.070	0.169	0.116	0.109	0.108	0.192	0.042	0.063	0.012
		$\mu(k)$	-0.053	-0.133	-0.128	-0.163	-0.090	-0.101	-0.308	-0.157	-0.193	-0.143	
		$\sigma(k)$	0.039	0.121	0.120	0.040	0.049	0.050	0.078	0.041	0.382	0.122	
	300	$P(k)$	0.070	0.085	0.148	0.089	0.129	0.040	0.233	0.039	0.121	0.045	0.011
		$\mu(k)$	-0.332	-0.185	-0.169	-0.172	-0.086	-0.185	-0.100	0.067	-0.081	-0.133	
		$\sigma(k)$	0.152	0.093	0.013	0.084	0.069	0.174	0.017	0.189	0.067	0.002	
	350	$P(k)$	0.095	0.191	0.020	0.062	0.012	0.109	0.187	0.109	0.116	0.097	0.013
		$\mu(k)$	-0.136	-0.109	0.0229	-0.234	0.116	-0.052	-0.112	-0.129	-0.064	-0.258	
		$\sigma(k)$	0.072	0.038	0.233	0.035	0.008	0.060	0.037	0.067	0.059	0.147	
Median	1	$P(k)$	0.121	0.042	0.009	0.103	0.427	0.013	0.118	0.099	0.029	0.038	0.016
		$\mu(k)$	-0.003	-0.829	-0.528	-0.071	-0.011	-0.517	-0.043	-0.070	0.007	-0.158	
		$\sigma(k)$	0.004	0.129	0.077	0.045	0.028	0.078	0.041	0.045	0.109	0.070	
	50	$P(k)$	0.457	0.021	0.089	0.007	0.048	0.109	0.088	0.037	0.082	0.061	0.011
		$\mu(k)$	-0.008	-0.418	-0.075	0.134	0.002	-0.012	-0.027	-0.261	-0.001	-0.028	
		$\sigma(k)$	0.015	0.031	0.018	0.001	0.018	0.036	0.032	0.052	0.035	0.032	
	100	$P(k)$	0.108	0.045	0.550	0.040	0.042	0.030	0.007	0.153	0.015	0.009	0.007
		$\mu(k)$	0.035	-0.276	-0.003	-0.030	-0.105	-0.072	-0.071	-0.034	-0.084	-0.079	
		$\sigma(k)$	0.027	0.060	0.011	0.039	0.033	0.037	0.001	0.012	0.037	0.001	
	150	$P(k)$	0.090	0.022	0.074	0.224	0.070	0.248	0.073	0.092	0.024	0.084	0.009
		$\mu(k)$	0.010	-0.057	-0.029	-0.012	-0.194	-0.010	-0.036	0.008	0.038	-0.013	
		$\sigma(k)$	0.023	0.026	0.037	0.015	0.087	0.014	0.036	0.025	0.109	0.035	
	200	$P(k)$	0.023	0.392	0.165	0.100	0.030	0.095	0.070	0.033	0.022	0.070	0.005
		$\mu(k)$	-0.206	-0.001	-0.022	0.009	-0.130	0.008	-0.046	-0.135	-0.025	0.016	
		$\sigma(k)$	0.071	0.006	0.005	0.022	0.045	0.023	0.006	0.044	0.031	0.021	
	250	$P(k)$	0.216	0.013	0.076	0.007	0.064	0.159	0.045	0.117	0.233	0.069	0.006
		$\mu(k)$	-0.005	-0.103	-0.008	-0.455	-0.011	-0.001	-0.187	0.002	-0.007	-0.041	
		$\sigma(k)$	0.019	0.001	0.037	0.001	0.037	0.001	0.051	0.022	0.018	0.029	
	300	$P(k)$	0.081	0.058	0.167	0.087	0.091	0.200	0.169	0.060	0.047	0.041	0.004
		$\mu(k)$	-0.035	0.012	-0.009	0.001	-0.001	-0.003	-0.005	-0.065	0.019	-0.084	
		$\sigma(k)$	0.007	0.024	0.016	0.018	0.007	0.005	0.017	0.126	0.024	0.052	
	350	$P(k)$	0.077	0.036	0.057	0.075	0.201	0.076	0.007	0.161	0.213	0.097	0.004
		$\mu(k)$	0.0010	-0.173	-0.056	-0.023	0.007	0.004	-0.273	-0.03	-0.009	-0.004	
		$\sigma(k)$	0.029	0.022	0.018	0.024	0.004	0.031	0.001	0.007	0.006	0.032	

Table A.III.2. *(continued)*

Attack/ Rank			Model parameters										Error
Sharpening	1	$P(k)$	0.069	0.129	0.136	0.078	0.159	0.020	0.133	0.116	0.076	0.084	0.026
		$\mu(k)$	0.204	0.085	0.080	0.201	0.101	0.005	0.093	0.081	0.071	0.104	
		$\sigma(k)$	0.110	0.069	0.067	0.052	0.045	0.003	0.068	0.069	0.067	0.072	
	50	$P(k)$	0.054	0.194	0.196	0.053	0.157	0.076	0.054	0.078	0.062	0.076	0.030
		$\mu(k)$	-0.053	0.062	0.042	0.162	0.107	0.054	0.059	0.155	0.087	-0.004	
		$\sigma(k)$	0.040	0.020	0.020	0.067	0.019	0.061	0.064	0.040	0.072	0.042	
	100	$P(k)$	0.007	0.124	0.151	0.121	0.083	0.128	0.098	0.047	0.048	0.195	0.029
		$\mu(k)$	-0.085	0.032	0.052	0.123	0.013	0.026	0.097	0.015	0.025	0.073	
		$\sigma(k)$	0.001	0.053	0.08	0.042	0.060	0.059	0.047	0.006	0.002	0.023	
	150	$P(k)$	0.047	0.073	0.046	0.120	0.130	0.224	0.080	0.095	0.103	0.081	0.026
		$\mu(k)$	0.002	0.060	0.097	0.061	0.081	0.040	0.038	0.065	0.009	0.062	
		$\sigma(k)$	0.002	0.004	0.003	0.066	0.063	0.027	0.003	0.066	0.054	0.033	
	200	$P(k)$	0.183	0.122	0.039	0.093	0.025	0.050	0.024	0.225	0.206	0.031	0.019
		$\mu(k)$	0.039	0.078	0.098	0.071	-0.037	0.119	-0.076	0.014	0.055	0.079	
		$\sigma(k)$	0.030	0.041	0.081	0.042	0.001	0.078	0.044	0.016	0.021	0.081	
	250	$P(k)$	0.084	0.181	0.077	0.177	0.096	0.078	0.075	0.073	0.084	0.074	0.023
		$\mu(k)$	0.003	0.052	0.051	0.022	0.042	0.060	0.038	0.072	0.033	0.055	
		$\sigma(k)$	0.048	0.023	0.072	0.033	0.055	0.048	0.023	0.056	0.012	0.033	
	300	$P(k)$	0.059	0.091	0.024	0.233	0.092	0.194	0.065	0.088	0.095	0.060	0.015
		$\mu(k)$	0.062	-0.014	0.057	0.038	0.016	0.015	0.080	0.086	0.056	0.019	
		$\sigma(k)$	0.030	0.039	0.001	0.014	0.045	0.009	0.056	0.023	0.029	0.046	
	350	$P(k)$	0.094	0.182	0.007	0.085	0.073	0.056	0.275	0.084	0.047	0.098	0.021
		$\mu(k)$	0.060	-0.004	-0.165	0.056	0.049	0.067	0.033	0.053	0.042	0.010	
		$\sigma(k)$	0.042	0.029	0.001	0.043	0.043	0.006	0.014	0.043	0.042	0.012	
StirMark	1	$P(k)$	0.100	0.085	0.084	0.089	0.226	0.084	0.062	0.092	0.092	0.086	0.040
		$\mu(k)$	-0.956	-0.382	-0.574	-0.730	-0.047	-0.713	-0.222	-0.335	-0.354	-0.865	
		$\sigma(k)$	0.355	0.219	0.334	0.377	0.064	0.375	0.106	0.292	0.216	0.369	
	50	$P(k)$	0.074	0.108	0.063	0.143	0.081	0.102	0.091	0.081	0.192	0.064	0.023
		$\mu(k)$	-0.165	-0.109	-0.306	-0.098	-0.176	-0.453	-0.586	-0.263	-0.043	-0.143	
		$\sigma(k)$	0.218	0.207	0.044	0.068	0.219	0.122	0.166	0.229	0.054	0.099	
	100	$P(k)$	0.040	0.073	0.096	0.100	0.234	0.085	0.074	0.139	0.120	0.040	0.021
		$\mu(k)$	-0.654	-0.350	-0.437	-0.057	-0.075	-0.025	-0.244	-0.221	-0.057	0.045	
		$\sigma(k)$	0.202	0.131	0.092	0.106	0.102	0.009	0.159	0.063	0.106	0.303	
	150	$P(k)$	0.037	0.099	0.112	0.117	0.412	0.080	0.043	0.044	0.048	0.007	0.021
		$\mu(k)$	-0.043	-0.169	-0.322	-0.187	-0.021	-0.302	-0.374	-0.070	-0.068	-0.204	
		$\sigma(k)$	0.178	0.058	0.101	0.048	0.042	0.108	0.247	0.183	0.183	0.001	
	200	$P(k)$	0.080	0.095	0.211	0.042	0.093	0.091	0.144	0.094	0.088	0.061	0.018
		$\mu(k)$	-0.075	-0.190	-0.067	0.146	-0.127	-0.081	-0.139	0.006	-0.318	-0.085	
		$\sigma(k)$	0.133	0.150	0.047	0.228	0.147	0.132	0.094	0.015	0.127	0.137	
	250	$P(k)$	0.021	0.059	0.096	0.223	0.160	0.106	0.007	0.081	0.098	0.148	0.012
		$\mu(k)$	-0.467	-0.355	-0.168	-0.019	-0.079	-0.084	-0.937	-0.023	-0.244	-0.009	
		$\sigma(k)$	0.015	0.026	0.12	0.025	0.048	0.105	0.001	0.120	0.0282	0.119	
	300	$P(k)$	0.013	0.064	0.132	0.080	0.109	0.093	0.013	0.353	0.093	0.049	0.015
		$\mu(k)$	-0.151	-0.329	-0.126	-0.218	-0.096	-0.135	-0.160	-0.020	-0.104	0.127	
		$\sigma(k)$	0.001	0.026	0.081	0.126	0.090	0.078	0.001	0.040	0.089	0.048	
	350	$P(k)$	0.014	0.177	0.085	0.071	0.071	0.033	0.084	0.127	0.224	0.115	0.021
		$\mu(k)$	-0.573	0.002	0.075	-0.092	-0.099	-0.117	-0.112	-0.228	-0.055	-0.119	
		$\sigma(k)$	0.020	0.019	0.111	0.094	0.103	0.004	0.099	0.078	0.029	0.059	

A.III.1.2. Model Validation

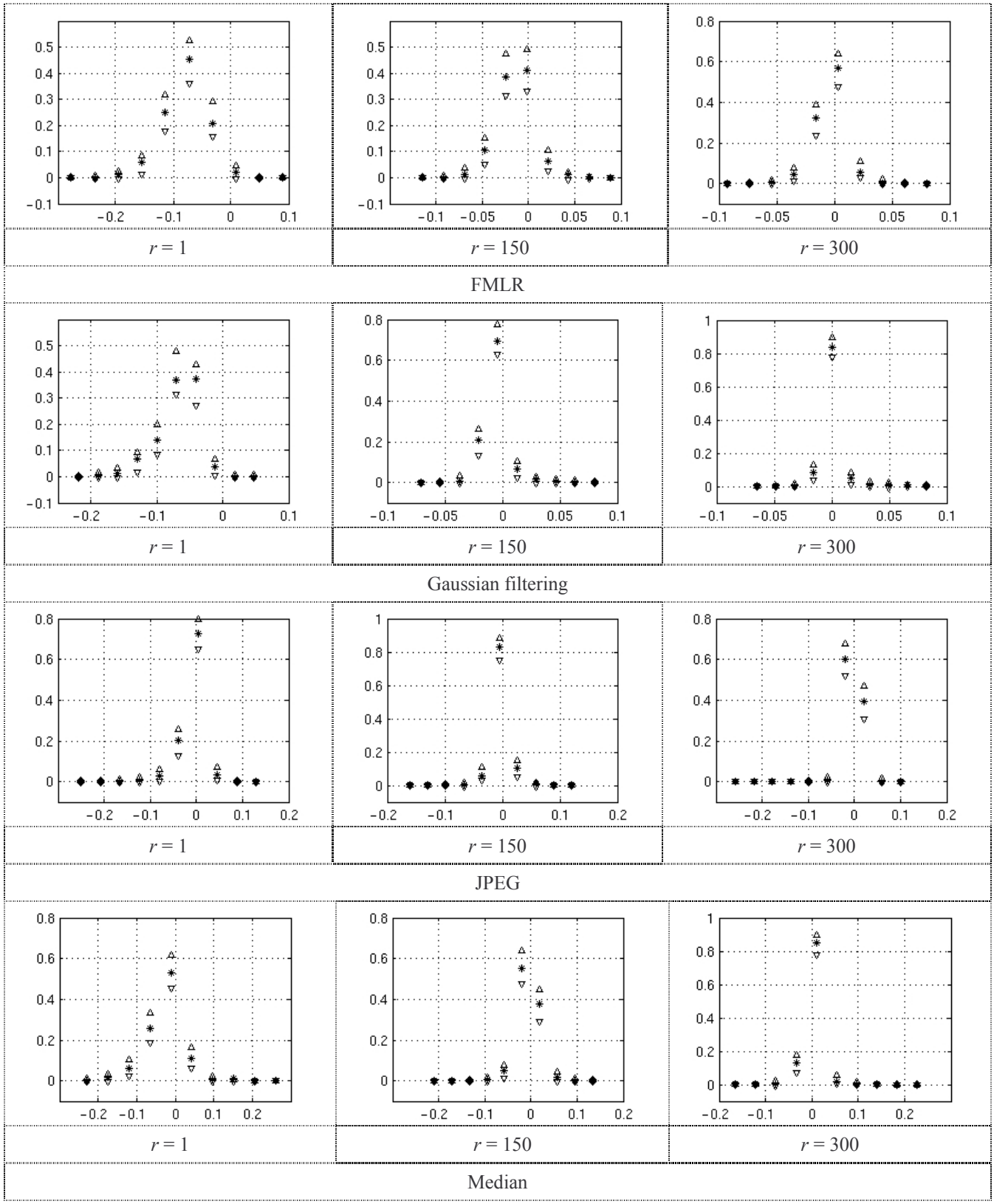


Figure A.III.1.a. Model validation for (9,7) DWT, for three ranks, low quality video, and for the FMLR, the Gaussian filtering, the JPEG and the median attacks: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

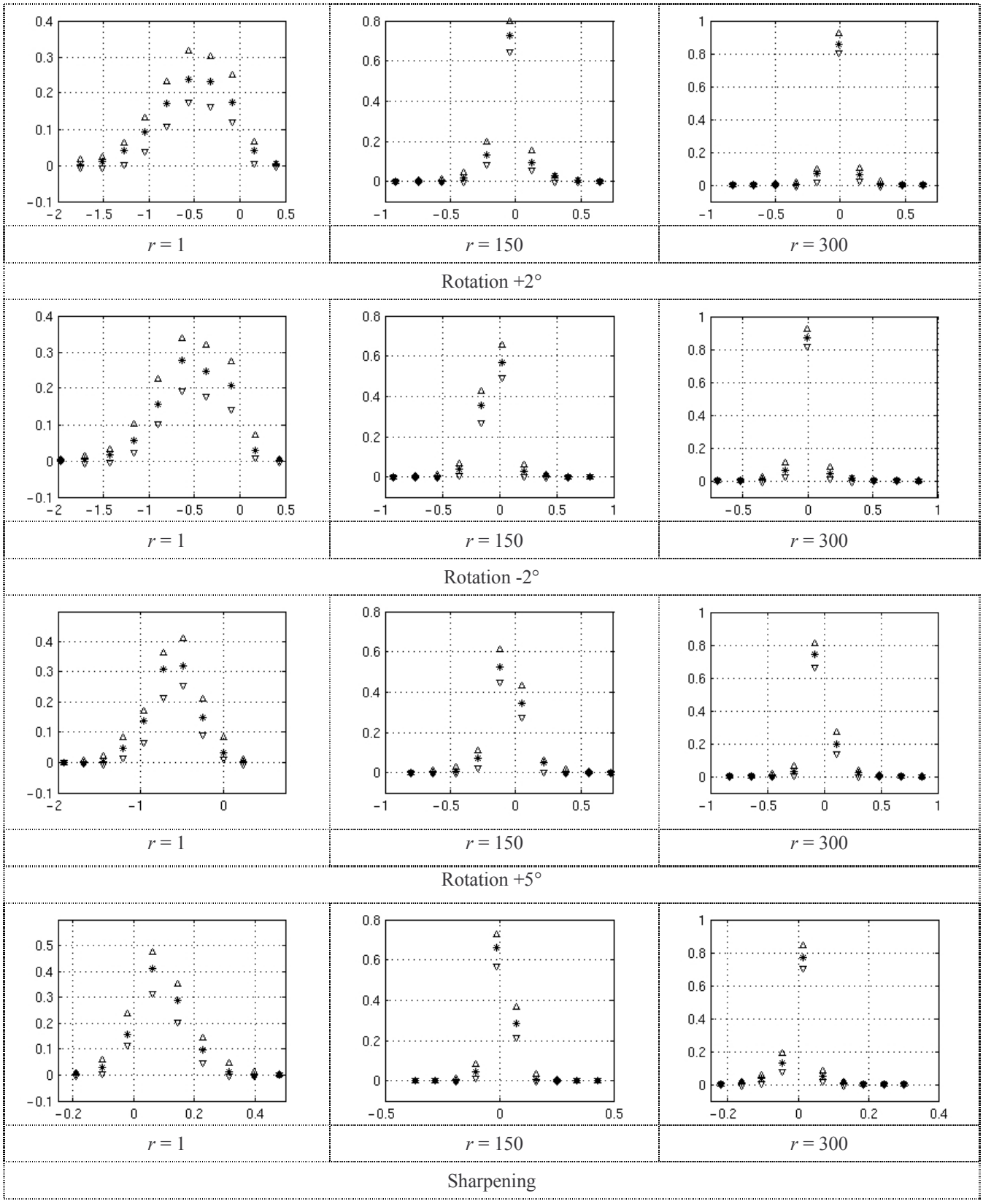


Figure A.III.1.b. Model validation for (9,7) DWT, for three ranks, low quality video, and for the median, the rotation $\pm 2°$, $+5°$, the sharpening attacks: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

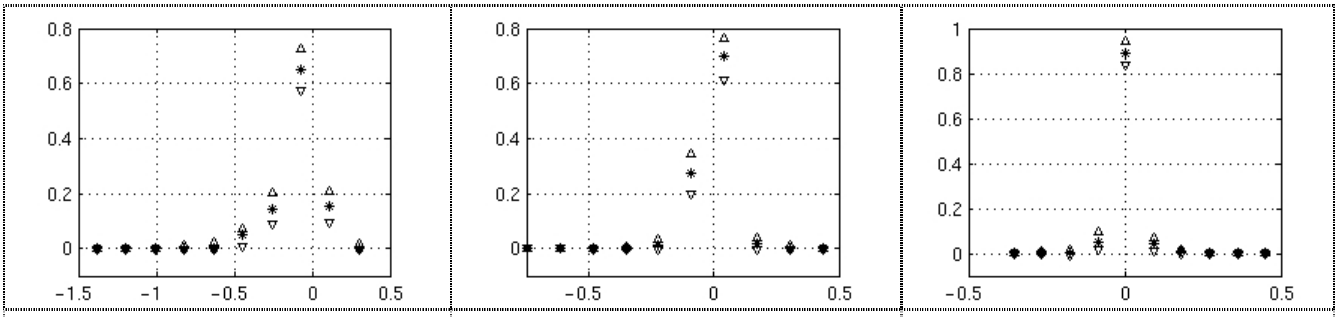


Figure A.III.1.c. Model validation for (9,7) DWT, for three ranks, low quality video, and for the StirMark attack: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

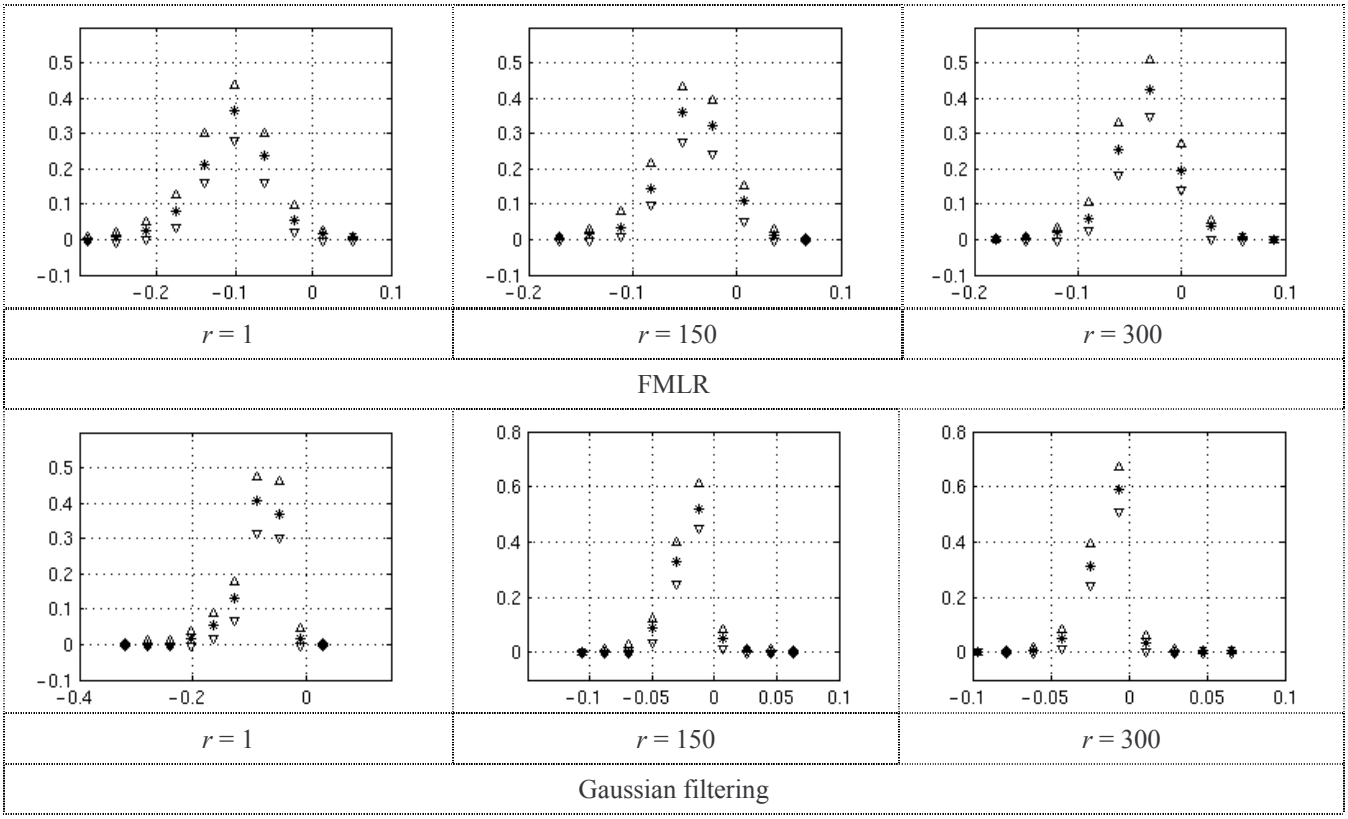


Figure A.III.2.a. Model validation for (9,7) DWT, for three ranks, high quality video, and for the FMLR, and the Gaussian filtering attacks: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

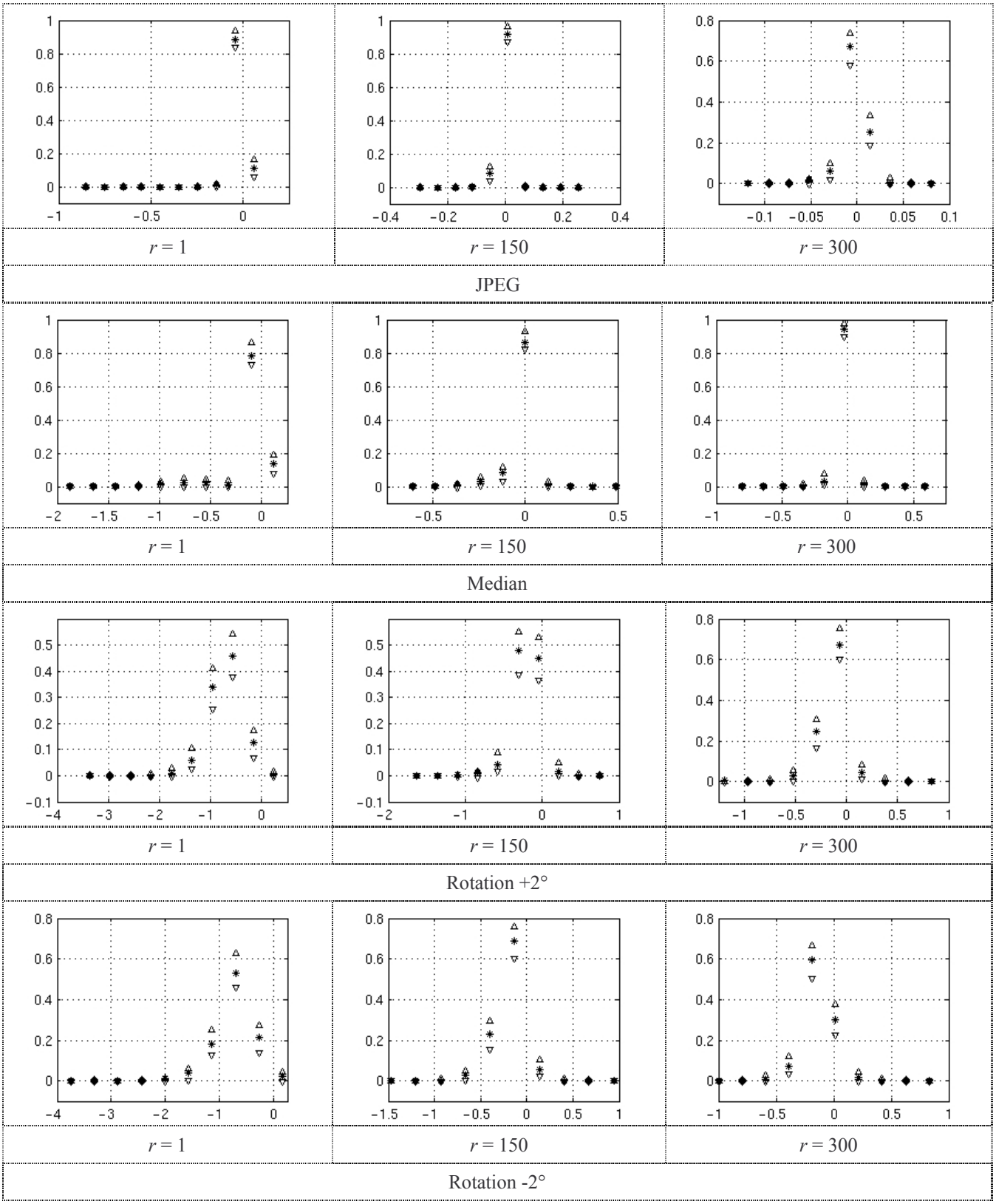


Figure A.III.2.b. Model validation for (9,7) DWT, for three ranks, high quality video, and for the JPEG, the median, and the rotation ± 2° attacks: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

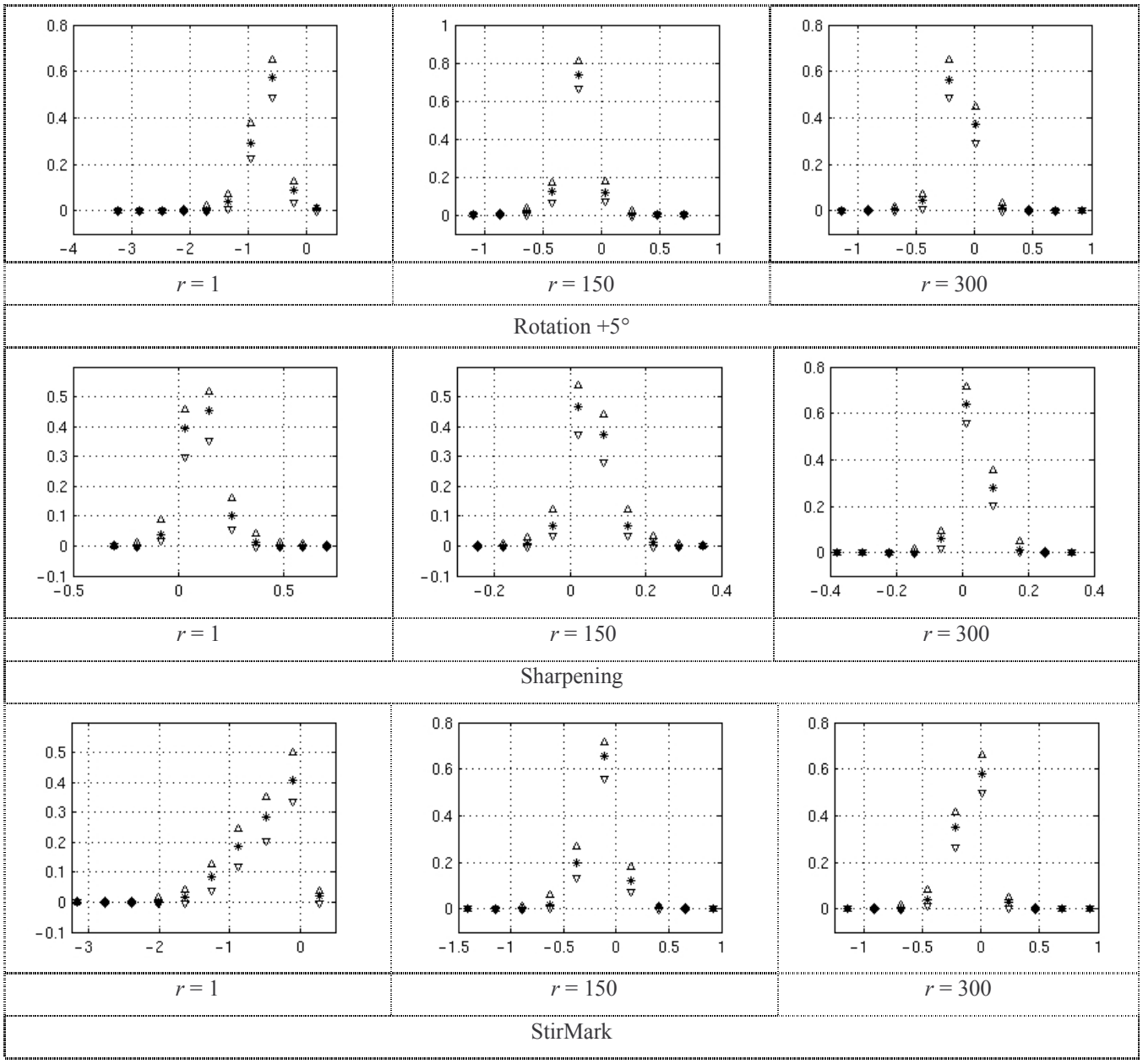


Figure A.III.2.c. Model validation for (9,7) DWT, for three ranks, high quality video, and for the rotation +5°, the sharpening, and the StirMark attacks: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by *.

A.III.2. The 2D-DCT coefficients modelling

A.III.2.1. Model Computation

Table A.III.3. IVM for DCT coefficients hierarchy, low quality video and different attacks.

Attack/ Rank		Model parameters											Error
Gaussian filtering	1	*P(k)*	0.118	0.062	0.091	0.102	0.105	0.101	0.047	0.054	0.212	0.109	0.041
		μ(k)	0.465	0.502	0.284	0.255	0.316	0.151	0.426	0.349	0.216	0.121	
		σ(k)	0.089	0.174	0.120	0.118	0.119	0.099	0.170	0.075	0.039	0.092	
	50	*P(k)*	0.046	0.32	0.119	0.030	0.094	0.042	0.050	0.144	0.054	0.398	0.014
		μ(k)	-0.029	0.069	0.125	0.103	0.037	0.319	0.042	0.052	0.053	0.027	
		σ(k)	0.079	0.102	0.052	0.001	0.030	0.101	0.098	0.025	0.100	0.029	
	100	*P(k)*	0.104	0.224	0.049	0.172	0.057	0.069	0.1756	0.047	0.070	0.032	0.017
		μ(k)	0.065	0.031	0.117	0.054	-0.010	0.166	0.048	0.417	-0.020	0.032	
		σ(k)	0.032	0.018	0.020	0.028	0.095	0.062	0.027	0.105	0.009	0.106	
	150	*P(k)*	0.053	0.264	0.107	0.060	0.117	0.098	0.104	0.140	0.033	0.024	0.018
		μ(k)	0.138	0.036	0.097	0.145	0.080	0.048	0.009	0.019	0.375	0.372	
		σ(k)	0.062	0.014	0.030	0.060	0.015	0.016	0.005	0.043	0.091	0.092	
	200	*P(k)*	0.054	0.052	0.049	0.166	0.014	0.091	0.017	0.045	0.327	0.184	0.014
		μ(k)	0.158	0.144	0.019	0.037	0.505	.0078	0.271	0.162	0.020	0.081	
		σ(k)	0.078	0.082	0.059	0.023	0.053	0.036	0.037	0.009	0.020	0.017	
	250	*P(k)*	0.039	0.012	0.120	0.018	0.399	0.097	0.012	0.112	0.027	0.052	0.018
		μ(k)	0.287	-0.031	0.111	0.209	0.034	0.130	0.041	0.046	-0.007	0.074	
		σ(k)	0.048	0.001	0.017	0.002	0.020	0.053	0.036	0.038	0.116	0.001	
	300	*P(k)*	0.066	0.056	0.016	0.088	0.053	0.179	0.089	0.104	0.124	0.226	0.014
		μ(k)	0.073	0.110	0.120	0.208	0.087	0.039	0.103	0.061	0.050	0.025	
		σ(k)	0.104	0.017	0.001	0.109	0.106	0.012	0.049	0.032	0.029	0.026	
	350	*P(k)*	0.130	0.172	0.098	0.026	0.034	0.128	0.044	0.082	0.219	0.067	0.016
		μ(k)	0.076	0.038	0.023	0.106	0.132	0.048	0.146	0.170	0.027	0.089	
		σ(k)	0.033	0.031	0.025	0.045	0.003	0.009	0.059	0.056	0.027	0.022	
JPEG	1	*P(k)*	0.195	0.139	0.094	0.077	0.089	0.071	0.137	0.109	0.044	0.044	0.028
		μ(k)	-0.016	-0.067	-0.005	0.008	-0.068	-0.006	-0.098	-0.048	-0.219	-0.130	
		σ(k)	0.044	0.059	0.104	0.102	0.092	0.105	0.063	0.053	0.056	0.004	
	50	*P(k)*	0.078	0.061	0.146	0.159	0.128	0.127	0.065	0.090	0.099	0.048	0.026
		μ(k)	-0.049	-0.041	-0.004	0.006	0.014	-0.020	0.027	-0.043	-0.012	0.011	
		σ(k)	0.043	0.006	0.030	0.028	0.026	0.033	0.058	0.044	0.029	0.060	
	100	*P(k)*	0.046	0.091	0.077	0.076	0.077	0.081	0.172	0.161	0.100	0.121	0.023
		μ(k)	-0.025	-0.015	-0.013	-0.011	0.014	-0.032	0.017	-0.015	-0.012	-0.012	
		σ(k)	0.003	0.042	0.042	0.041	0.002	0.039	0.035	0.042	0.042	0.042	
	150	*P(k)*	0.043	0.067	0.021	0.158	0.157	0.057	0.047	0.176	0.159	0.114	0.026
		μ(k)	-0.018	-0.021	0.125	0.002	-0.016	-0.020	-0.008	0.006	0.005	-0.053	
		σ(k)	0.038	0.038	0.015	0.027	0.020	0.038	0.037	0.028	0.028	0.031	
	200	*P(k)*	0.104	0.082	0.055	0.125	0.114	0.100	0.061	0.142	0.113	0.103	0.021
		μ(k)	0.022	0.002	0.008	-0.002	-0.002	-0.017	-0.070	-0.017	-0.013	-0.005	
		σ(k)	0.039	0.041	0.041	0.029	0.015	0.027	0.039	0.027	0.028	0.029	
	250	*P(k)*	0.169	0.117	0.045	0.079	0.057	0.113	0.136	0.092	0.119	0.074	0.024
		μ(k)	-0.001	-0.057	-0.043	-0.002	-0.015	0.002	0.003	-0.024	0.004	-0.002	
		σ(k)	0.015	0.035	0.001	0.053	0.002	0.018	0.023	0.037	0.023	0.053	
	300	*P(k)*	0.111	0.146	0.124	0.087	0.095	0.050	0.184	0.097	0.014	0.093	0.020
		μ(k)	-0.005	-0.013	-0.012	-0.001	-0.004	-0.028	0.001	-0.016	-0.115	0.002	
		σ(k)	0.035	0.031	0.006	0.039	0.037	0.030	0.020	0.032	0.002	0.034	
	350	*P(k)*	0.043	0.026	0.186	0.067	0.076	0.157	0.020	0.041	0.289	0.096	0.021
		μ(k)	-0.091	-0.008	0.023	-0.002	-0.001	-0.001	-0.030	-0.034	-0.023	0.007	
		σ(k)	0.019	0.026	0.023	0.022	0.023	0.009	0.033	0.033	0.017	0.026	

Table A.III.3. *(continued)*

Attack/ Rank		Model parameters										Error
Rotation +2°	1	$P(k)$	0.101	0.112	0.099	0.078	0.099	0.104	0.093	0.105	0.105	0.104	0.029
		$\mu(k)$	-0.408	-1.888	1.216	-0.802	-0.541	-1.070	-1.234	-0.481	-0.953	1.505	
		$\sigma(k)$	0.956	1.098	0.645	1.087	0.994	1.163	0.648	0.971	1.111	0.824	
	50	$P(k)$	0.041	0.068	0.079	0.068	0.189	0.140	0.149	0.094	0.124	0.047	0.029
		$\mu(k)$	-0.502	0.835	0.198	1.855	0.998	0.506	0.462	0.529	0.369	-0.885	
		$\sigma(k)$	1.164	0.754	0.113	0.544	0.361	0.498	0.504	0.724	0.578	1.116	
	100	$P(k)$	0.220	0.103	0.036	0.117	0.049	0.051	0.045	0.101	0.061	0.216	0.019
		$\mu(k)$	0.288	1.027	0.031	0.901	0.468	0.091	-0.632	0.468	0.370	0.554	
		$\sigma(k)$	0.283	0.331	0.015	0.365	1.381	0.317	0.148	0.425	0.415	0.183	
	150	$P(k)$	0.159	0.052	0.181	0.128	0.120	0.070	0.068	0.073	0.015	0.133	0.017
		$\mu(k)$	0.357	1.402	0.247	0.502	0.487	0.367	0.077	-0.172	0.931	0558	
		$\sigma(k)$	0.254	0.263	0.195	0.265	0.268	0.479	0.463	0.408	0.018	0.247	
	200	$P(k)$	0.098	0.195	0.087	0.068	0.171	0.076	0.088	0.046	0.086	0.084	0.012
		$\mu(k)$	0.153	0.272	0.372	0.194	0.488	0.282	0.189	0.237	0.302	0.530	
		$\sigma(k)$	0.796	0.094	0.341	0.327	0.147	0.345	0.323	0.019	0.346	0.313	
	250	$P(k)$	0.058	0.007	0.007	0.194	0.195	0.169	0.089	0.106	0.071	0.102	0.009
		$\mu(k)$	-0.131	1.949	1.109	0.227	0.299	0.307	0.359	-0.100	0.358	0.759	
		$\sigma(k)$	0.418	0.001	0.001	0.142	0.156	0.156	0.191	0.216	0.197	0.129	
	300	$P(k)$	0.074	0.028	0.133	0.086	0.082	0.082	0.301	0.079	0.050	0.085	0.013
		$\mu(k)$	0.262	0.911	0.275	0.323	0.369	0.558	0.161	0.360	-0.248	-0.183	
		$\sigma(k)$	0.166	0.016	0.092	0.165	0.058	0.116	0.094	0.164	0.316	0.086	
	350	$P(k)$	0.041	0.189	0.159	0.036	0.047	0.037	0.018	0.108	0.108	0.256	0.008
		$\mu(k)$	0.041	0.076	0.196	-0.042	-0.132	-0.173	0.844	0.353	0.458	0.319	
		$\sigma(k)$	0.848	0.043	0.043	0.247	0.217	0.053	0.032	0.155	0.132	0.065	
Rotation -2°	1	$P(k)$	0.131	0.084	0.101	0.085	0.108	0.098	0.101	0.091	0.084	0.115	0.029
		$\mu(k)$	-0.814	-1.813	-1.554	0.013	1.151	-0.494	-0.894	-0.436	0.385	-1.705	
		$\sigma(k)$	0.556	0.999	1.016	0.883	0.667	0.976	1.054	0.965	0.828	0.977	
	50	$P(k)$	0.007	0.144	0.086	0.053	0.148	0.095	0.120	0.157	0.076	0.114	0.027
		$\mu(k)$	-3.026	0.532	0.344	1.108	1.528	0.052	0.220	0.877	-0.343	0.159	
		$\sigma(k)$	0.001	0.263	0.500	0.939	0.222	0.609	0.405	0.137	0.578	0.361	
	100	$P(k)$	0.083	0.007	0.096	0.087	0.037	0.240	0.106	0.148	0.095	0.102	0.018
		$\mu(k)$	0.180	2.839	0.550	1.291	-0.474	0.482	-0.099	0.493	0.487	0.569	
		$\sigma(k)$	0.498	0.001	0.354	0.233	0.765	0.291	0.437	0.290	0.395	0.341	
	150	$P(k)$	0.021	0.116	0.121	0.105	0.041	0.057	0.106	0.162	0.072	0.198	0.016
		$\mu(k)$	-0.780	0.017	0.295	0.379	-0.362	0.606	0.281	0.603	1.256	0.525	
		$\sigma(k)$	0.117	0.114	0.158	0.232	0.045	0.350	0.159	0.222	0.342	0.238	
	200	$P(k)$	0.163	0.076	0.040	0.104	0.160	0.143	0.033	0.026	0.081	0.174	0.012
		$\mu(k)$	0.061	-0.374	1.162	0.501	0.556	0.307	-0.256	0.915	0.316	0.344	
		$\sigma(k)$	0.074	0.695	0.356	0.194	0.178	0.137	0.015	0.020	0.134	0.134	
	250	$P(k)$	0.153	0.044	0.028	0.196	0.087	0.226	0.065	0.014	0.132	0.053	0.009
		$\mu(k)$	0.348	0.470	-0.203	0.320	0.529	0.141	0.084	1.614	0.510	-0.542	
		$\sigma(k)$	0.159	0.269	0.044	0.097	0.265	0.143	0.119	0.017	0.268	0.528	
	300	$P(k)$	0.060	0.095	0.116	0.102	0.160	0.107	0.037	0.199	0.022	0.100	0.012
		$\mu(k)$	-0.341	-0.017	0.143	0.273	0.388	0.326	0.742	0.244	1.091	0.394	
		$\sigma(k)$	0.354	0.047	0.015	0.172	0.133	0.156	0.047	0.079	0.141	0.131	
	350	$P(k)$	0.139	0.146	0.062	0.061	0.051	0.154	0.114	0.090	0.162	0.023	0.008
		$\mu(k)$	0.183	0.305	0.422	0.327	0.171	0.214	0.348	0.337	0.067	-0.284	
		$\sigma(k)$	0.129	0.080	0.187	0.841	0.271	0.115	0.207	0.210	0.109	0.028	

Table A.III.3. *(continued)*

Attack/ Rank		Model parameters										Error
Rotation +5°	1	*P(k)*	0.078	0.088	0.095	0.084	0.071	0.122	0.162	0.081	0.084	0.136	0.032
		μ(k)	1.379	-2.651	-3.688	-2.059	1.267	-3.004	-0.675	-0.404	-1.082	-1.191	
		σ(k)	2.437	2.868	2.667	2.954	2.462	1.117	1.569	1.708	2.013	1.720	
	50	*P(k)*	0.052	0.080	0.056	0.080	0.110	0.067	0.060	0.240	0.140	0.116	0.013
		μ(k)	2.020	0.347	1.519	0.592	0.866	-0.874	1.957	0.670	0.950	0.915	
		σ(k)	0.360	0.903	0.478	0.826	0.409	0.760	1.281	0.347	0.411	0.416	
	100	*P(k)*	0.393	0.019	0.027	0.062	0.092	0.089	0.132	0.028	0.095	0.063	0.008
		μ(k)	0.409	-1.314	1.869	-0.411	1.283	0.592	0.811	1.505	0.417	1.080	
		σ(k)	0.168	0.942	0.690	0.275	0.309	0.201	0.049	0.784	0.180	0.338	
	150	*P(k)*	0.208	0.044	0.028	0.022	0.051	0.107	0.047	0.231	0.019	0.244	0.010
		μ(k)	0.025	1.461	1.036	-0.851	0.330	0.363	1.006	0.627	0.583	0.409	
		σ(k)	0.152	0.402	0.331	0.207	0.164	0.128	0.068	0.085	0.404	0.130	
	200	*P(k)*	0.091	0.134	0.051	0.029	0.077	0.161	0.039	0.070	0.119	0.228	0.009
		μ(k)	0.052	0.288	0.187	0.006	0.922	0.332	-0.410	0.299	0.193	0.499	
		σ(k)	0.222	0.140	0.268	1.921	0.160	0.128	0.400	0.287	0.200	0.107	
	250	*P(k)*	0.055	0.115	0.209	0.093	0.061	0.027	0.207	0.095	0.028	0.110	0.010
		μ(k)	0.985	0.193	0.308	0.082	0.444	-0.812	0.214	0.446	0.547	0.436	
		σ(k)	0.577	0.306	0.077	0.271	0.390	0.130	0.099	0.166	0.235	0.162	
	300	*P(k)*	0.077	0.202	0.078	0.049	0.156	0.052	0.254	0.040	0.051	0.041	0.004
		μ(k)	0.157	0.365	0.149	0.828	0.107	0.446	0.234	0.282	-0.337	0.778	
		σ(k)	0.239	0.106	0.237	0.320	0.069	0.080	0.072	0.425	0.353	0.084	
	350	*P(k)*	0.249	0.111	0.040	0.007	0.075	0.095	0.237	0.026	0.091	0.069	0.006
		μ(k)	0.306	0.199	-0.110	-0.594	0.157	0.247	0.160	0.424	0.454	0.224	
		σ(k)	0.082	0.131	0.027	0.001	0.213	0.185	0.070	1.728	0.341	0.251	
Median	1	*P(k)*	0.106	0.156	0.142	0.079	0.106	0.029	0.146	0.056	0.087	0.092	0.020
		μ(k)	-60.51	-59.59	-89.95	-55.61	-60.34	-39.42	-60.02	-36.29	-76.15	-67.05	
		σ(k)	10.19	6.797	6.792	9.509	1.102	5.806	7.097	5.662	1.325	1.287	
	50	*P(k)*	0.031	0.091	0.154	0.147	0.137	0.052	0.122	0.126	0.056	0.085	0.026
		μ(k)	-2.686	-1.995	-1.969	-1.874	-1.895	-1.166	-1.847	-1.550	-2.027	-1.818	
		σ(k)	0.417	0.148	0.262	0.283	0.282	0.223	0.289	0.187	0.259	0.297	
	100	*P(k)*	0.098	0.102	0.142	0.075	0.108	0.138	0.076	0.092	0.058	0.111	0.028
		μ(k)	-1.149	-1.046	-1.133	-1.197	-1.469	-1.093	-1.041	-0.994	-1.253	-1.189	
		σ(k)	0.204	0.247	0.166	0.025	0.165	0.172	0.251	0.251	0.198	0.199	
	150	*P(k)*	0.056	0.084	0.111	0.086	0.238	0.117	0.076	0.076	0.050	0.106	0.031
		μ(k)	-0.688	-0.779	-1.057	-0.841	-0.873	-0.933	-0.798	-0.923	-0.501	-0.890	
		σ(k)	0.080	0.115	0.144	0.187	0.088	0.169	0.195	0.174	0.194	0.176	
	200	*P(k)*	0.085	0.079	0.021	0.148	0.071	0.107	0.123	0.052	0.077	0.236	0.034
		μ(k)	-0.775	-0.807	-0.131	-0.529	-0.578	-0.691	-0.655	-0.773	-0.831	-0.742	
		σ(k)	0.165	0.175	-0.064	0.102	0.115	0.123	0.118	0.175	0.177	0.071	
	250	*P(k)*	0.070	0.087	0.083	0.193	0.096	0.112	0.080	0.111	0.112	0.055	0.034
		μ(k)	-0.578	-0.744	-0.641	-0.595	-0.535	-0.678	-0.492	-0.557	-0.536	-0.519	
		σ(k)	0.135	0.104	0.120	0.071	0.143	0.115	0.145	0.136	0.141	0.144	
	300	*P(k)*	0.101	0.143	0.104	0.156	0.148	0.079	0.076	0.007	0.105	0.081	0.034
		μ(k)	-0.437	-0.413	-0.557	-0.516	-0.559	-0.499	-0.706	-1.065	-0.482	-0.512	
		σ(k)	0.098	0.071	0.103	0.112	0.047	0.122	0.069	0.001	0.125	0.117	
	350	*P(k)*	0.129	0.200	0.097	0.091	0.097	0.021	0.144	0.031	0.009	0.181	0.028
		μ(k)	-0.522	-0.368	-0.465	-0.610	-0.532	-0.217	-0.448	-0.547	-0.187	-0.385	
		σ(k)	0.076	0.073	0.015	0.133	0.090	0.197	0.094	0.003	0.019	0.081	

Table A.III.3. *(continued)*

Attack/ Rank		Model parameters										Error
Sharpening	1	$P(k)$	0.101	0.037	0.157	0.088	0.079	0.078	0.072	0.167	0121	0.099	0.021
		$\mu(k)$	-1.169	-3.381	-1.937	-1.246	-1.436	-1.201	-1.400	-1/809	-2.707	-1.965	
		$\sigma(k)$	0.870	0.537	0.503	0.883	0.090	0.876	0.329	0.475	0.299	0.521	
	50	$P(k)$	0.125	0.058	0.098	0.114	0.017	0.114	0.091	0.238	0.028	0.117	0.018
		$\mu(k)$	-0.002	0.006	0.089	-0.541	-0.050	0.004	-0.013	-0.135	-0.469	-0.126	
		$\sigma(k)$	0.185	0.183	0.162	0.689	0.194	0.021	0.188	0.093	0.034	0.152	
	100	$P(k)$	0.007	0.203	0.085	0.065	0.102	0.088	0.075	0.064	0.110	0.200	0.019
		$\mu(k)$	-1.989	-0.073	-0.257	0.012	-0.066	-0.041	0.168	-0.581	-0.245	-0.069	
		$\sigma(k)$	0.001	0.079	0.085	0.138	0.152	0.147	0.101	0.106	0.172	0.078	
	150	$P(k)$	0.062	0.153	0.037	0.134	0.061	0.125	0.099	0.076	0.136	0.115	0.020
		$\mu(k)$	0.039	-0.144	-0.931	-0.092	-0.219	-0.086	-0.058	-0.417	-0.130	-0.022	
		$\sigma(k)$	0.125	0.133	0.319	0.025	0.182	0.152	0.149	0.215	0.147	0.044	
	200	$P(k)$	0.040	0.158	0.014	0.197	0.045	0.071	0.113	0.166	0.180	0.014	0.022
		$\mu(k)$	-0.133	-0.189	0.062	-0.001	-0.092	-0.547	-0.159	-0.208	-0.019	-1.055	
		$\sigma(k)$	0.006	0.076	0.001	0.154	0.012	0.123	0.089	0.187	0.041	0.045	
	250	$P(k)$	0.066	0.184	0.139	0.075	0.146	0.045	0.056	0.051	0.184	0.053	0.019
		$\mu(k)$	-0.229	-0.079	-0.194	-0.257	-0.141	-0.380	-0.182	-0.357	0.020	-0.174	
		$\sigma(k)$	0.170	0.036	0.053	0.125	0.091	0.297	0.238	0.297	0.070	0.241	
	300	$P(k)$	0.229	0.016	0.146	0.092	0.083	0.058	0.138	0.012	0.184	0.041	0.024
		$\mu(k)$	-0.090	-0391	-0.258	-0.118	0.017	-0.637	0.080	-0.488	-0.108	-0.150	
		$\sigma(k)$	0.036	0.004	0.061	0.117	0.037	0.233	0.112	0.005	0.117	0.405	
	350	$P(k)$	0.179	0.034	0.055	0.102	0.138	0.122	0.029	0.123	0.127	0.091	0.021
		$\mu(k)$	-0.099	-0.166	0.003	-0.154	-0.243	-0.081	-0.565	-0.036	-0.268	-0.178	
		$\sigma(k)$	0.037	0.003	0.005	0.153	0.161	0.133	0.261	0.122	0.155	0.063	
StirMark	1	$P(k)$	0.144	0.169	0.121	0.078	0.114	0.089	0.108	0.068	0.056	0.054	0.039
		$\mu(k)$	-0.201	-0.278	-0.276	-0.384	-0.196	-0.510	-0.163	-0.334	-0.667	0.300	
		$\sigma(k)$	0.293	0.264	0.300	0.296	0.294	0.162	0.284	0.350	0.423	0.554	
	50	$P(k)$	0.037	0.128	0.073	0.126	0.128	0.145	0.088	0.102	0.134	0.040	0.015
		$\mu(k)$	-0.306	0.081	-0.050	0.071	0.251	0.124	0.327	0.072	-0.064	-0.268	
		$\sigma(k)$	0.027	0.127	0.084	0.127	0.185	0.117	0.285	0.128	0.083	0.256	
	100	$P(k)$	0.174	0.096	0.027	0.103	0.122	0.017	0.141	0.120	0.067	0.130	0.019
		$\mu(k)$	0.234	0.101	0.871	0.098	0.016	0.150	-0.026	0.167	-0.255	-0.042	
		$\sigma(k)$	0.142	0.167	0.120	0.020	0.020	0.310	0.118	0.158	0.229	0.087	
	150	$P(k)$	0.246	0.028	0.101	0.023	0.018	0.049	0.149	0.137	0.165	0.083	0.022
		$\mu(k)$	0.018	0.840	-0.018	0.053	0.052	0.213	-0.025	0.281	0.065	0.125	
		$\sigma(k)$	0.081	0.138	0.213	0.006	0.041	0.033	0.086	0.126	0.215	0.016	
	200	$P(k)$	0.107	0.109	0.033	0.072	0.041	0.085	0.113	0.167	0.127	0.146	0.022
		$\mu(k)$	0.108	0.067	0.522	-0.133	0.384	0.253	0.056	0.017	-0.045	0.120	
		$\sigma(k)$	0.050	0.127	0.135	0.095	0.035	0.024	0.125	0.020	0.100	0.040	
	250	$P(k)$	0.099	0.114	0.103	0.213	0.123	0.122	0.045	0.034	0.100	0.048	0.020
		$\mu(k)$	0.332	0.140	-0.111	-0.006	0.069	0.066	0.227	0.152	0.156	0.038	
		$\sigma(k)$	0.167	0.089	0.034	0.053	0.083	0.028	0.099	0.015	0.094	0.222	
	300	$P(k)$	0.143	0.007	0.164	0.137	0.098	0.148	0.060	0.127	0.055	0.061	0.019
		$\mu(k)$	0.069	-0.715	0.072	0.014	0.259	0.022	-0.111	0.159	0.154	0.244	
		$\sigma(k)$	0.111	0.001	0.112	0.095	0.163	0.043	0.069	0.095	0.009	0.164	
	350	$P(k)$	0.043	0.093	0.021	0.054	0.177	0.048	0.070	0.155	0.144	0.194	0.021
		$\mu(k)$	0.163	0.141	0.337	0.082	-0.049	-0.210	0.266	0.097	0.169	0.011	
		$\sigma(k)$	0.135	0.080	0.005	0.118	0.037	0.049	0.128	0.019	0.038	0.023	

Table A.III.4. IVM for DCT coefficients hierarchy, high quality video and different attacks.

Attack/ Rank		Model parameters										Error
Gaussian filtering	1	$P(k)$	0.125	0.096	0.098	0.057	0.157	0.118	0.109	0.044	0.061	0.134	0.043
		$\mu(k)$	-0.444	-0.250	-0.249	-0.595	-0.315	-0.277	-0.256	0.066	-0.203	-0.076	
		$\sigma(k)$	0.063	0.107	0.107	0.015	0.032	0.109	0.108	0.153	0.093	0.071	
	50	$P(k)$	0.216	0.356	0.106	0.062	0.014	0.011	0.141	0.048	0.024	0.023	0.014
		$\mu(k)$	0.027	0.002	0.024	0.112	0.760	0.139	0.035	0.327	0.008	0.114	
		$\sigma(k)$	0.034	0.027	0.003	0.052	0.037	0.014	0.033	0.119	0.149	0.178	
	100	$P(k)$	0.042	0.031	0.036	0.163	0.043	0.144	0.107	0.084	0.309	0.040	0.005
		$\mu(k)$	0.341	0.010	0.141	0.027	0.081	0.014	0.024	0.004	0.015	-0.041	
		$\sigma(k)$	0.228	0.042	0.019	0.030	0.180	0.030	0.013	0.041	0.011	0.004	
	150	$P(k)$	0.221	0.0269	0.228	0.019	0.024	0.242	0.046	0.029	0.133	0.033	0.010
		$\mu(k)$	0.023	-0.052	0.016	-0.036	0.046	0.035	0.157	0.320	0.026	-0.030	
		$\sigma(k)$	0.026	0.084	0.021	0.084	0.051	0.028	0.022	0.033	0.027	0.039	
	200	$P(k)$	0.158	0.157	0.125	0.198	0.014	0.110	0.038	0.168	0.007	0.024	0.009
		$\mu(k)$	0.039	0.026	0.012	0.015	0.724	0.034	0.161	0.025	-0.043	0.246	
		$\sigma(k)$	0.061	0.025	0.012	0.023	0.087	0.024	0.091	0.015	0.001	0.068	
	250	$P(k)$	0.096	0.137	0.073	0.026	0.076	0.026	0.174	0.301	0.075	0.018	0.010
		$\mu(k)$	0.027	0.012	0.010	0.309	0.020	0.205	0.054	0.016	0.020	0.153	
		$\sigma(k)$	0.036	0.016	0.033	0.107	0.036	0.064	0.029	0.013	0.036	0.005	
	300	$P(k)$	0.014	0.184	0.092	0.040	0.103	0.007	0.101	0.196	0.160	0.102	0.010
		$\mu(k)$	0.086	0.022	0.002	0.049	0.024	0.432	0.077	0.028	0.010	0.067	
		$\sigma(k)$	0.001	0.023	0.026	0.001	0.005	0.001	0.089	0.021	0.011	0.091	
	350	$P(k)$	0.337	0.007	0.034	0.034	0.036	0.140	0.156	0.169	0.044	0.042	0.011
		$\mu(k)$	0.004	0.971	0.115	0.128	0.192	0.036	0.049	0.022	-0.058	0.040	
		$\sigma(k)$	0.010	0.001	0.008	0.068	0.078	0.018	0.018	0.009	0.022	0.019	
JPEG	1	$P(k)$	0.079	0.147	0.091	0.074	0.167	0.016	0.117	0.082	0.106	0.122	0.001
		$\mu(k)$	0.106	-0.078	0.018	-0.013	0.029	0.080	0.097	-0.276	-0.050	0.008	
		$\sigma(k)$	0.219	0.040	0.155	0.172	0.131	0.007	0.061	0.270	0.165	0.155	
	50	$P(k)$	0.015	0.105	0.123	0.218	0.116	0.033	0.158	0.143	0.068	0.021	0.001
		$\mu(k)$	0.252	0.036	-0.053	0.007	0.011	0.014	-0.001	0.071	0.013	0.017	
		$\sigma(k)$	0.072	0.028	0.017	0.021	0.031	0.031	0.025	0.019	0.031	0.031	
	100	$P(k)$	0.109	0.199	0.079	0.112	0.133	0.114	0.007	0.090	0.093	0.063	0.001
		$\mu(k)$	-0.014	0.066	-0.019	-0.022	0.027	0.006	-0.181	-0.025	0.005	0.033	
		$\sigma(k)$	0.029	0.015	0.028	0.028	0.007	0.030	0.001	0.028	0.004	0.027	
	150	$P(k)$	0.013	0.096	0.095	0.284	0.108	0.071	0.071	0.053	0.111	0.098	0.001
		$\mu(k)$	0.005	0.032	-0.032	-0.001	-0.034	0.021	0.010	0.013	-0.015	0.054	
		$\sigma(k)$	0.001	0.035	0.028	0.020	0.028	0.037	0.036	0.037	0.030	0.029	
	200	$P(k)$	0.080	0.111	0.116	0.075	0.083	0.041	0.101	0.252	0.101	0.042	0.001
		$\mu(k)$	0.024	-0.023	-0.014	0.021	0.046	0.005	-0.009	0.011	0.041	0.026	
		$\sigma(k)$	0.037	0.025	0.028	0.037	0.034	0.086	0.030	0.011	0.003	0.084	
	250	$P(k)$	0.145	0.158	0.041	0.208	0.068	0.057	0.057	0.143	0.071	0.052	0.001
		$\mu(k)$	-0.030	0.022	0.027	0.007	0.015	0.006	-0.006	0.029	0.025	0.015	
		$\sigma(k)$	0.009	0.021	0.054	0.011	0.023	0.021	0.049	0.019	0.054	0..023	
	300	$P(k)$	0.078	0.154	0.030	0.238	0.133	0.132	0.0295	0.045	0.124	0.035	0.001
		$\mu(k)$	-0.012	-0.006	0.084	0.032	0.002	-0.047	0.076	0.008	-0.003	0.011	
		$\sigma(k)$	0.003	0.021	0.023	0.015	0.023	0.010	0.026	0.023	0.022	0.023	
	350	$P(k)$	0.013	0.039	0.027	0.144	0.245	0.114	0.028	0.077	0.166	0.147	0.001
		$\mu(k)$	0.097	-0.067	-0.059	-0.005	0.029	0.010	0.020	-0.005	0.003	0.015	
		$\sigma(k)$	0.004	0.044	0.002	0.022	0.022	0.026	0.001	0.006	0.026	0.025	

Table A.III.4. *(continued)*

Attack/ Rank		Model parameters										Error
Rotation +2°	1	$P(k)$	0.097	0.121	0.056	0.077	0.140	0.087	0.057	0.122	0.099	0.142	0.029
		$\mu(k)$	-2.206	-3.308	-9.665	-4.512	-0.812	3.916	-5.936	-2.826	-0.276	-2.166	
		$\sigma(k)$	2.991	1.741	3.861	2.477	2.167	2.652	4.546	2.463	3.296	2.452	
	50	$P(k)$	0.112	0.064	0.114	0.103	0.087	0.112	0.102	0.097	0.086	0.123	0.025
		$\mu(k)$	1.218	-0.313	2.091	1.362	5.233	1.583	1.167	2.274	3.968	1.372	
		$\sigma(k)$	1.866	3.554	1.337	1.869	2.094	1.750	1.889	1.427	2.367	1.803	
	100	$P(k)$	0.182	0.098	0.045	0.102	0.067	0.088	0.095	0.113	0.112	0.097	0.019
		$\mu(k)$	1.133	1.302	3.482	1.172	0.216	-1.261	2.753	3.066	1.624	1.605	
		$\sigma(k)$	0.671	1.192	2.178	1.167	0.163	1.470	0.944	0.822	1.183	1.185	
	150	$P(k)$	0.185	0.078	0.069	0.103	0.093	0.105	0.122	0.098	0.039	0.108	0.019
		$\mu(k)$	1.107	2.076	2.533	1.754	1.534	1.379	-0.094	-1.047	4.269	1.454	
		$\sigma(k)$	0.305	1.192	0.813	0.810	1.075	0.876	0.247	1.092	1.049	0.861	
	200	$P(k)$	0.030	0.007	0.156	0.137	0.027	0.041	0.087	0.116	0.046	0.354	0.014
		$\mu(k)$	3.680	13.226	1.127	1.318	3.347	-1.712	2.266	0.267	0.711	1.063	
		$\sigma(k)$	0.837	0.001	0.744	1.060	0.862	0.579	0.323	0.552	0.762	0.448	
	250	$P(k)$	0.202	0.040	0.085	0.077	0.101	0.107	0.094	0.083	0.107	0.104	0.013
		$\mu(k)$	0.837	-1.081	0.662	0.235	0.828	2.139	1.354	0.626	1.719	1.172	
		$\sigma(k)$	0.393	0.906	0.659	0.880	0.665	1.271	1.340	0.309	0.243	0.595	
	300	$P(k)$	0.095	0.099	0.077	0.110	0.108	0.113	0.081	0.055	0.166	0.096	0.012
		$\mu(k)$	0.325	1.948	1.207	0.760	1.173	0.738	0.307	-1.408	0.690	1.092	
		$\sigma(k)$	0.668	1.032	0.549	0.785	0.797	0.781	0.661	2.098	0.149	0.823	
	350	$P(k)$	0.209	0.100	0.107	0.075	0.097	0.131	0.125	0.068	0.042	0.046	0.001
		$\mu(k)$	1.075	0.414	0.701	0.626	0.819	0.494	0.645	1.075	1.468	2.201	
		$\sigma(k)$	0.382	0.675	0.607	1.234	0.739	0.329	0.365	0.772	0.677	1.283	
Rotation -2°	1	$P(k)$	0.088	0.115	0.109	0.080	0.106	0.112	0.109	0.094	0.089	0.099	0.045
		$\mu(k)$	-3.301	0.533	-2.203	-5.328	-0.287	0.275	-2.476	-2.842	-3.848	-2.310	
		$\sigma(k)$	3.832	3.879	3.158	1.747	3.918	3.897	3.312	3.769	2.806	3.595	
	50	$P(k)$	0.071	0.175	0.046	0.124	0.099	0.110	0.086	0.070	0.119	0.097	0.025
		$\mu(k)$	0.498	2.582	-0.487	1.505	1.232	5.654	0.232	0.110	1.350	3.047	
		$\sigma(k)$	2.888	1.172	1.416	1.511	1.667	1.797	1.502	2.896	1.551	1.064	
	100	$P(k)$	0.044	0.146	0.117	0.127	0.090	0.098	0.145	0.101	0.050	0.082	0.019
		$\mu(k)$	3.849	1.686	0.392	2.371	2.299	0.574	0.850	0.775	4.879	2.012	
		$\sigma(k)$	1.338	1.041	1.608	0.778	0.835	1.655	0.544	1.705	1.685	1.067	
	150	$P(k)$	0.049	0.128	0.157	0.116	0.045	0.067	0.119	0.160	0.102	0.056	0.017
		$\mu(k)$	2.639	0.803	0.873	2.011	-1.534	1.763	1.202	1.103	1.108	3.814	
		$\sigma(k)$	1.074	0.858	0.872	0.677	1.662	1.102	0.891	0.892	0.899	0.779	
	200	$P(k)$	0.206	0.080	0.091	0.063	0.107	0.091	0.045	0.085	0.182	0.050	0.013
		$\mu(k)$	0.755	1.361	0.982	-1.284	1.430	-0.400	2.799	2.555	1.369	1.897	
		$\sigma(k)$	0.353	0.822	0.659	1.532	0.810	0.252	1.285	0.616	0.434	1.059	
	250	$P(k)$	0.143	0.099	0.134	0.148	0.115	0.041	0.014	0.123	0.132	0.052	0.012
		$\mu(k)$	1.012	-0.111	1.105	0.883	1.059	2.213	4.189	1.159	0.844	-0.803	
		$\sigma(k)$	0.653	1.087	0.679	0.654	0.778	0.143	1.048	0.718	0.258	1.968	
	300	$P(k)$	0.149	0.068	0.032	0.050	0.098	0.104	0.143	0.140	0.116	0.100	0.011
		$\mu(k)$	0.475	0.992	4.098	-1.181	0.828	1.192	0.484	0.733	2.099	1.033	
		$\sigma(k)$	0.556	0.054	1.051	0.893	0.645	0.560	0.563	0.521	0.358	0.610	
	350	$P(k)$	0.028	0.082	0.122	0.130	0.221	0.059	0.132	0.114	0.021	0.091	0.008
		$\mu(k)$	2.995	-0.277	0.746	1.106	0.715	1.480	0.444	0.893	-2.417	1.086	
		$\sigma(k)$	0.731	0.296	0.571	0.616	0.226	0.179	0.303	0.631	0.191	0.687	

Table A.III.4. *(continued)*

Attack/ Rank		Model parameters										Error
Rotation +5°	1	*P(k)*	0.165	0.089	0.128	0.102	0.056	0.099	0.108	0.065	0.133	0.071	0.024
		μ(k)	-8.653	-2.287	-3.337	-1.522	8.324	-1.705	-2.523	3.544	-6.088	-7.996	
		σ(k)	4.108	6.203	5.589	5.711	5.333	7.882	5.655	6.012	4.966	9.399	
	50	*P(k)*	0.365	0.073	0.074	0.079	0.007	0.065	0.075	0.064	0.108	0.089	0.011
		μ(k)	2.173	2.935	2.872	3.706	13.652	2.368	-1.991	-0.255	6.291	5.374	
		σ(k)	1.071	1.970	2.002	1.668	0.001	2.211	2.304	2.362	1.222	1.462	
	100	*P(k)*	0.059	0.089	0.133	0.100	0.098	0.124	0.078	0.123	0.151	0.045	0.012
		μ(k)	1.119	2.311	2.492	1.015	1.271	2.422	5.344	1.657	2.155	-1.162	
		σ(k)	7.318	1.087	1.093	0.720	0.331	1.088	1.678	0.968	1.009	0.221	
	150	*P(k)*	0.067	0.139	0.061	0.172	0.061	0.057	0.055	0.226	0.085	0.078	0.010
		μ(k)	1.811	1.195	1.921	1.636	2.106	1.804	1.948	0.965	-0.227	3.042	
		σ(k)	1.891	0.524	1.905	0.438	1.912	1.139	1.872	0.448	1.214	0.673	
	200	*P(k)*	0.167	0.079	0.131	0.019	0.111	0.125	0.066	0.090	0.113	0.098	0.005
		μ(k)	1.138	0.724	1.010	6.466	1.463	0.588	3.763	0.725	1.214	1.332	
		σ(k)	0.834	0.882	0.926	1.045	0.816	0.858	1.169	0.882	0.947	0.954	
	250	*P(k)*	0.314	0.165	0.049	0.057	0.007	0.051	0.067	0.085	0.086	0.120	0.008
		μ(k)	1.160	0.171	2.346	-1.026	8.976	2.622	0.738	1.501	1.574	1.119	
		σ(k)	0.368	0.193	1.531	0.618	0.001	1.500	0.537	0.693	0.687	0.525	
	300	*P(k)*	0.082	0.007	0.148	0.100	0.154	0.223	0.063	0.180	0.014	0.028	0.006
		μ(k)	0.634	6.245	0.975	1.181	0.481	1.285	0.791	0.803	-3.188	3.324	
		σ(k)	0.817	0.001	0.585	0.617	0.817	0.604	0.785	0.480	0.623	0.138	
	350	*P(k)*	0.060	0.033	0.242	0.083	0.089	0.080	0.141	0.154	0.090	0.046	0.003
		μ(k)	0.013	0.636	0.898	0.144	0.605	1.265	1.781	0.454	1.086	2.644	
		σ(k)	0.564	0.699	0.175	0.596	0.695	0.566	0.447	0.286	0.623	3.965	
Median	1	*P(k)*	0.118	0.083	0.017	0.075	0.095	0.103	0.138	0.036	0.109	0.224	0.002
		μ(k)	2.257	3.491	6.754	3.441	3.514	3.414	3.293	-8.559	3.345	2.198	
		σ(k)	0.613	0.890	0.634	0.904	0.884	0.910	0.925	6.021	0.921	0.632	
	50	*P(k)*	0.252	0.014	0.017	0.046	0.048	0.145	0.013	0.160	0.169	0.137	0.007
		μ(k)	0.013	0.227	0.188	0.098	2.284	-0.013	0.668	0.051	0.033	0.024	
		σ(k)	0.031	0.103	0.112	0.141	3.359	0.152	0.052	0.156	0.157	0.079	
	100	*P(k)*	0.104	0.113	0.114	0.038	0.049	0.024	0.293	0.153	0.098	0.014	0.003
		μ(k)	-0.042	0.021	0.063	1.683	-0.097	0.254	-0.008	0.111	-0.046	-0.046	
		σ(k)	0.101	0.096	0.084	0.965	0.084	0.021	0.036	0.067	0.100	0.100	
	150	*P(k)*	0.154	0.049	0.064	0.026	0.360	0.036	0.029	0.207	0.062	0.014	0.005
		μ(k)	0.102	0.216	-0.061	0.123	0.031	1.319	-0.204	-0.033	-0.050	3.081	
		σ(k)	0.024	0.091	0.039	0.123	0.034	0.213	0.013	0.060	0.162	0.597	
	200	*P(k)*	0.160	0.131	0.077	0.132	0.166	0.034	0.016	0.007	0.128	0.147	0.008
		μ(k)	0.023	-0.005	0.054	-0.011	0.0024	0.134	0.358	3.359	0.064	0.028	
		σ(k)	0.035	0.050	0.099	0.052	0.103	0.389	0.026	0.001	0.096	0.102	
	250	*P(k)*	0.014	0.114	0.083	0.180	0.199	0.007	0.132	0.064	0.014	0.192	0.001
		μ(k)	0.874	0.124	0.061	0.002	0.045	0.438	-0.021	0.067	1.730	-0.047	
		σ(k)	0.005	0.075	0.077	0.082	0.020	0.001	0.075	0.073	0.295	0.037	
	300	*P(k)*	0.018	0.019	0.181	0.074	0.019	0.119	0.114	0.050	0.243	0.164	0.005
		μ(k)	0.862	1.088	0.024	0.039	-0.087	0.013	-0.050	0.131	0.011	0.009	
		σ(k)	0.358	0.333	0.061	0.059	0.112	0.060	0.115	0.070	0.035	0.056	
	350	*P(k)*	0.161	0.021	0.062	0.125	0.043	0.007	0.054	0.162	0.154	0.210	0.006
		μ(k)	0.003	0.596	0.068	-0.063	0.155	1.385	0.026	0.007	0.018	0.067	
		σ(k)	0.065	0.029	0.212	0.052	0.029	0.001	0.064	0.022	0.065	0.035	

Table A.III.4. *(continued)*

Attack/ Rank		Model parameters										Error
Sharpening	1	$P(k)$	0.081	0.081	0.180	0.029	0.131	0.156	0.077	0.091	0.088	0.085	0.018
		$\mu(k)$	-1.921	-2.014	-2.045	-0.221	-1.433	-2.842	-1.846	-2.809	-2.245	-1.657	
		$\sigma(k)$	1.083	0.821	0.169	1.900	0.205	0.210	0.930	1.178	0.812	0.999	
	50	$P(k)$	0.247	0.045	0.202	0.075	0.055	0.087	0.053	0.045	0.161	0.029	0.019
		$\mu(k)$	-0.052	-0.636	0.157	0.096	-0.612	-0.221	0.318	0.160	-0.025	0.090	
		$\sigma(k)$	0.090	0.283	0.101	0.165	0.280	0.046	0.182	0.211	0.086	0.016	
	100	$P(k)$	0.185	0.065	0.121	0.069	0.076	0.147	0.077	0.059	0.114	0.087	0.012
		$\mu(k)$	-0.074	0.214	0.022	0.046	0.072	-0.061	-0.335	-0.743	-0.021	0.170	
		$\sigma(k)$	0.083	0.180	0.083	0.123	0.014	0.089	0.189	0.327	0.096	0.189	
	150	$P(k)$	0.076	0.060	0.054	0.296	0.030	0.022	0.307	0.049	0.072	0.033	0.012
		$\mu(k)$	-0.091	0.147	0.123	-0.037	0.286	-0.526	0.011	-0.317	-0.192	-0.217	
		$\sigma(k)$	0.100	0.037	0.181	0.063	0.011	0.032	0.067	0.65	0.153	0.010	
	200	$P(k)$	0.200	0.101	0.078	0.056	0.073	0.141	0.090	0.130	0.024	0.108	0.009
		$\mu(k)$	0.015	-0.051	-0.410	-0.029	0.097	-0.054	-0.025	-0.089	-0.081	-0.042	
		$\sigma(k)$	0.088	0.133	0.235	0.130	0.111	0.11	0.108	0.032	0.366	0.130	
	250	$P(k)$	0.219	0.0556	0.014	0.142	0.173	0.025	0.068	0.065	0.117	0.123	0.013
		$\mu(k)$	-0.073	-0.083	-0.789	0.001	-0.049	-0.206	-0.217	-0.216	0.131	0.065	
		$\sigma(k)$	0.032	0.229	0.017	0.044	0.062	0.018	0.218	0.219	0.144	0.030	
	300	$P(k)$	0.040	0.065	0.057	0.215	0.118	0.166	0.057	0.070	0.034	0.177	0.011
		$\mu(k)$	-0.033	-0.295	-0.738	-0.064	-0.151	-0.023	0.053	0.162	-0.045	-0.045	
		$\sigma(k)$	0.111	0.178	0.538	0.048	0.082	0.077	0.018	0.075	0.419	0.067	
	350	$P(k)$	0.173	0.015	0.112	0.026	0.007	0.407	0.023	0.073	0.099	0.064	0.012
		$\mu(k)$	-0.056	-0.237	-0.160	0.151	0.278	-0.029	-0.984	0.089	-0.336	0.061	
		$\sigma(k)$	0.074	0.026	0.028	0.001	0.001	0.047	0.216	0.051	0.179	0.164	
StirMark	1	$P(k)$	0.146	0.088	0.099	0.102	0.099	0.098	0.094	0.069	0.113	0.091	0.035
		$\mu(k)$	-1.706	-0.316	-1.773	-1.323	-1.053	-1.532	0.137	-2.167	-1.341	-2.525	
		$\sigma(k)$	0.726	0.836	0.850	0.916	0.927	0.981	0.733	1.118	0.790	1.059	
	50	$P(k)$	0.049	0.151	0.041	0.134	0.138	0.060	0.134	0.136	0.107	0.049	0.019
		$\mu(k)$	0.792	0.179	-0.879	0.122	-0.173	1.799	0.140	-0.153	0.667	0.607	
		$\sigma(k)$	0.884	0.481	1.138	0.537	0.597	0.652	0.529	0.388	0.373	0.894	
	100	$P(k)$	0.043	0.185	0.048	0.046	0.110	0.043	0.348	0.058	0.044	0.075	0.017
		$\mu(k)$	-0.594	0.742	-0.131	-0.563	-0.245	-0.267	0.155	0.855	-0.339	1.545	
		$\sigma(k)$	0.362	0.226	0.596	0.404	0.045	0.558	0.135	0.770	0.533	0.699	
	150	$P(k)$	0.064	0.149	0.140	0.040	0.029	0.293	0.156	0.070	0.028	0.030	0.023
		$\mu(k)$	1.032	0.086	0.118	0.585	-0.629	0.341	-0.227	1.030	-0.814	1.113	
		$\sigma(k)$	0.586	0.369	0.361	0.632	0.828	0.269	0.339	0.586	0.022	0.009	
	200	$P(k)$	0.088	0.166	0.067	0.065	0.069	0.122	0.062	0.181	0.063	0.117	0.019
		$\mu(k)$	1.305	0.422	-0.016	0.511	-0.214	0.007	0.409	0.147	-0.069	0.240	
		$\sigma(k)$	0.399	0.257	0.720	0.615	0.290	0.374	0.630	0.360	0.721	0.335	
	250	$P(k)$	0.111	0.098	0.071	0.030	0.141	0.111	0.136	0.137	0.042	0.122	0.020
		$\mu(k)$	0.436	-0.048	0.496	-0.283	0.017	0.183	0.523	0.121	0.314	0.595	
		$\sigma(k)$	0.853	0.044	0.087	0.017	0.50	0.057	0.513	0.546	0.118	0.897	
	300	$P(k)$	0.046	0.201	0.053	0.030	0.083	0.092	0.179	0.032	0.029	0.253	0.018
		$\mu(k)$	-0.268	0.001	-0.064	1.577	-0.538	0.822	0.508	0.303	1.381	0.262	
		$\sigma(k)$	0.013	0.148	0.369	0.758	0.279	0.156	0.301	0.005	0.109	0.222	
	350	$P(k)$	0.074	0.196	0.014	0.212	0.131	0.088	0.116	0.048	0.007	0.749	0.021
		$\mu(k)$	0.109	-0.914	0.208	-0.220	0.013	1.013	0.569	0.841	-2.150	0.528	
		$\sigma(k)$	0.191	0.069	0.221	0.306	0.349	0.457	0.124	0.036	0.001	0.001	

A.III.2.2. Model Validation

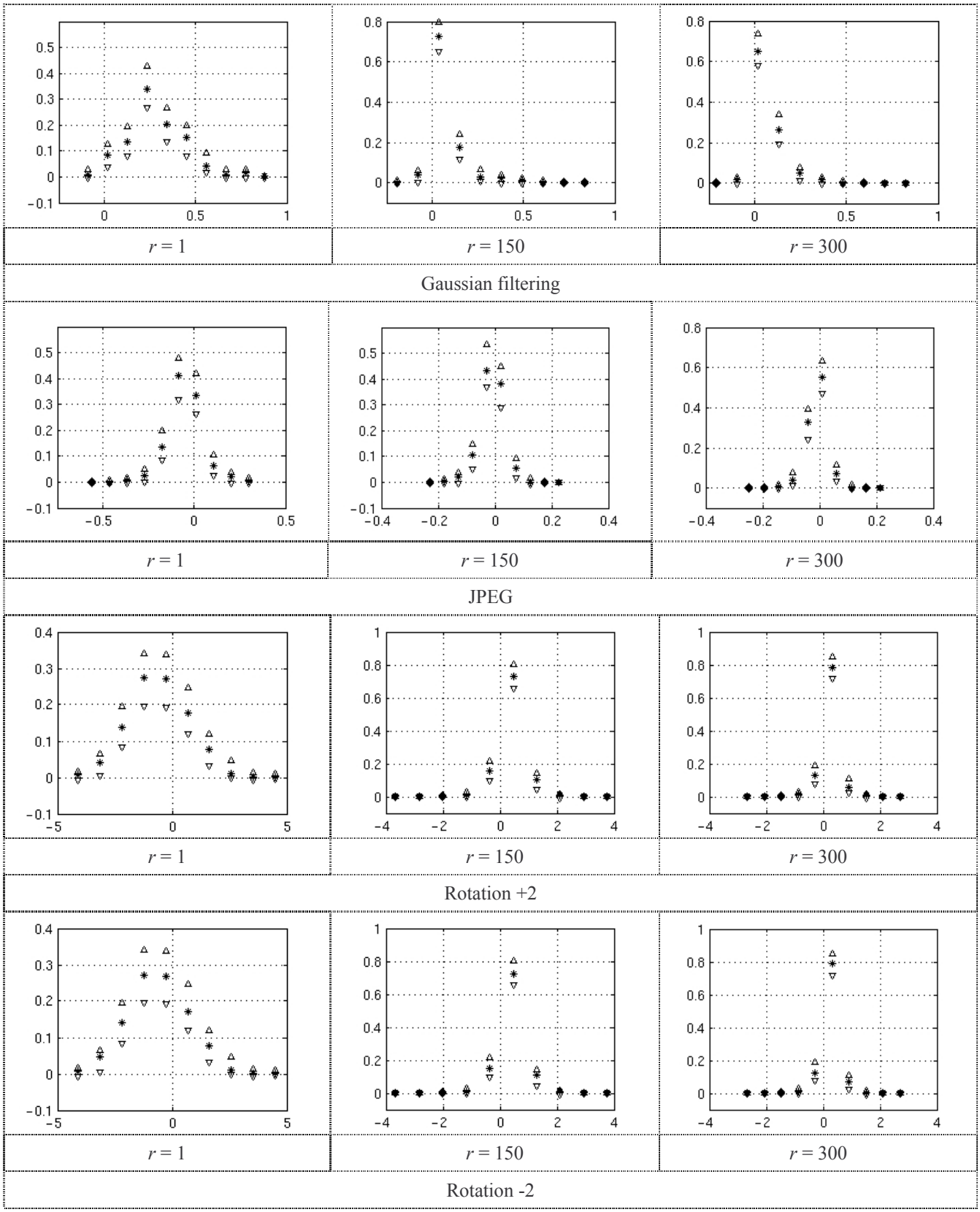


Figure A.III.3.a. Model validation: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by * (DCT on whole frames and low quality video).

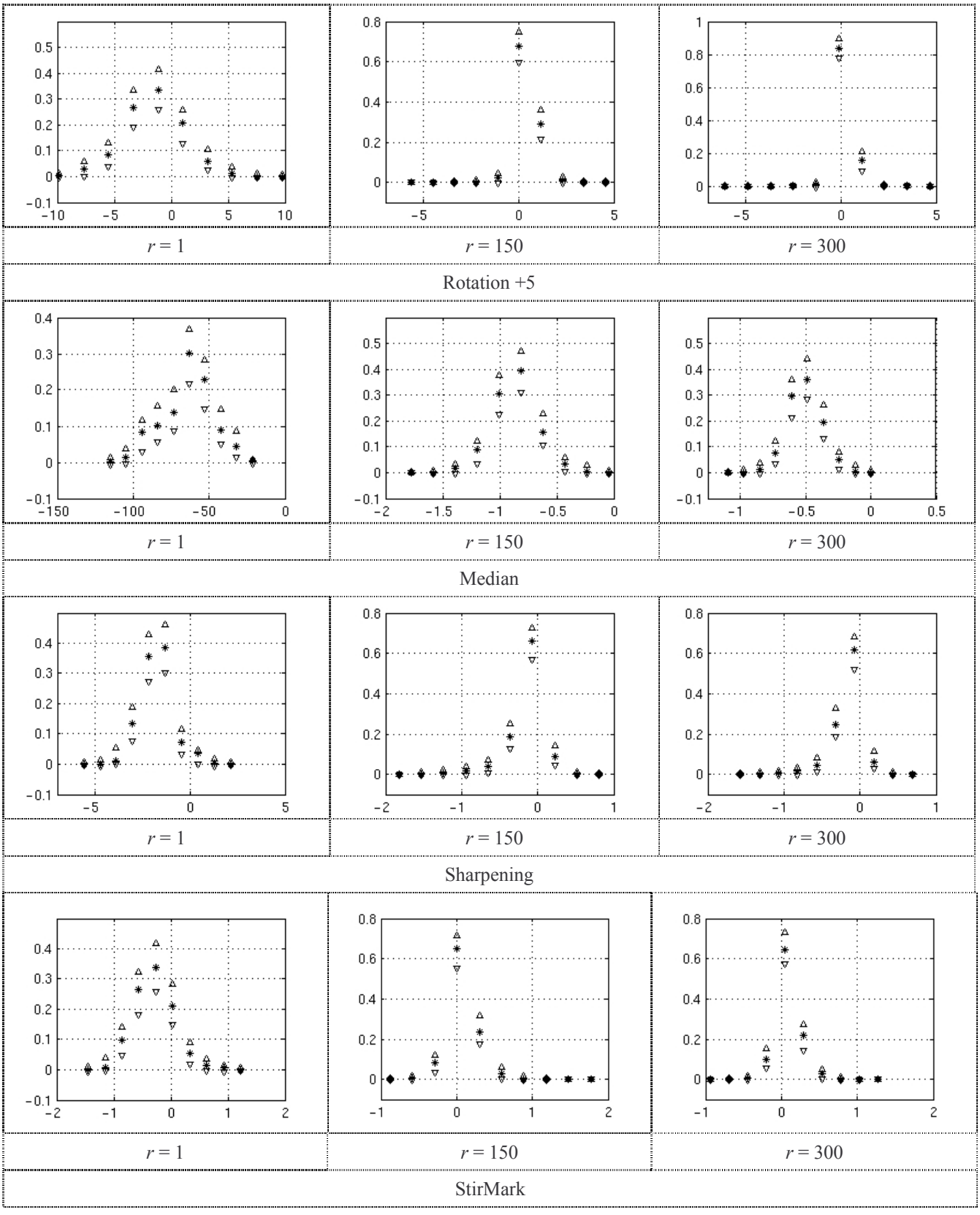


Figure A.III.3.b. Model validation: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by * (DCT on whole frames and low quality video).

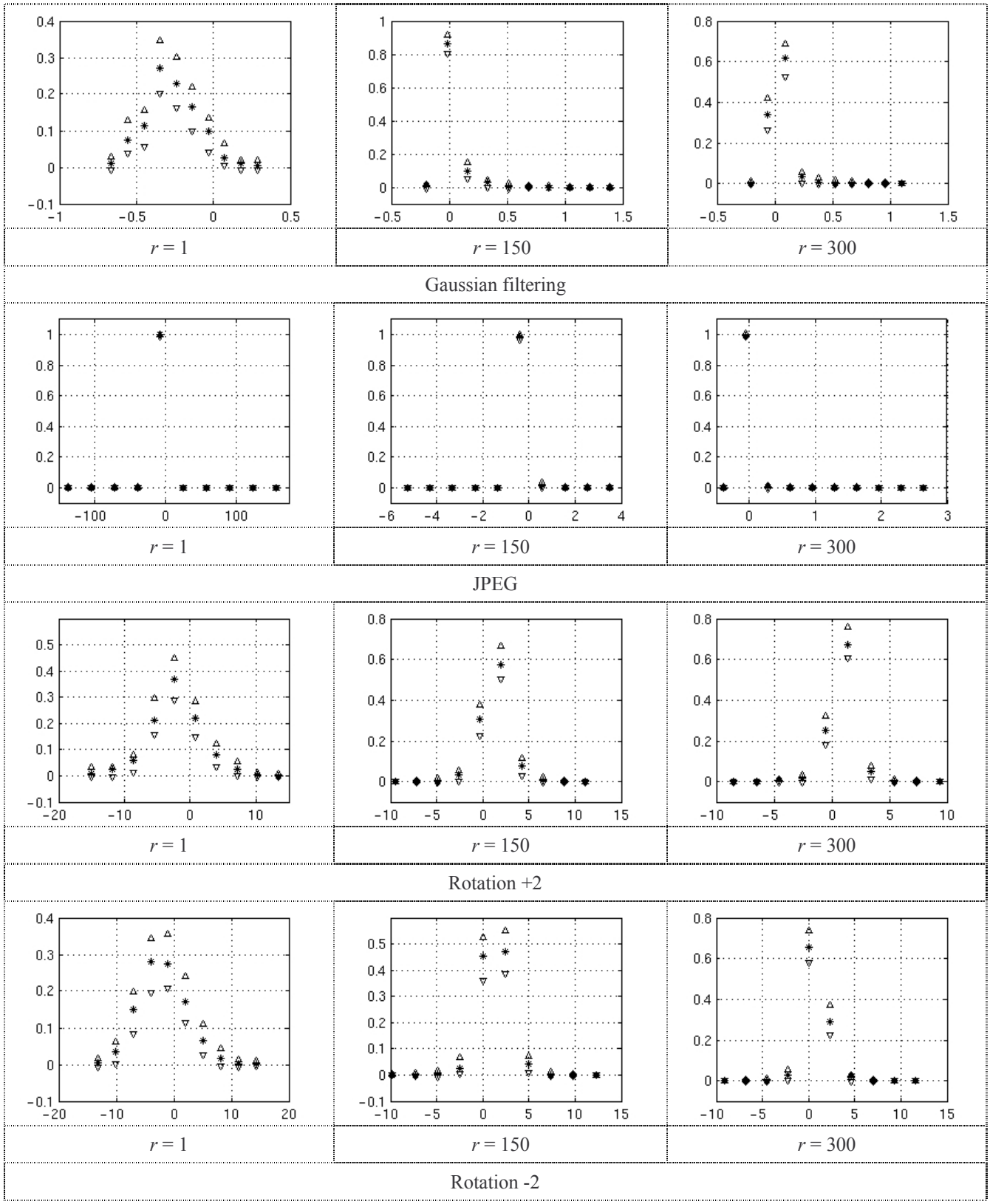


Figure A.III.4.a. Model validation: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by * (DCT on whole frames and high quality video).

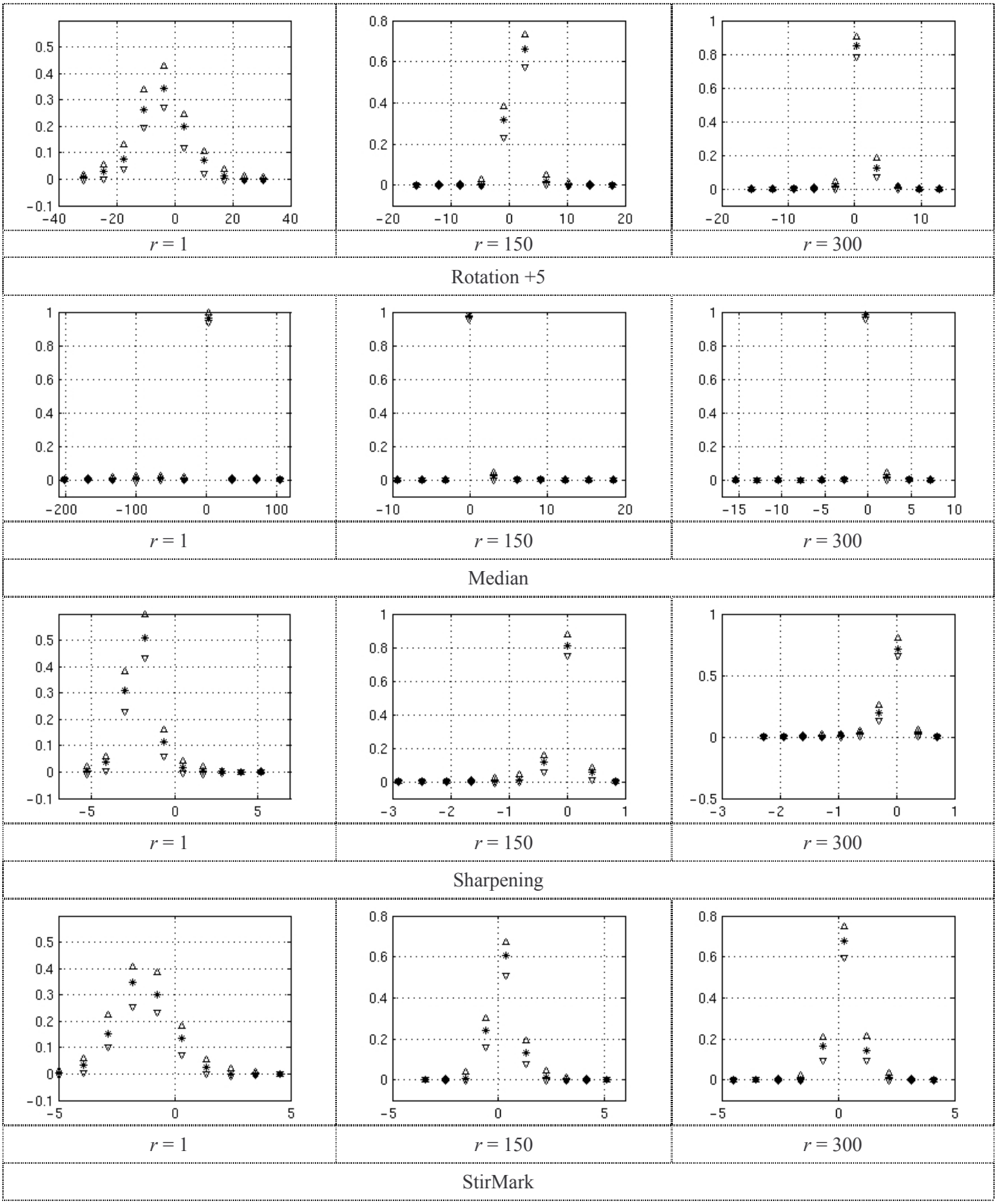


Figure A.III.4.b. Model validation: the 95% confidence limits are represented by ∇ and Δ, while the probabilities computed from model are represented by * (DCT on whole frames and high quality video).

Appendix IV

Shannon Capacity vs. Multimedia Applications

Table A.IV.1. Capacity limits for a video sequence (in bits per frame) for (9,7) DWT and SNR =30dB.

Attacks / C	Gaussian filtering	Rotation +2°	Rotation -2°	Rotation +5°	Sharpening	StirMark
C_{Gauss}	6.17	0.07	0.07	0.06	0.74	0.49
C_{low}	17.49	0.15	0.15	0.17	1.18	0.82
C_{upper}	267.87	180.54	181.64	199.19	125.75	177.07

Table A.IV.2. Capacity limits for a video sequence (in bits per frame) for three DWTs and SNR =30dB.

DWT	Attacks	C_{low}	C_{upper}	C_{Gauss}
(2,2)	Gaussian filtering	8.94	216.36	3.61
(4,4)		14.39	262.38	4.97
(9,7)		17.49	267.87	6.17
(2,2)	Rotation +2°	0.15	190.18	0.07
(4,4)		0.15	184.66	0.07
(9,7)		0.15	180.54	0.07
(2,2)	Rotation -2°	0.15	190.47	0.07
(4,4)		0.15	185.97	0.07
(9,7)		0.15	181.64	0.07
(2,2)	Rotation +5°	0.18	206.26	0.06
(4,4)		0.18	199.93	0.06
(9,7)		0.17	199.19	0.06
(2,2)	Sharpening	0.70	126.18	0.42
(4,4)		0.92	134.27	0.54
(9,7)		1.18	125.75	0.74
(2,2)	StirMark	0.64	193.72	0.38
(4,4)		0.68	181.88	0.40
(9,7)		0.82	177.07	0.49

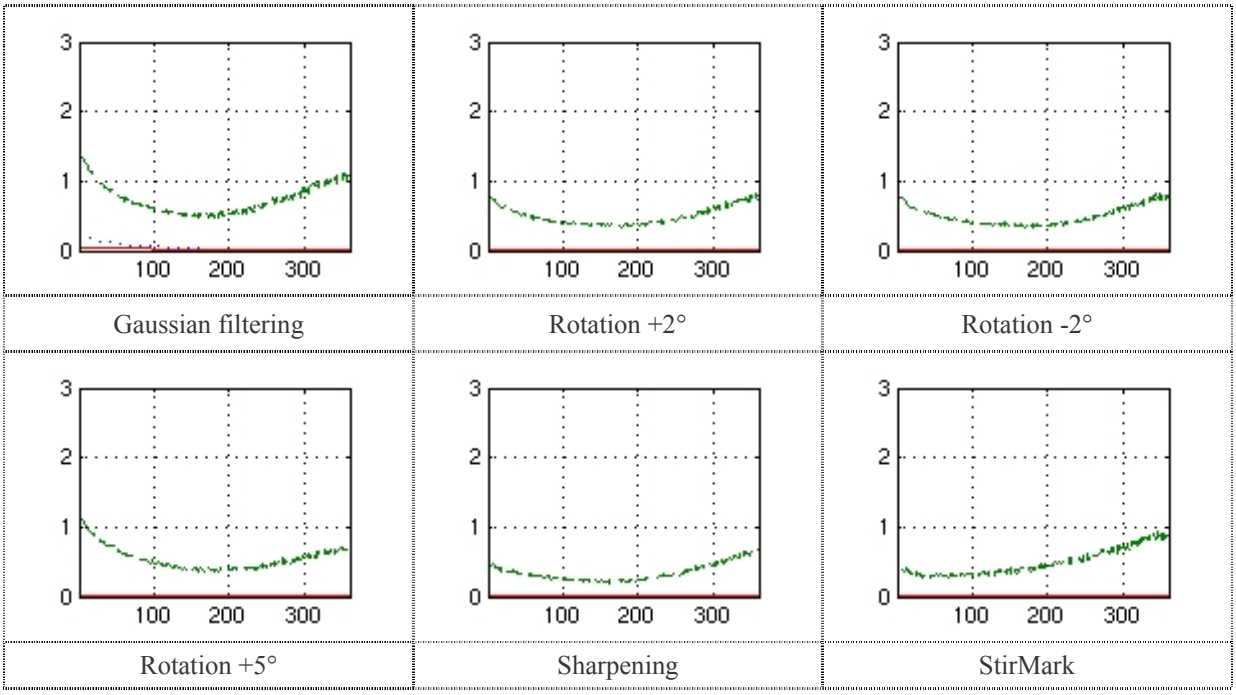


Figure A.IV.1. The capacity limits for (9,7) DWT whole frame and SNR=30dB. The upper limit (dashed line), lower limit (continuous line) and equivalent Gaussian white noise case capacity (dots) in bits per frame (ordinate) *versus* its hierarchy rank (abscissa) for six attacks.

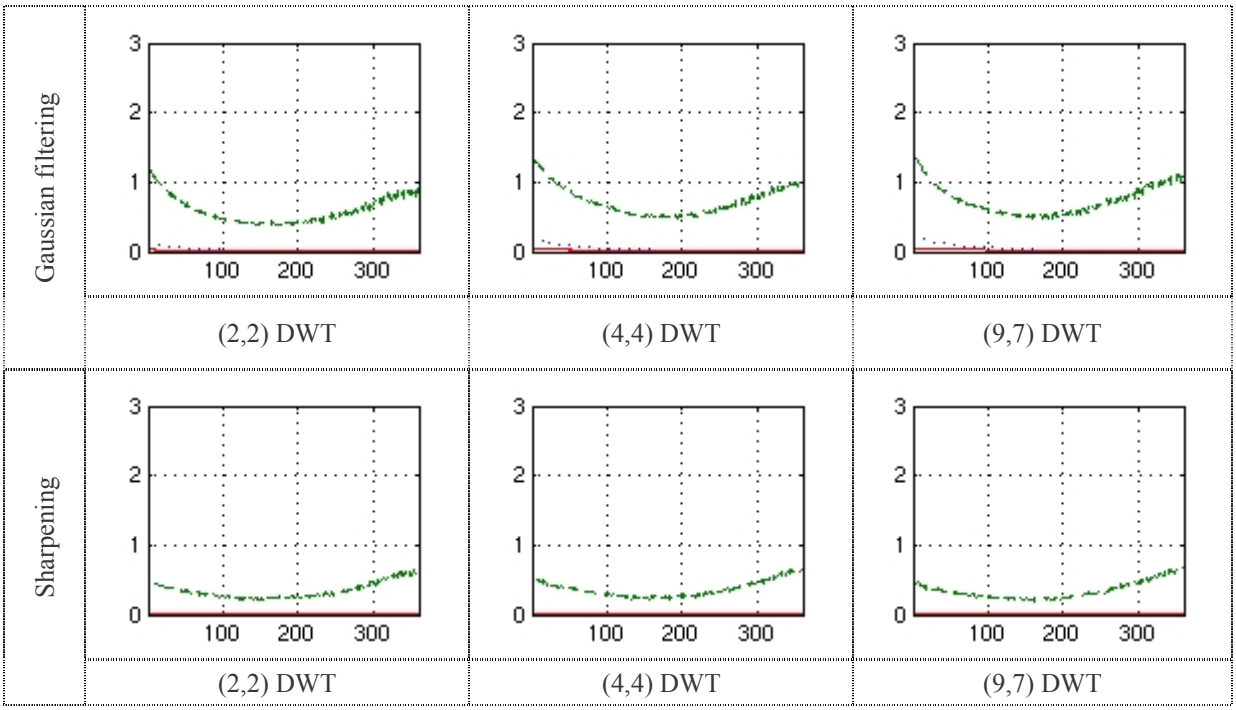


Figure A.IV.2.a. The capacity limits for three DWTs applied on whole frame and SNR=30dB. The upper limit (dashed line), lower limit (continuous line) and equivalent Gaussian white noise case capacity (dots) in bits per frame (ordinate) *versus* its hierarchy rank (abscissa) for Gaussian filtering and sharpening.

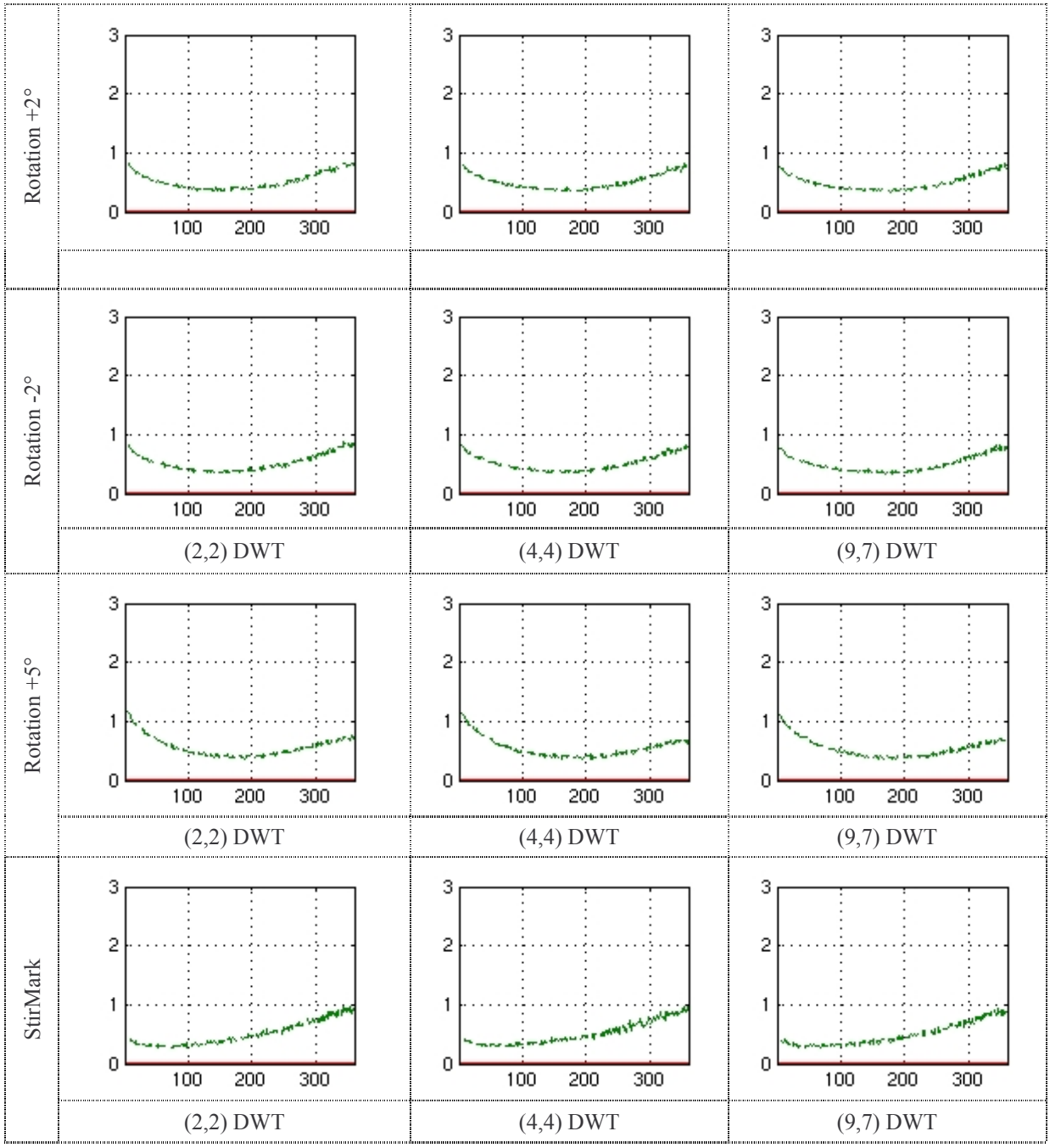


Figure A.IV.2.b. The capacity limits for three DWTs applied on whole frame and SNR=30dB. The upper limit (dashed line), lower limit (continuous line) and equivalent Gaussian white noise case capacity (dots) in bits per frame (ordinate) *versus* its hierarchy rank (abscissa) for different attacks.

Table A.IV.3. Capacity limits (in bits per frame) for DCT and SNR =30dB.

Attacks / *C*	Gaussian filtering	Rotation +2°	Rotation -2°	Rotation +5°	Sharpening	StirMark
C_{Gauss}	11.19	0.95	0.95	0.34	2.17	5.77
C_{low}	30.19	1.24	1.24	0.60	4.13	7.10
C_{upper}	354.23	183.40	183.40	241.26	198.58	192.58

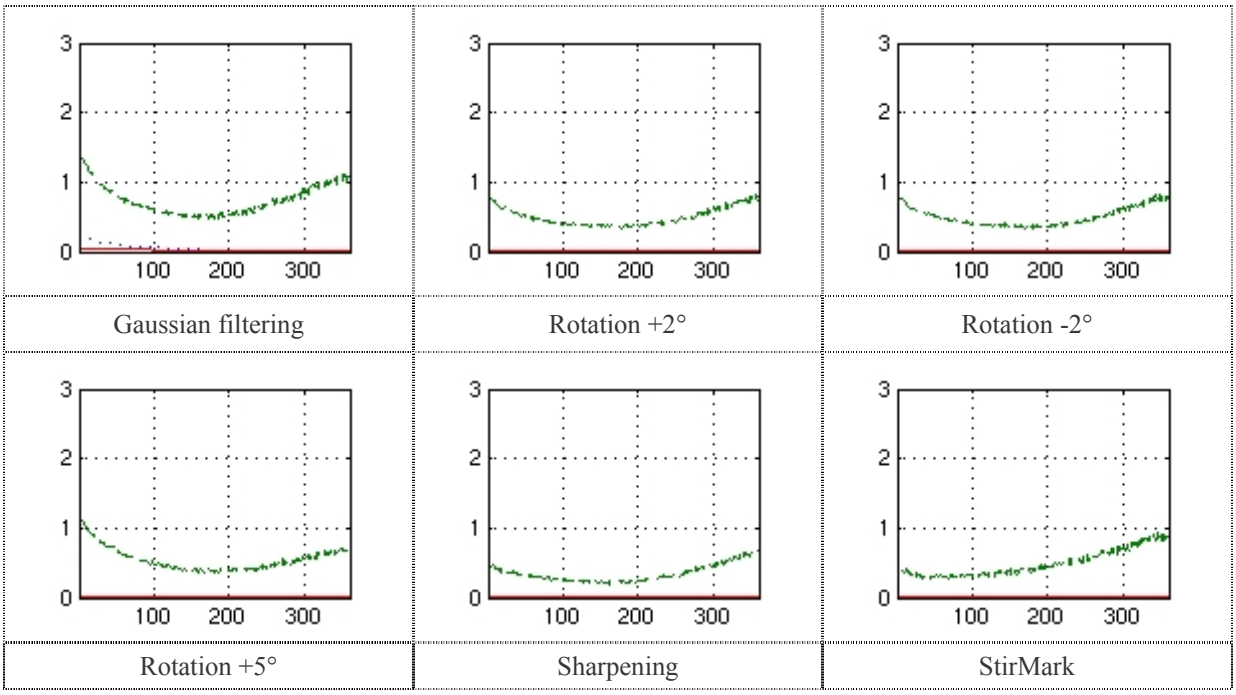


Figure A.IV.3. The capacity limits for DCT whole frame and SNR=30dB. The upper limit (dashed line), lower limit (continuous line) and equivalent Gaussian white noise case capacity (dots) in bits per frame (ordinate) *versus* its hierarchy rank (abscissa) for six attacks.

Table A.IV.4. Emerging watermarking applications beyond copyright protection field (Shannon limits, SNR =30dB and (9,7) DWT transform applied on whole frames).

Application/ Requirements	Attacks	Indexing 1 [bits/frame]	Subtitles 5 [bits/frame]	Hyper video 100 [bits/frame]	Interactive Digital TV 400 [bits/frame]
Viability of the (9,7) DWT transform	Filtering	**Yes** (lower limit)	**Yes** (lower limit is enough for Gaussian filtering, but not for sharpening, where it is necessary to consider the upper limit)	**No** (lower limit) **Yes** (upper limit)	**No**
	Rotations	**No** (lower limit) **Yes** (upper limit)			**No**
	StirMark	**No** (lower limit) **Yes** (upper limit)			**No**

Table A.IV.5. Emerging watermarking applications beyond copyright protection field (Shannon limits, SNR =30dB and DCT transform applied on whole frames).

	Attacks	Indexing 1 [bits/frame]	Subtitles 5 [bits/frame]	Hyper video 100 [bits/frame]	Interactive Digital TV 400 [bits/frame]
Viability of the DCT transform	Filtering	**Yes** (lower limit)	**Yes** (lower limit is enough for Gaussian filtering, but not for sharpening where it is necessary to consider the upper limit)	**No** (lower limit) **Yes** (upper limit)	**No**
	Rotations	**Yes** (lower limit)	**No** (lower limit) **Yes** (upper limit)	**No** (lower limit) **Yes** (upper limit)	**No**
	StirMark	**Yes** (lower limit)	**Yes** (lower limit)	**No** (lower limit) **Yes** (upper limit)	**No**

References

Published Papers

O. Dumitru, M. Mitrea, F. Prêteux, *"Probability Density Function Estimation for Natural Video Modelling in the DCT Domain"*, ELSEVIER – Computational Statistics & Data Analysis, manuscript submitted December 2009.

A. Chammen, **O. Dumitru**, M. Mitrea, F. Prêteux, *"Simulations Monte Carlo pour le tatouge robuste des séquence vidéo",* TAIMA 2009, Hammamet–Tunisia, May 2009, pp. 501-507.

O. Dumitru, M. Mitrea, F. Prêteux, *"Wavelet-based Video Modelling"*, The 50th International Symposium - ELMAR 2008, Zadar–Croatia, September 2008, pp. 105-108.

O. Dumitru, M. Mitrea, F. Prêteux, *"Theoretical Limits in DWT Video Watermarking"*, Proc. SPIE Vol. 7075, San Diego–USA, August 2008, pp. 7075 0C: 1-9.

M. Mitrea, **O. Dumitru**, S. Duță, F. Prêteux, A. Vlad, *"A Comparative Study on Video Watermarking Capacity",* Communications 2008, Bucharest–Romania, June 2008, pp. 335-338.

O. Dumitru, M. Mitrea, F. Prêteux, *"DCT Domain Video Watermarking: Attack Estimation and Capacity Evaluation"*, ICINCO 2008, Funchal–Portugal, May 2008, pp. 239-244.

O. Dumitru, M. Mitrea, F. Prêteux, *"Video Modelling in the DWT Domain"*, Proc. SPIE, Vol. 7000, Strasbourg–France, April 2008, pp. 7000 0P: 1-12.

O. Dumitru, M. Mitrea, F. Prêteux, A. Pathak, *"Probability Density Function Estimation for Video in the DCT Domain"* Proc. SPIE, Vol. 6812, San Jose–USA, January 2008, pp. 6812 0L: 1-9.

O. Dumitru, S. Duță, M. Mitrea, F. Prêteux, *"Gaussian Hypothesis for Video Watermarking Attacks: Drawbacks and Limitations"*, EUROCON 2007, Warsaw–Poland, September 2007, pp. 849-855.

O. Dumitru, M. Mitrea, F. Prêteux, *"Accurate Watermarking Capacity Evaluation",* Proc. SPIE, Vol. 6763, September 2007, Boston–USA, pp. 6763 03: 1–12.

M. Mitrea, **O. Dumitru**, F. Prêteux, A. Vlad, *"Zero Memory Information Sources Approximating to Video Watermarking Attacks",* Lecture Notes in Computer Science (LNCS), Vol. 4707, Part III, 2007, pp. 445–459.

O. Dumitru, M. Mitrea, F. Prêteux, *"Capacité du Tatouage Vidéo dans le Domaine des Ondelettes: Analyse Comparée",* TAIMA 2007, Hammamet–Tunisia, May 2007, pp. 355-360.

M. Mitrea, **O. Dumitru**, F. Prêteux, "*Video Watermarking Capacity in the DWT Hierarchy",* Proc. SPIE, Vol. 6576, Orlando–USA, April 2007, pp. 6576 0E: 1-10.

O. Dumitru, S. Duță, M. Mitrea, F. Prêteux, *"Commercial and Standard in Robust Multimedia Watermarking"*, WSEAS Transactions on Communications, Issue 9, Vol. 5, September 2006, pp. 1667 – 1673.