

UNIVERSITE MONTPELLIER II
SCIENCES ET TECHNIQUES DU LANGUEDOC

Habilitation à Diriger des Recherches

Discipline : *Informatique*

Ecole Doctorale : *Information, Structures, Systèmes*

présentée et soutenue publiquement
par

Marc CHAUMONT

TITRE :

**Schémas de tatouage d'images,
schémas de tatouage conjoint à la compression,
et schémas de dissimulation de données**

Soutenue le ... devant la commission d'examen

POSSIBLE COMPOSITION DU JURY

M.	Marc	ANTONINI	DR. CNRS
M.	Atila	BASKURT	Pr. INSA Lyon
M.	Jean-Pierre	GUEDON	Pr. Polytech Nantes
M.	Claude	LABIT	DR. INRIA
M.	Alain	JEAN-MARIE	DR. INRIA
M.	William	PUECH	Pr. IUT Béziers

Aux tétards et à la maman des tétards.

Remerciements

Table des matières

Tables des matières	7
Table des figures	11
Curriculum Vitae	13
Titres Universitaires et parcours	13
Brève biographie	14
Recherche	14
Résumé des travaux de recherche (période 2005-2011)	14
Publications principales	16
Encadrements	17
Encadrements de doctorants depuis 2005	17
Encadrements de stagiaires de Master 2 Recherche	18
Rayonnement et animation scientifique	19
Relecture de plus de 100 articles entre 2005 et fin 2011 (conférences ou revues)	19
Président de session de conférences	19
Commissions de spécialistes	19
Responsabilités diverses	20
Expertises	20
Tutoriaux à audience nationale et exposés de vulgarisation	20
ANR	20
Collaborations académique et industrielle	20
Prime	21
Enseignement	21
Responsabilités	21
Matière	21
Publications	21
Revue	22
Chapitre de livre	22
Tutoriaux dans des conférences internationales	22
Congrès internationaux avec actes	22
Congrès nationaux	25
Thèse	26
Introduction	29

I	Le tatouage d'images fixes	31
1	Le tatouage robuste d'images	33
1.1	Schéma général de tatouage informé	33
1.2	Quelques applications utilisant le tatouage numérique	34
1.3	L'espace d'insertion, le codage informé et l'insertion informée	36
1.4	Broken Arrows	37
2	Une approche DPTC basée rotation : RB-DPTC	41
2.1	Introduction	41
2.2	Le schéma DPTC original	42
2.3	Notre schéma RB-DPTC	43
2.3.1	L'espace d'insertion	43
2.3.2	L'étape d'insertion informée	44
2.3.3	Une extension psychovisuelle	46
2.4	Evaluations expérimentales	47
2.5	Conclusion	50
3	Une approche QIM Perceptuel (Multi-Hyper-Cube)	53
3.1	Introduction	53
3.2	L'algorithme Hyper-Cube	54
3.2.1	Slack de Watson modifiée	55
3.2.2	Codage / décodage du message	56
3.2.3	Calcul des slacks sur un voisinage	57
3.3	Utilisation de la TCQ : l'algorithme Multi-Hyper-Cube	59
3.4	Résultats	61
3.5	Conclusion	63
II	Le tatouage conjointement à la compression	67
4	Le tatouage conjoint à la compression	69
4.1	Brève présentation de JPEG2000	69
4.2	Tatouage dans JPEG2000	70
4.3	Brève présentation de H.264	71
4.4	Tatouage dans un flux vidéo	72
5	Une approche de tatouage conjointe à la compression JPEG2000	75
5.1	Introduction	75
5.2	La Quantification Codée par Treillis (TCQ)	75
5.3	Le schéma conjoint proposé	77
5.3.1	La méthode de dissimulation de données proposée	77
5.3.2	Le schéma conjoint JPEG2000 et dissimulation de données	79
5.3.3	La sélection des coefficients	80
5.4	Résultats expérimentaux	81
5.4.1	Évaluation des performances de <i>data-hiding</i> du schéma conjoint	81
5.4.2	Évaluation des performances de compression du schéma conjoint	82

5.4.3	Évaluation pratiques des coûts calculatoire	82
5.4.4	Vers un schéma robuste	82
5.5	Conclusion	85
6	Une approche avec prise en compte de l'optimisation RD dans H.264	87
6.1	Quelques mots sur le traçage de traîtres	87
6.2	Le tatouage dans H.264 (insertion d'un mot de code issu du code de Tardos) . . .	89
6.3	Résultats	91
6.4	Conclusion	93
III	La dissimulation de la couleur d'une image ; Une étude curieuse	95
7	Les deux grandes familles de dissimulation de la couleur	97
7.1	Introduction	97
7.2	L'approche de De Queiroz et Braun	98
7.3	L'approche de Campisi <i>et al.</i>	101
7.4	Les approches basées palettes	102
8	Une approche par modélisation floue et optimisation	105
8.1	Modélisation du problème de décomposition et résolution	105
8.2	La méthode d'insertion de données cachées	107
8.3	Résultats	107
8.4	Conclusion	110
9	Une approche par heuristique étendue au cas 512 couleurs	113
9.1	Introduction	113
9.2	La décomposition de l'image	114
9.2.1	La quantification couleur	114
9.2.2	L'algorithme de parcours en couches	115
9.2.3	Construction du message	116
9.3	Schéma de tatouage réversible	117
9.3.1	L'état <i>embarquant</i>	118
9.3.2	L'état <i>à-corriger</i>	118
9.3.3	L'état original	119
9.3.4	Les algorithmes de « codage » et de « décodage »	119
9.4	Résultats	120
9.5	Conclusion	122
Conclusion		125
Le tatouage robuste		125
Le tatouage conjoint à la compression		126
La dissimulation de données		127
Projet de recherche		129
Bibliographie		134

IV Quelques publications	145
Article revue IEEE Transactions on CSVT'2011	147
Article revue Springer SIVP'2011	164
Article revue Springer Annals Of Telecommunications 2011	183
Article I&ST SPIE'2009	198
Article IEEE ICME'2008	207
Article I&ST SPIE'2007	211

Table des figures

1.1	Schéma général de l'insertion.	34
1.2	Schéma général d'extraction aveugle.	34
1.3	Schéma détaillé de l'insertion informée.	36
1.4	Schéma général de tatouage par Broken Arrows.	38
2.1	Dirty Paper Trellis Codes appliqués sur une image 240×368	42
2.2	L'espace d'insertion pour notre schéma RB-DPTC.	44
2.3	Insertion basée rotation dans le plan de Miller, Cox et Bloom	45
2.4	Embedding scheme with a psychovisual mask	46
2.5	BER pour une attaque par ajout de bruit gaussien.	48
2.6	BER pour une attaque par filtrage gaussien.	49
2.7	BER pour une attaque valométrique de changement d'échelle.	49
2.8	BER pour une attaque par compression JPEG.	50
3.1	Schéma général de P-QIM pour un bloc de 8×8 pixels.	55
3.2	Machine à état du codeur convolutif 1/8-taux 2-mémoires.	56
3.3	Image 1 de la base de donnée BOWS-2 tatouée avec P-QIM et Hyper-Cube.	58
3.4	Position des blocs B et D par rapport à $\mathbf{X}^{filtré}$	59
3.5	La $i^{ème}$ transition dans un treillis à 4 états.	60
3.6	Illustration d'un réseau (<i>lattice</i>) pour un treillis à quatre états.	60
3.7	BER pour une attaque valométrique de changement d'échelle.	62
3.8	BER pour une attaque par compression JPEG.	63
3.9	BER pour une attaque par ajout de bruit gaussien.	64
3.10	BER pour une attaque par filtrage gaussien.	65
4.1	Schéma reprenant les principales étapes d'un codeur JPEG2000.	69
4.2	Schéma reprenant les grandes étapes d'un codeur basé bloc.	71
4.3	Les prédictions utilisées pour un bloc 4×4 dans H.264/AVC.	72
4.4	Diagramme détaillé des étapes d'un codeur H.264.	72
5.1	La structure du treillis utilisé dans JPEG2000.	76
5.2	Quantificateurs d'union définis dans JPEG2000.	77
5.3	Les principes de la QIM appliqués aux quantificateurs d'union de JPEG2000.	78
5.4	Schéma du système conjoint JPEG2000/dissimulation de données.	79
6.1	Schéma général d'insertion d'un mot de code de Tardos dans H.264.	90

6.2	Schéma général du système de tatouage intégré à H.264	91
7.1	Illustration d'une utilisation possible de la protection de la couleur.	98
7.2	Dissimulation de la couleur dans l'approche De Queiroz et Braun.	99
7.3	Extraction de l'image couleur dans l'approche de De Queiroz et Braun.	100
7.4	Application de l'approche de De Queiroz et Braun.	100
7.5	Dissimulation de la couleur dans l'approche de Campisi <i>et al.</i>	101
7.6	Extraction de l'image couleur dans l'approche de Campisi <i>et al.</i>	102
7.7	Application de l'approche de Campisi <i>et al.</i>	103
7.8	Décomposition d'une image couleur en une image d' <i>index</i> et une image couleur. .	104
8.1	Quelques étapes de l'approche par modélisation floue	108
8.2	Histogrammes.	109
9.1	Schéma général insertion/extraction avec une palette de 512 couleurs.	114
9.2	Vue du parcours en couches dans le cube RGB.	116
9.3	Étapes de l'approche 512 couleurs.	121
9.4	Exemple sur l'image <i>barbara</i>	122

Curriculum Vitae



Nom : CHAUMONT
Prénom : Marc
Grade : Maître de conférences
Etablissement : Université de Nîmes
Section CNU : 27
Date de naissance : 08/11/1976
Nationalité : Française
Situation familiale : Marié, 4 enfants
Adresse électronique : marc.chaumont@lirmm.fr
Adresse web : www.lirmm.fr/~chaumont

Unité d'appartenance :

LIRMM, UMR 5506, équipe ICAR (responsable William Puech). Le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) est une Unité Mixte de Recherche de l'Université Montpellier II (UMII) et du Centre National de la Recherche Scientifique (CNRS).

Activité de recherche :

Mots Clefs : Tatouage, Stéganographie, Sécurité, Compression, Segmentation, Suivi dans les vidéos.

Titres Universitaires et parcours

- Depuis septembre 2005 : Maître de conférences à l'université de Nîmes.
- Septembre 2004 - juillet 2005 : $\frac{1}{2}$ -ATER, à l'université de Pau et des Pays de l'Adour, IUT de Bayonne.
- Septembre 2003 - juillet 2004 : $\frac{1}{2}$ -ATER à l'université de Rennes1.
- Novembre 2003 : Doctorat de l'université de Rennes1 mention informatique avec la mention très honorable, « Représentation en objets vidéo pour un codage progressif et concurrentiel des séquences d'images ». Encadrement : S. PATEUX et H. NICOLAS (équipe TEMICS/IRISA). Soutenance le 13 novembre 2003 à l'IRISA. Jury : M. Claude LABIT (président), directeur de recherches à l'IRISA, M. Michel BARLAUD (rapporteur), professeur des universités à l'université de Nice, M. Jean-Marc CHASSERY (rapporteur), directeur de recherches au LIS, M. Atilla BASKURT (examineur), professeur des universités à l'université de Lyon, MM. Cécile DUFOUR (examinatrice), ingénieur de recherche Philips Digital Systems Laboratories Paris, M. Henri NICOLAS (directeur de travaux), chargé de recherches à l'IRISA, M. Stéphane PATEUX (co-directeur de travaux), chargé de recherches à l'IRISA.

- Juillet 2000 : DEA, Diplôme d'étude approfondie de l'IFSIC, Université de Rennes 1, option informatique.
- Juillet 1999 : Diplôme d'ingénieur de l'INSA de Rennes, option informatique.

Brève biographie

Diplômé de l'INSA de Rennes (Institut National des Sciences Appliquées) en 1999 avec l'option informatique, j'ai ensuite effectué mon service militaire à Saint-Cyr Coëtquidan en tant qu'enseignant scientifique du contingent. Durant la même période, j'ai obtenu un diplôme de DEA à l'IFSIC (Institut de Formation Supérieure en Informatique et Communication) de Rennes dans la filière réseaux. Le 13 novembre 2003, j'ai obtenu le diplôme de docteur de l'université de Rennes 1, mention informatique. La thèse portait sur la segmentation et la compression (dynamique et scalable) des objets vidéo dans le cadre de la compression vidéo. Durant l'année 2003-2004 j'ai occupé un poste de 1/2 ATER (Attaché Temporaire de l'Enseignement et de la Recherche) à l'IFSIC tout en poursuivant mon activité de recherche dans l'équipe TEMICS (TraitemEnt, Modélisation d'Images & CommunicationS) à l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) de Rennes. Durant l'année 2004-2005 j'ai occupé un poste de 1/2 ATER à l'IUT de Bayonne. Ma recherche a porté sur le suivi de visages par modèle 3D au sein de l'équipe TITS (Traitement d'Image, Traitement de Signal) dirigée par Franck Luthon au LIUPPA (Laboratoire Informatique Université Pau et Pays de l'Adour). En septembre 2005 j'ai été nommé enseignant-chercheur à l'université de Nîmes et j'ai intégré le LIRMM (Laboratoire d'Informatique, de Robotique et de Microelectronique de Montpellier). Mes domaines de recherche portent sur le tatouage de données (images, vidéo, modèle 3D), la stéganographie, la compression vidéo, et dans une moindre mesure la segmentation et le suivi dans les vidéos.

Recherche

Résumé des travaux de recherche (période 2005-2011)

Depuis mon intégration au LIRMM en 2005, mon axe principal de recherche s'intègre dans le domaine de la dissimulation de données dans les images et les vidéos. Ce domaine couvre de nombreux enjeux comme la protection des droits d'auteur, la sécurisation du multimédia, et l'enrichissement de médias par des métadonnées intégrées au support. Durant ces 6 années, j'ai développé trois axes de recherche dans ce domaine de la dissimulation de données.

Tout d'abord, je me suis intéressé à la dissimulation de données non robuste, c'est-à-dire sensible aux dégradations usuelles que subit une image (compression, modification du contraste, etc). Dans ce cadre, j'ai proposé une approche de dissimulation de la couleur d'une image dans une image en niveaux de gris. Toutes les propositions dans ce domaine sont des variations autour de la substitution de sous-bandes ondelettes de luminance par des sous-bande de chrominance. Une image couleur peut en effet se décomposer en trois images : une image de luminance et deux images de chrominance. L'information de couleur, présente dans les 2 images de chrominance, peut se dissimuler dans l'image de luminance : on applique une décomposition en ondelettes de l'image de luminance, et on remplace certaines sous-bandes de l'image de luminance par les images sous-échantillonnées de chrominance. Notre approche est fondamentalement différente : elle se base sur la quantification couleur d'une image et la dissimulation de la palette couleur dans

l'image en niveaux de gris. L'image couleur est vue comme une image d'index pointant vers une palette de couleurs. Notre proposition est de ré-organiser la palette et de la dissimuler dans l'image index. Deux variantes ont été proposées :

La première variante repose sur la formalisation du problème sous la forme d'une équation que l'on cherche à minimiser, puis par l'expression de la solution de manière formelle. L'approche donne une solution élégante permettant d'obtenir des résultats en distorsions similaires à ceux obtenus par les meilleures approches par substitution dans les ondelettes [Chaumont et al. 07c].

La seconde variante permet la dissimulation de 512 couleurs (au lieu de 256 couleurs) dans une image en niveaux de gris sur 8 bits [Chaumont et al. 09b]. Pour cela, nous quantifions l'image sur 512 couleurs, nous réarrangeons la palette pour que le mécanisme de tatouage soit possible, puis nous effectuons une dissimulation de la palette par un mécanisme de tatouage réversible. La qualité des images reconstruite dépasse de 2 dB les autres approches de l'état de l'art.

Depuis quelques années, il y a une grande effervescence autour de la dissimulation de la couleur dans une image en niveaux de gris. Depuis 2002, la majeure partie des approches consiste à faire de la substitution des sous-bandes d'ondelettes de luminance par des sous-bandes d'ondelettes des chrominances. Notre approche forme un courant alternatif aux autres approches et donne en général de meilleurs résultats.

Ma première contribution à la dissimulation de données aborde le volet non robuste de la dissimulation. Par la suite, je me suis intéressé au problème de la dissimulation robuste de données. J'ai développé deux grands axes de recherche : d'une part la comparaison rigoureuse des deux grands courants pour la dissimulation de données : approches treillis et approches quantification ; d'autre part l'étude de l'apport de la dissimulation conjointe à la compression.

La littérature est foisonnante d'approches de dissimulation robuste, avec des principes et des fondements théoriques très variés. La plupart de ces approches sont validées par des expérimentations sur des données artificielles et pas des images.

Un objectif que j'ai poursuivi est l'étude de l'application de ces approches dans le domaine de l'imagerie numérique. Cette étude a débuté par l'analyse précise de l'ensemble des propositions existantes, et la sélection de celles les plus adéquates pour les caractéristiques propres au domaine de l'image. Deux grandes catégories d'approches pertinentes pour les images ont ainsi pu être identifiées. Les deux approches théoriquement reconnues comme les meilleures du point de vue de la robustesse, l'une basée treillis [Chaumont 10a] et l'autre basée quantification [Chaumont et al. 11a], ont été réimplémentées et améliorées par l'intégration des spécificités propres à leur application à des images, puis testées dans les mêmes conditions sur des bases d'images significatives. Dans ces travaux, un objectif était de se positionner dans le contexte d'une insertion à haut débit, c'est-à-dire d'une grande quantité d'informations, alors que la plupart des travaux existants portent sur une insertion à bas débit.

Les résultats obtenus montrent que lorsque les conditions expérimentales sont similaires, les deux schémas ont des performances similaires. L'approche par treillis se comporte mieux vis-à-vis de l'attaque valométrique de changement d'échelle et est plus sûre. L'approche basée quantification est en revanche moins complexe en coût de calcul. La similitude des performances des deux approches laisse ouverte les perspectives d'amélioration pour chacun des deux schémas. Les approches basées treillis ont un énorme potentiel puisque le paramétrage, le code en lui-même, et l'apport des codes correcteurs sont insuffisamment étudiés. Les approches par quantification gagneraient à introduire le principe de codage turbo proposé dans T-TCQ et une plus grande sécurité.

La deuxième partie de mes travaux dans le domaine de la dissimulation robuste de données visait à répondre à la question : la dissimulation robuste de données conjointe à la com-

pression apporte-t-elle de la robustesse ou un gain de complexité calculatoire par rapport à une dissimulation indépendante de la compression ? Cette question a été abordée d'une part pour les images (JPEG2000) [Goudia et al. 11b] [Goudia et al. 12] et d'autre part pour les vidéos (H.264) [Shahid et al. 11a] [Shahid et al. 10b].

Mon expertise en matière de compression vidéos (issue de mes travaux de doctorat) a permis de proposer rapidement une approche de dissimulation conjointe à la compression qui a servi de base à ces travaux. Dans cette approche, la dissimulation a lieu lors de la phase de quantification, de manière parfaitement intégrée au mécanisme de compression, précisément lors de la phase d'optimisation débit-distorsion de la compression. Cette approche permet, pour un débit donné, d'avoir la dégradation la plus faible possible dans la vidéo tatouée (c'est-à-dire avec insertion de données). Pour la dissimulation conjointe à la compression dans les images, une nouvelle approche a été développée, intégrant les données à cacher lors de la phase de quantification.

L'expérimentation de ces approches de dissimulation robuste conjointes à la compression, pour les images et pour les vidéos, ont permis de tirer comme conclusion que le seul avantage de la dissimulation conjointe à la compression est une réduction de la complexité calculatoire du mécanisme global : notamment, une approche conjointe ne permet pas de gagner en robustesse.

Un des apports principal de mes contributions sur ce thème est une validation rigoureuse du principe de dissimulation conjointe à la compression, sur de grandes bases d'images ou de vidéos, et intégrant plusieurs aspects comme la robustesse aux attaques ou la préservation de la qualité de compression. En effet, la plupart des travaux existants pour les approches conjointes n'effectuent pas d'expérimentations sur la robustesse de leur approche, et ne permettent donc pas de conclure sur l'apport en robustesse des approches conjointes.

Publications principales

Reuves internationales :

- [Shahid et al. 11b] Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I & P frames.
IEEE Transactions on Circuits and Systems for Video Technology,
Mars 2011, 17 pages,
Shahid (Z.), Chaumont (M.) et Puech (W.).
- [Shahid et al. 11a] Considering the Reconstruction Loop for Data Hiding of Intra- and Inter-frames of H.264/AVC.
Signal, Image and Video Processing, Springer,
Avril 2011, 19 pages,
Shahid (Z.), Chaumont (M.) et Puech (W.).
- [Goudia et al. 11b] Joint Trellis Coded Quantization Watermarking for JPEG2000 images.
Annals of telecommunications, Signal, Image and multimedia processing, Springer,
Décembre 2011, 26 pages,
Goudia (D.), Chaumont (M.), Puech (W.) et Hadj Said (N.).

Chapitres de livre :

- [Goudia et al. 12] Quantization Watermarking for Joint Compression and Data Hiding Schemes.
Watermarking / Book 1, édité par Dr. Mithun Das Gupta,
InTech, 2012, 26 pages,
Goudia (D.), Chaumont (M.), Puech (W.), et Hadj Said (N.).

- [Shahid et al. 11c] Scalable Video Coding. *Effective Video Coding for Multimedia Applications*, édité par Sudhakar Radhakrishnan, InTech, 2011, 18 pages, Shahid (Z.), Chaumont (M.) et Puech (W.).
- [Chaumont et al. 09b] Protecting the Color Information by Hiding it. *Recent Advances in Signal Processing*, édité par Ashraf A Zaher, InTech, 2009, 22 pages, Chaumont (M.) et Puech (W.).

Conférences (Meilleures conférences du domaine IEEE ICIP, IEEE ICME, . . .) :

- [Chaumont et al. 11a] TCQ Practical Evaluation in the Hyper-Cube Watermarking Framework. *IEEE International Conference on Multimedia and Expo, ICME'2011*, Barcelona, Spain, Juillet 2011, 6 pages, Chaumont (M.) et Goudia (D.).
- [Shahid et al. 10b] Spread Spectrum-Based Watermarking for Tardos Code-Based Fingerprinting of H.264/AVC Video. *IEEE International Conference on Image Processing, ICIP'2010*, Hong-Kong, China, Septembre 2010, 4 pages, Shahid (Z.), Chaumont (M.) et Puech (W.).
- [Shahid et al. 10a] Selective Encryption of C2DVLC of AVS Video Coding Standard for I & P Frames. *IEEE International Conference on Multimedia & Expo, ICME'2010*, Singapore, Juillet 2010, 6 pages, Shahid (Z.), Chaumont (M.) et Puech (W.).
- [Chaumont 10a] A Novel Embedding Technique For Dirty Paper Trellis Watermarking. *Visual Information Processing and Communication, Part of IS&T/SPIE 22th Annual Symposium on Electronic Imaging, VIPC'2010, SPIE'2010*, vol. 7543, San Jose, California, USA, Janvier 2010, 9 pages, Chaumont (M.).
- [Shahid et al. 09] Fast Protection of H.264/AVC by Selective Encryption of CABAC. *IEEE International Conference on Multimedia and Expo, ICME'2009*, New York City, Juin 2009, 4 pages, Shahid (Z.), Chaumont (M.) et Puech (W.).
- [Chaumont et al. 07c] A Grey-Level Image Embedding its Color Palette. *IEEE International Conference on Image Processing, ICIP'2007*, vol. I, pp. 389–392, San Antonio, Texas, USA, Septembre 2007, 4 pages, Chaumont (M.) et Puech (W.).

La liste des publications est donnée à la fin du CV, et elles sont disponibles à l'adresse suivante : <http://www.lirmm.fr/~chaumont/Publications.html>.

Encadrements

Encadrements de doctorants depuis 2005

- Sarra KOUIDER (oct. 2010 - . . .) « Codes correcteurs et sécurité en stéganographie ». W. Puech, M. Chaumont (Financement Ministère Enseignement/Recherche Algérien). Taux Encadrement : 90%.
- Dalila GOUDIA (sept. 2008 - décembre 2011) « Tatouage conjoint à la compression d'images fixes par les ondelettes ». W. Puech, M. Chaumont, M. Hadj Said Naima (co-tutelle avec le Laboratoire Signal Image PArole (SIMPA), Algérie). Taux Encadrement : 70%. Thèse soutenue en décembre 2011.
- Zafar Javed SHAHID (janv 2008 - nov 2010) : « Protection de vidéos par compression et tatouage scalables ». W. Puech et M. Chaumont. Taux Encadrement : 50%. Thèse soutenue en novembre 2010.

- Peter MEUEL (octobre 2006 - décembre 2009) : « Enrichissement et protection de séquences vidéos ». W puech et M. Chaumont. Taux Encadrement : 50%. Thèse soutenue en décembre 2009.

Encadrements de stagiaires de Master 2 Recherche en Informatique à l'université Montpellier II

2011-2012 :

- Iuliia TKACHENKO : « Stégananalyse universelle sans *source-cover mismatch* ».
- Florian COURTADE : « Segmentation automatique d'images numériques : application à la détection et à la géo-localisation des tombes dans un cimetière ».

2009-2010 :

- Hugo ALATRISTA SALAS : « La stéganographie moderne : L'art de communication secrète ». Poursuite en thèse à Montpellier.
- Maha GHARBI ARAB : « Comparaison à haut débit du tatouage 'on-off keying' avec le(s) schéma(s) de tatouage Dirty Paper Trellis Codes et/ou Scalar Costa Scheme ». Elle est retournée en Tunisie.
- Sarra KOUIDER : « Extraction de caractéristiques pour l'analyse biométrique 3D d'un visage ». Poursuite en thèse encadré par W.Puech et moi-même.

2008-2009 :

- Omar BERREZOUG : « Tatouage robuste aux attaques de désynchronisations ». Une publication à MajecSTIC. Il est retourné en Algérie.
- Vinh Truong Hoang : « Attaque de système de tatouage ». Il est retourné au Vietnam.

2007-2008 :

- Nicolas TOURNIER : « Protection de vidéos par compression et tatouage hiérarchiques ». Une publication à CORESA. Poursuite en thèse dans l'équipe ICAR.
- Eric ELIAS : « Etes-vous un hacker dans l'âme ? Attaque d'un système de tatouage BOWS2 ». Il a abandonné les études pour des raisons familiales.

2006-2007 :

- Faraz Ahmed ZAIDI : « Protection de la couleur d'une image par insertion de données cachées ». Poursuite en thèse à Bordeaux.
- Minh VU DUC : « Sécurisation des visages dans les séquences d'images par cryptage sélectif ». Il est retourné au Vietnam.

2005-2006 :

- Peter MEUEL : « Tatouage vidéo avec utilisation du codeur H.264 ». Poursuite en thèse encadré par W.Puech et moi-même.
- Khizar HAYAT : « Insertion de données multi-résolution dans des images elles mêmes multi-résolues ». Ces travaux ont donné lieu à deux publications (CORESA 2006 et EI SPIE2007). Poursuite en thèse dans l'équipe ICAR.

Rayonnement et animation scientifique

Relecture de plus de 100 articles entre 2005 et fin 2011 (conférences ou revues)

- Re-lecteur de la conférence ICIP (IEEE International Conference on Image Processing) depuis 2007,
- Re-lecteur et TPC Member de la conférence EUSIPCO (The European Signal Processing Conference) depuis 2007,
- Re-lecteur de la conférence et membre du comité de programme CORESA (COMpression et REprésentation des Signaux Audiovisuels) depuis 2007,
- Re-lecteur pour IWDW'2007 (International Workshop on Digital Watermarking),
- Re-lecteur pour la revue Hindawi - International Journal of Digital Multimedia Broadcasting 2008,
- Re-lecteur pour la revue Journal of Electronic Imaging (SPIE - The International Society for Optical Engineering) 2009, 2011,
- Re-lecteur pour la revue Computerized Medical Imaging and Graphics 2009,
- Re-lecteur pour la revue Elsevier Pattern Recognition Letters 2009,
- Re-lecteur pour Singaporean-French IPAL Symposium 2009,
- Re-lecteur et membre du comité scientifique de programme de IPTA (International Conference on Image Processing Theory, Tools and Applications) depuis 2010,
- Re-lecteur pour la revue IEEE Signal Processing Letters - 2009, 2010,
- Re-lecteur pour la revue Elsevier Journal of Visual Communication and Image Representation 2009, 2010,
- Re-lecteur pour la revue IET Information Security Journal 2010, 2011,
- Re-lecteur pour la revue IEEE Transaction on Image Forensics and Security 2010, 2011,
- Re-lecteur pour la revue IS&T Journal of Imaging Science and Technology 2010,
- Re-lecteur pour la revue Elsevier Journal of Systems and Software 2010, 2011,
- Re-lecteur pour la revue Elsevier Signal Processing Image Communication 2010,
- Re-lecteur pour la conférence ICCCT (International Conference on Computer and Communication Technology) 2011.

Président de session de conférences

- Président de la session orale OS5 « Video Coding », jeudi 8 juillet, 14h-16h, IPTA'2010, International Conference on Image Processing Theory, Tools and Applications, July 7-10, Paris, France, http://ipta10.ibisc.univ-evry.fr/doku.php?id=technical_program.
- Président de la session orale « Compression vidéo », mercredi 27 octobre 2010, 15h20-16h10, CORESA'2010, COMpression et REprésentation des Signaux Audiovisuels, Lyon, France, 26-27 Octobre, 2010, <http://liris.cnrs.fr/coresa10/index.php?page=programme>.

Commissions de spécialistes

- Membre du Pool d'expert de Montpellier depuis sa création en 2008,
- Membre de Commission de spécialistes de la section 27 de l'Université de Provence (CMI, 39 rue Joliot-Curie 13453 Marseille) en 2007,
- Membre de Commission de spécialistes du département science de l'Université de Nîmes en 2007.

Responsabilités diverses

- Co-animateur du groupe sécurité du LIRMM depuis mars 2012,
- Membre de la commission de recherche de l'université de Nîmes depuis le 14 janvier 2011,
- Membre du Comité d'organisation du CORESA 2007 (Compression et REprésentation des Signaux Audiovisuels) qui a eu lieu à Montpellier, le 8 et 9 novembre 2007.

Expertises

- Expertise d'une demande de Crédit Impôt Recherche pour l'année 2010 (en novembre 2011),
- Expertise de 2 dossiers ANR en 2007 et 2008,
- Expertise liée à la compression pour la société Meeting One en 2006.

Tutoriaux à audience nationale et exposés de vulgarisation

- Tutoriel invité au groupe de travail sécurité en octobre 2010 LIRMM sur le tatouage dans les vidéos,
- Tutoriel invité pour la réunion du GDR-ISIS Paris en 2009 sur le tatouage robuste aux désynchronisations,
- Présentation pour les portes ouvertes à l'université de Nîmes depuis 2008,
- Exposé (1h30) sur le tatouage, la stéganographie et la science du forensics pour la fête de la science en 2009.

ANR

- Participation à 50% au projet TSAR (Transfert Sécurisé d'images d'Art haute Résolution) porté par l'IRCCyN retenu par l'ANR dans le cadre du programme ARA SSIA (Sécurité, Systèmes Embarqués et Intelligence Ambiante) 2005-2008. Ce projet réunit l'IRCCyN (Nantes), le LIS (Grenoble), l'IETR (Rennes), le C2RMF (Louvre, Paris) et le LIRMM (Montpellier),
- Participation à 50% au projet ANR Vooddo porté par la société Vodnet (spécialiste de la diffusion de vidéo en peer-to-peer) 2007-2010, et encadrement de Zafar Shahid en thèse dans le cadre de cette ANR,

Collaborations industrielle

- Collaboration avec la société Vodnet (spécialiste de la diffusion de vidéo en peer-to-peer) dans le cadre de la thèse de Zafar Shahid (ANR Voodoo),
- Collaboration avec la société Meeting One (spécialiste dans des services de visioconférence) en 2006, 2007 (expertise),
- Collaboration avec la société Sigma Méditerranée (spécialiste dans la mise en place de systèmes de vidéo numérique à distance) en 2006 et 2007 dans le cadre d'un montage de projet finalement non retenu,
- Collaboration avec une association artistique (projet start-up 2L-Pendha EMBRUN) (projet de suivi de visage pour une plate-forme d'interface homme-machine dans un objectif artis-

- tique) en 2006, 2007 (prolongement du travail de [Chaumont et al. 07a] et encadrement de stagiaires),
- Collaboration avec la société Setinnov de septembre à novembre 2010 (étude à l'élaboration d'une compression multi-flux pour le streaming synchronisé de flux à différentes résolution ou bien pour des flux pour la restitution en 3D),
 - Collaboration avec la société Floware sur de la compression sans perte et la transmission à la volée de tout type de fichier, septembre 2010 - juillet 2011 (expertise),
 - Collaboration avec la société Berger-Levrault pour l'étude de la segmentation automatique d'images numériques avec application à la détection et à la géo-localisation des tombes dans un cimetière, janvier 2012-septembre 2012 (expertise et encadrement d'un stagiaire).

Prime

- Bénéficiaire de la Prime d'Excellence Scientifique (PES) depuis le 1^{er} octobre 2010.

Enseignement

Responsabilités

- Responsable de 6 unités d'enseignement : (Traitement du signal M1-Montpellier, Graphe L3MI, Réseaux-Compression L3MI, Module Projet L3MI, Structure de données en Java L2MI, Traitement Numérique en Maple L1),
- Co-responsable de la licence Math-Informatique de l'université de Nîmes depuis 2009
- Président de jury des L3 Math Info et membre des jurys L1 et L2 Math Info.

Matière enseignées

Depuis septembre 1999, je donne des cours, des TDs, et des TP à différents niveaux et sur différentes matières, et avec différents langages (C, C++, Java, Ada, Visual Basic). J'ai également encadré ou co-encadré 43 étudiants (40 sujets différents) pour des projets IUT, L3, M1 sur de nombreux domaines (Intelligence artificielle, Réseau, Jeux, Modélisation, Contraintes, Images, ...). Le tableau 1 résume les disciplines dans lesquelles j'ai enseigné. Certains supports de cours sont à l'adresse <http://www.lirmm.fr/~chaumont/Lecture.html> et à l'adresse <http://www.lirmm.fr/~chaumont/download/cours/>.

Publications

Ci-dessous sont listés les 3 journaux, les 3 chapitres de livre, les 35 articles-conférences internationales et les 14 articles-conférences nationales dans lesquels j'ai publié depuis le début de ma carrière. L'ensemble de ces publications s'inscrit dans le domaine de traitement de l'image. Les 3 journaux ont un *impact factor* bon à très bon. On retrouve les meilleures conférences du domaine (taux d'acceptation entre 23% et 50% : IEEE ICIP, IEEE ICME, IEEE ICASSP, ...) et les conférences nationales du domaine.

Revues

- Shahid (Z.), Chaumont (M.) et Puech (W.). – Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I & P frames. *IEEE Transactions on Circuits and Systems for Video Technology*, 9, Mars 2011, 17 pages. 5-Year Impact Factor = 3.187 (JCR : Journal Citation Reports de Thomson Reuters 2009).
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Considering the Reconstruction Loop for Data Hiding of Intra- and Inter-frames of H.264/AVC. *Signal, Image and Video Processing*, Springer, pp. 1–19, Avril 2011, 19 pages Impact Factors = 0.617 (JCR : Journal Citation Reports de Thomson Reuters 2010).
- Goudia (D.), Chaumont (M.), Puech (W.) et Hadj Said (N.). – Joint Trellis Coded Quantization Watermarking for JPEG2000 images. *Annals of telecommunications, Signal, Image and multimedia processing*, Springer, DOI : 10.1007/s12243-011-0280-9, 26 pages, Décembre 2011. Impact Factors = 0.602 (JCR : Journal Citation Reports de Thomson Reuters 2010).

Chapitre de livre

- Goudia (D.), Chaumont (M.), Puech (W.), et Hadj Said (N.). – Quantization Watermarking for Joint Compression and Data Hiding Schemes. *Watermarking / Book 1*, édité par Dr. Mithun Das Gupta, ISBN 979-953-307-583-8 – InTech, 2012, 26 pages.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Scalable Video Coding. *Effective Video Coding for Multimedia Applications*, édité par Sudhakar Radhakrishnan, pp. 3–20. – InTech, 2011, 18 pages.
- Chaumont (M.) et Puech (W.). – Protecting the color information by hiding it. *Recent Advances in Signal Processing*, édité par Ashraf A Zaher, pp. 101–122. – InTech, 2009, 22 pages. (Chapitre consulté plus de 200 fois).

Tutoriaux dans des conférences internationales

- Chaumont (M.). – *Invited Paper - Tutorial* : Ensuring Security of H.264 Videos by Using Watermarking. – *Mobile Multimedia/Image Processing, Security, and Applications, Part of SPIE Defense, Security, and Sensing, DSS'2011, SPIE'2011*, vol. 8063, Orlando, Florida, USA, Avril 2011, 10 pages.
- Chaumont (M.). – *Invited Paper - Tutorial* : H.264 Video Watermarking : Applications, Principles, Deadlocks, and Future. – *International Conference on Image Processing Theory, Tools and Applications, IPTA'2010*, Paris, France, Juillet 2010, Abstract.

Congrès internationaux avec actes

- Chaumont (M.) and S. Kouider (S.). – Steganalysis by Ensemble Classifiers with Boosting by Regression, and Post-Selection of Features. – *IEEE International Conference on Image Processing, ICIP'2012*, Lake Buena Vista (suburb of Orlando), Florida, USA, September 30 - October 3, 2012, 4 pages.
- Goudia (D.), Chaumont (M.), Puech (W.) et Hadj Said (N.). – A Joint Trellis Coded Quantization (TCQ) Data Hiding Scheme in the JPEG2000 Part 2 Coding Framework. – *The 19th European Signal Processing Conference, EUSIPCO'2011*, Barcelona, Spain, Septembre 2011, 5 pages.

- Chaumont (M.) et Goudia (D.). – TCQ Practical Evaluation in the Hyper-Cube Watermarking Framework. – *IEEE International Conference on Multimedia and Expo, ICME'2011*, Barcelona, Spain, Juillet 2011, 6 pages, Taux d'acceptation = 30%.
- Chaumont (M.), Goudia (D.) et Puech (W.). – Hyper-Cube Watermarking Scheme. – *Visual Information Processing and Communication II, Part of IS&T/SPIE 23th Annual Symposium on Electronic Imaging, VIPC'2011, SPIE'2011*, vol. 7882, pp. 10–18, San Francisco, California, USA, Janvier 2011, 9 pages.
- Goudia (D.), Chaumont (M.), Puech (W.) et Hadj Said (N.). – A joint JPEG2000 Compression and Watermarking System Using a TCQ-Based Quantization Scheme. – *Visual Information Processing and Communication II, Part of IS&T/SPIE 23th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7882, San Francisco, California, USA, Janvier 2011, 8 pages.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Spread Spectrum-Based Watermarking for Tardos Code-Based Fingerprinting of H.264/AVC Video. – *IEEE International Conference on Image Processing, ICIP'2010*, Hong-Kong, China, Septembre 2010, 4 pages, Taux d'acceptation = 47%.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Over The Real-Time Selective Encryption of AVS Video Coding Standard. – *The 18th European Signal Processing Conference, EUSIPCO'2010*, Aalborg, Denmark, Août 2010, 5 pages, Taux d'acceptation = 48.8%.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Selective Encryption of C2DVLC of AVS Video Coding Standard for I & P Frames. – *IEEE International Conference on Multimedia & Expo, ICME'2010*, Singapore, Juillet 2010, 6 pages, Taux d'acceptation = 30%.
- Chaumont (M.). – A Novel Embedding Technique For Dirty Paper Trellis Watermarking. – *Visual Information Processing and Communication, Part of IS&T/SPIE 22th Annual Symposium on Electronic Imaging, VIPC'2010, SPIE'2010*, vol. 7543, San Jose, California, USA, Janvier 2010, 9 pages.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Selective and Scalable Encryption of Enhancement Layers for Dyadic Scalable H.264/AVC by Scrambling of Scan Patterns . – *IEEE International Conference on Image Processing, ICIP'2009*, Cairo, Egypt, Novembre 2009, 4 pages, Taux d'acceptation = 45%.
- Chaumont (M.). – Psychovisual Rotation-based DPTC Watermarking Scheme. – *17th European Signal Processing Conference, EUSIPCO'2009*, Glasgow, Scotland, Août. 2009, 5 pages.
- Shahid (Z.), Meuel (P.), Chaumont (M.) et Puech (W.). – Considering the Reconstruction Loop for Watermarking of Intra and Inter Frames of H.264/AVC. – *The 17th European Signal Processing Conference, EUSIPCO'2009*, Glasgow, Scotland, Août 2009, 5 pages.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Fast Protection of H.264/AVC by Selective Encryption of CABAC For I & P Frames. – *The 17th European Signal Processing Conference, EUSIPCO'2009*, Glasgow, Scotland, Août 2009, 5 pages.
- Chaumont (M.). – Fast Embedding Technique for Dirty Paper Trellis Watermarking. – Ho (Anthony T.S.), Shi (Yun Q.), Kim (H.J.) et Barni (Mauro) (édité par), *8th International Workshop on Digital Watermarking, IWDW'2009*, vol. 5703 of *Lecture Notes in Computer Science*, pp. 110–120, University of Surrey, Guildford, United Kingdom, Aug. 2009. Springer, 11 pages.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Fast Protection of H.264/AVC by Selective Encryption of CABAC. – *IEEE International Conference on Multimedia and Expo*,

- ICME'2009*, New York City, Juin 2009, 4 pages, Taux d'acceptation présentation orale = 23%.
- Shahid (Z.), Chaumont (M.) et Puech (W.). – Fast Protection of H.264/AVC by Selective Encryption. – *Singaporean-French IPAL Symposium, SinFra'2009*, Fusionopolis, Singapore, Février 2009, 11 pages.
 - Chaumont (M.) et Puech (W.). – A High Capacity Reversible Watermarking Scheme. – *IS&T/SPIE 21th Annual Symposium on Electronic Imaging, Visual Communications and Image Processing, VCIP'2009, SPIE'2009*, vol. 7257, San Jose, California, USA, Janvier 2009, 9 pages.
 - Shahid (Z.), Chaumont (M.) et Puech (W.). – An Adaptive Scan of High Frequency Subbands of Dyadic Intra Frame in MPEG4-AVC/H.264 Scalable Video Coding. – *Visual Communications and Image Processing, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2009*, vol. 7257, San Jose, California, USA, Janvier 2009, 9 pages.
 - Chaumont (M.) et Puech (W.). – A 8-Bits-Grey-Level Image Embedding its 512 Color Palette. – *16th European Signal Processing Conference, EUSIPCO'2008*, Lausanne, Switzerland, Août 2008, 5 pages, Taux d'acceptation = 50.7%.
 - Chaumont (M.) et Puech (W.). – Attack By Colorization of a Grey-Level Image Hiding its Color Palette. – *IEEE International Conference on Multimedia & Expo, ICME'2008*, Hannover, Germany, Juin 2008, 4 pages, Taux d'acceptation = 50%.
 - Puech (W.), Chaumont (M.) et Strauss (O.). – A Reversible Data Hiding Method for Encrypted Image. – *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Part of IS&T/SPIE 20th Annual Symposium on Electronic Imaging, SPIE'2008*, vol. 6819, Janvier 2008, 9 pages.
 - Chaumont (M.) et Puech (W.). – A Grey-Level Image Embedding its Color Palette. – *IEEE International Conference on Image Processing, ICIP'2007*, vol. I, pp. 389–392, San Antonio, Texas, USA, Septembre 2007, 4 pages, Taux d'acceptation = 46%.
 - Meuel (P.), Chaumont (M.) et Puech (W.). – Data Hiding in H.264 Video For Lossless Reconstruction of Region. – *The 15th European Signal Processing Conference, EUSIPCO'2007*, Poznan, Poland, Septembre 2007, 5 pages.
 - Chaumont (M.) et Puech (W.). – Fast Protection of the Color of High Dimension Digital Painting Images. – *Eighth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'2007*, pp. 60–63, Santorini, Greece, Juin 2007, 4 pages.
 - Chaumont (M.) et Puech (W.). – A Fast and Efficient Method to Protect Color Images. – *Visual Communications and Image Processing, Part of IS&T/SPIE 19th Annual Symposium on Electronic Imaging, VCIP'2007, SPIE'2007*, vol. 6508, San Jose, California, USA, Janvier 2007, 9 pages.
 - Chaumont (M.) et Puech (W.). – 3D face model tracking based on a multiresolution active search. – *Visual Communications and Image Processing, Part of IS&T/SPIE 19th Annual Symposium on Electronic Imaging, VCIP'2007, SPIE'2007*, vol. 6508, San Jose, California, USA, Janvier 2007, 8 pages.
 - Hayat (K.), Puech (W.), Gesquière (G.) et Chaumont (M.). – Wavelet-based data-hiding of DEM in the context of real-time 3D visualization. – *Visualization and Data Analysis, Part of the IS&T/SPIE Symposium on Electronic Imaging, SPIE'2007*, San Jose, California, Janvier 2007, 10 pages.
 - Chaumont (M.) et Puech (W.). – A DCT-Based Data-Hiding Method to Embed the Color Information in a JPEG Grey Level Image. – *The European Signal Processing Conference,*

- EUSIPCO'2006*, Pise, Italie, Septembre 2006, 5 pages.
- Rodrigues (J.M.), Puech (W.), Meuel (P), Bajard (J.-C.) et Chaumont (M.). – Face Protection with Fast Selective Encryption in a Video Sequence. – *IEE conference on Crime and Security (Imaging for Crime Detection and Prevention), ICDP'2006*, Juin 2006, 6 pages.
 - Chaumont (M.) et Puech (W.). – A Color Image in a Grey-Level Image. – *IS&T Third European Conference on Colour in Graphics, Imaging, and Vision, CGIV'2006*, pp. 226–231, Leeds, UK, Juin 2006, 7 pages.
 - Beaumesnil (B.), Chaumont (M.) et Luthon (F.). – Liptracking and MPEG4 Animation with Feedback Control. – *IEEE International Conference on Acoustics Speech Signal Processing, ICASSP'2006*, Toulouse, France, Mai 2006, 4 pages, Taux d'acceptation = 48%.
 - Chaumont (M.) et Beaumesnil (B.). – Robust and real-time 3d-face model extraction. – *IEEE International Conference on Image Processing, ICIP'2005*, pp. 461–464, Genova, Italie, Septembre 2005, 4 pages, Taux d'acceptation = 46.2%.
 - Chaumont (M.), Pateux (S.) et Nicolas (H.). – Object-Based Video Coding Using a Dynamic Coding Approach. – *IEEE International Conference on Image Processing, ICIP'2004*, pp. 1105–1108, Singapore, Octobre 2004, 4 pages, Taux d'acceptation = 44.2%.
 - Chaumont (M.), Cammas (N.) et Pateux (S.). – Fully Scalable Object Based Video Coder Based on Analysis-Synthesis Scheme. – *IEEE International Conference on Image Processing, ICIP'2003*, Barcelona, Spain, Septembre 2003, 4 pages, Taux d'acceptation = 42.8%.
 - Chaumont (M.), Pateux (S.) et Nicolas (H.). – Efficient Lossy Contour Coding Using Spatio-Temporal Consistency. – *IEEE Picture Coding Symposium, PCS'2003*, pp. 289–294, Saint-Malo, France, Avril 2003, 6 pages.
 - Chaumont (M.), Pateux (S.) et Nicolas (H.). – Segmentation of Non-Rigid Video Objects Using Long Term Temporal Consistency. – *IEEE International Conference on Image Processing, ICIP'2002*, Rochester, USA, Septembre 2002, 4 pages, Taux d'acceptation = 58.3%.

Congrès nationaux

- Kouider (S.), Chaumont (M.), et Puech (W.). – Stéganographie Adaptative par Oracle (ASO). – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2010*, Lille, France, Mai, 2012, 6 pages.
- Diop (I.), Farssi (S. M.), Chaumont (M.), Khouma (O.), et Diouf (H. B.). – Utilisation des codes LDPC en stéganographie. – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2010*, Lille, France, Mai, 2012, 7 pages.
- Goudia (D.), Chaumont (M.) et Puech (W.). – Un Schéma conjoint de dissimulation de données (Data Hiding) dans JPEG2000 basé sur la quantification codée par treillis (TCQ). – *23ème édition du colloque GRETSI sur le traitement du signal et des images*, Bordeaux, Septembre 2011, 4 pages.
- Goudia (D.), Chaumont (M.), Puech (W.) et Said (N. Hadj). – Tatouage et compression conjoint dans JPEG2000 avec un algorithme de quantification codée par treillis (TCQ), "top 5 paper"; article sélectionné pour publication d'une version étendue dans une revue internationale, <http://liris.cnrs.fr/coresa10/>. – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2010*, Lyon, France, Août 2010, 6 pages.
- Berrezoug (O.) et Chaumont (M.). – Tatouage robuste aux attaques de désynchronisations. – *MANifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la*

- Communication (conférence organisée par des doctorants pour des chercheurs débutants : Master 2, doctorants, post-docs, ATER...), MajecSTIC'2009, Avignon, France, Novembre 2009, 8 pages.*
- Chaumont (M.). – Une nouvelle technique pour le tatouage par Dirty Paper Trellis Code (DPTC). – *22ème Colloque GRETSI sur le traitement du signal et des images, GRETSI'2009*, Dijon, France, Septembre 2009, 4 pages.
 - N. Tournier, M. Chaumont (W. Puech). – Tatouage informé hiérarchique d'un message hiérarchisé (en vue de la protection vidéo). – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2009*, Toulouse, France, Mars 2009, 6 pages.
 - Meuel (P.), Chaumont (M.) et Puech (W.). – Insertion de données cachées pour une reconstruction sans perte de Régions d'Intérêts dans des vidéos H.264. – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2007*, Montpellier, France, Novembre 2007, 6 pages.
 - Chaumont (M.) et Puech (W.). – Insertion d'une palette couleur dans une image en niveaux de gris. – *21ème Colloque GRETSI sur le traitement du signal et des images, GRETSI'2007*, Troyes, France, Septembre 2007, 4 pages.
 - Chaumont (M.) et Puech (W.). – Une image couleur cachée dans une image en niveaux de gris. – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2006*, Rennes, France, Novembre 2006, 7 pages.
 - Hayat (K.), Puech (W.), Gesquière (G.) et Chaumont (M.). – Visualisation 3D temps-réel à distance de MNT par insertion de données cachées basée ondelettes. – *COmpression et REprésentation des Signaux Audiovisuels, CORESA'2006*, Rennes, France, Janvier 2006, 7 pages.
 - Chaumont (M.). – Extraction d'un modèle 3d de visage en temps-réel et de manière robuste. – *20ème Colloque GRETSI sur le traitement du signal et des images, GRETSI'2005*, pp. 675–678, Louvain-la-Neuve, France, Septembre 2005, 4 pages.
 - Beaumesnil (B.), Luthon (F.) et Chaumont (M.). – Extraction temps-réel de contours labiaux par segmentation vidéo robuste en vue d'animation 3d. – *20ème Colloque GRETSI sur le traitement du signal et des images, GRETSI'2005*, pp. 489–492, Louvain-la-Neuve, France, Septembre 2005, 4 pages.
 - Chaumont (M.), Pateux (S.) et Nicolas (H.). – Codage efficace de contour avec perte utilisant la consistance spatio-temporelle. – *Traitement du signal et des images, GRETSI'2003*, Paris, France, Septembre 2003, 4 pages.

Thèse

- Chaumont (M.). – *Représentation en objets vidéo pour un codage progressif et concurrentiel des séquences d'images*. – PhD. Thesis, Université de Rennes 1, France, Novembre 2003. Supervisé par H. Nicolas et S. Pateux.

TABLE 1 – Résumé des enseignements dispensés

Établissement	Année	Matière	Contenu	Niveau	Nombre d'heure
Université de Nîmes	2005-2011	Algorithmique des graphes	Structures et algorithmes programmation en Java	Bac+3	120h CM, 128h TD 50h TP
Université de Nîmes	2005-2011	Calcul numérique	Algorithmique en Maple	Bac+1	86h CM, 64h TD 40h TP
Université de Nîmes	2005-2011	Traitement du signal	Échantillonnage, quantification convolution	Bac+3	42h CM
Université de Nîmes	2005-2011	Compression de données	Codage statistique et par dictionnaire (programmation en C)	Bac+3	48h CM, 92h TP
Université de Nîmes	2009-2011	Réseaux	Couches OSI (programmation socket en Java)	Bac+3	14.5h CM, 9.5h TD 18h TP
Université de Nîmes	2005-2011	Projet informatique	33 étudiants	Bac+3	121h TP
Université de Nîmes	2007-2011	Structures de données	Tri, pile, file, liste, ensemble arbre, table de hash	Bac+2	48h CM, 72h TD, 40h TP
Université de Nîmes	2005-2007	Algorithmique	Tri, pile, file, ensemble, grammaire	Bac+2	51h TD
Université de Nîmes	2007-2011	Méthodologie	Exposés scientifiques	Bac+2	9h CM
Université de Nîmes	2007-2008 2009-2010	Tutorat	Encadrement d'un tuteur étudiant	Bac+2	15h CM
Université de Montpellier 2	2007-2011	FMIN34 « Images compression et insertion de données cachées » - Tatouage M2R Info	Tatouage de données	Bac+5	24h CM
Université de Montpellier 2	2007-2011	UMSI 386 « Traitement du signal, réseau, théorie de l'information » - Codes détecteurs et correcteurs d'erreurs - M2 EEA	Codes correcteurs	Bac+5	24h CM, 10.5h TD
Université de Montpellier 2	2007-2011	FMIN 211 « Transmission des données et Traitement du signal » - Signal - M1 Info	Introduction au signal	Bac+4	12h CM, 12h TD, 12h TP
Université de Montpellier 2	2010-2011 2006-2007	Encadrement stage M1 EEA	2 étudiants	Bac+4	
Université de Montpellier 2	2005-2010	(co-)encadrement stages M2R	11 étudiants	Bac+5	27h eqTD
Université de Montpellier 2	2009-2010	Encadrement stage TER	4 étudiants	Bac+4	11h eqTD
Université de Montpellier 2	2006-2007	Encadrement stage L3Pro	1 étudiant	Bac+3	
Université de Montpellier 2	2006-2007	Encadrement stage ENS	1 étudiant	Bac+2	
UFR Sciences Pau $\frac{1}{2}$ ATER	2004-2005	Traitement et analyse d'images	Initiation aux traitement d'images et à la compression	Bac+5	3h cours 3h TD, 5h TP
IUT Bayonne $\frac{1}{2}$ ATER	2004-2005	Algorithmique et programmation	Initiation à l'algorithmique en Java	Bac+1	24h TD 25.5h TP
IUT Bayonne $\frac{1}{2}$ ATER	2004-2005	Algorithmique et programmation	Initiation à l'algorithmique en ADA	Bac+2	27h TD
Université de Rennes 1 (IFSIC) $\frac{1}{2}$ ATER	2003-2004	Système d'information et génie logiciel	Modélisation d'un logiciel de gestion d'intervention des pompiers	Bac+4	20h TD 48h TP
Université de Rennes 1 (IFSIC) $\frac{1}{2}$ ATER	2003-2004	Réseaux	Initiation aux couches Liaison, Réseau et Transport	Bac+5	12h TD 8h TP
Université de Rennes 1 (IFSIC) $\frac{1}{2}$ ATER	2003-2004	Compression vidéo	Initiation aux codages d'image et de vidéo	Bac+5	18h TP
Université de Rennes 1 (IFSIC) $\frac{1}{2}$ ATER	2003-2004	Traitement et analyse d'images	Initiation aux images numérique	Bac+4	8h TD 8h TP
INSA de Rennes (Moniteur)	2000-2003	Algorithmique et programmation	Initiation à l'algorithmique en Java	Bac+2	63h cours-TD 192h TP
Saint-Cyr Coëtquidan (Scientifique du contingent)	1999-2000	Algorithmique et programmation	C, C++, Pascal, Visual-Basic, Merise, bureautique	Bac+1 Bac+2	150h de TD/TP

Introduction

Le tatouage numérique moderne est apparu au début des années 90 [Tanaka et al. 90]. Durant une dizaine d'années, il y a eu un fort engouement pour cette jeune discipline de la part des communautés travaillant sur le signal, les télécommunications, la théorie de l'information, et les images. En plus de l'intérêt scientifique, de nombreux industriels ont vu dans cette nouvelle discipline une solution venant en soutien de la cryptographie pour sécuriser les médias comme l'image, le son, la vidéo... Ces industriels ont encouragé la recherche dans le domaine. Cette période est donc foisonnante de propositions. Avec le recul on peut considérer que c'est une période où la communauté a acquis une grande expérience de « manipulation des données » en vue du tatouage.

Malgré cette euphorie, il manquait cruellement d'une formalisation du processus de tatouage. La majorité des propositions avaient pour objectif de faire du tatouage robuste tout en préservant la qualité perceptuelle du média. La formalisation du tatouage robuste est donc apparue à la fin des années 90 avec la redécouverte d'un papier de Max Costa [Costa 83] sur la transmission fiable d'un message sur un canal subissant deux sources de dégradation (l'une connue de l'émetteur, l'autre inconnue de l'émetteur). Pour la communauté du tatouage, le travail de Max Costa était particulièrement émulateur, car il permettait d'envisager des mécanismes de tatouage bien plus performants que ce qui avait été proposé auparavant [Cox et al. 97]. D'autre part, le travail de Max Costa donnait des bornes théoriques, mais ne donnait pas d'approche pratique. Il y avait donc la place pour de nouvelles approches de tatouage : les schémas informés, appelés également schémas à information de bord ou bien encore schémas avec information adjacente.

On a donc vu apparaître deux grandes familles de schémas de tatouage informés à partir du début des années 2000. Ces deux familles ont évidemment donné lieu à un grand nombre de publications visant à améliorer incrémentalement les propositions initiales. À partir de 2005, la sécurité de ces schémas a également été étudiée [Cayre et al. 05], [Bas et al. 08a], [PérezFreire et al. 07]. Une question qui restait pour moi en suspend était de comprendre quelles étaient les performances pratiques respectives de ces schémas. Quelles performances en termes de robustesse pouvions-nous attendre de tels schémas ? Jusqu'à quel payload¹ pouvions-nous aller pour une dégradation raisonnable ? Quel algorithme était le plus performant ? Comment comparer de manière équitable et avec quel critère ? Quels étaient les paramètres qu'il fallait figer pour pouvoir obtenir une comparaison ? Bref, toutes ces questions plutôt d'ordre pratique permettent de répondre à la question posée au début des années 90 : est-ce que le tatouage numérique apporte un niveau de robustesse et de sécurité suffisant pour être envisagé en appoint de la cryptographie pour sécuriser les applications ? Que la réponse soit positive ou négative ne remet pas en question l'intérêt de faire du tatouage. En effet, le tatouage peut être utilisé pour des applications qui ne sont pas liées à la sécurité. La première partie du manuscrit traite de ces questions à travers des évaluations expérimentales.

1. payload : quantité de bits insérée.

tales. Deux propositions incrémentales sont également présentées et nous ont permis d'apporter une pierre supplémentaire à l'édifice.

Le modèle retenu par la communauté du tatouage considère un signal source auquel est ajouté un signal de tatouage. Bien souvent, on prend comme signal source une image au format « raw », ou bien pour une vidéo, une séquence d'images « raw ». Tout traitement survenant après la phase de tatouage est donc considéré comme une attaque sur le système de tatouage. La plupart du temps, les images sont échangées au format gif, bmp, png ou jpg,... et les vidéos aux formats MPEG1, MPEG2, MPEG4, H.264, VC-1, WMV, RealVideo... Tous ces formats compriment et dégradent les données. Pour un système de tatouage, la dégradation due à la compression représente une attaque au système de tatouage survenant avant même que le média tatoué soit distribué. Or, la logique du tatouage informé voudrait que toutes les dégradations qui surviennent avant la distribution du média soient prises en compte lors de la phase de tatouage. La phase de compression et en particulier la phase de quantification devrait donc être considérée. On peut donc très justement se demander si une collaboration entre le mécanisme de tatouage et le mécanisme de compression pourrait améliorer les performances en termes de robustesse. Le tatouage joint à la compression est un aspect qui peut sembler logique, mais qui n'a pas été énormément traité dans la littérature. Il existe beaucoup de solutions qui sont mal intégrées dans le mécanisme de compression. Bien souvent, le module de tatouage et le module de quantification sont deux modules différents qui « luttent l'un contre l'autre ». Lorsque le mécanisme est bien intégré, on peut espérer que les performances restent bonnes, à la fois en terme de compression, mais aussi en terme de robustesse du système de tatouage. La deuxième partie du manuscrit traite de cet aspect « joint » à travers la présentation 1- d'une approche de tatouage conjointement à la quantification au sein de jpeg2000, et 2- d'une approche de tatouage intégrée dans le mécanisme d'optimisation débit distorsion au sein du codeur H.264.

Bien souvent le tatouage est envisagé pour des applications liées à la sécurité. Cela dit, le tatouage peut également être utilisé dans le but d'augmenter les fonctionnalités d'une application. Dans ce cas, la robustesse n'est pas forcément requise et on parle alors plutôt de dissimulation de données (data-hiding). On trouve dans la littérature de nombreuses applications originales de dissimulation de données : l'insertion dans une vidéo de paramètres d'animation d'un clone ('enrichment'), l'insertion d'une image sous résolue dans l'image pour la récupération d'erreurs de transmission ('error concealment'), l'insertion d'information dans une vidéo pour contrôler un robot visualisant la vidéo ('device control'), l'insertion d'un paramètre de contrôle au sein d'une vidéo H.264 pour optimiser la performance en compression ('compression improvment'), l'insertion d'annotations servant à la surveillance/analyse du contenu d'une diffusion radio ('broadcast monitoring'), ... Dans le même esprit, nous avons proposé une application de dissimulation de la couleur dans une image en niveaux de gris. À ma connaissance, le premier travail relatif à la dissimulation de la couleur date de 2002 [Campisi et al. 02] et consiste à substituer certaines sous-bandes ondelettes de la luminance par des sous-bandes de chrominances. L'objectif était d'améliorer les performances de compression d'images couleur. De nombreuses autres approches très similaires dans l'esprit sont encore aujourd'hui publiées. Notre proposition est totalement différente des approches utilisant des substitutions de coefficients ondelettes. Elle repose sur de la quantification couleur et de la dissimulation avec des images utilisant une palette couleur. La troisième partie de ce manuscrit traite de cette contribution de la dissimulation de la couleur dans une image en niveau de gris à travers : 1- une approche par modélisation/optimisation du problème, et 2- une approche plus heuristique avec dissimulation d'une palette de 512 couleurs.

Première partie

Le tatouage d'images fixes

Chapitre 1

Le tatouage robuste d'images

Le tatouage robuste n'a été formalisé que très récemment. Il a fallu la redécouverte des travaux de Max Costa [Costa 83] dans les années 1998-1999 [Cox et al. 99] pour que l'on considère le tatouage comme un problème de communication fiable sur un canal bruité et que l'on montre théoriquement que les performances des systèmes de tatouage pouvaient très nettement être améliorés. Dans ce chapitre, nous ne reviendrons pas sur les systèmes de tatouage développés lors des dix premières années de la discipline. Nous nous intéressons aux propositions de tatouage informé apparues depuis les années 2000. Les systèmes développés dans la décennie 2000-2010 reprennent évidemment les enseignements de la période 1990-1998. Pour ne pas alourdir le chapitre, nous n'incluons pas d'explication sur la technique d'étalement de spectre dont l'utilisation en tatouage est apparue en 1997 [Cox et al. 97]. Les principes présents dans l'étalement de spectre sont réutilisés dans les algorithmes informés de Broken Arrows [Furon et al. 08] que nous présentons en Section 1.4 et de RB-DPTC (Rotation-Based Dirty Paper Trellis Code) présenté dans le chapitre 2.

Dans ce chapitre nous rappelons brièvement en section 1.1 en quoi consiste le tatouage robuste. Nous présentons ensuite en section 1.2 quelques exemples d'applications réalistes qui utilisent intelligemment le tatouage numérique. Nous donnons en section 1.3 trois caractéristiques communes aux systèmes de tatouage informés modernes : l'espace d'insertion, le codage informé et l'insertion informée. Nous donnons ensuite en section 1.4 une illustration de ces caractéristiques à travers la description de l'algorithme de Broken Arrows [Furon et al. 08]. Nous en profitons également pour aborder un autre point important pour un système de tatouage moderne : la prise en compte de l'impact psychovisuel.

1.1 Schéma général de tatouage informé

Le tatouage robuste a été formalisé comme la communication d'un message sur un canal **non fiable**. On entend par **non fiable** le fait que le canal (par exemple un canal hertzien) peut subir des perturbations (par exemple du bruit électromagnétique). Lors d'une communication sur un canal non fiable, le message transmis (par exemple une suite de bits) n'est pas forcément identique au message reçu. L'objectif du tatouage robuste est donc de transmettre un message entre un émetteur (algorithme d'insertion) et un récepteur (algorithme d'extraction et/ou de détection) de manière fiable, c'est-à-dire en étant insensible aux perturbations du canal. On retrouve donc ici un problème similaire à celui de la théorie de l'information et de la communication dans lequel on a recourt à

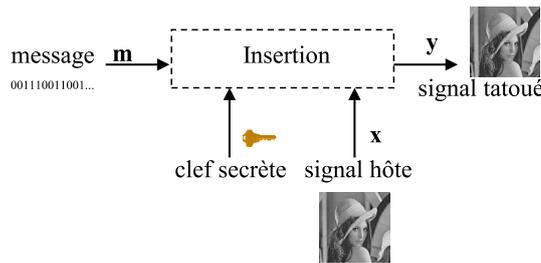


FIGURE 1.1 – Schéma général de l'insertion.

l'utilisation de codes correcteurs et détecteurs d'erreurs. Le schéma général du tatouage robuste **informé** pour une image est donné figure 1.1. Un message binaire m est « transporté » par un signal hôte x (c'est-à-dire une représentation de l'image) qui est transformé en un signal tatoué y . L'image tatouée peut alors être amenée à subir des dégradations non malveillantes ou malveillantes ; on dira que l'image tatouée est « attaquée ». La détection et/ou extraction du message peut alors être effectuée sur l'image tatouée attaquée. Dans de nombreuses applications, la détection/extraction est effectuée sans l'utilisation de l'image originale. On parle d'extraction aveugle. Nous nous limiterons à ce cas, sachant que l'extraction avec utilisation de l'image originale ne change pas la partie insertion et n'introduit quasiment pas de changement lors de l'extraction. La figure 1.2 illustre la phase d'extraction aveugle lorsqu'un signal tatoué y est attaqué par un signal n inconnu lors de l'extraction. Le message m' est alors extrait à partir du signal reçu z .

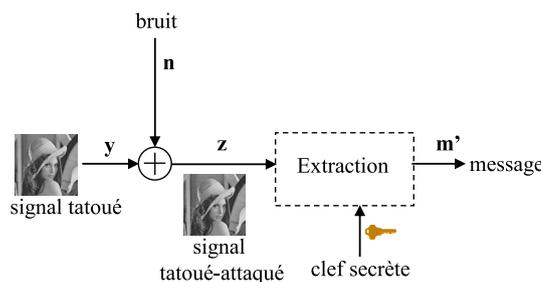


FIGURE 1.2 – Schéma général d'extraction aveugle.

1.2 Quelques applications utilisant le tatouage numérique

Avant de rentrer plus en détail dans les différents schémas d'insertion, il est intéressant de donner un aperçu des quelques applications envisagées ou réalistes qui peuvent bénéficier du tatouage numérique. Le tableau 1.1 résume l'ensemble des problématiques sur lesquelles le tatouage peut apporter une solution ou une partie de solution. On peut regrouper ces problématiques en deux ensembles : l'un lié à la sécurité et nécessitant pratiquement toujours des schémas très robustes et très sûrs, l'autre lié à l'enrichissement et ne nécessitant pas nécessairement autant de robustesse ni de sûreté.

Il est évident que les applications liées à la sécurité sont celles qui ont le plus drainé d'« affaires ». Comme nous l'indiquons dans l'introduction, de nombreux industriels ont vu dans le tatouage une solution pour la protection des droits d'auteurs venant en soutien de la cryptographie

TABLE 1.1 – Classification des applications de tatouage.

Relatif à la sécurité	Relatif à l'amélioration de médias
Protection du droit d'auteur	Analyse/surveillance d'un flux numérique
Traçage de traîtres	Contrôle de périphérique
Authentification	Enrichissement (fonctionnalités et/ou métadatas) avec compatibilité ascendante
Contrôle de copie	Amélioration des performances de compression Correction d'erreurs après transmission

pour sécuriser les médias comme l'image, le son, la vidéo. Ces industriels ont encouragé financièrement la recherche dans le domaine. Ce sont également les applications ayant les plus fortes contraintes d'invisibilité, de robustesse, de « payload »¹, de temps réel, de sûreté.

Pour la problématique de **protection du droit d'auteur** (copyright), le tatouage permet d'insérer dans le média une information permettant de retrouver l'auteur du média tatoué. On utilise principalement des approches 0-bit comme celle de Broken Arrows [Furon et al. 08] (voir la description en section 1.4). Pour la problématique du **traçage de traîtres** [Furon 09], étant donnée la dimension des codes anti-collusions, les applications sont principalement envisageables pour du tatouage vidéo [Shahid et al. 11d], [Chaumont 10b]. Le chapitre 6 aborde le traçage de traître à travers la proposition d'un système de tatouage intégré à H.264 et utilisant le code anti-collusion de Tardos. L'objectif est d'intégrer au sein d'une vidéo l'identifiant de l'acheteur lors d'une vente. Pour être robuste aux attaques par collusion² il faut utiliser une approche jointe de tatouage numérique et de code anti-collusion comme celle de [Xie et al. 08] ou [Shahid et al. 11d]. Les approches d'**authentification** utilisent majoritairement des approches fragiles ou semi-fragiles et ne sont donc pas robustes. Bien souvent le résultat d'une fonction de hash, calculé sur un ensemble de coefficients représentatifs de l'image, est inséré de manière non robuste. Lors de l'extraction, si les coefficients représentatifs ont été modifiés, alors la signature qui était embarquée et la signature calculée sont différentes. Dans ce cas, l'image est considérée comme non authentique. Enfin, la problématique du **contrôle de copie** s'apparente de manière plus générale au contrôle de périphérique. La solution retenue par l'industrie du divertissement pour la lutte contre le piratage de DVDs illustre très bien cette problématique [Doërr 05]. Pour empêcher la copie pirate de DVDs, les constructeurs d'enregistreurs DVDs devaient intégrer un système de détection de présence de tatouage. L'enregistreur DVDs n'autorisait pas l'enregistrement s'il y avait la présence d'un signal de tatouage interdisant la copie. La sécurisation des lecteurs et des enregistreurs passait également par le chiffrement des DVDs³. Cela dit, on se rend compte en pratique que pour des applications aussi grand public, il est très difficile de mettre en place une solution viable. Il faut que la solution soit sûre à la fois aux niveaux logiciel et matériel, il faut prendre en compte la pression des utilisateurs en terme de compatibilité et de facilité d'utilisation, il faut réussir à imposer une norme

1. payload : quantité de bits insérée.

2. Une attaque par collusion met en jeu plusieurs pirates (des traîtres selon la terminologie) qui vont créer une nouvelle vidéo, en mélangeant les différentes versions de la même vidéo, ceci dans l'espoir de faire disparaître l'identifiant contenu dans chaque version.

3. Toute la chaîne de sécurité a rapidement été cassée et en particulier suite aux travaux de Johansen *et al.* qui ont trouvé les clefs du système de chiffrement du DVD en passant par une approche de rétro-ingénierie d'un lecteur DVDs [Patrizio 99].

qui soit respectée par les différents fournisseurs de technologie, enfin, il faut réussir à ce que les différentes législations empêchent la vente de solution détournant la norme de sécurisation.

Il y a également beaucoup d'applications qui ne sont pas directement liées à la sécurité, mais plutôt à l'enrichissement et dont certaines sont listées dans le tableau 1.1. Certaines sont détaillées dans le livre de Cox *et al.* [Cox et al. 07a], d'autres sont plus marginales et sont à chercher dans la littérature comme l'amélioration de performance de compression [Campisi et al. 02], [Thiesse et al. 10] ou la correction d'erreurs de transmission [Adsumilli et al. 05] sur un canal non fiable. Ces applications ne nécessitent pas forcément d'avoir de la robustesse. On parlera d'ailleurs pour certaines applications d'insertion de données cachées (data-hiding) plutôt que de tatouage. Nous reviendrons dans la partie III du manuscrit sur les applications d'enrichissement à travers la description d'une application de dissimulation de la couleur dans une image en niveaux de gris.

1.3 L'espace d'insertion, le codage informé et l'insertion informée

Comme nous l'avons vu dans la section 1.1 et sur la figure 1.1 un message binaire m est embarqué dans un signal hôte x pour obtenir un signal tatoué y . Le signal x est ce que l'on appelle *l'espace d'insertion*. C'est un vecteur qui représente l'image et qui est obtenu en utilisant tout ou une partie des pixels. Le choix de l'espace d'insertion c'est-à-dire des pixels à utiliser, des transformations à appliquer et des coefficients à retenir permet de définir le vecteur x . Parmi les schémas informés, les trois principaux représentants utilisent soit une partie des coefficients ondelette (BA : Broken Arrows [Furon et al. 08]) soit une partie des coefficients DCT (DPTC : Dirty Paper Trellis Code [Miller et al. 04], P-QIM : Perceptual-QIM [Li et al. 07]). L'algorithme de BA applique une décomposition ondelette en trois niveaux et utilise pour le tatouage toutes les sous-bandes sauf la sous-bande basse résolution. L'algorithme DPTC applique une transformée DCT 8×8 et utilise pour le tatouage les 12 premiers coefficients DCT de chaque bloc. Ces espaces d'insertion sont choisis de telle sorte qu'ils permettent d'obtenir des coefficients suffisamment stables (robustes). Ce sont en effet des coefficients de moyenne fréquence et/ou des coefficients qui lorsqu'ils sont modifiés n'ont pas un très fort impact psychovisuel. Une fois que le signal x est généré on a alors bien souvent un mécanisme de tatouage qui devient indépendant du type de média et donc qui est identique pour de l'image, du son, de la vidéo. Ce mécanisme peut se décomposer en deux étapes successives : l'étape de **codage informé** et l'étape d'**insertion informée**. La figure 1.3 reprend ces deux étapes du mécanisme d'insertion en reprenant les notations utilisées dans ce manuscrit.

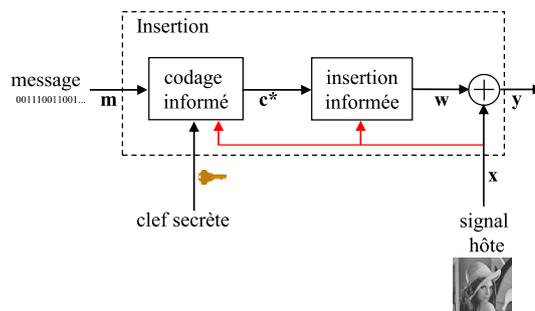


FIGURE 1.3 – Schéma détaillé de l'insertion informée.

La première étape (le **codage informé**), consiste à **coder** le message. Ce processus de codage

n'est pas exactement le même que pour un codage de type code correcteur. En effet, dans le cadre du codage informé, l'application de codage dispose pour chaque message binaire d'une multitude de mots de code \mathbf{c}^i représentant le message (alors que pour les codes correcteurs nous avons une fonction qui associe un unique mot de code à un message). L'étape de **codage informé** prend en entrée le vecteur binaire \mathbf{m} , le vecteur de coefficients réels \mathbf{x} et retourne en sortie le mot de code \mathbf{c}^* qui est un vecteur de coefficients réels de même dimension que \mathbf{x} . Notons d'ores et déjà que le mot de code \mathbf{c}^* est très proche de \mathbf{x} , c'est-à-dire corrélé au vecteur \mathbf{x} . Que cela soit pour Broken Arrows, pour P-QIM ou pour DPTC ce principe est le même. La section 1.4 décrit cette étape pour le cas de l'algorithme BA.

La deuxième étape est l'**insertion informée**. D'un point de vue géométrique, cela consiste à déplacer le vecteur \mathbf{x} vers le mot de code \mathbf{c}^* . Autour du mot de code \mathbf{c}^* , existe une région (appelé région de Voronoï) dont la signification est similaire à la région de décodage lorsque l'on utilise des codes détecteurs et correcteurs d'erreurs. Cette région correspond à l'ensemble des vecteurs (c'est-à-dire les mots) qui seront reconnus lors de l'extraction comme étant associés au vecteur mot de code \mathbf{c}^* . En fonction de la mesure de corrélation utilisée lors de la détection, on va soit déplacer le vecteur vers un hyper-cône dont l'axe central est le mot de code \mathbf{c}^* (cas de l'algorithme BA), soit déplacer le vecteur \mathbf{x} vers la région de Voronoï délimité par des hyper-plans (cas de l'algorithme DPTC), soit déplacer vers l'hyper-sphère centrée en \mathbf{c}^* (cas de l'algorithme P-QIM). La section 1.4 décrit cette étape pour le cas de l'algorithme BA.

Pour donner une illustration un peu plus précise du choix de l'espace d'insertion, de la phase de codage informé et de la phase d'insertion informée, nous allons détailler l'algorithme de Broken Arrows. Cet algorithme est un peu particulier puisqu'il n'y a pas à proprement parler d'information transmise entre l'émetteur (algorithme d'insertion) et le récepteur (algorithme de détection). En effet, l'émetteur insère un signal qui est connu du détecteur. On parle donc de schéma de tatouage 0-bits dans le sens où aucun bit n'est transmis. Bien qu'aucun bit ne soit transmis, on retrouve les notions d'espace d'insertion, de codage informé et d'insertion informée. Notons que le schéma BA aborde également le problème de la dégradation psychovisuelle, la notion de robustesse, et la notion de sécurité. Nous aborderons ces aspects à travers l'explication du schéma de BA. Il faut bien comprendre que tous ces aspects sont essentiels à la mise au point d'un système de tatouage mature. On peut d'ailleurs dire que l'algorithme de BA (et ses quelques améliorations [Xie et al. 10a], [Xie et al. 10b]) représente l'un des rares outils mature arrivant environ 20 ans après la naissance de la discipline.

Notons que nous détaillerons un peu plus les deux principaux algorithmes multi-bits informés dans les deux chapitres suivants. Le chapitre 2 reprend les principes de l'algorithme DPTC et propose principalement une autre approche pour l'insertion informée. Le chapitre 3 reprend les principes de l'algorithme P-QIM et propose de l'étendre à une approche basée TCQ.

1.4 Broken Arrows

Le schéma de tatouage de BA (Broken Arrows [Furon et al. 08]) est donné Figure 1.4 pour la partie insertion. Le schéma est illustré avec une image en niveaux de gris 512×512 sur 8 bits. Dans un premier temps, l'image subit une transformation en ondelettes 9/7 de Daubechies en 3 niveaux. L'ensemble des coefficients ondelettes exceptés ceux de la sous-bande basse fréquence est alors stocké dans le vecteur hôte \mathbf{x} . On a donc un espace d'insertion (le vecteur \mathbf{x}) qui est dans un domaine spatio-fréquentiel et qui ne prend pas en compte les très basses résolutions.

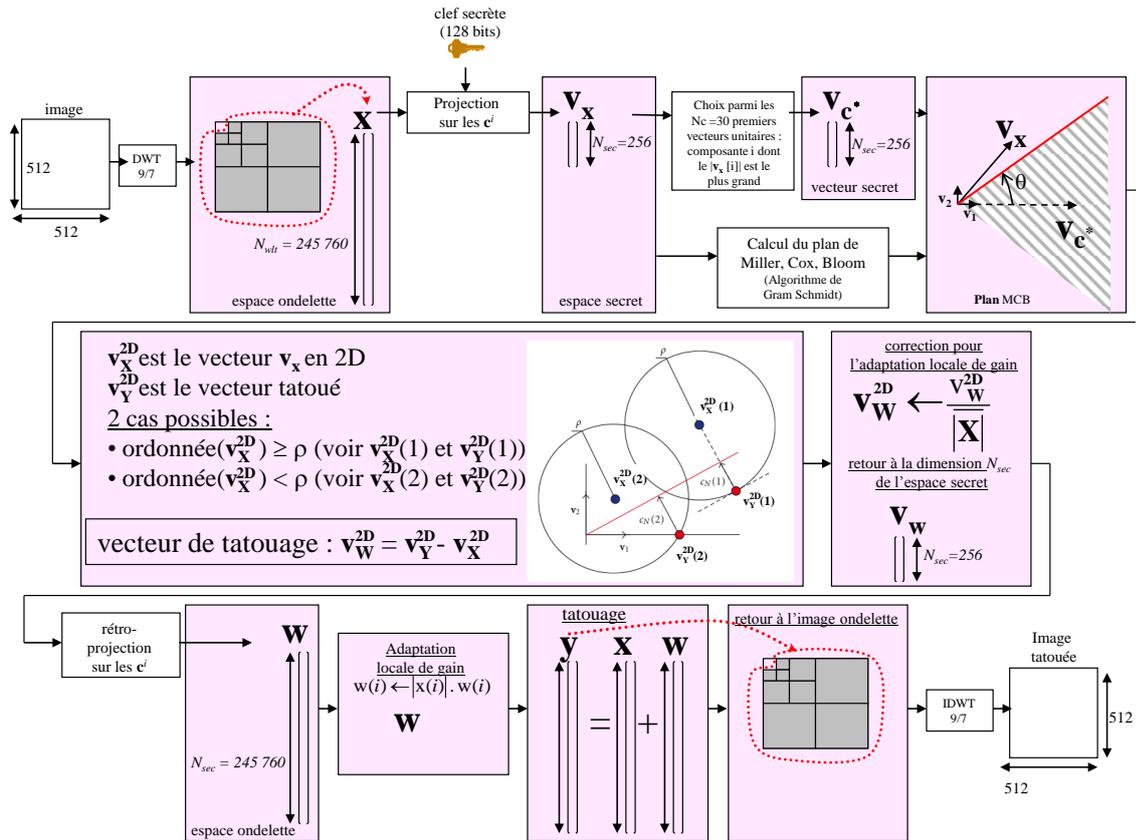


FIGURE 1.4 – Schéma général de tatouage par Broken Arrows [Furon et al. 08] pour une image en niveaux de gris 512x512.

Le vecteur hôte \mathbf{x} est alors « comparé » à N_{sec} vecteurs secrets noté $\mathbf{c}^i, i \in [1, N_{sec}]$. Ces vecteurs secrets sont obtenus : par l'utilisation d'un générateur pseudo-aléatoire qui tire des valeurs dans $\{+1, -1\}$, puis par normalisation à 1 des vecteurs. Les N_{sec} vecteurs secrets $\mathbf{c}^i, i \in [1, N_{sec}]$ sont quasi-orthogonaux. Notons que ces vecteurs secrets sont générés en initialisant le générateur avec une clef secrète. Cette même clef secrète sera nécessaire lors de la phase de détection pour détecter la présence de la marque. L'ensemble \mathcal{C} composé des N_{sec} vecteurs secrets correspond au livre du code (codebook) contenant l'ensemble des mots de code disponibles. Cet ensemble \mathcal{C} est donc également disponible du côté récepteur si celui-ci dispose de la clef secrète. En effet, le récepteur re-génère le livre du code ; c'est-à-dire re-génère les vecteurs secrets.

De la même façon que pour le tatouage multi-bit informé (voir Section 1.3), on retrouve une phase de type codage informé. Nous avons en entrée le vecteur hôte \mathbf{x} et l'ensemble des mots de code de \mathcal{C} . Il faut donc déterminer le mot de code $\mathbf{c}^* \in \mathcal{C}$ « le plus proche » de \mathbf{x} . On détermine cette proximité en calculant le produit scalaire entre \mathbf{x} et un sous-ensemble N_c de l'ensemble des

vecteurs de \mathcal{C}^4 et l'on détermine le vecteur \mathbf{c}^* le plus corrélé :

$$\begin{aligned}\mathbf{c}^* &= \text{sign}(\mathbf{x} \cdot \mathbf{c}') \cdot \mathbf{c}' \\ \mathbf{c}' &= \arg_{\mathbf{c}^i} \max_{i \in \{1, \dots, N_c\}} |\mathbf{x} \cdot \mathbf{c}^i|\end{aligned}\quad (1.1)$$

Au lieu de passer à la phase suivante qui consiste à faire l'insertion informée en prenant \mathbf{x} et \mathbf{c}^* en entrée et retournant le vecteur tatoué \mathbf{y} , BA ajoute une étape qui permet de sécuriser le schéma. Cette étape consiste à déterminer un nouvel espace pour l'insertion : l'espace secret. L'espace secret est obtenu en projetant le vecteur \mathbf{x} (par produit scalaire) sur l'ensemble des mots de code de \mathcal{C} . On obtient alors un vecteur secret $\mathbf{v}_x \in \mathbb{R}^{N_{sec}}$. L'insertion informée dans ce nouvel espace nécessite de raisonner sur cet espace secret. Il faut en effet choisir un mot de code qui soit « le plus proche » de \mathbf{v}_x . Comme on l'a vu à l'équation 1.1, dans l'espace d'insertion \mathbf{x} c'est le vecteur \mathbf{c}^* le plus corrélé en valeur absolue (parmi les N_c premiers vecteurs) qui est choisi.

Dans le même esprit, dans l'espace d'insertion secret \mathbf{v}_x , on choisit le vecteur **unitaire** « le plus proche » de \mathbf{v}_x parmi les N_c premiers vecteurs unitaires. Ce vecteur unitaire « le plus proche » de \mathbf{v}_x est un vecteur dont toutes les composantes sont nulles, exceptée la composante i telle que $|\mathbf{v}_x[i]|$ est maximum, qui est positionnée à $+/-1$. Notons que le fait de passer par l'espace secret permet d'avoir un étalement du signal de tatouage lors du retour dans l'espace ondelette \mathbf{x} . De la même façon que pour une approche par étalement de spectre, la puissance du signal de tatouage est alors répartie sur de nombreux coefficients. Notons \mathbf{v}_{c^*} unitaire le vecteur le plus corrélé à \mathbf{v}_x .

Une fois que le vecteur \mathbf{v}_x et le vecteur \mathbf{v}_{c^*} ont été déterminés, on passe à la phase d'insertion informée. Cette phase consiste à déplacer le vecteur \mathbf{v}_x vers la région de Voronoï de \mathbf{v}_{c^*} (voir Figure 1.4). La norme du vecteur de déplacement \mathbf{v}_w (ce vecteur de déplacement est le vecteur de tatouage dans l'espace secret) est réglable en fonction du PSNR (calculé entre l'image hôte et l'image tatouée) désiré à la fin de l'insertion via la formule (voir hypothèses et justifications dans [Furon et al. 08] ; équation 25 pour le cas de l'insertion proportionnelle) :

$$\|\mathbf{v}_w\| = \frac{\overline{|\mathbf{x}|}}{\sqrt{\overline{\mathbf{x}^2}}} 255 \sqrt{W \cdot H} 10^{-PSNR/20} \quad (1.2)$$

avec $\|\mathbf{v}_w\|$ la norme du vecteur de tatouage dans l'espace secret, $\overline{|\mathbf{x}|}$ la moyenne des valeurs absolues des coefficients de \mathbf{x} et $\sqrt{\overline{\mathbf{x}^2}}$ la racine carrée de la moyenne des carrés des coefficients de \mathbf{x} , W la largeur de l'image, et H la hauteur de l'image.

De plus, la région de Voronoï de \mathbf{v}_{c^*} est un hyper-cône d'angle θ . Les hypothèses fournies dans [Furon et al. 08] permettent de relier le paramètre θ à la probabilité de faux positif par la formule :

$$P_{fp} \leq N_c \frac{S_{N_{sec}-2}(\theta)}{S_{N_{sec}-2}(\pi/2)} \quad (1.3)$$

avec $S_{N_{sec}-2}(\theta)$ l'angle solide associé à θ en dimension $N_{sec} - 2$ [Miller et al. 99]. Lors de la compétition BOWS2, le paramètre θ était fixé à 1.2154 ce qui correspond à une probabilité inférieure ou égale à $P_{fp} = 3 \cdot 10^{-7}$ pour $N_c = 30$ cones. Lors de l'insertion informée, si le PSNR final (environ 43 dB) et la probabilité de faux positif (environ 10^{-6}) sont fixés, on se retrouve avec un vecteur \mathbf{v}_x qu'il faut déplacer de la norme maximum donnée à l'équation 1.2 pour aller

4. Dans l'algorithme original, on détermine la proximité à seulement $N_c = 30$ premiers vecteurs de \mathcal{C} .

pénétrer à l'intérieur d'un cône centré en \mathbf{v}_{c^*} et d'angle fixé. Le problème se résume alors à décider de l'endroit où l'on doit déplacer le vecteur \mathbf{v}_x à l'intérieur du cône avec la contrainte d'avoir le maximum de robustesse. Le problème est résolu de manière géométrique en 2D dans le plan formé par les vecteurs \mathbf{v}_x et \mathbf{v}_{c^*} (dans le chapitre 2 nous revenons sur la construction de ce plan 2D) et consiste à plonger dans le cône orthogonalement à la surface du cône. Si l'axe du cône est atteint alors le déplacement se fait co-linéairement à l'axe. Il est intéressant de noter que Furon et Bas montrent que le critère de robustesse lors de l'insertion [Furon et al. 08] est celui de la pire attaque. L'approche est donc plus générale que l'approche décrite dans [Cox et al. 07b] qui consiste à projeter sur une hyper-hyperbole.

Une fois que le vecteur de déplacement \mathbf{c}_w est calculé dans l'espace secret, celui-ci est divisé par la moyenne des valeurs absolues de \mathbf{x} . Cette opération est effectuée, car l'insertion dans l'espace ondelette est effectuée proportionnellement aux amplitudes des coefficients de \mathbf{x} de la manière suivante :

$$\mathbf{y}[i] = \mathbf{x}[i] + |\mathbf{x}[i]| \cdot \mathbf{w}[i], \quad (1.4)$$

avec \mathbf{w} le vecteur \mathbf{c}_w redimensionné puis rétroprojeté dans l'espace ondelette. Ce qu'il est surtout intéressant de noter dans l'équation 1.4, c'est le fait que chaque coefficient ondelette est modifié proportionnellement à son amplitude. De cette manière, le signal de tatouage \mathbf{w} est très bien intégré au signal \mathbf{x} et l'impact psychovisuel bien mieux pris en compte qu'en effectuant une simple addition de \mathbf{x} et de \mathbf{w} .

Pour résumer cette présentation succincte de l'algorithme de Broken Arrows, on rappellera les mots clefs importants de cet algorithme informé : espace d'insertion dans le domaine des ondelettes, génération pseudo-aléatoire des mots de code pour le codage informé, utilisation d'un espace secret, insertion informée robuste avec un raisonnement dans le plan 2D, insertion proportionnelle (psychovisuelle).

Dans les deux chapitres suivants nous allons aborder les deux principaux algorithmes multi-bits informés. Ces deux algorithmes reprennent les concepts d'espace d'insertion bien choisi, de codage informé et d'insertion informée. Nous proposons à travers ces deux chapitres de décrire une amélioration des algorithmes de base, mais également de déterminer quelle est l'approche la plus robuste, et/ou la plus sûre pour une dégradation fixée et un payload fixé. Pour pouvoir faire ces comparaisons, il a fallu mettre en place une plate-forme de test. Il est à noter que ces comparaisons n'étaient pas encore bien établies dans la littérature.

Chapitre 2

Une approche DPTC basée rotation : RB-DPTC

Résumé

Le schéma de tatouage par codes à papier sale par treillis (Dirty Paper Trellis Code : DPTC [Miller et al. 04]), publié en 2004, est un des schémas à forte capacité parmi les plus performants. Il possède cependant deux inconvénients majeurs : sa faiblesse en terme de sécurité et sa complexité en coût de calcul. Nous proposons de traiter ces deux problèmes par l'utilisation d'un espace d'insertion plus sûr et par l'utilisation d'une technique d'insertion plus rapide. L'espace d'insertion est construit par projections des coefficients ondelettes sur des porteuses secrètes. Cela renforce la sécurité, et cela permet d'obtenir de bonnes propriétés psycho-visuelles. L'insertion, quant à elle, repose sur une rotation dichotomique dans le plan de Cox, Miller et Bloom. Cette insertion donne de meilleures performances par rapport aux approches d'insertion de faible complexité. Quatre attaques différentes sont utilisées pour l'évaluation et les résultats obtenus montrent un bon comportement du schéma en terme de robustesse et de complexité opératoire.

2.1 Introduction

Les schémas informés (également appelés schémas à information adjacente) sont apparus en 1998 lorsque le travail de Costa [Costa 83] a été redécouvert. On distingue deux grandes catégories de systèmes de tatouage informés multi-bits : les schémas basés codes à lattice, également appelés codes en réseau, et plus couramment appelés schéma basés quantification (DC-QIM [Chen et al. 01], SCS [Eggers et al. 03]...) et les schémas basés treillis (DPTC [Miller et al. 04]).

L'algorithme original basé DPTC est connu pour sa bonne robustesse et sa haute capacité d'insertion, mais possède deux grosses faiblesses : l'étape d'insertion informée utilise une approche Monte-Carlo très complexe en coût de calcul et le schéma montre quelques faiblesses de sécurité vis-à-vis d'attaques par collusion [Bas et al. 08a]. Dans ce papier nous proposons un schéma DPTC au moins aussi sûr et moins complexe.

Lin *et al.* [Lin et al. 05] proposent de remplacer l'approche Monte-Carlo par une technique non-optimale mais de faible complexité. Nous proposons une solution encore plus efficace. Nous utilisons le domaine ondelettes qui ne produit pas les « effets de bloc » du domaine DCT (domaine utilisé dans l'approche DPTC initiale). Pour renforcer la sécurité et rendre encore plus difficile

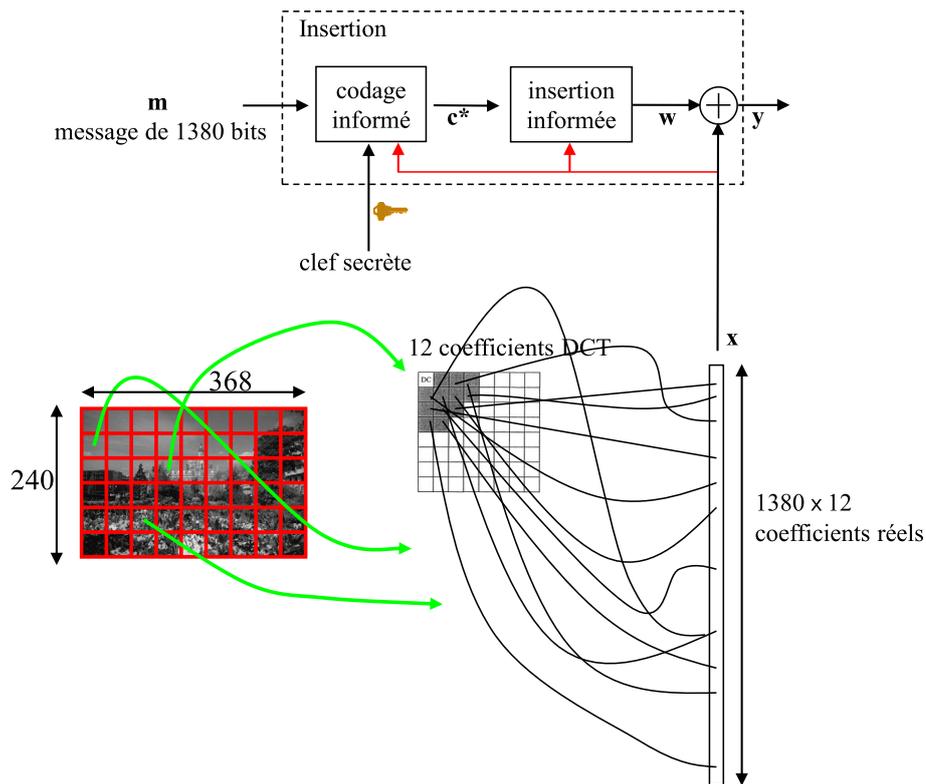


FIGURE 2.1 – Dirty Paper Trellis Codes appliqués sur une image 240×368 .

l'attaque présentée dans [Bas et al. 08a], nous réalisons l'insertion du signal de tatouage dans un espace secret. Finalement, puisque notre technique est rapide et que l'espace d'insertion est adapté, nous augmentons la taille du treillis (plus exactement nous augmentons la taille du *livre du code*¹) ce qui nous permet d'augmenter la robustesse [Wang et al. 06] et de réduire la distorsion.

Dans la section 2.2, nous rappelons le principe du DPTC original [Miller et al. 04]. Dans la section 2.3 nous présentons l'espace d'insertion et l'algorithme d'insertion. Enfin, dans la section 2.4 nous comparons le DPTC original [Miller et al. 04], l'approche de Lin *et al.* [Lin et al. 05], notre approche basée rotation (Rotation Based Dirty Paper Trellis Code : RB-DPTC) et une version psychovisuelle (PR-RB-DPTC).

2.2 Le schéma DPTC original

Le schéma DPTC original [Miller et al. 04] appliqué sur une image $N = 240 \times 368$ est illustré sur la figure 2.1.

La première étape du schéma consiste à transformer l'image en une représentation spatio-fréquentielle (par transformée DCT) pour obtenir le *signal hôte* \mathbf{x} . Dans le schéma original, on applique à l'image une transformée DCT 8×8 , puis les douze premiers coefficients ACs de chaque bloc DCT sont extraits et ordonnés pseudo-aléatoirement dans un vecteur \mathbf{x} de taille $12 \times N/64 = 3 \times N/16$.

1. livre du code : *codebook* en anglais.

La deuxième étape du schéma DPTC est le *codage informé*. Le message binaire \mathbf{m} à insérer est codé en un mot de code \mathbf{c}^* en prenant en compte le *signal hôte* \mathbf{x} . Pour réaliser ce codage, un treillis non déterministe et l’algorithme de Viterbi sont utilisés (cette partie est détaillée dans [Miller et al. 04] et dans [Chaumont 10a]).

La dernière étape du schéma DPTC est l’*insertion informée*. Cette étape consiste à modifier le *signal hôte* \mathbf{x} pour le « déplacer » dans la région de Voronoï du mot de code \mathbf{c}^* . Les auteurs de [Miller et al. 04] utilisent une approche Monte-Carlo (avec un critère de robustesse prédéfini) pour effectuer le « déplacement » du *signal hôte* \mathbf{x} dans la région de Voronoï du mot de code \mathbf{c}^* . L’approche est itérative et consiste à attaquer (i.e. dégrader) le *signal tatoué* \mathbf{y} puis contre-attaquer en régénérant un nouveau *signal tatoué* \mathbf{y} . Cette technique nécessite un grand nombre d’appels à l’algorithme de Viterbi. Même avec les optimisations proposées dans [Miller et al. 04], la complexité en temps de calcul est élevée et c’est actuellement une forte limitation, que cela soit pour son évaluation intensive ou bien pour son utilisation au sein d’un système logiciel ou matériel².

2.3 Notre schéma RB-DPTC

Dans cette section, nous présentons notre nouvel espace d’insertion et notre nouvelle approche pour l’insertion.

2.3.1 L’espace d’insertion

Le travail récent de Bas et Doërr [Bas et al. 08a] sur la sécurité de l’approche DPTC originale [Miller et al. 04] montre que dans le cadre de Kerckhoffs [Kerckhoffs 83], c’est-à-dire lorsque les algorithmes d’insertion et d’extraction sont publics et donc à la disposition d’attaquants, le *livre du code* \mathcal{C} peut-être retrouvé³ en observant un grand nombre d’images tatouées (tatouées avec l’utilisation de la même clef secrète). Ce résultat est obtenu sur une version simplifiée de l’algorithme DPTC⁴, mais montre néanmoins l’existence d’une certaine faiblesse de sécurité dans l’algorithme. L’espace privé que nous utiliserons dans notre schéma permet de cacher la structure du treillis et devrait rendre plus difficile une attaque à la sécurité du type de celle proposée par Bas et Doërr [Bas et al. 08a]. En outre, il est certainement très difficile d’estimer les projections secrètes de la même manière que dans [Bas et al. 09] car le nombre de mots-de-code est trop grand (pour un treillis composé de 128 états, de 128 arcs par état, et d’un payload⁵ de 1024 bits, cela fait plus de 10^{387} mots-de-code).

La figure 2.2 illustre l’espace d’insertion de notre proposition Rotation-Based Dirty Trellis Codes (RB-DPTC). Notre nouvel espace d’insertion est obtenu tout d’abord par une transformation ondelettes de l’image, puis par une projection du *signal hôte* \mathbf{x} de dimension N_{wlt} (\mathbf{x} est la concaténation des coefficients des sous-bandes, exceptés les coefficients de la sous-bande LL) sur N_{sec} porteuses (notées \mathbf{u}_i avec $i \in [1, N_{sec}]$). La projection est juste un produit scalaire. Remarquons également que la complexité de la projection peut facilement être réduite en une complexité

2. Sur un PC à 3Ghz, en fonction du seuil de robustesse, l’insertion peut prendre de 30 min à 90 min sur une image 256×256 .

3. Plus exactement, ce sont les coefficients attachés aux arcs du treillis qui peuvent être assez bien estimés.

4. il n’y a pas de ré-ordonnancement pseudo-aléatoire des coefficients DCTs.

5. payload : quantité de bits insérée.

linéaire avec une approche de Space Division Multiplexing [Chaumont 09a]. Le vecteur obtenu \mathbf{v}_x de dimension N_{sec} peut alors être utilisé pour le *codage informé* et l'*insertion informée*.

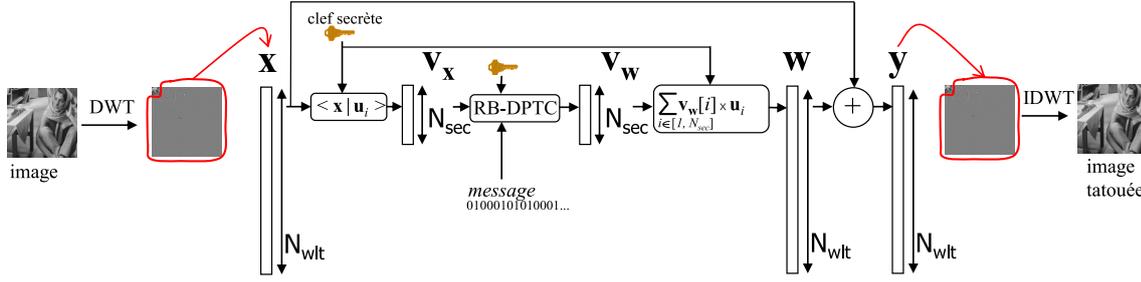


FIGURE 2.2 – L’espace d’insertion pour notre schéma Rotation Based Dirty Paper Trellis Code (RB-DPTC).

Cet espace d’insertion est au moins aussi sûr que l’original car il permet de disperser le signal de tatouage sur tout le domaine fréquentiel (il n’y a donc pas de super robustesse [Craver et al. 07]). De plus, la projection sur les N_{sec} porteuses donne à l’espace d’insertion un aspect Gaussien connu pour ses bonnes capacités de canal [Costa 83]. Finalement, le domaine ondelettes a de bonnes propriétés psycho-visuelles et génère des effets moins gênants (ou du moins, moins tranchés) que les effets de bloc du schéma DPTC original [Miller et al. 04].

2.3.2 L’étape d’insertion informée

Le codage informé que nous utilisons est le même que celui du DPTC original, mais il est réalisé à partir du vecteur hôte \mathbf{v}_x (c’est-à-dire à partir de l’espace secret). Le codage-informé permet d’obtenir le mot de code \mathbf{c}^* . Pour accélérer l’insertion et conserver un bon compromis robustesse-distorsion, nous proposons une solution non optimale, mais meilleure que l’approche de Lin *et al.* [Lin et al. 05].

Rappelons qu’au décodage, le mot de code le plus corrélé $\tilde{\mathbf{c}}^*$ est obtenu en exécutant l’algorithme de Viterbi sur le treillis « complet ». Ce mot de code $\tilde{\mathbf{c}}^*$ appartient au *livre du code* \mathcal{C} et maximise la corrélation avec le vecteur tatoué-attaqué $\tilde{\mathbf{v}}_y$:

$$\begin{aligned}\tilde{\mathbf{c}}^* &= \arg_{\mathbf{c}^i \in \mathcal{C}} \max (\tilde{\mathbf{v}}_y \cdot \mathbf{c}^i) \\ &= \arg_{\mathbf{c}^i \in \mathcal{C}} \max (||\tilde{\mathbf{v}}_y|| \cdot ||\mathbf{c}^i|| \cdot \cos \theta_i),\end{aligned}$$

avec θ_i l’angle entre $\tilde{\mathbf{v}}_y$ et \mathbf{c}^i . Sachant que tous les mots de code possèdent la même norme, l’algorithme de Viterbi extrait donc le mot de code $\mathbf{c}^i \in \mathcal{C}$ formant le plus petit angle $|\theta_i|$ avec $\tilde{\mathbf{v}}_y$. En supposant que le bruit d’attaque est un bruit blanc gaussien, l’attaque n’a aucune influence sur la corrélation et l’on retrouve, lors de l’extraction, le mot de code \mathbf{c}^* utilisé lors de l’insertion.

Pour insérer le message m lors du codage, il suffit donc de réduire l’angle entre le *vecteur hôte* \mathbf{v}_x et le mot de code \mathbf{c}^* , jusqu’à obtenir le plus petit angle (parmi tous les autres angles possibles $(\widehat{\mathbf{v}_x, \mathbf{c}^i})$).

Pour réduire l’angle entre \mathbf{v}_x et \mathbf{c}^* , nous exprimons d’abord les deux vecteurs dans le plan de Miller, Cox et Bloom (*abrév.* plan MCB) [Cox et al. 07b]. La figure 2.3 illustre ce plan. Le plan MCB est défini par une base ortho-normalisée $(\mathbf{v}_1, \mathbf{v}_2)$ telle que le \mathbf{v}_x et \mathbf{c}^* appartiennent à ce

plan (algorithme de Gram-Schmidt) :

$$\mathbf{v}_1 = \frac{\mathbf{c}^*}{\|\mathbf{c}^*\|}, \quad \mathbf{v}_2 = \frac{\mathbf{v}_x - (\mathbf{v}_x \cdot \mathbf{v}_1)\mathbf{v}_1}{\|\mathbf{v}_x - (\mathbf{v}_x \cdot \mathbf{v}_1)\mathbf{v}_1\|}$$

Dans le plan MCB, les coordonnées 2D du *vecteur hôte* \mathbf{v}_x sont :

$$\begin{aligned} \mathbf{v}_x^{2D}[1] &= \mathbf{v}_x \cdot \mathbf{v}_1, \\ \mathbf{v}_x^{2D}[2] &= \mathbf{v}_x \cdot \mathbf{v}_2, \end{aligned}$$

et les coordonnées 2D du mot de code \mathbf{c}^* sont :

$$\begin{aligned} \mathbf{c}_{2D}^*[1] &= \|\mathbf{c}^*\|, \\ \mathbf{c}_{2D}^*[2] &= 0. \end{aligned}$$

Une rotation du *vecteur hôte* \mathbf{v}_x^{2D} d'un angle θ dans le plan MCB est telle que :

$$\begin{aligned} \mathbf{v}_y^{2D}[1] &= \cos\theta \cdot \mathbf{v}_x^{2D}[1] - \sin\theta \cdot \mathbf{v}_x^{2D}[2], \\ \mathbf{v}_y^{2D}[2] &= \sin\theta \cdot \mathbf{v}_x^{2D}[1] + \cos\theta \cdot \mathbf{v}_x^{2D}[2]. \end{aligned}$$

Si nous réduisons la valeur absolue de l'angle entre le *vecteur hôte* \mathbf{v}_x et le mot de code \mathbf{c}^* dans le plan MCB, cela augmente la corrélation $\mathbf{v}_x \cdot \mathbf{c}^*$. Avec une approche dichotomique sur l'angle de rotation, on peut rapidement trouver une frontière de Voronoï. L'algorithme consiste à itérer une dizaine de fois les instructions suivantes :

1. tourner \mathbf{v}_x et obtenir \mathbf{v}_y dans le plan MCB,
2. exécuter l'algorithme de Viterbi sur le treillis « complet » et tester si \mathbf{v}_y appartient ou non à la région de Voronoï de \mathbf{c}^* i.e. vérifier si le vecteur décodé est égal ou non à \mathbf{c}^* ,
3. modifier l'angle de rotation en fonction du résultat d'appartenance à la région de Voronoï region et retourner en 1.

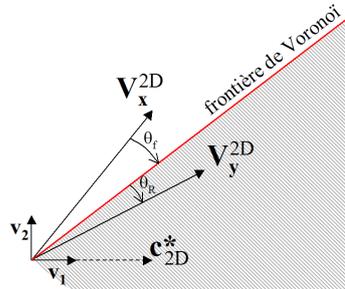


FIGURE 2.3 – Insertion basée rotation dans le plan de Miller, Cox et Bloom

Une fois que l'angle frontière θ_f dans le MCB est trouvé, on améliore la robustesse d'insertion en pénétrant à l'intérieur de la région de Voronoï avec un angle θ_R . Notre insertion informée est donc une rotation du *vecteur hôte* \mathbf{v}_x d'un angle orienté égal à $\max(\theta_f + \theta_R, \widehat{(\mathbf{v}_x, \mathbf{c}^*)})$. La figure 2.3 illustre \mathbf{v}_x , \mathbf{v}_y , θ_f et θ_R dans le plan MCB. Notons qu'un des critères de sécurité proposé dans [Xie et al. 10b] pour contrecarrer l'attaque par analyse en composante principale de [Bas et al. 09], et tenter d'améliorer l'algorithme de Broken Arrows consiste à imposer $\|\mathbf{v}_y\| = \|\mathbf{v}_x\|$. C'est exactement ce que nous faisons en effectuant une rotation de \mathbf{v}_x ; la norme de \mathbf{v}_y est égale à la norme de \mathbf{v}_x .

2.3.3 Une extension psychovisuelle

Afin que l'impact du tatouage soit psychovisuellement invisible, il est classique de « mettre en forme » le signal de tatouage à travers l'utilisation d'un masquage psychovisuel. Grosso modo, la puissance du signal de tatouage doit être réduite dans les régions uniformes et doit être augmentée dans les régions texturées ou les régions de contours. L'extension psychovisuelle que nous présentons brièvement ici est traitée plus en détails dans [Chaumont 09b]. L'algorithme est appelé Psychovisual Rudimentary mask RB-DPTC (PR-RB-DPTC). Notons que dans l'algorithme PR-RB-DPTC nous avons également ajouté une phase de codage du message par un code détecteur/correcteur d'erreurs de rendement 1/2 [Chaumont 09b]. La figure 2.4 donne le schéma général d'insertion avec l'utilisation d'un masque psychovisuel. On voit donc se greffer trois grandes étapes par rapport au schéma de la figure 2.2 :

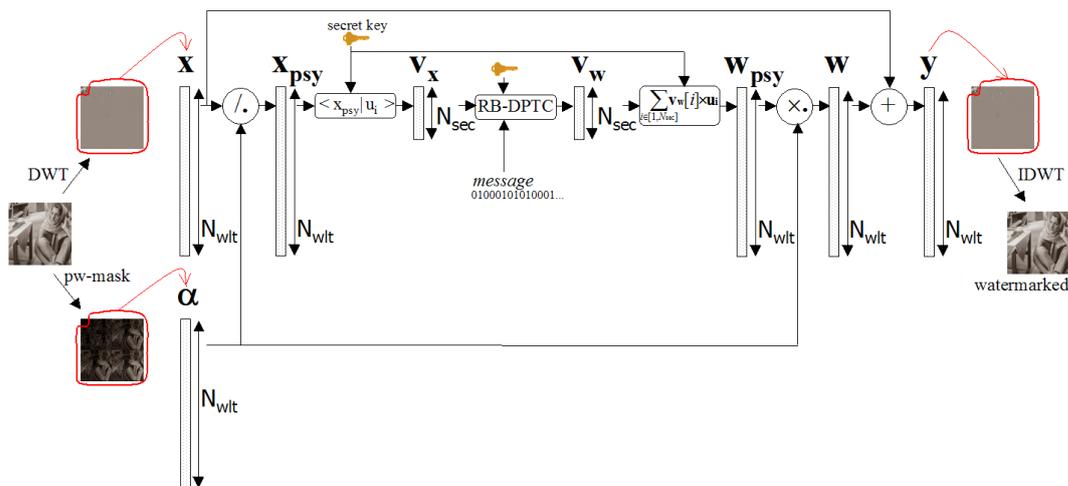


FIGURE 2.4 – Embedding scheme with a psychovisual mask

1. la construction d'un espace psychovisuel \mathbf{x}_{psy} . Dans cet espace, les coefficients sont psychovisuellement équivalents. Les coefficients sont donc tous traités de la même façon. Cet espace psychovisuel est tel que $\forall i \in [1, N_{wlt}], \mathbf{x}_{psy}[i] = \mathbf{x}[i]/\alpha[i]$, avec α le masque psychovisuel ;
2. la mise en forme du signal de tatouage grâce au masque α : $\forall i \in [1, N_{wlt}], \mathbf{w}[i] = \mathbf{w}_{psy}[i] \times \alpha[i]$. Cette mise en forme réduit l'impact psychovisuel du tatouage dans les régions où celui-ci aurait été visible. Par exemple, la valeur de α sera faible dans les régions où l'intensité lumineuse est uniforme, afin de réduire la puissance de \mathbf{w} dans ces régions ;
3. l'insertion du signal de tatouage « mis en forme ». Cette insertion est telle que $\forall i \in [1, N_{wlt}], \mathbf{y}[i] = \mathbf{w}[i] + \mathbf{x}[i]$.

Lors du décodage, le décodeur doit recalculer le masque psychovisuel. C'est une particularité de l'approche que nous proposons. Dans l'algorithme BA ou DPTC, l'impact psychovisuel est pris en compte lors de l'insertion et il n'y a pas de calcul de masque lors de l'extraction. Cela dit, les approches utilisées dans BA et DPTC sont difficilement réapplicables pour l'approche RB-DPTC car les hypothèses ne sont pas adaptées ou bien parce que la complexité calculatoire est élevée. L'approche de [Le Guelvouit 09] pourrait, être envisagée et consiste à effectuer la mise en forme

du signal de tatouage lors de la phase de rétroprojection sur les porteuses. Notons que l'approche que nous proposons permet d'utiliser n'importe quel masque psychovisuel de la littérature.

Le calcul du masque psychovisuel lors de l'extraction est une phase délicate, car le masque recalculé doit être le même que celui utilisé lors de l'insertion. Si le masque recalculé diffère trop du masque calculé lors de l'insertion, l'extraction du message inséré risque d'être erronée. Ainsi, le masque doit en plus d'être un masque psychovisuel, posséder des propriétés de robustesse aux différentes attaques d'un système de tatouage. Notons qu'à ma connaissance aucune recherche n'a pour le moment traité des masques psychovisuels robustes.

Ainsi, le décodeur extrait le vecteur ondelette à partir de l'image tatouée attaquée, divise chaque composante i par $\alpha[i]$ (α est recalculé au décodeur), projette le vecteur résultant sur les porteuses secrètes puis récupère le mot de code le plus proche (et donc le message) à partir du livre du code \mathcal{C} .

2.4 Evaluations expérimentales

Les expérimentations ont été réalisées sur les 100 premières images de la base de données de BOWS-2⁶ avec des images redimensionnées en 256×256 ⁷. Ces images sont des photos d'amateurs en niveaux de gris codées sur 8 bits.

Quatre algorithmes sont évalués : l'algorithme **DPTC original** dans le domaine DCT, l'algorithme inspiré de **Lin et al. basé cône** dans le domaine ondelette, notre algorithme basé rotation **RB-DPTC**, et la version **PR-RB-DPTC** ajoutant un masque psychovisuel et un code correcteur.

La structure du treillis possède 128 états avec 128 arcs par états pour Lin et al. basé cône, RB-DPTC et PR-RB-DPTC, et 64 états avec 64 arcs par états pour DPTC original. Les étiquettes des arcs de sortie sont tirées d'une distribution Gaussienne. Il y a 12 coefficients par arc pour Lin et al. basé cône, DPTC, RB-DPTC et 10 coefficients par arc pour PR-RB-DPTC. Le payload est le même que dans l'article DPTC [Miller et al. 04], c'est-à-dire 1 bit pour 64 pixels. Le nombre de bits insérés est donc de 1024 pour une image 256×256 .

Pour l'algorithme inspiré de Lin et al. basé cône, et les algorithmes RB-DPTC et PR-RB-DPTC, nous utilisons le même espace d'insertion. La transformée ondelettes est une 9/7 Daubechies avec $l = 3$ niveaux de décomposition. Exceptée la sous-bande LL, toutes les autres sous-bandes sont utilisées pour former le *signal hôte* \mathbf{x} . Avec des images 256×256 , la taille de l'espace ondelettes est de $N_{wlt} = 64\ 512$ coefficients. Sachant que la capacité est de $1/64$ bits par pixel et que le nombre de coefficients de sortie pour un arc est $N_{arc} = 12$ coefficients pour Lin et al. et RB-DPTC, la taille de l'espace privé est donc de $N_{sec} = 1024 \times 12 = 12\ 288$ coefficients. Pour PR-RB-DPTC la taille de l'espace privé est de $N_{sec} = 1024 \times 10 = 20\ 480$ coefficients. Les porteuses \mathbf{u}_i sont construites à partir de séquences pseudo-aléatoires bipolaires normalisées.

Quatre attaques à la robustesse ont été expérimentées : l'attaque par ajout de bruit Gaussien, l'attaque par filtrage, l'attaque valométrique de changement d'échelle et l'attaque par compression JPEG. Ces quatre attaques sont décrites en détails dans [Miller et al. 04]. Le Taux d'Erreur Binaire (Bit Error Rate : BER) est calculé à partir du message extrait et il est égal au nombre de bits erronés divisé par le nombre total de bits insérés. Le BER est calculé pour chaque attaque.

Quatre algorithmes sont en compétition pour un PSNR d'insertion proche de 42.6 dB : le **DPTC original** (dans l'espace DCT) avec un PSNR moyen d'insertion = 42.6 dB, l'algorithme de

6. la base de données de BOWS2 est téléchargeable à l'adresse <http://bows2.gipsa-lab.inpg.fr/>.

7. le sous-échantillonnage d'images a été réalisé avec le programme xview et utilisant l'interpolation de Lanczos.

Lin *et al.* **basé cône** (dans l'espace ondelette secret) avec la robustesse positionnée à un bruit de puissance $n^2 = 1$ (ce qui correspond à $R_t = 1$ dans l'article [Lin *et al.* 05]) et un PSNR moyen d'insertion = 34.2 dB (remarquons qu'il est impossible d'augmenter le PSNR de Lin *et al.*, leur technique est très sous-optimale lorsque l'on expérimente avec des images réelles), l'algorithme **basé rotation RB-DPTC** (dans l'espace ondelette secret) avec un angle de pénétration de $\theta_R = 0.1$ radian et un PSNR moyen d'insertion = 42.4 dB, et enfin l'algorithme PR-RB-DPTC (dans l'espace ondelette secret) avec un angle de pénétration de $\theta_R = 0.09$ radian et un PSNR moyen d'insertion = 41.75 dB.

Les figures 2.5, 2.6 2.7, 2.8 nous donnent les résultats de BER pour les quatre attaques. D'emblée, nous écartons les résultats de l'approche de Lin *et al.* puisqu'il est impossible d'avoir un PSNR moyen supérieur à 34.2 dB. L'algorithme de Lin *et al.* ne peut donc pas être utilisé comme substitut rapide à l'algorithme DPTC original, puisqu'il n'est pas capable d'atteindre un PSNR raisonnable de 42 dB. La comparaison est donc uniquement réalisée entre l'algorithme DPTC original, notre algorithme basé rotation ainsi que sa version psychovisuelle.

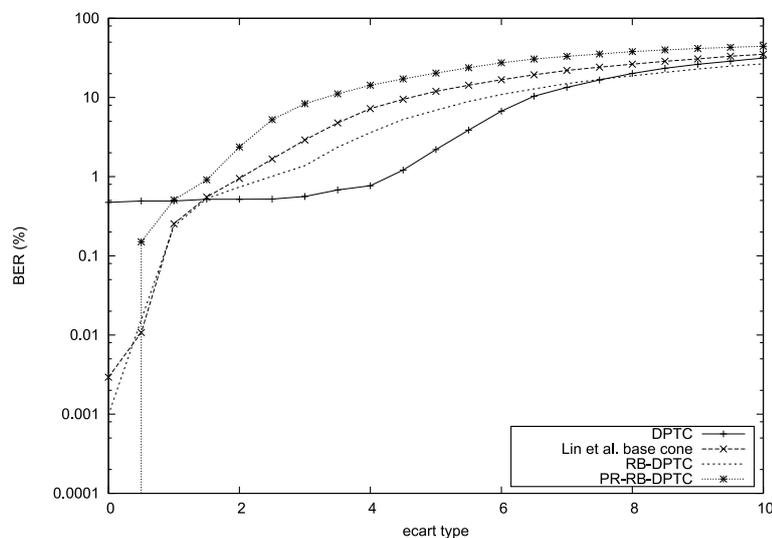


FIGURE 2.5 – BER pour une attaque par ajout de bruit gaussien.

On peut constater que l'algorithme DPTC possède un taux d'erreur de 0.5% bits erronées même lorsqu'il n'y a pas d'attaque. Ce taux pourrait être réduit en augmentant le nombre d'itérations lors de la phase d'insertion informée par Monte-Carlo, ainsi qu'en affinant la granularité d'incrément de la puissance d'attaque dans l'algorithme de Monte-Carlo. La durée d'expérimentation est déjà de l'ordre de deux semaines d'exécution sur un PC à 3GHz et augmenter le nombre d'itérations ainsi que la granularité de la puissance d'attaque entrainerait un très fort accroissement de cette durée. Le DPTC originale n'est clairement pas adapté à une insertion au payload de 1 bit pour 64 pixels.

L'algorithme basé rotation RB-DPTC a des taux d'erreur de 0.001 % lorsqu'il n'y a pas d'insertion, ce qui est bien meilleur que pour le DPTC original. On constate également que l'ajout d'un code correcteur en plus d'un masque psychovisuel (PR-RB-DPTC) permet d'obtenir un taux d'erreur nul lorsqu'il n'y a pas d'attaque. De manière générale, on constate que le taux d'erreur atteint les 10% pour des puissances d'attaque moyennes. C'est un constat que l'on peut avoir pour

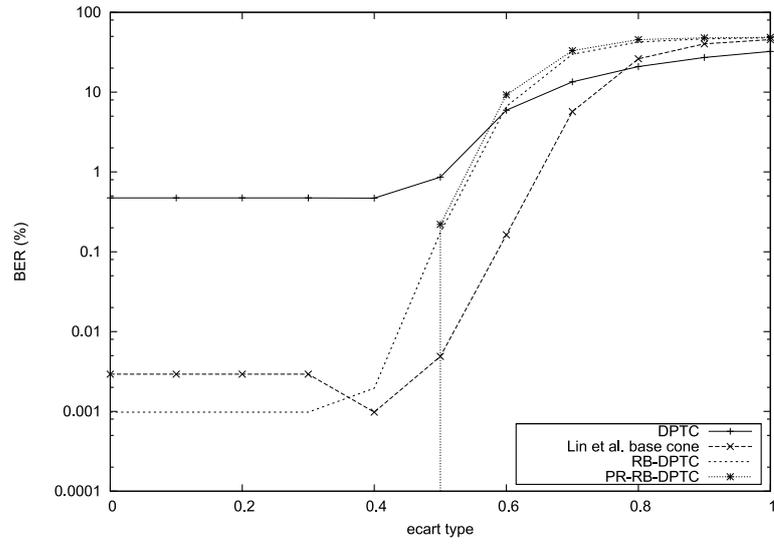


FIGURE 2.6 – BER pour une attaque par filtrage gaussien.

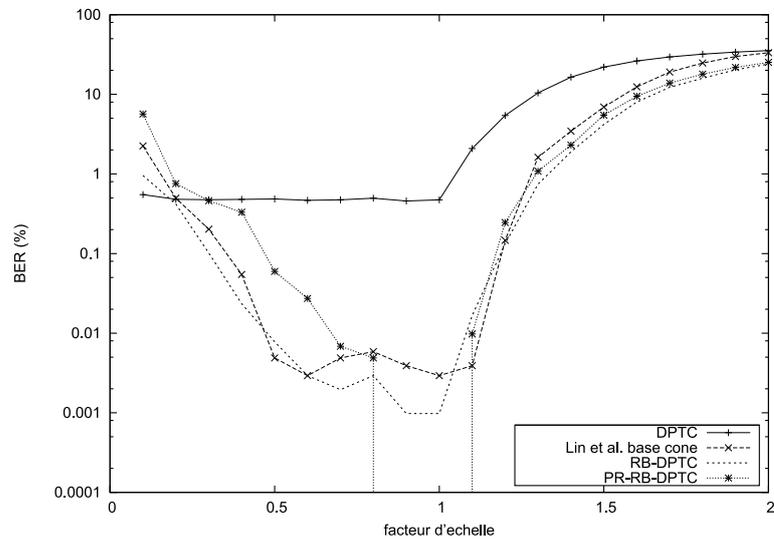


FIGURE 2.7 – BER pour une attaque valométrique de changement d'échelle.

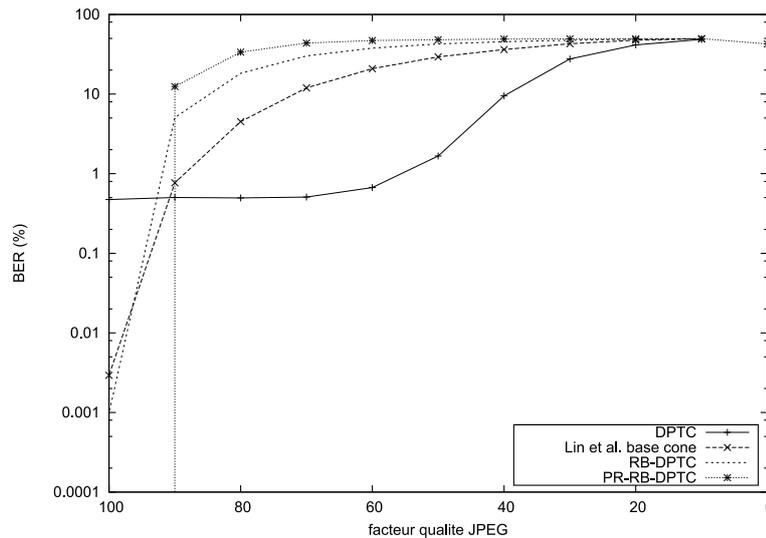


FIGURE 2.8 – BER pour une attaque par compression JPEG.

toutes les approches qui effectuent un tatouage à un payload aussi élevé. Si l'on regarde de plus près les courbes, pour l'attaque valométrique de changement d'échelle et l'attaque de filtrage, le BER varie de la même façon que pour l'algorithme DPTC tout en étant bien plus petit à faible puissance d'attaque. Pour l'attaque par filtrage, les approches RB-DPTC et PR-RB-DPTC ont un BER qui atteint les 10% bien plus rapidement que l'approche DPTC, bien que le BER soit initialement plus faible lorsque l'attaque est de faible puissance. Enfin, on constate que les approches RB-DPTC et PR-RB-DPTC sont sensibles à l'attaque JPEG puisque le seuil d'erreur de 10% est franchi pour un facteur de qualité supérieur à 80. Cette sensibilité s'explique par l'insertion dans des ondelettes de hautes fréquences.

Pour résumer, les approches RB-DPTC ou PR-RB-DPTC sont intéressantes lorsque l'on désire effectuer des insertions à fort payload avec une approche basée treillis. Les approches permettent de réduire la complexité (une insertion ne prend que quelques secondes sur un PC à 3GHz) de l'algorithme DPTC original et ont de très bonnes performances en robustesse. De plus, nos approches garantissent une bonne sécurité via l'utilisation d'un espace d'insertion sûr [Furon et al. 08].

2.5 Conclusion

Comme nous l'évoquons dans l'introduction, un des objectifs de notre travail sur le tatouage robuste d'images est de parvenir à positionner les deux schémas modernes de tatouage d'images robuste à fort payload : les schémas par treillis (chapitre 2) et les schémas par quantification (chapitre 3). Afin de réaliser une comparaison rigoureuse, ces deux approches doivent être testées avec les mêmes caractéristiques les rendant les plus efficaces : les schémas doivent donc prendre en compte l'aspect psychovisuel, être informés, intégrer des codes correcteurs, et être à faible complexité. Or la plupart des propositions de la littérature équipent l'une ou l'autre des deux approches de certaines de ces caractéristiques, mais jamais toutes. Un premier travail a donc consisté à mettre en place une approche basée quantification et une approche basée treillis, avec ces quatre

caractéristiques.

Dans ce chapitre, nous avons présenté l'approche mise en place pour un schéma basé treillis. Cette approche se base sur un nouvel algorithme de codes à papier sale par treillis (Dirty Paper Trellis Code : DPTC). L'idée de cet algorithme était de remplacer le module d'insertion basé sur une optimisation de Monte-Carlo par une approche basée sur une rotation du vecteur hôte dans le plan MCB. En effet, l'optimisation de Monte-Carlo est réputée complexe et empêche d'effectuer des expérimentations à grande échelle. L'approche que j'ai proposée, bien que sous-optimale du point de vue de l'optimisation robustesse-distorsion, est beaucoup moins complexe pour des résultats peu éloignés en robustesse, et a donc permis de réaliser des expérimentations à large échelle.

Plus précisément, par rapport à l'algorithme original, nous utilisons le domaine ondelette au lieu du domaine DCT. La sécurité est au moins aussi bonne que dans l'algorithme DPTC original grâce à l'ajout d'un espace d'insertion secret. Cet espace secret est obtenu en projetant les coefficients ondelettes sur des porteuses orthogonales. Cette projection garantit également (pendant la rétro-projection) une dispersion du signal de tatouage sur tous les coefficients ondelettes. Nous présentons également une nouvelle approche d'insertion basée rotation. L'objectif est de déplacer le *vecteur hôte* dans la région de Voronoï du mot de code représentant le message. Le déplacement consiste à faire faire une rotation au *vecteur hôte* par rapport à l'axe défini par le mot de code. L'angle de rotation utilisé pour déterminer la frontière de Voronoï est obtenu de manière dichotomique. Le *vecteur hôte* est ensuite « tourné » de sorte qu'il y ait une pénétration dans la région de Voronoï d'un angle prédéfini. Les résultats sont meilleurs que l'approche rapide de Lin *et al.* et sont bons en comparaison de l'approche originale [Miller et al. 04]. De plus, notre approche est de bien plus faible complexité calculatoire que l'approche originale [Miller et al. 04].

Nous avons également proposé une version psychovisuelle de l'algorithme RB-DPTC en utilisant un masque psychovisuel et un code correcteur d'erreur. L'article [Chaumont 09b] détaille cet aspect et montre que pour un SSIM de 98% [Wang et al. 04], les résultats en robustesse sont meilleurs que pour l'approche RB-DPTC.

De nombreuses améliorations sont envisageables. Une des premières améliorations à apporter consiste à améliorer la robustesse du schéma face à JPEG. L'utilisation d'un espace d'insertion basé DCT pourrait améliorer la robustesse. La sensibilité de l'espace secret à l'attaque de Westfeld [Bas et al. 09] a été traitée par l'utilisation d'un masquage local [Xie et al. 10a]. Cette approche pourrait être ajoutée pour augmenter le niveau de sécurité. Le schéma doit alors être réglé sur des payload plus faibles, car cet ajout réduit la dimension de l'espace secret. L'approche de [Le Guelvouit 09] pour mettre en forme psychovisuellement le signal de tatouage lors de la rétroprojection sur les porteuses est prometteuse et permettrait peut être de gagner en robustesse. Lors de la phase d'extraction du message, cela permettrait également d'éviter d'avoir à recalculer le masque puis à utiliser ce masque, détérioré par les attaques, pour pouvoir extraire le message. La rotation telle que nous l'avons définie pourrait être guidée par une valeur de dégradation comme le SSIM [Wang et al. 04]. Une analyse plus poussée de la sécurité pourrait être envisagée. Enfin, le code DPTC peut encore largement être amélioré comme cela a déjà été proposé dans [Wang et al. 06], [Wang et al. 07].

Chapitre 3

Une approche QIM Perceptuel (Multi-Hyper-Cube)

Résumé

En 2007, Li et Cox [Li et al. 07] démontraient que leur schéma appelé Perceptual-QIM (P-QIM) était l'une des solutions les plus abouties lorsque l'on souhaite faire du tatouage d'image multi-bits à partir d'une approche basée quantification. Dans ce chapitre nous poursuivons l'étude par le biais de deux algorithmes : l'approche Hyper-Cube et l'approche Multi-Hyper-Cube. Dans un premier temps, nous exposons le schéma Hyper-Cube et son positionnement par rapport à P-QIM. Nous améliorons ensuite l'approche Hyper-Cube en y intégrant un module TCQ (Quantification codée par treillis). Les résultats obtenus sont bons quel que soit le type d'attaque et permettent de conclure que le schéma de tatouage Multi-Hyper-Cube est actuellement l'une des techniques les plus abouties pour effectuer un tatouage robuste, à haute capacité et prenant en compte les aspects psychovisuels.

3.1 Introduction

Dans ce chapitre, nous nous intéressons à la famille basée quantification (DC-QIM, SCS, P-QIM [Chen et al. 01], [Eggers et al. 03], [Li et al. 07], ...). Les schémas basés quantification sont connus pour leur faible complexité calculatoire et leurs bonnes capacités d'insertion, mais aussi pour une grande sensibilité aux attaques valométriques de changement d'échelle. Cette grande sensibilité a cependant été réduite par l'ajout du principe RDM [PérezGonzález et al. 04].

L'un des schémas de tatouage d'images basé quantification les plus efficaces est actuellement le tatouage P-QIM (Perceptual-QIM) [Li et al. 07]. Cette affirmation repose sur la comparaison d'algorithmes non robustes aux désynchronisations dont le payload¹ est très élevé (1 bit inséré dans 64 pixels).

L'algorithme P-QIM utilise un tatouage par scalaire QIM [Chen et al. 01] avec un codage du message par répétition. Le principe RDM est intégré au schéma et permet de le rendre plus robuste aux attaques valométriques de changement d'échelle. L'originalité du schéma réside dans l'utilisation du modèle psycho-visuel modifié de Watson [Watson 93] pour assurer l'invariance

1. payload : quantité de bits insérée.

aux attaques valométriques de changement d'échelle, mais également pour prendre en compte l'impact psychovisuel.

Dans ce chapitre, nous proposons de poursuivre le travail autour de l'algorithme P-QIM en améliorant l'aspect psychovisuel. Nous proposons également d'intégrer un code correcteur et ainsi obtenir une cascade de décodage lors de l'extraction du message inséré. Ces améliorations sont regroupées au sein de l'algorithme Hyper-Cube qui est présenté en section 3.2. Nous proposons ensuite dans la section 3.3 de faire le pont entre l'approche basée treillis DPTC, et les approches basées quantification, en substituant le module QIM par un module basé TCQ (Quantification codée par treillis).

3.2 L'algorithme Hyper-Cube

Que cela soit pour l'algorithme P-QIM [Li et al. 07] ou Hyper-Cube [Chaumont et al. 11b], le schéma général est sensiblement le même. L'image est découpée en blocs 8×8 , et pour chaque bloc on insère un ou plusieurs bits. Le schéma de tatouage, à l'insertion, est résumé pour un bloc 8×8 sur la figure 3.1. La DCT d'un bloc \mathbf{X} est calculée, puis les n premiers coefficients ACs issus du parcours en zig-zag sont rangés dans un vecteur que nous appelons signal hôte et que nous notons $\mathbf{x} \in \mathbb{R}^n$. Les n coefficients de \mathbf{x} sont tatoués par scalaire QIM. On insère donc les n bits du vecteur $\mathbf{m} \in \{0, 1\}^n$ dans les n bits du vecteur \mathbf{x} . Le vecteur \mathbf{m} est composé de bits du *message* préalablement encodé par un code correcteur (voir section 3.2.2). Chaque coefficient $\mathbf{x}[i], i \in \{1, \dots, n\}$ embarque un bit $\mathbf{m}[i], i \in \{1, \dots, n\}$ et l'on obtient le signal tatoué $\mathbf{y} \in \mathbb{R}^n$ en quantifiant chaque dimension :

$$\forall i \in \{1, \dots, n\}, \mathbf{y}[i] = Q_{\mathbf{m}[i]}(\mathbf{x}[i], \Delta_i), \quad (3.1)$$

avec Δ_i le pas de quantification associé au $i^{\text{ème}}$ coefficient et Q une fonction de quantification. Le pas de quantification est obtenu à partir des *slacks* de Watson qui ont été calculées sur un bloc précédemment tatoué (voir section 3.2.1).

Pour un coefficient $\mathbf{x}[i]$, et un pas de quantification Δ_i , les quantificateurs Q_0 et Q_1 sont :

$$\begin{aligned} Q_0(\mathbf{x}[i], \Delta_i) &= 2\Delta_i \times \text{round} \left(\frac{\mathbf{x}[i]}{2\Delta_i} \right), \\ Q_1(\mathbf{x}[i], \Delta_i) &= 2\Delta_i \times \text{round} \left(\frac{\mathbf{x}[i] - \Delta_i}{2\Delta_i} \right) + \Delta_i. \end{aligned} \quad (3.2)$$

Il est à noter que l'approche P-QIM [Li et al. 07] ajoute une séquence de *dithering* dans l'équation 3.1. Le *dithering* consiste à bruitez chaque échantillon $\mathbf{x}[i]$ par une valeur, connue à l'insertion et à l'extraction, appartenant à $[-\Delta_i, +\Delta_i]$ et générée de manière pseudo-aléatoire. Dans l'approche Hyper-Cube [Chaumont et al. 11b] la sécurité du *message* est assurée grâce au brassage (entrelaçage) pseudo-aléatoire du résultat du codage du *message* (voir section 3.2.2 et la figure 3.1). Nous n'avons pas non plus intégré le principe d'*insertion informée* (voir chapitre 1), présent dans SCS [Eggers et al. 03] et DM-QIM [Chen et al. 01], qui consiste à *déplacer* le signal \mathbf{x} vers le point de quantification sans toutefois atteindre ce point. En effet, pour le niveau de puissance des attaques que nous considérons, le principe d'*insertion informée* n'améliorerait que très faiblement la robustesse.

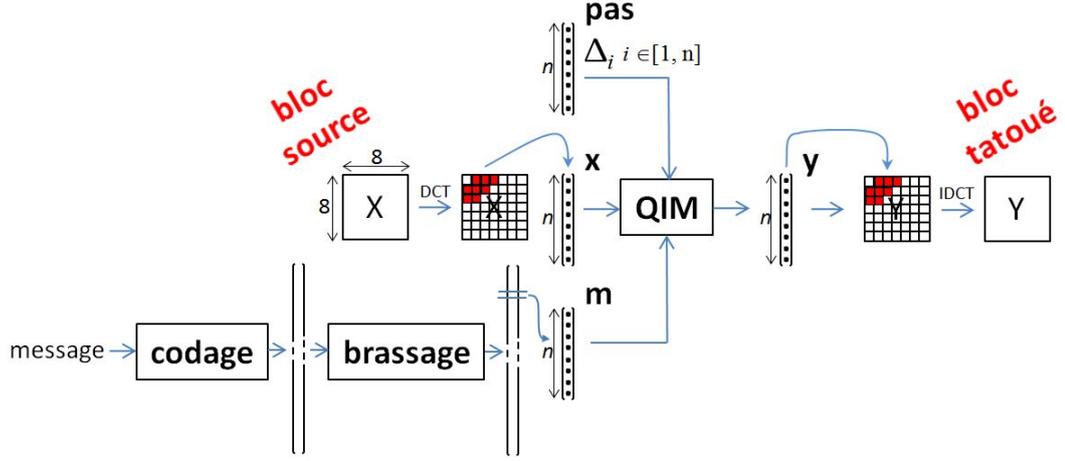


FIGURE 3.1 – Schéma général de P-QIM pour un bloc de 8×8 pixels.

3.2.1 Slack de Watson modifiée

Les *slacks* de Watson sont calculées pour chaque coefficient de chaque bloc DCT. La valeur d'une *slack* est un scalaire positif qui estime le niveau de distorsion que peut subir un coefficient sans être perceptible par le système visuel humain (plus exactement, pour un niveau de distorsion il doit y avoir un maximum de 50% d'expérimentateurs détectant la distorsion). Une faible valeur de *slack* signifie que le coefficient DCT considéré ne peut être que faiblement modifié, inversement, une grande valeur de *slack* signifie que le coefficient DCT peut être fortement modifié.

Pour un bloc DCT 8×8 donné, la *slack* de Watson **modifiée** associée au coefficient DCT $x \in \mathbb{R}$ de position $i \in \{0, \dots, 63\}$ est [Li et al. 07] :

$$s(x, i) = \max(t_L^M[i], |x|^{0.7} t_L^M[i]^{0.3}), \quad (3.3)$$

avec t_L^M le masquage de luminosité :

$$t_L^M[i] = t[i] \left(\frac{C[0]}{C_0} \right)^{0.649} \left(\frac{C_0}{128} \right), \quad (3.4)$$

avec $C[0]$ le coefficient DC du bloc DCT, C_0 la moyenne de tous les coefficients DCs de l'image, et $t[i]$ la valeur de sensibilité associée à la position i [Watson 93].

Par rapport à la *slack* de Watson [Watson 93], la fonction *slack* de Watson **modifiée** varie linéairement en échelle avec la variation en échelle du coefficient :

$$\forall x \in \mathbb{R}, i \in [0, 63], \forall \nu \in \mathbb{R}, s(\nu.x, i) = \nu.s(x, i). \quad (3.5)$$

Le pas de quantification Δ_i utilisé par les quantificateurs Q_0 et Q_1 pour effectuer le tatouage (voir équation 3.1) est :

$$\Delta_i = G_{HC} \times s(x, i),$$

avec $G_{HC} \in \mathbb{R}$ une constante permettant de régler la force d'insertion du schéma de tatouage. Une attaque valométrique multipliant l'amplitude des pixels d'un scalaire $\nu \in \mathbb{R}_+$ entrainera une

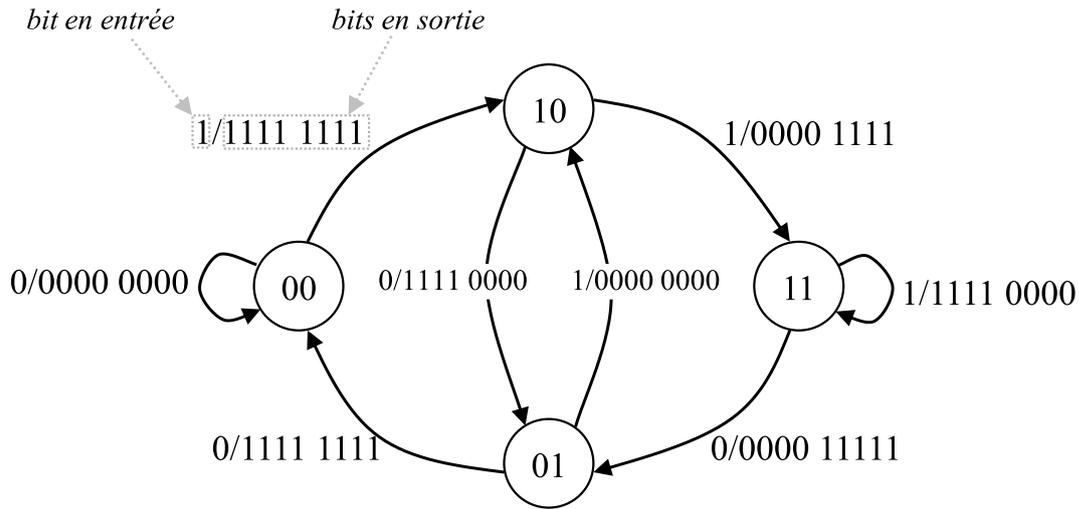


FIGURE 3.2 – Machine à état du codeur convolutif 1/8-taux 2-mémoires.

modification du pas de quantification d’un même facteur ν . Ainsi, le schéma de tatouage devient théoriquement invariant à l’attaque valométrique et prend en compte l’aspect psychovisuel. Dans le schéma P-QIM [Li et al. 07], la *slack* $s(x, i)$ est calculée sur le bloc gauche précédemment tatoué. Dans l’approche Hyper-Cube [Chaumont et al. 11b], nous calculons cette valeur à partir du bloc gauche ou du bloc supérieur (voir section 3.2.3).

3.2.2 Codage / décodage du message

À l’insertion, le *message* binaire est encodé par un code convolutif (dans l’algorithme P-QIM le *message* est implicitement encodé par une répétition des bits). Le taux du code convolutif est $1/n$ et le codeur est représenté par la « boîte » codage sur la figure 3.1. Le mot de code résultant du codage est alors mélangé (« boîte » brassage sur la figure 3.1)². Le vecteur binaire obtenu est alors découpé en petits vecteurs de taille n . Chaque vecteur de taille n est embarqué dans un bloc DCT 8×8 . Sur la figure 3.1, et pour la simplicité de l’explication, tous les petits vecteurs sont notés \mathbf{m} . Pour un bloc DCT, le signal de tatouage \mathbf{y} est obtenu en quantifiant chaque composante du signal \mathbf{x} avec les quantificateurs $\{Q_{\mathbf{m}[i]}\}, i \in \{1, \dots, n\}$ comme indiqué à l’équation 3.1.

L’utilisation de codes efficaces permet d’améliorer les performances du schéma de tatouage. Dans l’algorithme Hyper-Cube, nous insérons 1 bit dans $n = 8$ coefficients ACs. Nous utilisons un code convolutif 1/8-taux 2-mémoires. Le code 1/8-taux 2-mémoires est dérivé à partir d’un code convolutif 1/2-taux 2-mémoires en répétant quatre fois chaque bit de sortie. Le schéma de la machine à état du code 1/8-taux 2-mémoires est présenté à la figure 3.2.

À l’extraction, deux décodeurs sont mis en cascade pour décoder le message. Le premier décodeur est alimenté avec les vecteurs \mathbf{z} extraits de chaque bloc DCT tatoué-attaqué. Pour chaque bloc, il y a calcul de $2n$ distances euclidiennes : les distances $\mathbf{d}_0[i] = (\mathbf{z}[i] - Q_0(\mathbf{z}[i], \Delta_i))^2, i \in \{1, \dots, n\}$ calculées entre les coefficients $\mathbf{z}[i]$ et les mots correspondant à un bit 0 inséré, et les distances $\mathbf{d}_1[i] = (\mathbf{z}[i] - Q_1(\mathbf{z}[i], \Delta_i))^2, i \in \{1, \dots, n\}$ calculées entre les coefficients $\mathbf{z}[i]$ et les mots correspondant à un bit 1 inséré. Le second décodeur est un décodeur convolutif. Il prend les

2. Le mélange permet de distribuer pseudo-aléatoirement les bits sur toute l’image.

distances d_0 et d_1 calculées pour tous les blocs DCT, dé-entrelace les distances, puis les ajoute soigneusement afin d'étiqueter les arcs du treillis du code convolutif. Ensuite le décodeur calcule le chemin du treillis ayant la distance la plus faible. Pour cela nous utilisons l'algorithme de Viterbi [Viterbi 95].

3.2.3 Calcul des slacks sur un voisinage

La méthode proposée dans P-QIM utilise pour le tatouage du bloc courant les *slacks* du bloc précédemment tatoué (le bloc à gauche du bloc courant s'il y en a un et le bloc du dessus sinon). Ainsi, il n'y a aucune dérive entre les valeurs des *slacks* à l'insertion et les valeurs des *slacks* à l'extraction. En contrepartie, pour certaines images un effet de « trainée de blocs » peut apparaître aux endroits de fort contour. Les figures 3.3.a et 3.3.b représentent l'image 1 de la base BOWS-2 tatouée avec la technique P-QIM (seuls les 9 coefficients ACs d'un bloc sont utilisés et la distance SSIM³ [Wang et al. 04] de tatouage est fixée à 98%). On remarque la présence d'effets de blocs dans la zone de ciel à côté de l'église. Dans cette zone de l'image, les *slacks* utilisées pour la zone de ciel sont des *slacks* calculées sur la zone de l'église. Il en résulte une mauvaise quantification psychovisuelle et donc un aspect psychovisuel désagréable.

Pour supprimer ce défaut, il faut utiliser pour chaque bloc une *slack* plus appropriée tout en évitant une trop grande dérive entre la *slack* utilisée à l'insertion et celle utilisée à l'extraction. Pour ce faire, nous déterminons pour chaque bloc, le bloc voisin le plus proche au sens de la distance L2. Le calcul est effectué dans le domaine spatial en comparant chaque bloc voisin déjà tatoué avec le bloc courant dont on a supprimé les coefficients DCT utilisés pour le tatouage. Nous limitons le voisinage de recherche pour réduire la probabilité de dérive entre les valeurs des *slacks* à l'insertion et à l'extraction.

Après différentes expérimentations, il apparaît qu'un bon compromis consiste à ne considérer que 2 blocs : le bloc au dessus du bloc courant, noté bloc **B**, et le bloc à gauche du bloc courant, noté bloc **D**. Pour déterminer lequel du bloc **B** ou **D** sera utilisé pour calculer la *slack* qui servira au bloc courant, noté bloc **X** :

1. nous filtrons **X** pour obtenir $\mathbf{X}^{filtré}$. Le filtrage de **X** consiste à appliquer la DCT, mettre à zéro les n premiers coefficients ACs du parcours en zig-zag, et appliquer l'IDCT ;
2. nous calculons la distance Euclidienne entre **B** et $\mathbf{X}^{filtré}$, et également celle entre **D** et $\mathbf{X}^{filtré}$,
3. et nous retournons le bloc (**B** ou **D**) de plus petite distance.

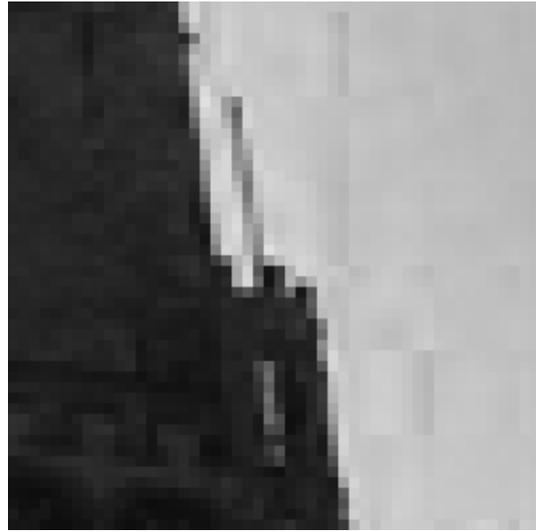
La figure 3.4 illustre la position de **B** et **D** par rapport à $\mathbf{X}^{filtré}$.

L'algorithme Hyper-Cube permet donc de mieux choisir le bloc sur lequel les *slacks* sont calculées. Nous pouvons observer à la figure 3.3(c) qu'avec ce mécanisme de choix du « meilleur bloc », l'effet de « trainée de blocs » est totalement supprimé. Remarquons que la distance SSIM n'est pas totalement adaptée pour mesurer la dégradation liée à ces systèmes de tatouage (i.e. P-QIM, Hyper-Cube et Multi-Hyper-Cube) puisque pour les figures 3.3(a) et 3.3(c), la distance psychovisuelle SSIM est la même (98%) et pourtant il ne fait pas de doute que l'image 3.3(c) est plus agréable visuellement. La distance de Watson pourrait être utilisée, mais le reproche majeur

3. Le SSIM est l'un des modèles de mesure de distorsion les plus corrélés au Système de Vision Humain. Les valeurs de SSIM sont comprises dans l'intervalle [0,1]. Plus la dégradation est grande et plus la valeur de SSIM est faible. Une valeur de SSIM de 1 signifie que l'image n'est pas dégradée. Pour calculer la valeur de SSIM nous utilisons l'implémentation C++ de Mehdi Rabah disponible à l'adresse <http://mehdi.rabah.free.fr/SSIM/>.



(a) Image 1 tatouée par P-QIM.



(b) Grossissement d'une partie de (a).



(c) Image 1 tatouée par Hyper-Cube.



(d) Grossissement d'une partie de (c).

FIGURE 3.3 – Image 1 de la base de donnée BOWS-2 tatouée à SSIM=98% et payload=1/64 ; (a) avec P-QIM ; (c) avec Hyper-Cube SSIM=98%.

que nous lui faisons est sa trop forte spécificité à mesurer des dégradations liées à la quantification JPEG. Notons que la distance de Watson, de la même façon que la distance SSIM, calcule une erreur moyenne qui ne prend pas assez en compte les fortes erreurs locales comme par exemple les « trainées de blocs ».

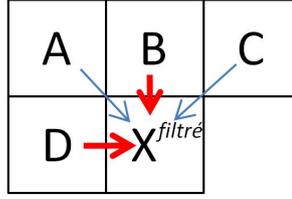


FIGURE 3.4 – Position des blocs **B** et **D** par rapport à $\mathbf{X}^{\text{filtré}}$.

3.3 Utilisation de la TCQ : l’algorithme Multi-Hyper-Cube

Dans [Chaumont et al. 11a], nous proposons d’utiliser un module TCQ (Quantification codée par treillis) en remplacement du module QIM (voir figure 3.1), pour améliorer les performances de l’approche Hyper-Cube. Cela permet de faire le pont entre l’approche basée treillis (DPTC) et les approches basées quantification (P-QIM [Li et al. 07]). Le Guelvoudit [Le Guelvoudit 09] a également proposé une approche basée TCQ, mais celle-ci n’est pas robuste aux attaques valométriques. Abrardo *et al.* [Abrardo et al. 06] ont proposé une approche utilisant la TCQ qui est robuste aux attaques valométriques, mais l’approche n’a pas été évaluée sur des images (mais sur des signaux gaussiens), et les aspects psychovisuels n’ont pas été pris en compte.

La TCQ (Quantification codée par treillis) est une technique de quantification utilisant un ensemble de quantificateurs organisés au sein d’une machine à états et agissant de la même manière qu’un codeur convolutif. La machine à états représente les transitions possibles étant donnée une séquence de coefficients en entrée. La machine à états peut être représentée dans son évolution temporelle par un treillis. Habituellement, un treillis est construit en plaçant tous les états en colonne. Chaque transition est représentée par un arc reliant l’état à l’instant t et l’état à l’instant $t + 1$. À chaque arc est associé un quantificateur. Dans le cas du tatouage par TCQ, un coefficient $\mathbf{x}[i] \in \mathbb{R}$ et un bit à insérer $\mathbf{m}[i] \in \{0, 1\}$ en entrée provoquent une transition vers un nouvel état et génèrent en sortie la quantification du coefficient $\mathbf{x}[i]$ qui n’est rien d’autre que le coefficient tatoué $\mathbf{y}[i]$. La figure 3.5 illustre le cas d’un treillis à 4 états.

De manière plus formelle, la fonction de transition f du treillis est telle que :

$$\begin{aligned} \mathcal{S} \times \{0, 1\} &\longrightarrow \mathcal{S} \\ f : (s, \mathbf{m}[i]) &\longmapsto s', \end{aligned}$$

avec $\mathcal{S} = \{0, 1, \dots, S - 1\}$ l’ensemble des états du treillis, $s \in \mathcal{S}$ l’origine de l’arc de transition, $s' \in \mathcal{S}$ l’extrémité de l’arc de transition, et $\mathbf{m}[i], i \in \{1, \dots, n\}$, le $i^{\text{ème}}$ bit de \mathbf{m} .

Chaque arc est étiqueté par un quantificateur :

$$\begin{aligned} \mathcal{S} \times \{0, 1\} \times \mathbb{R} \times \mathbb{R} &\longrightarrow \mathbb{R} \\ Q : (s, \mathbf{m}[i], \mathbf{x}[i], \Delta_i) &\longmapsto \mathbf{y}[i], \end{aligned}$$

avec Δ_i le pas de quantification. Pour simplifier, nous notons les quantificateurs $Q_{\mathbf{m}[i]}(s, \mathbf{x}[i], \Delta_i)$. Sur la figure 3.5, chaque arc est étiqueté par un unique quantificateur.

Les quantificateurs $Q_{\mathbf{m}[i]}(s, \mathbf{x}[i], \Delta_i)$ sont définis pour un état $s \in \mathcal{S}$, pour un coefficient $\mathbf{x}[i]$,

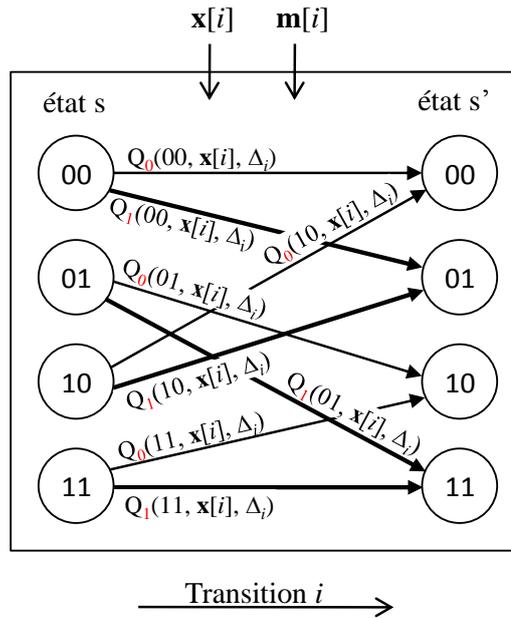


FIGURE 3.5 – La $i^{\text{ème}}$ transition dans un treillis à 4 états.

pour un pas de quantification Δ_i , et pour un bit $m[i]$ valant 0 ou 1 par :

$$\begin{aligned}
 Q_0(\mathbf{x}[i], s, \Delta_i) &= 2\Delta_i \times \text{round} \left(\frac{\mathbf{x}[i] - \delta}{2\Delta_i} \right) + \delta, \\
 Q_1(\mathbf{x}[i], s, \Delta_i) &= 2\Delta_i \times \text{round} \left(\frac{\mathbf{x}[i] - \Delta_i - \delta}{2\Delta_i} \right) + \Delta_i + \delta, \\
 \text{avec } \delta &= \frac{\Delta_i \times s}{S}.
 \end{aligned} \tag{3.6}$$

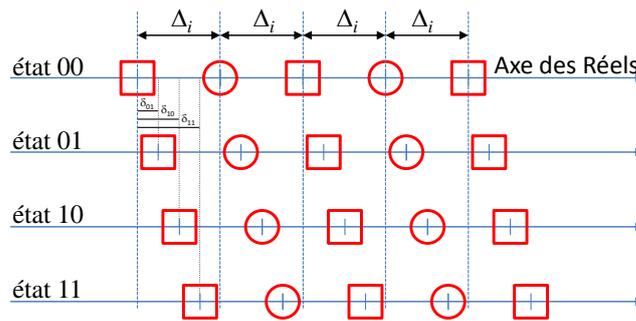


FIGURE 3.6 – Illustration d'un réseau (*lattice*) pour un treillis à quatre états. Les cercles rouges représentent les mots de code obtenus à l'aide des quantificateurs Q_0 (équation 3.6) et les carrés rouges représentent les mots de code obtenus à l'aide des quantificateurs Q_1 (équation 3.6).

La figure 3.6 montre la partition de l'axe des réels dans le cas d'un treillis à quatre états.

Les cercles rouges représentent les mots de code associés aux bits d'entrée de valeur 0 et les carrés rouges représentent les mots de code associés aux bits d'entrée de valeur 1. Pour un état donné $s \in \mathcal{S}$, la distance minimale entre les mots de code pour une transition 0 et les mots de code pour une transition 1 est égale à Δ_i . Nous pouvons également remarquer que les mots de code sont légèrement translatés entre états. Cette translation provient du terme δ de l'équation 3.6. Cette translation rend l'approche TCQ intéressante, car en fonction du chemin emprunté pour traverser le treillis, la quantification n'est pas la même. Parmi toutes les transitions possibles, on a alors plus de chance de trouver un mot de code très proche du coefficient hôte que lorsque nous n'avons qu'un seul quantificateur (cas du QIM). Pour une robustesse fixée, la distorsion obtenue en utilisant une approche par TCQ est alors inférieure à celle d'une approche par QIM. Le terme de translation est donc important pour que le système soit fonctionnel.

L'insertion dans un bloc DCT d'un petit vecteur \mathbf{m} est obtenue en élaguant pour chaque transition i les arcs ne correspondant pas au bit $\mathbf{m}[i]$ puis en déterminant le chemin minimisant l'erreur $\sum_{i=1}^{i=n} (\mathbf{y}[i] - \mathbf{x}[i])^2$. Le chemin minimisant l'erreur est obtenu en utilisant l'algorithme de Viterbi [Viterbi 95]. Pour le décodage, le treillis n'est pas élagué et l'algorithme de Viterbi est utilisé pour retourner l'estimation du vecteur binaire \mathbf{m} . L'ensemble des petits vecteurs binaires \mathbf{m} est concaténé, puis désentrelacé, et enfin transmis au deuxième décodeur qui retourne le *message*.

3.4 Résultats

Les expérimentations ont été réalisées sur les 100 premières images de la base de données de BOWS-2⁴ avec des images redimensionnées en 256×256 ⁵.

Quatre attaques à la robustesse ont été expérimentées : l'attaque par ajout de bruit Gaussien, l'attaque par filtrage, l'attaque valométrique de changement d'échelle, et l'attaque par compression JPEG. Le BER est calculé pour chaque attaque. Nous avons fixé la dégradation à une valeur SSIM de 98% [Wang et al. 04].

Les deux grandes familles de tatouage robuste multi-bits sont les codes en réseau (*lattice*) également connus sous le nom de codes basés quantification, et les codes à papiers sales (*Dirty Paper Trellis Codes*). Afin d'analyser la performance de notre approche, nommée **Multi-Hyper-Cube**, nous la mettons en compétition avec des algorithmes représentatifs des deux grandes familles. L'approche par code à papiers sales [Miller et al. 04] est représentée par le **PR-RB-DPTC** [Chaumont 09b]. Cet algorithme possède une faible complexité calculatoire ce qui nous permet de mener les expérimentations en un temps inférieur à une heure (voir chapitre 2 pour la présentation de PR-RB-DPTC). L'approche basée quantification est représentée par l'**Hyper-Cube** [Chaumont et al. 11b]. L'algorithme Hyper-Cube est une adaptation de l'algorithme P-QIM et fait partie des algorithmes les plus performants. Enfin, nous avons également intégré dans les tests l'approche par **Turbo-TCQ** [Le Guelvouit 09] qui fait le lien entre les deux familles (codes à papiers sales et codes en réseau) et utilise le principe turbo provenant des codes correcteurs.

Notons que ces quatre algorithmes sont testés sur des images réelles, et non sur des signaux gaussiens. En outre, ils ont une complexité faible qui est de l'ordre de $\mathcal{O}(\text{taille})$ avec *taille* la taille de l'image. Le temps de calcul est de l'ordre de quelques secondes pour une image CIF 360×288 sur un ordinateur portable à faible coût (CPU Intel Core 2, P8600, 2.4 GHz).

Les résultats de l'attaque valométrique par changement d'échelle sont donnés à la figure 3.7.

4. La base de données de BOWS-2 est téléchargeable à l'adresse <http://bows2.gipsa-lab.inpg.fr/>.

5. Le sous-échantillonnage d'images a été réalisé avec le programme `xnview` et utilisant l'interpolation de Lanczos.

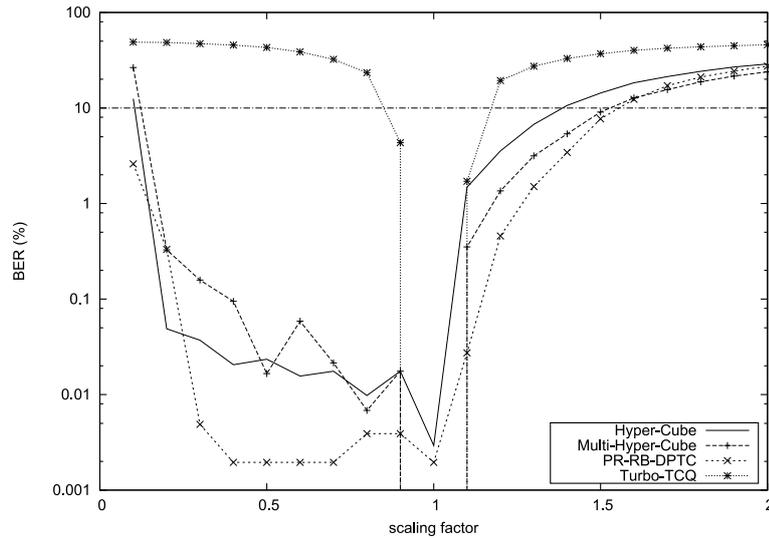


FIGURE 3.7 – BER pour une attaque valumétrique de changement d'échelle.

Pour toutes les autres attaques, la Turbo-TCQ surpasse les autres approches, mais pour l'attaque valumétrique, la Turbo-TCQ n'est pas du tout robuste. Le Guelvout avait observé ce phénomène dans [Le Guelvout 09] et c'est un comportement classique des approches basées quantification. Afin de supprimer cette sensibilité, nous pouvons par exemple utiliser le principe RDM (Rational Dither Modulation). Le principe RDM est intégré dans les approches Hyper-Cube et Multi-Hyper-Cube, et nous observons effectivement un meilleur comportement face à l'attaque valumétrique. Pour une modification valumétrique d'un facteur de 0.9, il y a en moyenne 0,018 bit erroné sur 100 bits transmis pour l'approche Multi-Hyper-Cube tandis qu'il y a 4,33 bits erronés sur 100 bits transmis pour l'approche Turbo-TCQ. Notons que les courbes de l'Hyper-Cube et du Multi-Hyper-Cube sont très proches avec un léger avantage pour le Multi-Hyper-Cube qui possède un BER nul quand il n'y a pas d'attaque. Enfin, notons que le PR-RB-DPTC [Chaumont 09b] possède la meilleure performance face à l'attaque valumétrique, en particulier pour une attaque valumétrique d'un facteur inférieur à 1. Ce très bon comportement était observé dans [Miller et al. 04].

Les résultats de l'attaque par compression JPEG sont donnés à la figure 3.8. Habituellement, les courbes de l'Hyper-Cube (et Multi-Hyper-Cube) et de PR-RB-DPTC sont souvent très proches, excepté pour l'attaque par compression JPEG où le PR-RB-DPTC n'est pas assez robuste. L'algorithme original DPTC [Miller et al. 04] est plus robuste, mais sa complexité calculatoire le rend peu pratique pour un payload de 1 bit inséré dans 64 pixels. En outre, les autres propositions d'améliorations comme celle de [Lin et al. 05] ne sont pas efficaces dans la pratique (voir [Chaumont 10a]). Cela montre que dans la pratique, la technique Hyper-Cube (et Multi-Hyper-Cube) est plus intéressante que l'approche PR-RB-DPTC lorsque le payload est élevé.

Les résultats sont donnés à la figure 3.9 pour l'attaque par ajout de bruit gaussien, et à la figure 3.10 pour l'attaque par filtrage. L'approche par Turbo-TCQ a de très bonnes performances, les autres approches ont des performances moindres. L'approche Multi-Hyper-Cube est intéressante à faible puissance d'attaque puisque le BER est nul.

Pour résumer, les deux approches qui possèdent de bonnes performances quelles que soient les attaques sont l'Hyper-Cube et le Multi-Hyper-Cube. Le Multi-Hyper-Cube améliore significativement l'approche Hyper-Cube quand il y a une attaque de puissance très faible. En effet,

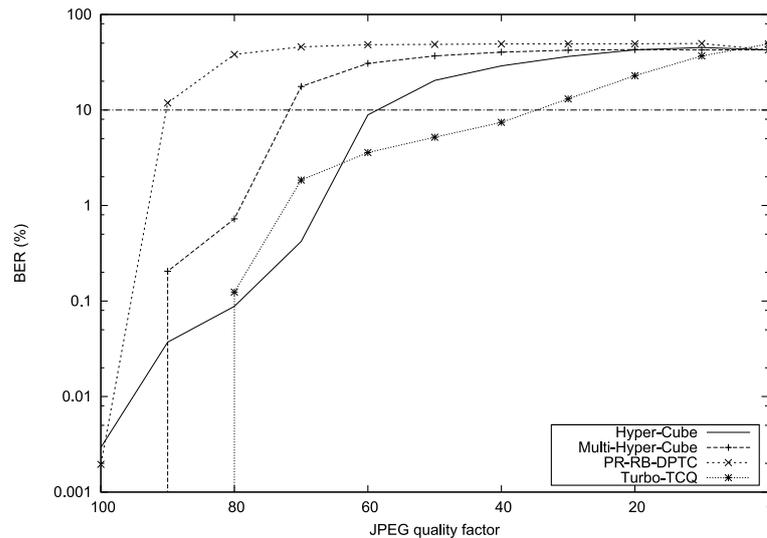


FIGURE 3.8 – BER pour une attaque par compression JPEG.

pour l'ensemble des 100 images, tous les bits ont été récupérés pour des attaques de faible puissance. Nous ne pouvons cependant pas conclure que l'algorithme Multi-Hyper-Cube surpasse l'algorithme Hyper-Cube. En effet, lorsque les puissances d'attaque sont plus fortes, le BER du Multi-Hyper-Cube n'est pas toujours inférieur au BER de l'Hyper-Cube. L'utilisation d'un module TCQ permet d'ajouter plus de quantificateurs, mais en contrepartie, cela ajoute sans doute plus d'instabilité de décodage lorsqu'il y a une attaque de puissance moyenne.

Le Multi-Hyper-Cube a des comportements qui sont similaires à ceux d'un code correcteur. Lorsque la puissance d'attaque est trop forte, la probabilité d'erreur est rapidement supérieure à 0,1 bit erroné sur 100 bits transmis. Cette croissance rapide du BER est encore plus visible avec l'approche par Turbo-TCQ pour les attaques de filtrage et valométriques. Le BER augmente soudainement à des valeurs supérieures à 1 bit erroné sur 100 bits transmis. C'est un comportement classique avec des approches utilisant des codes correcteurs quasi-optimaux. En conclusion, l'approche par Multi-Hyper-Cube donne un BER nul pour des attaques de faible puissance, mais n'est pas meilleur que l'Hyper-Cube pour les attaques de puissance moyenne.

3.5 Conclusion

Dans ce chapitre nous avons présenté l'algorithme basé quantification appelé Hyper-Cube. Nous avons également évalué le remplacement du module de tatouage QIM par un module TCQ. La TCQ permet d'augmenter le nombre de quantificateurs. Cet ajout permet d'améliorer sensiblement la robustesse. Ce nouvel algorithme, appelé Multi-Hyper-Cube est comparé à trois algorithmes représentatifs de l'état de l'art dont le payload est de 1 bit pour 64 pixels et qui prennent en compte l'aspect psychovisuel. Les résultats montrent que le schéma résiste correctement aux attaques bruit gaussien, filtrage, compression JPEG, et valométriques. Ce comportement constant quel que soit l'attaque n'est pas observé pour les autres algorithmes. L'algorithme PR-RB-DPTC est sensible à la compression JPEG, et l'algorithme Turbo-TCQ est sensible à l'attaque valométrique. Notons finalement que pour les attaques de faible puissance, le BER de l'approche Multi-

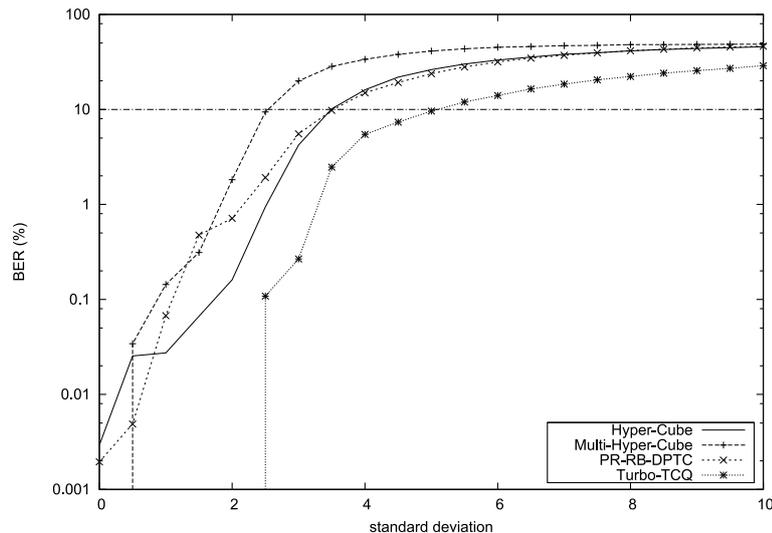


FIGURE 3.9 – BER pour une attaque par ajout de bruit gaussien.

Hyper-Cube est nul.

De nombreuses améliorations peuvent encore être apportées au schéma Multi-Hyper-Cube. Les coefficients DCT utilisés pour le tatouage ont été choisis empiriquement. Il faudrait déterminer l'ensemble de coefficients permettant d'obtenir la plus grande robustesse. Cette étude devrait être menée conjointement avec la conception du code correcteur. Le choix d'un code correcteur à faible densité de parité de contrôle (LDPC) ou un turbo-code permettrait probablement d'atteindre des performances s'approchant de celles de l'algorithme Turbo-TCQ. Une gestion habile des deux codeurs / décodeurs pourrait également augmenter les performances globales. Nous pensons également que l'intégration d'un QIM vectoriel [Bardyn et al. 09], l'ajout d'étalement, et/ou l'utilisation d'une *insertion informée* pourrait améliorer l'approche. On peut aussi envisager de rendre le schéma robuste à des attaques supplémentaires. Les auteurs de [Zhu et al. 08] proposent par exemple de cumuler robustesse au changement d'échelle et robustesse au changement constant de luminosité. La robustesse globale est ainsi améliorée. Enfin, les *slacks* de Watson ne sont pas totalement satisfaisantes car leur robustesse aux attaques n'est pas toujours assurée. Il y a donc un défi à relever concernant la définition de modèles psychovisuels robustes ou peu sensibles aux attaques.

Ce chapitre vient clore la partie sur le tatouage robuste d'images. Les évaluations de la robustesse pour la famille de tatouage robuste par treillis (chapitre 2) et celle par quantification (chapitre 3) ont été effectuées en utilisant quatre attaques classiques [Miller et al. 04] sur la base d'images BOWS-2 [Bas et al. 08b]. Nous nous sommes attachés à utiliser des algorithmes de faible complexité représentant bien l'état de l'art. Une telle comparaison systématique n'avait pas été établie pour des images avec en plus des contraintes psychovisuelles, et un fort payload.

Les résultats obtenus donnent une vision claire, dans des conditions réalistes d'utilisation, de l'état de l'art des schémas de tatouage à haut débit et faible complexité. Ces expérimentations montrent que, tant pour les schémas par quantification que pour ceux par treillis, il existe des approches efficaces (faible complexité, fort payload = 1/64 bpp, faible dégradation = SSIM de 98%, prenant en compte l'aspect psychovisuel) par exemple PR-RB-DPTC ou Multi-Hyper-Cube. Les expérimentations montrent également que l'utilisation de codes correcteurs utilisant le principe

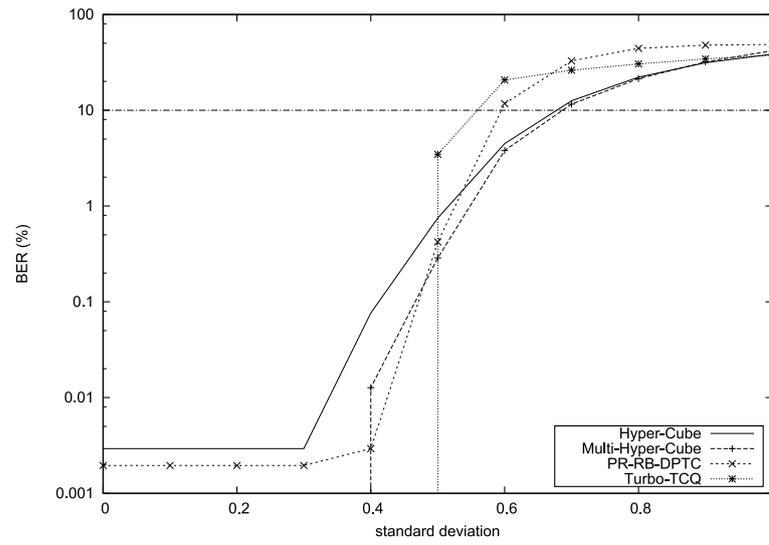


FIGURE 3.10 – BER pour une attaque par filtrage gaussien.

turbo permet d'améliorer la robustesse pour une distorsion donnée ; l'ajout du principe turbo aux deux grandes familles de schémas permettrait de gagner en robustesse, à l'instar de la Turbo-TCQ.

Deuxième partie

Le tatouage conjointement à la compression

Chapitre 4

Le tatouage conjoint à la compression

Résumé

Ce chapitre traite du tatouage conjoint à la compression et rappelle brièvement quelques notions liées à la compression images JPEG2000 et la compression vidéos H.264. Comme nous l'avons déjà plusieurs fois évoqué, la plupart des images, des vidéos, etc, sont stockées et échangées sous une forme compressée et le plus souvent la compression se fait avec perte. Si le tatouage est effectué avant l'étape de compression, alors le signal de tatouage subit une première dégradation (attaque) alors que le média (image, son, vidéo...) n'a pas encore été distribué. Il apparaît donc naturel de faire conjointement la compression et le tatouage. Dans les deux chapitres suivants, nous présenterons une approche de tatouage conjointe à une compression JPEG2000, et une approche de tatouage conjointement à une compression H.264. Ce chapitre rappelle les grandes étapes de la compression JPEG2000 et H.264 et donne les grandes classes d'approche de tatouage pour ces deux formats.

4.1 Brève présentation de JPEG2000

JPEG2000 [JPE00] est un standard de compression développé par le Joint Photographic Experts Group qui prend en charge la compression avec perte ou sans perte des images en niveaux de gris ou en couleur. Initialement, ce standard de compression devait remplacer le standard JPEG [JPE91] mais la complexité calculatoire des codeurs JPEG2000 et la réticence de l'industrie à adopter un nouveau standard ont fait qu'il a tardé à être adopté. Son essor récent vient principalement de l'industrie du cinéma numérique haute définition.

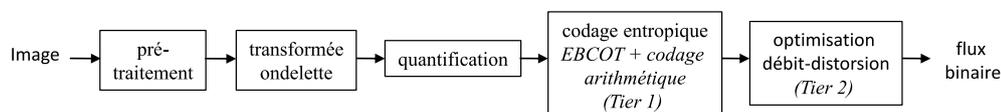


FIGURE 4.1 – Schéma reprenant les principales étapes d'un codeur JPEG2000.

Les principales étapes du codage JPEG2000 avec pertes sont données sur la figure 4.1. JPEG-2000 laisse la possibilité de changer l'espace couleur de l'image et d'adapter les données d'entrée (étape de pré-traitement). La première grande étape consiste à appliquer une transformée ondelette

avec ou sans perte. Lorsque l'on utilise JPEG2000 dans sa version avec pertes, on a tendance à utiliser la transformée en ondelette 9/7 de Daubechies [Daubechies et al. 98]. Notons que la norme JPEG2000 laisse alors la possibilité de découper l'image en zones rectangulaires (les tuiles) qui seront traitées séparément. Cette option est intéressante dans le cas d'images de grandes dimensions qui nécessitent beaucoup de ressources mémoire et processeur. La deuxième grande étape consiste à quantifier les valeurs des coefficients ondelette. Le standard propose d'utiliser un quantificateur scalaire uniforme à zone morte. Dans la partie 2 du standard (partie extension) [JPE04] il est possible d'utiliser une Quantification Codée par Treillis (Treillis Coded Quantization : TCQ). Nous reviendrons au chapitre 5 sur la TCQ puisque c'est cette technique de quantification que nous utiliserons pour réaliser le tatouage conjointement à la compression. La troisième partie consiste à effectuer le codage entropique en appliquant le codage par plans de bits EBCOT (Embedded Block Coding with Optimal Truncation) [Taubman 00] suivi d'un codage arithmétique. La dernière étape consiste à regrouper les données en paquet en utilisant une politique d'optimisation débit-distorsion. La construction des paquets permet de hiérarchiser l'information en résolution, en débit ou en qualité. Cette dernière étape est appelée la phase d'allocation de débit et certaines informations (passes de codage) sont supprimées pour augmenter le taux de compression. Cette étape introduit une dégradation supplémentaire des données, en plus des dégradations survenant lors de l'étape de quantification. Cette dégradation est à prendre en compte lors de la conception d'un schéma de tatouage joint. Une fois que les paquets sont obtenus, on les assemble pour former le flux JPEG2000.

4.2 Tatouage dans JPEG2000

Pour faciliter la classification des différentes approches de tatouage, on peut les différencier en fonction du lieu où le tatouage est effectué lors de la compression : avant la compression, pendant ou après la quantification, et lors du codage entropique.

Avant la compression, nous retrouvons les approches images comme l'étalement de spectre [Cox et al. 97], le tatouage par DPTC [Miller et al. 04], le tatouage PQIM [Li et al. 07] (voir la partie I du manuscrit). Ces approches peuvent d'ailleurs être appliquées après l'étape de transformation ondelette de JPEG2000. On peut par exemple citer l'approche Ouled Zaid *et al.* qui consiste à effectuer un tatouage des coefficients ondelette en utilisant une approche de type QIM [OuledZaid et al. 09].

La plupart des algorithmes de tatouage réalisés pendant ou après la phase de quantification utilisent une approche par quantification [Meerwald 01], [Li et al. 03] de type QIM. Meerwald effectue l'insertion par QIM avec *dithering*, après la phase de quantification et de région d'intérêt. Un même bit est inséré par *code-block*. Li et Zahang [Li et al. 03] insèrent également après l'étape de quantification. Pour faire survivre le signal de tatouage à l'étape d'allocation de débit, le pas de quantification est déterminé à partir du débit cible fixé par l'utilisateur.

On peut également réaliser une insertion de données cachées pendant ou après le codage entropique. Dans ce cas, l'insertion n'est pas robuste. Chen *et al.* [Chen et al. 10] ont proposé de réaliser l'opération de dissimulation de données directement au niveau du flux binaire JPEG2000, après la phase d'allocation de débit, en simulant une nouvelle phase d'allocation de débit. Le nouveau débit binaire doit être plus petit que le débit cible. Cette nouvelle phase induit une réorganisation des couches de qualité de manière à libérer de l'espace. Cet espace est ensuite utilisé pour dissimuler des données. Su *et al.* [Su et al. 03] insèrent l'information cachée dans le flux

binaires JPEG2000 en exploitant l'option *lazy mode* de JPEG2000. L'opération de dissimulation de données est également effectuée après la phase d'allocation de débit en modifiant les bits se trouvant au niveau des passes de codage de raffinement. Le principal inconvénient de cette méthode est qu'elle ne peut s'effectuer que sous le mode *lazy*. De plus, il est nécessaire que le débit cible soit supérieur à 2 bpp afin de permettre l'activation de ce mode.

4.3 Brève présentation de H.264

H.264 ou MPEG-4 Part 10 [H2603] est le codeur de l'état-de-l'art pour le codage vidéo¹. La première version du standard a été approuvée en 2003. La norme provient de deux organisations : l'ITU-T et l'ISO/IEC. Le codage H.264 permet d'obtenir jusqu'à 50% de gain en débit par rapport à un codage MPEG2 ou MPEG4-Part 2 simple profile [Richardson 10]. Nous ne rentrerons pas dans les détails des différents profils de H.264 ni dans les améliorations qui ont été proposées par rapport à MPEG2 et MPEG4 Part 2. Ce qu'il faut retenir c'est que depuis la normalisation de MPEG1 en 1988, le mécanisme de codage repose sur la décomposition de l'image en blocs, puis par l'application de prédictions (spatiales ou temporelles), pour finir par la quantification et le codage des coefficients. On parle d'ailleurs de codeur basé blocs. Pour simplifier l'explication, nous donnons ici un schéma extrêmement simplifié reprenant les étapes importantes du codage vidéo sur la Figure 4.2.

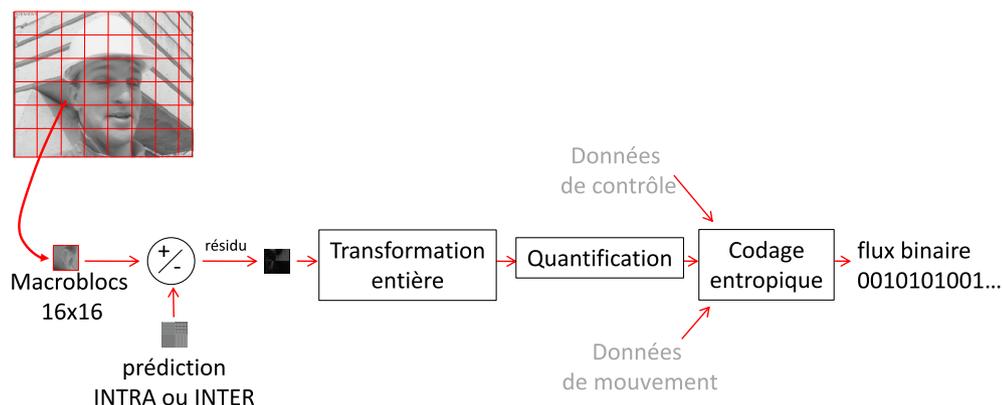


FIGURE 4.2 – Schéma reprenant les grandes étapes d'un codeur basé bloc.

L'image courante à coder est divisée en macroblocs de taille 16×16 . Chaque macrobloc est composé d'un nombre variable de blocs qui vont être prédits temporellement ou spatialement selon que la « frame » est une « frame » Intra ou une « frame » Inter. La figure 4.3 donne l'exemple de différentes prédictions Intra dans le cas où les blocs sont de taille 4×4 . Le bloc de taille 4×4 est prédit en utilisant l'une des méthodes de prédiction et l'on calcule alors la différence du bloc courant par le bloc de prédiction. On obtient donc un bloc composé de résidus et celui-ci va alors

1. Les groupes d'experts de l'ITU-T (Video Coding Experts Group : VCEG) et de l'ISO/IEC (Moving Picture Experts Group : MPEG) ont lancé un appel à proposition en janvier 2010 pour définir le futur standard de compression. Certaines des propositions ont permis, dans certains cas, de réduire le débit par deux par rapport à H.264 High Profile à même qualité visuelle. L'ITU-T et l'ISO/IEC ont donc lancé la standardisation du nouveau codeur « High Efficiency Video Coding : HEVC ». La norme de ce nouveau standard est prévue pour juillet 2012. Il est fort probable que certaines propositions testées sur le logiciel KTA [KTA11] soient reprises dans ce nouveau standard.

subir une transformation fréquentielle (la transformation entière). Le bloc obtenu est alors quantifié puis les coefficients sont codés entropiquement par le codeur CAVLC ou CABAC.

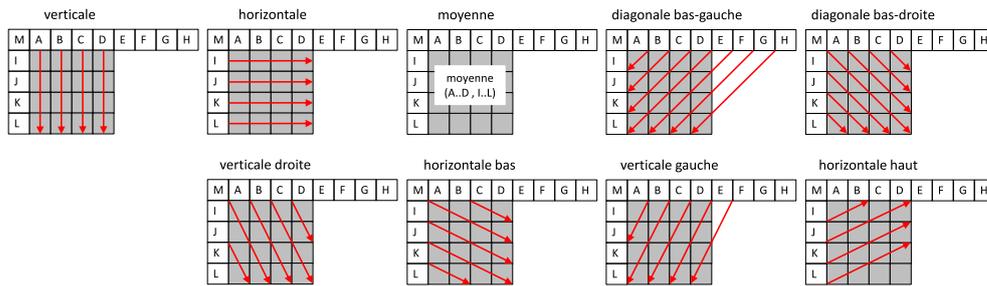


FIGURE 4.3 – Les prédictions utilisées pour un bloc 4×4 dans H.264/AVC.

Dans le cas d'une prédiction temporelle un bloc résidu est également calculé. Le processus de transformation, quantification, et codage est alors sensiblement le même. Lorsqu'il y a prédiction temporelle (cas des « frames » Inter) la prédiction consiste à déterminer dans une « frame » déjà codée (précédente ou suivante temporellement) le bloc le plus proche en distance absolue ou L2. La position de ce bloc le plus proche est alors représentée par un vecteur de déplacement qui est relatif au bloc courant que l'on souhaite prédire. Ce vecteur déplacement (vecteur de mouvement) devra lui aussi être codé.

Le diagramme 4.4 donne le diagramme complet d'un codeur et d'un décodeur H.264. Un pixel $f(i, j)$ provenant d'un bloc est prédit par un pixel précédemment codé-décodé (provenant d'un bloc déjà codé-décodé). On obtient donc un coefficient de résidu $e(i, j)$. Le coefficient de résidu est alors transformé via la transformation entière. On obtient le coefficient $g(u, v)$. Le coefficient $g(u, v)$ est alors quantifié, puis codé par le codeur entropique et enfin décodé (boucle de décodage). On peut constater que le codeur a besoin de décodé le pixel pour que celui-ci puisse servir de prédicteur pour des pixels non encore codés. Pour ce que nous avons besoin de faire par la suite, il n'est pas utile de rentrer plus dans la description du codec H.264. Nous allons maintenant donner quelques références sur le tatouage au sein d'une vidéo H.264.

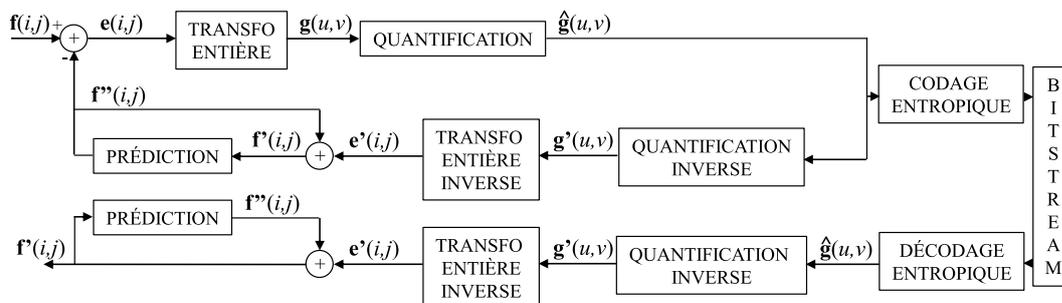


FIGURE 4.4 – Diagramme détaillé des étapes d'un codeur H.264.

4.4 Tatouage dans un flux vidéo

Pour faciliter la classification des différentes approches, on peut les différencier en fonction du lieu où le tatouage est effectué lors de la compression de la vidéo : avant la compression, avant la

quantification, après la quantification et lors du codage entropique.

Avant la compression nous retrouvons les approches images comme l'étalement de spectre [Cox et al. 97], le tatouage par DPTC [Miller et al. 04], le tatouage PQIM [Li et al. 07] (voir la partie I du manuscrit). Il existe également des approches qui prennent en compte la dimension temporelle comme le tatouage dans l'histogramme de luminances moyenne des « frames » [Haitsma et al. 01], [Chen et al. 09] ou bien le tatouage sur un volume fréquentiel comme par exemple sur une DFT 3D [Deguillaume et al. 99].

Avant la phase de quantification il est possible de modifier les coefficients DCT de luminances en prenant soin d'avoir un paramètre de puissance d'insertion calculé en fonction du paramètre de quantification choisi par l'utilisateur pour effectuer la compression. On peut citer par exemple la très intelligente proposition de Golikeri *et al.* [Golikeri et al. 07]. Les auteurs de [Golikeri et al. 07] proposent d'insérer 1 bit par macro-bloc, d'utiliser un masque psycho-visuel et d'utiliser une approche mixant étalement et quantification (la technique de tatouage est le ST-SCS). Cette approche est d'ailleurs l'une des approches intégrées à un flux vidéo (H.264) parmi les plus robustes. On peut également insérer le signal dans les vecteurs mouvement [Zhang et al. 01]. Enfin, Linnartz et Talstra [Linnartz et al. 98] ont proposé une méthode consistant à modifier la structure de GOP (Group Of Picture) en variant le nombre de « frames » I-P-B au sein d'un GOP. Cette dernière méthode n'est pas robuste à un ré-encodage.

Après la phase de quantification, on peut distinguer deux problèmes différents. Soit l'insertion est effectuée sur un flux en cours de codage, soit l'insertion est effectuée sur un flux déjà encodé. Pour une insertion dans un flux déjà encodé, Noorkami et Mersereau [Noorkami et al. 08] ont proposé une très belle approche : robuste, 0-bits, utilisant un masque psychovisuel, avec une insertion dans les coefficients ACs et une détection pouvant être effectuée sans connaissance des positions des coefficients tatoués. Nous avons également proposé une méthode qui a lieu après la phase de quantification [Shahid et al. 10b] et prend en compte l'aspect performance de compression en utilisant l'optimisation débit-distorsion. Nous reviendrons sur le principe de tatouage utilisé dans cette approche dans le chapitre 6. L'insertion peut également être effectuée sur un flux déjà compressé. La difficulté vient du fait que l'on a peu de marge pour faire des modifications. On peut citer par exemple [Hartung et al. 98] et [Gong et al. 08]. Ces techniques sont intéressantes lorsque que l'on souhaite intégrer un signal de tatouage sans avoir à décoder, tatouer, puis ré-encoder la vidéo.

Enfin, il est possible de faire le tatouage durant la phase de codage entropique. Ce genre d'approche est difficile à mettre en place, car il faut rentrer dans la structure très complexe de CABAC ou CAVLC. Mobasseri et Raikar [Mobasseri et al. 07], Zou et Bloom [Zou et al. 09] entre autres ont proposé ce genre d'approche. Le payload est évidemment très faible et ces schémas ne sont pas robustes. En contrepartie, le débit du fichier obtenu n'est pas modifié. Il faut également noter que certaines approches ont tendance à générer une dérive (drift) entre ce qui a été codé et ce qui sera décodé. Cela implique alors une dégradation significative de la qualité du signal vidéo reconstruit.

De ce que nous venons de présenter, que cela soit pour le tatouage dans JPEG2000 ou dans H.264, les mécanismes les plus robustes sont ceux localisés juste avant ou après la phase de quantification. Parmi toutes les solutions proposées, la plupart « luttent » contre le module de quantification et le module d'allocation de débit. Dans les deux chapitres suivants, nous proposons d'effectuer le tatouage conjointement avec le module de quantification. L'objectif de cette collaboration entre la quantification et le tatouage est d'obtenir de meilleures performances en robustesse (pour le module de tatouage) et également en débit-distorsion (pour le module de compression). Le cha-

pitre 5 propose un mécanisme de tatouage au sein de JPEG2000 en effectuant conjointement la quantification et le tatouage. Le chapitre 6 traite de l'intégration d'un mécanisme de tatouage au sein de H.264 en intégrant l'impact du signal de tatouage dans l'optimisation débit-distorsion.

Chapitre 5

Une approche de tatouage conjointe à la compression JPEG2000

Résumé

Dans ce chapitre nous présentons un schéma conjoint de quantification et de tatouage au sein de JPEG2000. Cette approche a été publiée à EUSIPCO 2011 [Goudia et al. 11a] et est également détaillée dans le manuscrit de thèse de Dalila Goudia [Goudia 11]. Les schémas de tatouage au sein de JPEG 2000 (voir chapitre 4) ajoutent tous un module supplémentaire au codeur. Notre schéma, via l'utilisation de la TCQ (quantification codée par treillis), permet simultanément de tatouer et de quantifier. De plus, l'insertion du message réduit faiblement la qualité, le taux de compression varie peu, et le schéma est robuste à la phase de contrôle de taux.

5.1 Introduction

En effectuant simultanément l'étape de tatouage et l'étape de quantification, nous réglons à la fois la robustesse du signal de tatouage vis-à-vis de la quantification et la distorsion additionnelle due à l'insertion du signal de tatouage. L'approche que nous proposons consiste à modifier légèrement l'étape de quantification basée TCQ de JPEG2000 partie 2 [JPE04] pour réaliser l'insertion. En section 5.2 nous présentons la quantification TCQ dans JPEG2000 partie 2, en section 5.3 nous détaillons le schéma, en section 5.4 nous analysons les résultats expérimentaux, puis nous concluons.

5.2 La Quantification Codée par Treillis (TCQ)

La Quantification Codée par Treillis (TCQ) est une technique de quantification rapide proposée par Marcellin et Fisher [Marcellin et al. 90] basée sur l'idée de partitionnement d'ensembles de Ungerboeck [Ungerboeck. 82] pour combiner la modulation et le codage canal. La TCQ consiste à partitionner un dictionnaire de quantification en sous-dictionnaires complémentaires puis à attribuer ces sous-dictionnaires aux transitions d'un treillis. La TCQ de JPEG2000 utilise quatre sous-dictionnaires qui sont notés D_0 , D_1 , D_2 et D_3 . Chaque arc du treillis est donc étiqueté par un de ces sous-dictionnaires.

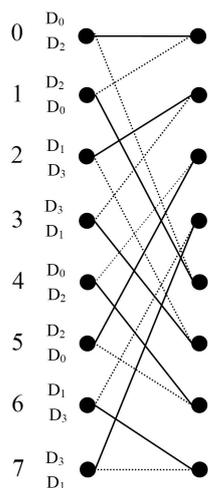


FIGURE 5.1 – La structure du treillis utilisé dans JPEG2000.

À partir d'un état initial, le chemin dans le treillis peut être décrit par une séquence binaire puisqu'il n'y a que deux transitions possibles d'un état à un autre (voir figure 5.1). Afin de quantifier la séquence source \mathbf{x} , l'algorithme de Viterbi [Viterbi 95] est utilisé pour trouver le chemin de coût minimal à travers le treillis. Le chemin de coût minimal correspond au chemin générant le vecteur quantifié $\hat{\mathbf{x}}$ le plus proche en distance L2 du vecteur source \mathbf{x} . L'algorithme de Viterbi produit donc une séquence de bits indiquant le chemin optimal, et une séquence d'indices résultant de la quantification.

À la réception, le décodeur reconstruit la source de la manière suivante. La séquence de bits nous informe sur le chemin à suivre dans le treillis ainsi que sur la suite de sous-dictionnaires, et la séquence d'indices permet, connaissant les sous-dictionnaires, d'appliquer la quantification inverse. À chaque transition i , le bit de chemin permet de retrouver l'état suivant et donc le sous-dictionnaire de quantification utilisé. L'indice permet de reconstruire l'échantillon source qui sera retourné en sortie du codeur.

Plusieurs techniques de quantification sont proposées dans la partie 2 du standard JPEG2000. Parmi celles-ci, nous pouvons citer l'ECTCQ (Entropy Coded TCQ) [Su et al. 03], [JPE00] qui est l'une des variantes de la TCQ. Le treillis employé est un treillis à huit états où chaque état possède deux arcs comme illustré sur la figure 5.1. Les quantificateurs ou sous-dictionnaires associés à chaque état sont combinés pour former les quantificateurs d'union : $A_0 = D_0 \cup D_2$, $A_1 = D_1 \cup D_3$. Les deux quantificateurs d'union A_0 et A_1 sont illustrés sur la figure 5.2. Les valeurs de reconstruction $\hat{\mathbf{x}}$ sont données pour chaque quantificateur d'union. Les indices $q(A_j)$, $j \in \{0, 1\}$, correspondant à ces valeurs de reconstruction sont également donnés. Chaque état dans le treillis est associé à l'un des deux quantificateurs d'union.

L'algorithme de Viterbi génère une séquence d'indices, et un chemin représenté par une séquence binaire. En pratique, il n'est pas nécessaire de transmettre le chemin puisqu'il est implicitement contenu dans la séquence d'indices. Ainsi, dans JPEG2000, seuls les indices sont transmis et le bit de poids faible de chaque indice correspond au bit de chemin qui est également un indicateur du sous-dictionnaire utilisé (D_0 ou D_2 , et D_1 ou D_3). Ce détail pratique se révèle extrêmement important. En effet, lors de la phase de contrôle de taux, certains plans de bits de poids faibles

des indices peuvent être tronqués. Il y a donc destruction de l'information de cheminement. Cela peut avoir un impact catastrophique sur un schéma de tatouage basé TCQ. Nous verrons en section 5.3.3 comment nous contournons simplement ce problème.

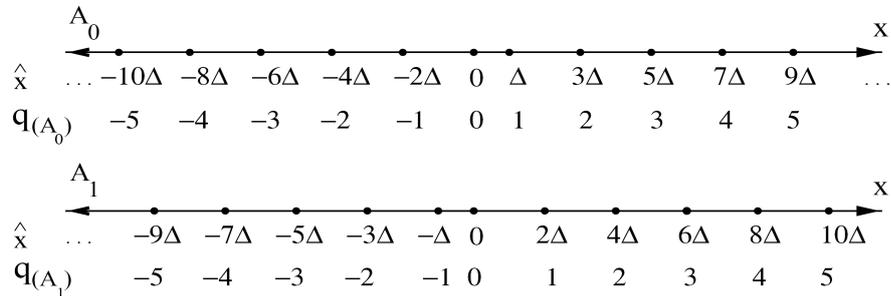


FIGURE 5.2 – Quantificateurs d'union définis dans JPEG2000.

5.3 Le schéma conjoint proposé

Lors de la compression JPEG2000, chaque sous-bande contenant les coefficients d'ondelettes quantifiés est découpée en entités rectangulaires appelées *code-blocks*. Chacun de ces *code-blocks* est alors codé indépendamment des autres au moyen d'un algorithme appelé EBCOT (Embedded Block Coding with Optimal Truncation). Chaque *code-block* est codé plan de bits par plan de bits, en commençant par les bits de poids les plus forts. Tous ces plans de bits sont codés par trois passes de codage successives. Durant la phase de contrôle taux, également appelée allocation de débit, on associe au flux binaire obtenu pour chacune des passes de codage, un point de troncature. L'algorithme de contrôle de taux se charge de trouver, dans le flux binaire correspondant à chaque *code-block*, le point de troncature optimal permettant d'atteindre le débit spécifié dans les paramètres de codage. Les passes de codage sont ensuite regroupées en paquets qui sont ajoutés au flux de données final ou *code-stream*. Nous devons donc prendre en compte cette troncature du flux binaire afin qu'il n'y ait pas de dégradation du message caché.

5.3.1 La méthode de dissimulation de données proposée

Notre stratégie de dissimulation de données s'appuie sur les principes de la QIM (Quantization Index Modulation) [Chen et al. 01] et sur l'utilisation d'un treillis. C'est une technique basée quantification utilisant la quantification TCQ pour insérer des données cachées. Le principe général est le suivant : nous disposons d'une séquence de quantificateurs. Le choix des quantificateurs est déterminé en fonction des données à dissimuler. Les échantillons du signal hôte sont alors quantifiés à l'aide des quantificateurs sélectionnés.

Il y a deux arcs sortant par état. Sur la figure 5.3, pour chaque état, l'arc en gras est étiqueté par D_0 ou D_1 et l'autre arc est étiqueté par D_2 ou D_3 . Pour insérer le bit 0 nous utilisons le quantificateur D_0 ou D_1 , et pour insérer le bit 1 nous utilisons le quantificateur D_2 ou D_3 . Cette sélection revient à choisir les arcs empruntés lors de la traversée du treillis. Ainsi, on supprime les arcs étiquetés par D_2 ou D_3 lorsque l'on insère un 0 et on supprime les arcs étiquetés par D_0 ou D_1 lorsque l'on insère un 1. Nous nous retrouverons donc avec un treillis élagué et les coefficients

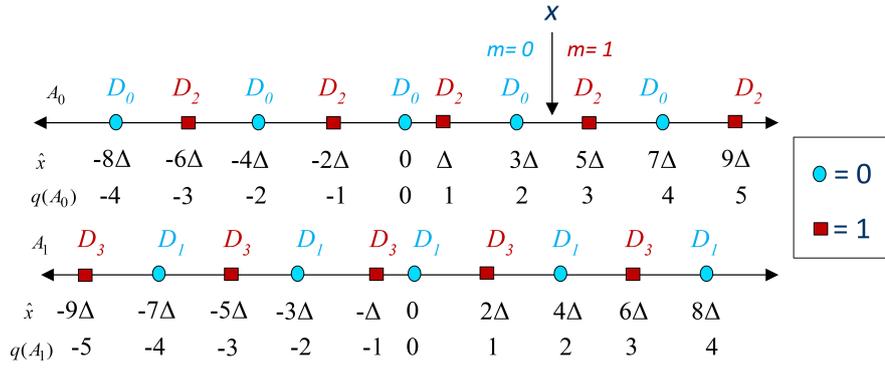


FIGURE 5.3 – Les principes de la QIM appliqués aux quantificateurs d’union de JPEG2000.

d’ondelettes sont quantifiés en lançant l’algorithme de Viterbi sur ce treillis élagué. Le chemin de coût minimal représenté par une séquence binaire est alors équivalent au message caché.

Lorsqu’on l’intègre cette approche dans le schéma de codage de JPEG2000, l’étape de contrôle de taux doit être prise en compte. En effet, la perte partielle du chemin dans le treillis signifie que les données cachées ne seront pas correctement extraites durant la phase de décompression. Sans mesure adéquate, c’est ce qui se produit puisqu’un certain nombre de bits de poids faible des indices de quantification sont supprimés après la phase de contrôle de taux.

Nous proposons donc d’insérer l’information cachée seulement dans les coefficients qui ont le plus de chance de survivre à la phase de contrôle de taux. Ces coefficients sont appelés *coefficients sélectionnés*. En pratique, cela revient à insérer (et donc élaguer le treillis) uniquement au niveau des transitions correspondant à ces *coefficients sélectionnés*. En outre, pour être sûr que le plan de bits LSB restera inchangé après le contrôle de taux, celui-ci est déplacé de sa position initiale vers un plan de bits supérieur.

Le message binaire à cacher est noté $\mathbf{m} \in \{0, 1\}^{|\mathbf{m}|}$. Dans le but de sécuriser le message, nous le mélangeons de manière pseudo-aléatoire. Le tatouage se fait indépendamment au niveau de chaque *code-block*. Pour chaque *code-block*, le treillis est élagué seulement pour les transitions correspondant aux *coefficients d’ondelettes sélectionnés*. L’étape de quantification produit la séquence d’indices de quantification $\tilde{\mathbf{x}}$ donnée par :

$$\tilde{\mathbf{x}}[i] = Q_{D_j}(\mathbf{x}[i]) \quad (5.1)$$

où Q_{D_j} est la fonction de quantification et $D_j, j \in \{0, 1, 2, 3\}$ est le quantificateur utilisé pour quantifier $\mathbf{x}[i]$. Le quantificateur est sélectionné suivant le bit à cacher.

Lors de la décompression JPEG2000, le message caché est extrait durant l’étape de quantification inverse. Plus précisément, l’algorithme examine sur le chemin optimal, les arcs associés aux *coefficients sélectionnés*. Si l’arc est étiqueté par D_0 ou D_1 , le bit 0 est extrait, sinon le bit 1 est extrait. Pour chaque *code-block*, les valeurs de reconstruction $\hat{\mathbf{x}}$ sont obtenues de la manière suivante :

$$\hat{\mathbf{x}}[i] = \bar{Q}_{D_j}^{-1}(\tilde{\mathbf{x}}[i]) \quad (5.2)$$

où \bar{Q}^{-1} est la fonction de quantification inverse. Notons qu’il n’est pas nécessaire de générer les valeurs dé-quantifiées pour extraire le message caché.

5.3.2 Le schéma conjoint JPEG2000 et dissimulation de données

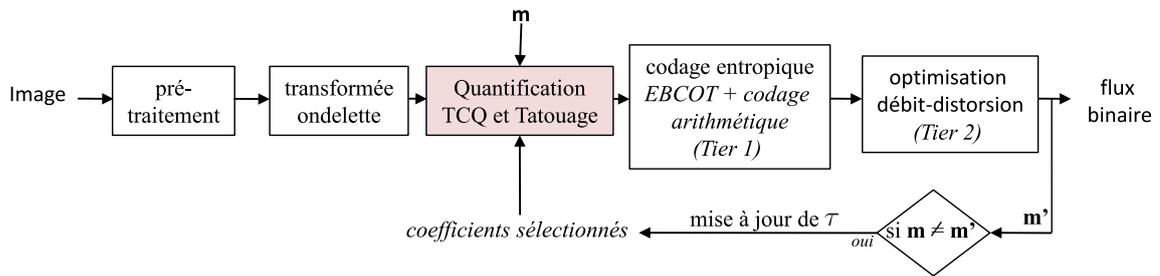


FIGURE 5.4 – Schéma de fonctionnement du système conjoint codage JPEG2000/dissimulation de données.

Le schéma de fonctionnement de notre système de compression JPEG2000/dissimulation de données conjoint est illustré sur la figure. 5.4. L'image subit des opérations de pré-traitements (décalage des données et transformée couleur). Celle-ci est ensuite décomposée à l'aide de la transformée en ondelettes discrète. Les coefficients inclus dans le processus de dissimulation de données sont ceux des sous-bandes de détails HL, LH et HH de tous les niveaux de résolution excepté le premier niveau. Le critère de sélection utilisé dans la sélection des coefficients est détaillé dans la sous-section suivante. Le nombre de *coefficients sélectionnés* permet de déterminer la quantité d'information à cacher. Les données sont cachées durant le processus de quantification. Ensuite vient l'étape de codage entropique, suivie de la phase de contrôle de taux qui arrange les flux de chaque *code-block* sous la forme de couches de qualité afin de former le flux binaire JPEG2000.

Suivant le contenu de l'image et le débit cible, certains coefficients embarquant des bits des données cachées seront tronqués lors de la phase d'optimisation débit-distorsion. Afin que l'extraction soit correcte, une opération de vérification est lancée après l'étape de contrôle de taux afin de tester s'il n'y a pas eu de perte d'information. Ce processus consiste à effectuer les opérations de décodage entropique et d'extraction des données cachées. Si l'information cachée n'a pas été correctement récupérée, l'étape de sélection des coefficients d'ondelettes est ré-itérée. Le critère de sélection est modifié pour chaque *code-block* où les données cachées sont erronées, et ce, de manière à exclure les *coefficients sélectionnés* qui ont été touchés par la phase de contrôle de taux. Nous sélectionnons moins de *coefficients sélectionnés* que lors de la précédente itération. Les étapes de quantification TCQ, EBCOT, contrôle de taux et vérification sont répétées à chaque fois que le critère de sélection est modifié, et ce, jusqu'à ce qu'il ait une insertion puis une extraction sans erreurs. Le *payload* est déterminé par le nombre de *coefficients sélectionnés*. Nous aurons donc un *payload* pour chaque débit binaire : les *payloads* obtenus à bas débit seront plus petits que ceux obtenus à haut débit.

Pour réaliser l'extraction des données cachées, le flux binaire est décodé à l'aide du décodeur EBCOT puis le message caché est extrait durant la quantification inverse. Seuls les *coefficients sélectionnés* lors de la quantification embarquent un bit. L'image décodée est obtenue après l'application de la transformée en ondelettes inverse et les opérations de post-traitements.

5.3.3 La sélection des coefficients

Comme nous l'avons expliqué en section 5.3.1, le bit de poids faible d'un *indice TCQ sélectionné* permet de déduire le quantificateur utilisé et également un bit du message. Un indice TCQ \tilde{x} peut être représenté en représentation binaire de la manière suivante :

$$\tilde{x} = s\tilde{x}_{L-1}\tilde{x}_{L-2}\dots\tilde{x}_1\tilde{x}_0 \quad (5.3)$$

où s est le signe, \tilde{x}_{L-1} le bit de poids le plus fort (MSB) et \tilde{x}_0 est le bit de poids le plus faible (LSB) de \tilde{x} . L est le nombre de bits nécessaires pour représenter tous les indices de quantification dans le *code-block*. Le calcul du seuil de sélection τ pour chaque *code-block* permet de sélectionner une séquence de coefficients significatifs \mathbf{S} . En supposant qu'il y a L plans de bits dans le *code-block* courant \mathbf{C} , τ est calculée de la manière suivante :

$$\tau = \lfloor \alpha \times L \rfloor \quad (5.4)$$

où α est un paramètre réel compris entre 0 et 1 initialisé avec une valeur prédéfinie pour chaque sous-bande. Le tableau 5.1 donne les valeurs de α pour certains débits cible. La sélection des coefficients inclus dans le processus de dissimulation de données est effectuée de la manière suivante :

$$\text{si } \lceil \log_2(|\tilde{\mathbf{x}}[i]| + 1) \rceil > \tau, \text{ alors } \tilde{\mathbf{x}}[i] \in \mathbf{S} \quad (5.5)$$

où $\lceil \log_2(|\tilde{\mathbf{x}}[i]| + 1) \rceil$ est le nombre de bits utilisés pour représenter l'indice \tilde{x} du $i^{\text{ème}}$ coefficient d'ondelette du *code-block* \mathbf{C} . Nous sélectionnons donc les coefficients d'ondelettes dont les indices TCQ possèdent un nombre de plans de bits supérieur à τ . Dans le cas où la phase de contrôle de taux provoque une dégradation du message caché, la valeur de τ est incrémentée pour itérer une nouvelle sélection des coefficients et une nouvelle insertion. Afin d'être sûr que le chemin ne sera

Niveau de résolution	Sous-bande	Débit binaire (bpp)	α
3	1	2.5	0.20
		1.6	0.33
		0.2	0.50
4	2	2.5	0.33
		1.6	0.33
		0.2	0.75

TABLE 5.1 – Valeurs initiales de α pour quelques débits binaires.

pas partiellement perdu après la phase de contrôle de taux, surtout à bas débit, nous proposons de déplacer le plan de bits LSB des indices TCQ des *coefficients sélectionnés* vers la position $L - 2$ (équation 5.3). Le seuil de sélection τ pour chaque *code-block* peut être stocké dans l'entête du fichier JPEG2000. De cette manière, nous sommes capables, durant la décompression JPEG2000, de retrouver les positions des indices TCQ correspondant aux *coefficients sélectionnés*. Nous n'avons donc pas besoin de sauvegarder ces positions dans un fichier et de les transmettre comme information adjacente au décodeur. Notons que le coût additionnel est négligeable par rapport au coût de l'image. Pour un débit de 0.2 bpp, un *code-block* de taille 64×64 , et $L = 8$, le ratio entre le coût de codage du seuil et le coût de codage moyen du *code-block* est $\frac{3}{0.2 \times 64 \times 64} \approx 0.0037$.

5.4 Résultats expérimentaux

Nous utilisons la librairie OpenJPEG [Ope11] pour implémenter notre schéma. Cette librairie est libre de droit, et écrite en langage C, et implémente la norme JPEG2000 partie 1. Nous avons remplacé le module de quantification scalaire uniforme par un module de quantification TCQ compatible avec la partie 2 de la norme. Les paramètres de compression sont les suivants : transformée en ondelette 9/7 avec 5 niveaux de décomposition, taille des code-blocks égale à 64×64 du premier au troisième niveau de résolution, égale à 32×32 pour le quatrième niveau, et égale à 16×16 pour le cinquième niveau. Le taux de compression varie entre 2.5 bpp et 0.2 bpp. Les sous-bandes de détails HL, LH et HH du second au cinquième niveau de résolution sont retenues pour l'insertion. Nous avons testé l'approche sur des images de taille 512×512 ce qui donne un total de 21 *code-blocks* utilisables pour le processus d'insertion. Notons que le coût de codage de seuils de sélection τ est négligeable puisqu'en prenant un codage du seuil sur 4 bits, le coût de codage des 21 seuils est seulement de 84 bits. Cette information est placée dans l'entête du fichier JPEG2000.

5.4.1 Évaluation des performances de *data-hiding* du schéma conjoint

Notre approche a été testée sur 200 images choisies aléatoirement dans la base d'images BOWS2¹ [Bas et al. 08b]. Les images ont une taille de 512×512 pixels et sont en niveaux de gris avec 8 bits/pixel. Pour chaque débit binaire, nous avons calculé le *payload* du message inséré. Les valeurs moyenne, minimale et maximale des *payloads* sont données dans le tableau 5.2.

Débit binaire	2.5 bpp	2 bpp	1.6 bpp	1 bpp	0.5 bpp	0.2 bpp
<i>Payload</i> moyen	11254	11203	11172	7384	4213	1573
<i>Payload</i> minimal	1266	1266	1266	1266	926	422
<i>Payload</i> maximal	36718	26180	21903	13470	7530	2621

TABLE 5.2 – *Payloads* obtenus à l'aide de notre schéma conjoint sur 200 images de la base BOWS-2.

Les *payloads* obtenus sont importants. Le *payload* moyen est supérieur à 11 000 bits pour des débits compris entre 2.5 bpp et 1.6 bpp. À 2.5 bpp, le *payload* minimal est de 1 266 bits alors que le *payload* maximal est de 36 718 bits. Cette grande différence entre les *payloads* minimal et maximal s'explique parce que le nombre de *coefficients sélectionnés* dépend du contenu de l'image. Une image avec de larges plages uniformes et peu de contraste donnera des coefficients d'ondelettes de faible amplitude. À l'inverse, une image texturée avec de grandes différences de contraste possède bien plus d'information et fournira un grand nombre de coefficients d'ondelettes de forte amplitude. En d'autres termes, nous pouvons insérer un volume de données cachées bien plus important dans des images fortement texturées par rapport à des images faiblement texturées. Le taux de compression influe également sur la capacité d'insertion du message. À partir de 1 bpp, le *payload* est beaucoup moins important par rapport aux hauts débits. Cela provient de la troncature des *bitstreams* de chaque *code-block* à bas débits qui laisse peu de coefficients d'ondelettes candidats à l'insertion. Le *payload* diminue donc à mesure que le débit décroît.

1. La base d'image BOWS2 est disponible à l'adresse : <http://bows2.gipsa-lab.inpg.fr>

5.4.2 Évaluation des performances de compression du schéma conjoint

L'évaluation des performances visuelles de notre schéma conjoint est effectuée en comparant les images compressées avec le codeur JPEG2000 classique et les images compressées/augmentées à l'aide de notre schéma conjoint. Les expérimentations ont été effectuées sur 7 images tests de taille 512 x 512 : Lena, Goldhill, Girl, Barbara, Bike, Peppers et Clown.

Le Tableau 5.3 présente les valeurs obtenues par le schéma conjoint proposé et par le codeur JPEG2000 en termes de PSNR et de SSIM. La dégradation de la qualité visuelle induite par l'insertion des données cachées est relativement petite lorsqu'on compare le PSNR (respectivement le SSIM) de l'image compressée augmentée avec celui obtenu à partir de l'image compressée avec JPEG2000. Une différence de 1.29 dB (0,69 %) est constatée entre les deux valeurs de PSNR (SSIM) pour l'image Goldhill à 2 bpp, et 0.92 dB (1,64 %) pour l'image Barbara à 0.2 bpp. Pour d'autres images, la différence est plus importante. À titre d'exemple, il existe une différence de PSNR de 2.62 dB pour l'image Lena à 2.5 bpp. La différence de SSIM n'est cependant que de 0.54 %.

5.4.3 Évaluation pratiques des coûts calculatoire

Un critère également important est le temps de calcul. Nous considérons d'abord la phase de compression. Le schéma conjoint proposé utilise un algorithme itératif pour insérer le message. Les étapes de quantification TCQ, codage Tier 1, codage Tier 2 et vérification sont répétées jusqu'à ce que le message puisse être extrait sans erreur. Plus le nombre d'itérations augmente, plus le temps de calcul est important. Les résultats présentés dans le tableau 5.4 sont obtenus en utilisant un ordinateur Intel dual core 2 GHZ et CPU avec 3 Go de RAM. À mesure que le débit binaire décroît, le nombre d'itérations, et donc, le temps de calcul augmente. Concernant le codeur JPEG2000 partie 2, le temps de compression moyen pour une image de taille 512 × 512 est d'environ 1,90 s. En comparant avec les temps de calcul obtenus par notre schéma conjoint, nous constatons que JPEG2000 est beaucoup plus rapide que notre système conjoint lors de la phase de compression. Concernant la phase de décompression, les temps de calculs pour les deux codeurs sont similaires car l'extraction du message s'effectue durant l'étape de quantification inverse (environ 0.55 s pour une image 512 x 512).

5.4.4 Vers un schéma robuste

Le schéma que nous venons de présenter est un schéma d'insertion de données cachées sans robustesse. Ses performances égalent les schémas de ce type comme par exemple l'approche de Chen *et al.* [Chen et al. 10]. Le schéma a également été adapté dans la thèse de Dalila Goudia pour tester la robustesse du message face à quatre types d'attaques sur une image décompressée. Ce schéma est appelé T-EPAD (Tatouage avec Extraction Pendant et Après Décompression) et est présenté de la page 98 à la page 104 [Goudia 11]. Par rapport à ce que nous venons de présenter, un code correcteur est ajouté et le payload est réduit. Le protocole d'évaluation consiste :

1. à insérer le message conjointement à la compression JPEG2000,
2. à décompresser l'image pour obtenir une image »brute« (matrice de pixels),
3. à exécuter une attaque à la robustesse (ajout de bruit gaussien, filtrage gaussien, attaque valométrique de changement d'échelle, compression JPEG),

Image test	Débit binaire (bpp)	PSNR (dB)		SSIM	
		JPEG2000	conjoint	JPEG2000	conjoint
Lena	2.5	47.47	44.85	0.9836	0.9782
	2	45.33	43.83	0.9749	0.9726
	1.6	43.38	42.53	0.9661	0.9634
	1	41.55	40.57	0.9490	0.9436
	0.5	40.03	37.87	0.9226	0.9130
	0.2	36.56	35.29	0.8700	0.8560
Goldhill	2.5	43.63	42.32	0.9838	0.9769
	2	43.33	42.04	0.9707	0.9638
	1.6	43.30	42.03	0.9600	0.9532
	1	41.63	41.79	0.9271	0.9186
	0.5	39.87	40.36	0.8704	0.8610
	0.2	38.74	38.49	0.7638	0.7478
Girl	2.5	42.58	40.57	0.9806	0.9767
	2	40.10	37.79	0.9607	0.9559
	1.6	38.68	36.96	0.9461	0.9412
	1	34.72	34.68	0.8984	0.8952
	0.5	33.13	32.15	0.8291	0.8214
	0.2	31.63	31.08	0.7586	0.7521
Barbara	2.5	42.78	40.30	0.9854	0.9813
	2	41.15	38.72	0.9782	0.9740
	1.6	39.41	37.32	0.9701	0.9657
	1	35.93	33.53	0.9507	0.9427
	0.5	30.73	28.67	0.9042	0.8907
	0.2	26.23	25.31	0.7894	0.7730
Bike	2.5	43.23	41.48	0.9775	0.9734
	2	39.64	40.26	0.9547	0.9512
	1.6	39.33	38.56	0.9343	0.9253
	1	38.11	37.03	0.8852	0.8726
	0.5	36.51	33.82	0.8031	0.7789
	0.2	33.52	33.54	0.6737	0.6671
Clown	2.5	44.08	41.16	0.9885	0.9825
	2	42.78	40.06	0.9817	0.9775
	1.6	40.77	38.80	0.9741	0.9669
	1	38.71	36.34	0.9596	0.9486
	0.5	35.76	33.29	0.9249	0.9081
	0.2	31.09	29.93	0.8384	0.8160
Peppers	2.5	43.13	41.53	0.9800	0.9753
	2	39.69	38.66	0.9649	0.9617
	1.6	39.20	37.50	0.9531	0.9485
	1	39.03	36.20	0.9273	0.9227
	0.5	36.50	32.47	0.8878	0.8785
	0.2	29.35	27.08	0.8378	0.8226

TABLE 5.3 – Comparaison de la qualité de l'image décompressée en termes de PSNR et de SSIM entre JPEG2000 et le schéma conjoint d'insertion de données cachées.

Débit (bpp)	Image test	Nombre d'itérations	Temps de calcul (s)	Image test	Nombre d'itérations	Temps de calcul (s)
2.5	Lena	1	3.68	Goldhill	1	3.78
2		1	3.42		1	3.66
1.6		1	3.31		1	3.53
1		1	3.13		2	6.39
0.5		2	5.83		5	14.84
0.2		2	5.56		6	16.70
2.5	Girl	1	3.47	Barbara	1	3.80
2		1	3.39		1	3.66
1.6		1	3.27		1	3.56
1		3	9.01		2	6.50
0.5		2	5.88		3	8.88
0.2		5	13.85		3	8.49
2.5	Bike	1	4.08	Clown	1	3.66
2		2	7.43		1	3.46
1.6		3	11.24		1	3.40
1		3	10.20		1	3.19
0.5		4	12.52		2	5.94
0.2		3	9.09		5	13.77
2.5	Peppers	1	3.57			
2		1	3.43			
1.6		1	3.34			
1		1	3.13			
0.5		2	6.06			
0.2		2	5.78			

TABLE 5.4 – Temps de calcul pour la compression et l'insertion du message et nombre d'itérations de l'algorithme itératif.

4. à relancer le processus de compression JPEG2000 jusqu'à l'étape de sélection des coefficients et de quantification par TCQ pour extraire les bits du message (l'algorithme de Viterbi est lancé sur le treillis complet ; les bits de chemin des transitions associées aux coefficients sélectionnés représentent le message codé).

Il faut noter que pour cette expérimentation la carte de coefficients sélectionnée est une information transmise à l'algorithme d'extraction. En effet, sans cette information l'algorithme d'extraction doit déterminer quels sont les coefficients ondelette porteurs du message. Or ce calcul mène à une sélection de coefficients ondelette différente de celle calculée lors de l'insertion. Si l'on décide de ne pas transmettre la carte de coefficients sélectionnés, l'algorithme d'extraction déterminera de manière incorrecte une carte qui va impliquer la sélection de coefficients ne portant pas d'information et également la non-sélection de coefficients portant l'information. On voit donc qu'il est nécessaire de mettre en place un mécanisme de gestion des effacements si l'on souhaite supprimer la transmission de la carte au décodeur, c'est-à-dire obtenir un schéma de tatouage à détection aveugle. Les codes correcteurs à gestion d'effacement (section IV.C de [Solanki et al. 06]), sont donc une perspective à explorer pour avoir dans le même temps un schéma robuste et ne nécessitant pas d'information adjacente lors de l'extraction. Cette piste est laissée en perspective.

Dans le schéma T-EPAD, l'algorithme d'extraction utilise la carte qui est transmise par le codeur JPEG2000 (on a donc un schéma à détection non aveugle). Le payload est fixé à 1024 bits pour des images 512×512 . Les couches de qualité sont configurées de manière à atteindre une distorsion de 45 dB. L'approche est testée sur 200 images choisies aléatoirement dans la base d'images BOWS2 [Bas et al. 08b]. Les résultats obtenus sont décevants puisqu'il n'y a pas de robustesse. Des attaques de faibles puissances impliquent immédiatement des taux d'erreur binaire supérieurs à 10% des bits du message. Ces résultats peuvent cependant être améliorés en choisissant un meilleur paramétrage (un pas de quantification adapté au tatouage, en choisissant mieux les sous-bandes, en réduisant le payload, ...). Ce travail est mis en perspective de la thèse de Dalila Goudia.

5.5 Conclusion

Dans ce chapitre, nous avons présenté un schéma conjoint de compression JPEG2000 et de dissimulation de données basé sur la TCQ. Les données cachées sont dissimulées dans les indices TCQ. La perte des bits du message caché causée par la phase de contrôle de taux est évitée grâce à une sélection des coefficients inclus dans le processus de dissimulation de données et grâce au déplacement des plans de bits LSB vers une autre position. Le schéma conjoint proposé présente de bonnes performances en termes de qualité visuelle et de capacité d'insertion. Les applications possibles sont l'augmentation ou l'enrichissement des contenus des images JPEG2000.

Si l'on regarde les performances en terme de *data-hiding* sans robustesse, le schéma est parmi les plus performants de l'état de l'art. Il est de faible complexité, dégrade très peu les performances visuelles ainsi que les performances en compression. Par ailleurs, c'est un schéma qui ne nécessite pas de protocole compliqué pour transmettre une information adjacente puisque cette information est intégrée dans l'entête du fichier. C'est donc un schéma pratique qui est utilisable en l'état.

Si l'on s'intéresse à ses performances en terme de tatouage robuste, il est nécessaire d'ajouter des codes correcteurs, de réduire le *payload*, de sélectionner plus finement les sous-bande à utiliser, etc., pour obtenir des résultats en robustesse satisfaisants. En l'état, le schéma T-EPAD est bien moins robuste que ce que l'on pourrait espérer. Notons qu'un autre schéma (T-EAD) a égale-

ment été proposé dans la thèse de Dalila Goudia² qui nécessite la transmission d'une information adjacente lors du décodage, mais dont les performances en termes de robustesse sont meilleures.

Par ailleurs, une forte limitation de ces schémas est la difficulté qu'il y a à les étendre à des schémas à détection complètement aveugles. Pour le schéma que nous venons de présenter, il est en effet nécessaire de connaître les seuils pour retrouver les *coefficients sélectionnés* lors de l'extraction. Une extension envisageable est d'utiliser des codes correcteurs à gestion d'effacement (section IV.C de [Solanki et al. 06]), ce qui est un moyen d'avoir un schéma à la fois robuste et ne nécessitant pas d'information adjacente lors de l'extraction.

2. Dans l'approche T-EAD l'insertion est finalisée lors de la décompression JPEG2000 ; cette proposition s'éloigne de la conception classique du tatouage comme un problème de « communication fiable ».

Chapitre 6

Une approche avec prise en compte de l'optimisation RD dans H.264

Résumé

Ce chapitre présente une approche de traçage de traîtres (en anglais « traitor tracing », « transactional watermarking », « content serialisation », « users forensics », ou encore « active fingerprinting ») intégrée au codeur vidéo de l'état de l'art H.264/AVC. Cette approche a été publiée à ICIP'2010 dans [Shahid et al. 10b] et également développée dans la thèse de Zafar Shahid [Shahid 10]. Ce travail réunit deux concepts importants : le tatouage conjoint à la compression H.264 et l'utilisation d'un code anti-collusion. La partie qui nous intéressera le plus concerne le tatouage. Ce travail met en lumière, à travers de nombreuses expérimentations et une implémentation opérationnelle, la difficulté à maintenir les nombreuses contraintes d'un système de tatouage transactionnel (système de traçage de traîtres) pour une vidéo : la robustesse aux attaques photométriques, la robustesse aux désynchronisations, la robustesse aux collusions, la sécurité, le payload (et donc un code anticollusion de petite taille), l'invisibilité, la complexité.

6.1 Quelques mots sur le traçage de traîtres

L'objectif du traçage de traîtres est de dissuader l'utilisation illégale d'un produit. À la suite d'une transaction entre un vendeur et un/des acheteur(s), le traçage de traîtres permet au vendeur de déterminer le/les acheteur(s) qui redistribue(nt) illégalement le produit. Lors de la vente d'un film, le vendeur intègre un mot de code qui lui permettra d'identifier l'acheteur à partir de la vidéo. Si le vendeur découvre que son film est redistribué illégalement, la technique de traçage de traîtres doit lui permettre d'identifier le/les acheteur(s) qui sont responsables de cette redistribution. L'approche de traçage de traîtres doit fournir un mécanisme d'accusation sûre qui permette au vendeur d'engager des sanctions ou des poursuites judiciaires à l'encontre du/des fraudeur(s) (que l'on appelle les traîtres). Le mécanisme de traçage de traîtres doit donc fournir un mécanisme d'accusation dans lequel aucun innocent ne peut être accusé. En particulier, il doit être impossible pour un traître d'usurper l'identité d'un innocent. Un groupe de traîtres peut construire une nouvelle version de la vidéo par collusion (mélange) de leur version dans l'espoir de faire disparaître l'identifiant intégré à leur vidéo. Dans ce cas, le mécanisme de traçage de traîtres doit permettre d'identifier au moins un traître.

Dans la littérature, le traçage de traîtres a été initialement étudié par la communauté cryptographique. Un système de code anticollusion doit fournir une construction qui permet de retrouver au moins un traître sans erreur possible. À la différence des codes anti-collusion déterministes, les codes probabilistes permettent de fixer la probabilité d'accuser un innocent quelle que soit la stratégie de collusion [Tardos 03]. On peut donc fixer cette probabilité à une valeur suffisamment faible et créer un code plus sûr.

Un code anti-collusion est défini par les paramètres suivants :

- le nombre d'utilisateurs n ,
- le nombre de *colluders* (traîtres) c ,
- la probabilité de faux positifs (probabilité d'accuser un innocent) ϵ_1 ,
- la probabilité de faux négatifs (probabilité de ne pas accuser un traître) ϵ_2 ,
- la longueur de code m obtenue à partir des paramètres précédents.

Il faut noter que la longueur de code m influence grandement l'utilisabilité dans un schéma joint à une technique de tatouage. En tatouage vidéo, il est difficile d'insérer de manière robuste (robustesse photométrique, robustesse valométrique, robustesse aux désynchronisations spatiales et temporelles) plus d'une dizaine de bits dans une minute de vidéo. Le standard dans l'industrie du cinéma numérique haute définition est de 35 bits insérés dans 15 minutes de film. Malheureusement, les contraintes des vendeurs impliquent un code de grande dimension. Les vendeurs ont besoin d'avoir un mécanisme résistant à des coalitions de centaines de traîtres ($c > 100$), fournissant des mots de code pour des millions voir des milliards d'utilisateurs ($n > 10^6$), et dont la probabilité d'accusation d'un innocent (faux positifs) ϵ_1 doit être extrêmement faible. La probabilité de non-détection d'un coupable (faux négatifs) ϵ_2 est beaucoup moins critique et l'on peut tolérer une valeur assez grande.

En 2003, Tardos [Tardos 03] a proposé un code probabiliste pour la problématique du traçage de traîtres. La longueur du code était estimée à $m = 100c^2 \ln(\frac{n}{\epsilon_1})$ pour n utilisateurs, c *colluders*, et une probabilité d'accuser un innocent ϵ_1 . Notons, que la constante 100 dans la longueur m du code a été par la suite réduite dans [Škorić et al. 08b], [Škorić et al. 08a], [Blayer et al. 08], etc. Les auteurs de [Škorić et al. 08b], [Škorić et al. 08a], [Blayer et al. 08] entre autre, ont repris les travaux de Tardos [Tardos 03], revu les hypothèses et optimisé les paramètres sans fondamentalement changer ni la construction du code ni l'algorithme d'accusation.

La génération d'un code de Tardos de dimension m est extrêmement simple et s'effectue en deux étapes :

- **Génération de m valeurs de probabilité :** On initialise un générateur pseudoaléatoire avec une clef secrète puis l'on génère m valeurs réelles comprises entre 0 et 1 dont la distribution est $f(p) = \frac{1}{\pi\sqrt{p(1-p)}}$ avec $p \in [0, 1]$. Ces m valeurs sont notées $\{p(i)\}_{1 \leq i \leq m}$ et correspondent à la probabilité d'avoir un 1 pour le i -ème bit d'un mot de code ;
- **Génération du code de Tardos :** Pour n utilisateurs, on génère une matrice \mathbf{S} de taille $m \times n$. Une colonne représente un mot de code associé à un j -ème utilisateur. Les lignes de la matrice sont remplies par des 0 et des 1 de manière à ce que $Prob[\mathbf{S}(i, j) = 1] = p(i)$.

Le processus d'accusation est également extrêmement simple puisqu'il suffit de calculer un score d'accusation A_j entre le mot extrait \mathbf{z} et le mot de code d'un utilisateur j :

$$A_j = \sum_{i=1}^m U(\mathbf{z}(i), \mathbf{S}(i, j), p(i)), \quad (6.1)$$

avec la fonction U définie pour $p \in [0, 1]$ par :

$$\begin{aligned} U(1, 1, p) &= \sqrt{(1-p)/p}, & U(1, 0, p) &= -\sqrt{p/(1-p)}, \\ U(0, 0, p) &= \sqrt{p/(1-p)}, & U(0, 1, p) &= -\sqrt{(1-p)/p}. \end{aligned}$$

Le score des traîtres peut être modélisé par une distribution gaussienne centrée en $\mu = \frac{2m}{c\pi}$ alors que le score des innocents peut être modélisé par une gaussienne centrée en 0. Il suffit donc de fixer un seuil au-dessus duquel on considérera que l'on a un traître. Notons que Céro *et al.* [Céro *et al.* 08] ont proposé un seuillage précis qui relie ϵ_1 et ce seuil. Bien que la génération du code de Tardos et l'accusation soient extrêmement simples, le mécanisme d'accusation est complexe en coût de calcul (dans le pire des cas il y a $n \times m$ appels à la fonction U) et complexe en coût de stockage si l'on souhaite stocker la matrice \mathbf{S} (cela fait $n \times m$ bits). De plus, la longueur du code augmente quadratiquement avec le nombre de *colluders* m , logarithmiquement avec la diminution de ϵ_1 et logarithmiquement avec l'augmentation du nombre d'utilisateurs n . Par exemple, pour $c = 10$, $n = 10^4$, et $\epsilon_1 = 10^{-5}$, on a $m = 100c^2 \ln(n/\epsilon_1) \approx 200\,000$ bits. Cela fait une complexité de calcul de l'accusation, au pire, d'environ 2 milliards d'appels à U et un coût de stockage d'environ 2 Giga-bits si l'on souhaite stocker la matrice \mathbf{S} . Si c passe à 20, $m \approx 800\,000$ bits. Cela correspond à une complexité de calcul de l'accusation, au pire, d'environ 8 milliards d'appels à U et un coût de stockage d'environ 8 Giga-bits si l'on souhaite stocker la matrice \mathbf{S} .

6.2 Le tatouage dans H.264 (insertion d'un mot de code issu du code de Tardos)

Comme nous l'avons déjà évoqué dans le chapitre 4, le tatouage dans H.264 peut intervenir à différents emplacements (avant quantification, après quantification et lors du codage entropique [Chaumont 10b]). Pour assurer la robustesse du signal de tatouage, il faut effectuer l'opération d'insertion du tatouage avant ou après l'opération de quantification. De plus, pour maintenir les bonnes performances en débit et en distortion, il faut avoir un mécanisme joint de tatouage et de compression. Le mécanisme que nous avons retenu est basé sur un tatouage par étalement de spectre (l'hypothèse de *marking assumption* [Tardos 03] nécessaire au bon fonctionnement du mécanisme d'accusation de Tardos est alors respectée). L'approche que nous proposons a pour but d'étudier la faisabilité de l'intégration d'un schéma de traçage au sein d'une vidéo. Si l'on souhaite obtenir des meilleures performances (en terme de robustesse photométrique, mais aussi de sécurité) on peut aisément reprendre le principe de tatouage informé proposé dans Broken Arrows [Furon *et al.* 08] pour remplacer le module de tatouage par étalement de spectre.

L'approche de tatouage consiste à découper chaque « frame » en régions qui vont chacune embarquer 1 bit. De manière pratique, chaque région sera stockée dans ce que l'on appelle un « slice ». Un « slice » est une partie du flux binaire H.264 (c'est une portion du fichier obtenue après compression) à laquelle on peut accéder très facilement et dont le contenu est décodable indépendamment des autres « slices ». La figure 6.1 montre le découpage d'une « frame » vidéo CIF 352x288 en 10 régions¹, chacune embarquant 1 bit. Notons que le processus de tatouage peut être effectué *offline*. Il suffit en effet pour chaque « slice » composant la vidéo de générer 2 versions différentes : un « slice » contenant le bit 0 et l'autre contenant le bit 1. Lorsqu'une vidéo est distribuée à un acheteur, le vendeur attribue un mot de code de Tardos à l'acheteur

1. Dans nos expérimentation nous avons choisi un code de longueur raisonnable [Furon 09] $m = 20 \cdot c^2 \cdot \ln(\frac{n}{\epsilon_1})$.

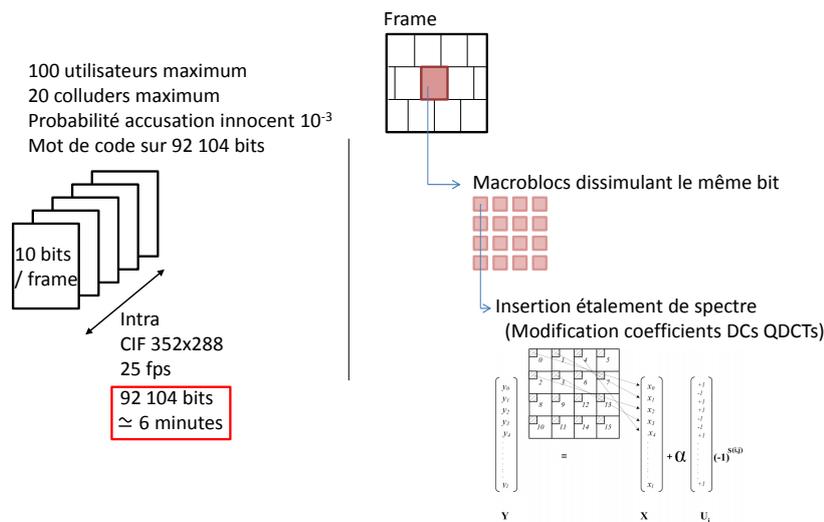


FIGURE 6.1 – Schéma général d'insertion d'un mot de code de Tardos dans H.264.

puis reconstitue la vidéo en utilisant les « slices » qui ont été tatoués avant la transaction et enfin distribue la vidéo à l'acheteur.

Le mécanisme de tatouage est très simple. Une fois que la découpe en « slice » est effectuée, l'insertion d'un bit dans le « slice » s'effectue par étalement de spectre. Le « slice » est composé de plusieurs macroblocs. L'ensemble des macroblocs du « slice » va être modifié de sorte qu'ils contiennent le même bit. Le principe de tatouage est le même pour tous les macroblocs. Les coefficients quantifiés DCs des blocs du macrobloc forment le vecteur hôte \mathbf{x} . On applique alors le tatouage par étalement de spectre (voir équation 6.2). On génère une porteuse secrète $\mathbf{u}_i \in \{-1, 1\}^n$ (avec n la dimension du vecteur hôte), on module le i -ème bit du mot de code de Tardos $S(i, j)$ du j -ème utilisateur pour avoir -1 si $S(i, j) = 1$ et $+1$ si $S(i, j) = 0$, puis on étale le bit modulé sur la porteuse (multiplication de la porteuse et du bit modulé) pour obtenir le vecteur de tatouage. Le vecteur de tatouage est alors amplifié ou réduit par multiplication par le facteur de force d'insertion (scalaire α) et enfin ajouté au vecteur hôte \mathbf{x} pour obtenir le vecteur tatoué \mathbf{y} :

$$\mathbf{y} = \mathbf{x} + \alpha \cdot \mathbf{u}_i \cdot (-1)^{S(i,j)} \quad (6.2)$$

Le vecteur \mathbf{y} correspond alors aux nouveaux coefficients DCT quantifiés. Il faut noter qu'une approche qui modifie les coefficients DCs produit des effets de blocs désagréables. Dans l'article [Shahid et al. 11d] nous avons proposé d'améliorer l'approche en ajoutant les coefficients ACs mais également en sélectionnant les coefficients à modifier. La procédure de sélection des coefficients nécessite d'effectuer des seuillages spécifiques à chaque vidéo. L'introduction de seuil variable est problématique puisque ces seuils doivent être ré-estimés correctement lors de l'extraction. On peut raisonnablement penser que l'utilisation d'un masque psychovisuel suffisamment robuste, et l'utilisation d'un code correcteur prenant en compte les effacements, permettrait de régler le problème.

Lors de l'étape d'extraction, pour chaque macrobloc, les coefficients ACs sont ré-extraits pour

former le vecteur tatoué-attaqué \mathbf{z} . La corrélation entre la porteuse \mathbf{u}_i et le vecteur \mathbf{z} nous donne le bit inséré (et donc le i -ème bit du mot de code attribué au j -ème utilisateur et noté $\tilde{\mathbf{S}}(i, j)$) :

$$\tilde{\mathbf{S}}(i, j) = \begin{cases} 0, & \text{if } \sum_{k=0}^n \mathbf{z}[k] \cdot \mathbf{u}_i[k] > 0 \\ 1, & \text{if } \sum_{k=0}^n \mathbf{z}[k] \cdot \mathbf{u}_i[k] < 0. \end{cases} \quad (6.3)$$

Un point important de l'intégration du tatouage par étalement de spectre dans le codeur H.264 est son intégration au sein de la phase de compression. Lorsqu'un bloc est en cours de traitement, le codeur doit choisir le mode de prédiction du bloc qui va donner le meilleur compromis débit-distorsion. Pour sélectionner le mode de prédiction, il faut :

- appliquer les différentes prédictions du bloc,
- calculer les différents blocs d'erreurs,
- tatouer les blocs d'erreurs,
- calculer pour toutes ces versions du bloc d'erreur le débit et la distorsion,
- enfin, sélectionner la meilleure prédiction en choisissant la prédiction pour laquelle la valeur de la fonction Lagrangienne de débit-distorsion est la plus petite.

Une fois que la prédiction du bloc est choisie, on applique réellement le tatouage. On voit bien qu'en faisant comme ceci, le signal de tatouage est parfaitement intégré dans le flux vidéo. L'impact dû au tatouage est pris en compte à travers l'optimisation débit-distorsion du codeur. Par ailleurs, il n'y a pas de dérive entre ce qui est calculé au codage et ce qui est calculé au décodage. Sur la figure 6.2, nous retrouvons les différentes étapes, décrites dans le chapitre 4, pour un codeur H.264. L'étape de tatouage a également été ajoutée au schéma. Remarquons que la figure n'est cependant pas suffisante pour décrire le mécanisme d'intégration tel que nous venons de l'expliquer.

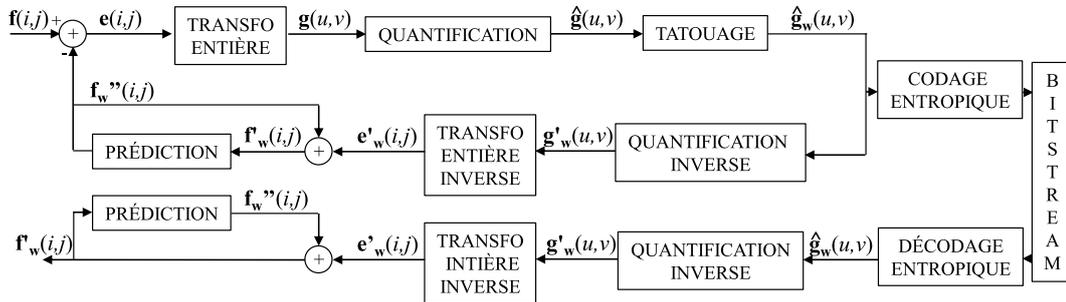


FIGURE 6.2 – Schéma général du système de tatouage intégré à H.264

6.3 Résultats

Dans cette section nous donnons les résultats de l'article [Shahid et al. 10b]. Nous avons utilisé l'implémentation de JSVM 10.2 de H.264 avec des vidéos en résolution CIF 352×288 . Nous avons choisi le mode intra 4×4 pour le codage Intra, sélectionné le codeur CAVLC, fixé la quantification à $QP = 18$, et inséré le mot de code de Tardos à la fois dans les luminances et les chrominances. Neuf séquences (*bus*, *city*, *foreman*, *football*, *soccer*, *harbour*, *ice*, *mobile*, *crew*) ont été concaténées et répétées pour former la vidéo à tatouer. Les paramètres du code de Tardos sont $n = 100$ utilisateurs, $\epsilon_1 = 10^{-3}$, $c = 20$ colluders et une longueur de code

$m = 20 \cdot c^2 \cdot \ln\left(\frac{n}{\epsilon_1}\right) = 92104$ [Furon 09]². On insère 10 bits par « frame ». Il faut donc 9211 « frames » pour insérer complètement le mot de code de Tardos ce qui fait environ 6 minutes pour une vidéo à 25 images par seconde.

Lorsqu'un vendeur désire tracer la provenance d'une vidéo, il va extraire le mot de code de Tardos inséré dans la vidéo et ensuite lancer son processus d'accusation. Comme nous l'avons expliqué en section 6.1, le vendeur va calculer un score d'accusation A_j pour chaque utilisateur j . Si le score dépasse le seuil d'accusation, l'utilisateur j est alors considéré comme coupable de redistribution illégale de la vidéo. Le score des traîtres peut être modélisé par une distribution gaussienne centrée en $\mu = \frac{2m}{c\pi}$ et de variance m , alors que le score des innocents peut être modélisé par une gaussienne centrée en 0. En première approximation nous avons fixé le seuil d'accusation à $\mu - \sqrt{m} = \frac{2m}{c\pi} - \sqrt{m}$. Un seuillage plus adéquat peut être calculé en utilisant l'approche proposée dans [Cérou et al. 08].

Pour évaluer le schéma, seules les attaques par collusion dans le domaine spatial ont été testées. Il faudrait bien entendu également évaluer la robustesse du schéma (aux attaques photométriques) et également sa sécurité (par exemple face à la collusion Intra). L'objectif premier de ce travail était d'évaluer à la fois la faisabilité d'un schéma de traçage de traîtres intégré à H.264 mais également de proposer une solution de tatouage conjointement à la compression. En ce sens, les résultats obtenus ici sont prometteurs.

Les attaques par collusion ont toutes été réalisées dans le domaine spatial. Les traîtres disposent tous d'une version différente de la même vidéo et créent une nouvelle vidéo en utilisant une stratégie particulière. Chaque pixel de la nouvelle vidéo est obtenu en utilisant l'une des stratégies suivantes : calcul de la moyenne, calcul du minimum, calcul du maximum, calcul de la valeur médiane, calcul du minmax (moyenne de la valeur maximum et minimum) et calcul du modneg (valeur minimum plus valeur maximum moins valeur médiane). La table 6.1 reprend l'ensemble des attaques en explicitant les formules. f_k correspond à la « frame » du *colluder* k , \mathcal{T} correspond à l'ensemble des *colluders* (traîtres), et K correspond au nombre de *colluders*.

TABLE 6.1 – Liste des différentes stratégies d'attaque de la part des *colluders*.

$f_{min} = \min\{f_k\}_{k \in \mathcal{T}}$	$f_{max} = \max\{f_k\}_{k \in \mathcal{T}}$
$f_{moy} = \sum_{k \in \mathcal{T}} \frac{f_k}{K}$	$f_{med} = median\{f_k\}_{k \in \mathcal{T}}$
$f_{minmax} = \frac{f_{min} + f_{max}}{2}$	$f_{modneg} = f_{min} + f_{max} - f_{med}$

Le PSNR des vidéos tatouées est de 35 dB et pratiquement toutes les stratégies d'attaque mènent à un PSNR proche ou même supérieur à 35 dB pour des collusions de 1 à 20 *colluders*.

2. Il faut remarquer que dans le cadre d'une application à plusieurs millions d'utilisateurs, et une centaine de *colluders*, la probabilité ϵ_1 devrait être inférieure à 10^{-6} . La longueur du code serait alors trop grande pour pouvoir envisager de l'utiliser dans un système de tatouage.

Seule la stratégie modneg fait chuter le PSNR à 15dB. Cette dernière stratégie est donc peu intéressante puisque la qualité de la vidéo pirate est tellement faible qu'elle en devient inutile.

La table 6.2 montre le nombre de *colluders* qui ont été successivement détectés après analyse de la vidéo pirate pour les différentes stratégies de collusion et pour un nombre variable de *colluders*. La stratégie la plus efficace est l'attaque modneg mais c'est aussi une attaque qui n'est pas réaliste. Pour toutes les autres attaques, quelle que soit le nombre de *colluders*, le processus d'accusation retrouve pratiquement tous les traîtres. Sachant que dans une application de traçage de traîtres, le plus important est de déterminer au moins un traître, l'approche fonctionne extrêmement bien.

TABLE 6.2 – Nombre de *colluders* détectés à partir d'une copie pirate issue d'une stratégie de collusion à K *colluders*.

K	Nombre de <i>colluders</i> détectés pour chaque stratégie d'attaque					
	moyenne	min	max	median	minmax	modneg
2	2	2	2	2	2	2
5	5	5	5	5	5	5
8	8	8	8	8	8	6
11	11	10	10	10	10	7
14	14	13	13	13	13	9
17	16	15	16	16	16	10
20	18	18	18	19	18	11

6.4 Conclusion

Ce travail a mis en lumière la faisabilité d'un système de traçage de traîtres par tatouage vidéo au sein de H.264. En effet, à ma connaissance, il n'y a pas dans la littérature d'autres systèmes de tatouage intégré à H.264 avec une utilisation des codes de Tardos. Le code de Tardos a été utilisé pour assurer le traçage de traîtres d'une vidéo pirate obtenue par collusion. Le système de tatouage et le code ont été conçus pour être réalistes dans le cas d'un très petit nombre d'utilisateurs ($n = 100$). Aucune des stratégies de collusion n'a mis en défaut le processus d'accusation.

Il est évident que l'intégration d'un système de traçage de traîtres dans une vidéo compressée n'est pas encore un problème totalement résolu. Cela dit, la proposition de tatouage de code de Tardos par étalement de spectre au sein de H.264 fonctionne bien et permet de mettre en lumière le travail qu'il reste à faire pour obtenir des systèmes plus matures. On voit bien, par exemple, que la longueur des codes anti-collusion doit être réduite.

Si l'on revient à la seule problématique du tatouage dans H.264, on peut facilement se convaincre que la complexité du mécanisme de tatouage est très faible. Les expérimentations détaillées dans la thèse de Zafar Shahid [Shahid 10] montrent en pratique que les coûts de calcul additionnels pour effectuer le tatouage sont négligeables par rapport à la phase de compression. On peut donc dire que le tatouage conjoint à la compression est une approche intéressante si l'on cherche à réduire la complexité tout en ayant un tatouage robuste.

Pour terminer, on notera que la sécurité des systèmes de tatouage joints à la compression reste un point qui n'a pas été évalué dans la littérature à la vue des connaissances nouvelles en sécu-

rité [Cayre et al. 05], [PérezFreire et al. 06], [Chaumont 11]. C'est évidemment un point crucial qui doit être évalué dans le cadre d'application comme le traçage de traître. De même, il serait important de rendre les schémas de tatouage vidéo robustes aux attaques désynchronisantes.

Troisième partie

La dissimulation de la couleur d'une image ; Une étude curieuse

Chapitre 7

Les deux grandes familles de dissimulation de la couleur

Résumé

Cette partie du manuscrit traite de dissimulation de données et non de tatouage robuste. Nous proposons de dissimuler la couleur d'une image dans sa version en niveaux de gris. L'approche permet de distribuer une version dégradée (la version en niveaux de gris) d'une image tout en laissant la possibilité à un utilisateur autorisé de ré-extraire une version améliorée (la version en couleur). Dans ce chapitre, nous présentons deux approches bien connues de dissimulation de la couleur dans une image en niveaux de gris : l'approche de [Campisi et al. 02] et l'approche de [Queiroz et al. 06] qui ressemble très fortement à celle de Campisi *et al.*. Ces deux approches consistent à substituer certaines sous-bandes ondelette de luminance par des sous-bandes ondelette de chrominance. Dans les chapitres suivants, nous présenterons deux propositions permettant de dissimuler la couleur d'une image dans sa version en niveaux de gris avec une approche utilisant la représentation d'une image par une palette couleur et une image d'*index*. Notons que cette partie du manuscrit reprend en grande partie le chapitre de livre [Chaumont et al. 09b].

7.1 Introduction

Jusqu'à encore deux-trois ans, il existait peu de travaux autour de la dissimulation de la couleur dans une image en niveaux de gris. Le travail de [Campisi et al. 02] est précurseur et permet d'augmenter les performances de compression d'images couleur. De Queiroz et Braun [Queiroz et al. 06] ont proposé quant à eux de dissimuler la couleur pour pouvoir imprimer des documents en « noir et blanc » et également récupérer la couleur à partir de l'image « noir et blanc » scannée. Nous avons également proposé de dissimuler la couleur d'une image pour « protéger/sécuriser » la valeur commerciale de l'image [Chaumont et al. 06]. Le scénario envisagé est de donner un libre accès à la version dégradée (l'image en niveaux de gris) et de donner un accès sécurisé à une version améliorée (l'image en couleur). La figure 7.1 donne une utilisation possible de la protection de la couleur où seuls les propriétaires de la clef secrète peuvent reconstruire l'image couleur.

Les approches de [Campisi et al. 02], [Queiroz et al. 06] pour dissimuler la couleur dans une image en niveaux de gris, sont basées sur la décomposition en ondelettes et la substitution de sous-

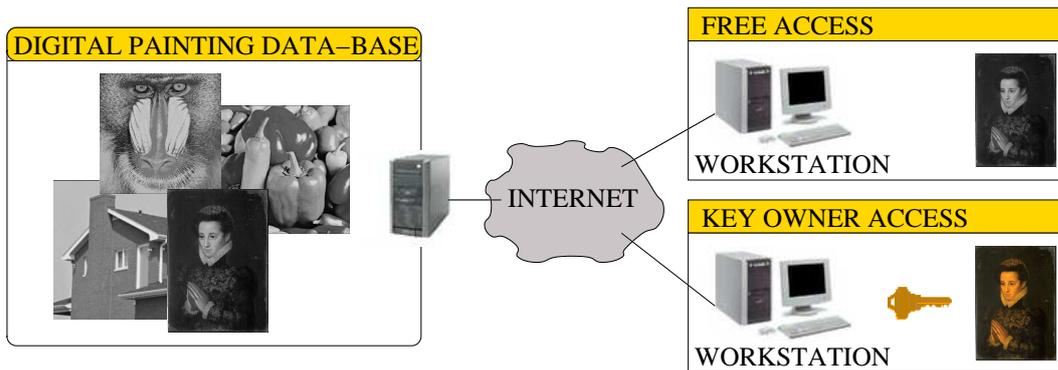


FIGURE 7.1 – Illustration d’une utilisation possible de la protection de la couleur.

bandes. La section 7.2 présente l’approche pour l’impression de [Queiroz et al. 06] et la section 7.3 présente l’approche pour la compression perceptuelle de [Campisi et al. 02]. Les quelques autres papiers publiés sur le sujet contiennent de très petites améliorations. On peut tout de même citer quelques publications récentes pour prolonger l’étude. De Queiroz [Queiroz 10] étudie l’influence des énergies de chaque sous-bande pour améliorer l’impact de l’insertion, ainsi que l’influence du processus « print & scan ». Les propositions [Ko et al. 08], [Horiuchi et al. 10] reprennent des idées déjà introduites dans Campisi *et al.* [Campisi et al. 02] (paquet d’ondelettes, zone d’insertion, *scaling* des plans de chrominances) et comparent expérimentalement les différentes approches.

7.2 L’approche de De Queiroz et Braun

La solution proposée par [Queiroz et al. 06] est spécifique au problème d’impression d’images. Quand un document possédant des images en couleur doit être distribué, il est très souvent imprimé sur des imprimantes « noir et blanc ». Les graphiques, certaines images couleur, les schémas, etc. peuvent devenir illisibles. La solution proposée consiste à transformer une image couleur en une image compréhensible en niveaux de gris, même en présence de nombreuses couleurs de même valeur de luminance. Ce qui nous concerne le plus dans la solution de [Queiroz et al. 06] est la relative réversibilité de leur approche. Il est possible de récupérer une approximation visuellement proche de l’image couleur originale à partir de l’image en niveaux de gris.

L’approche proposée par De Queiroz et Braun [Queiroz et al. 06] illustrée figure 7.2 est très simple et consiste à :

1. passer l’image de l’espace couleur RGB à l’espace de couleur Y, Cr, Cb,
2. sous-échantillonner par quatre les deux plans Cr et Cb,
3. déterminer deux plans Cr^+ et Cr^- (resp. Cb^+ et Cb^-) à partir de la version sous-échantillonnée de Cr (resp. Cb). Le plan Cr^+ (resp. Cb^+) est une copie du plan sous-échantillonné Cr (resp. Cb) dont les valeurs négatives sont positionnées à zéro et le plan Cr^- (resp. Cb^-) est une copie du plan sous-échantillonné de Cr (resp. Cb) dont les valeurs positives sont mises à zéro ;
4. décomposer le plan Y en deux niveaux de décomposition ondelette,

5. substituer la sous-bande LH1 par Cb^+ , HL1 par Cr^+ , HH1 par Cr^- et HH2 par une version de Cb^- sous-échantillonnée par quatre,
6. inverser la transformée ondelette, et obtenir une image en niveaux de gris dissimulant l'information couleur.

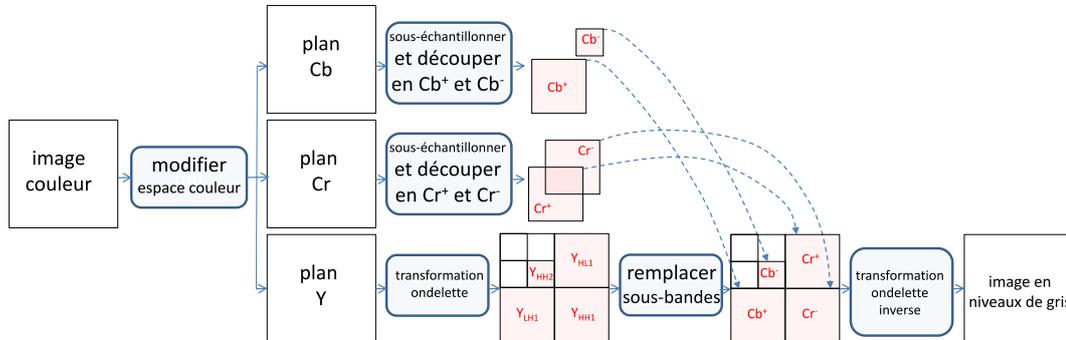


FIGURE 7.2 – Dissimulation de la couleur dans l'approche de De Queiroz et Braun [Queiroz et al. 06].

Pour extraire l'information couleur dissimulée dans l'image en niveaux de gris, c'est-à-dire régénérer les trois plans Y, Cr, Cb il faut :

1. appliquer une transformation ondelette en deux niveaux,
2. extraire Cb^+ , Cr^+ , Cr^- , et Cb^- ,
3. sur-échantillonner Cb^- ,
4. recalculer Cr et Cb tel que $Cr = |Cr^+| - |Cr^-|$ et $Cb = |Cb^+| - |Cb^-|$, et sur-échantillonner Cr et Cb afin d'obtenir les plans de chrominance de dimensions égales à la dimension originale,
5. mettre à zéro les sous-bandes HL1, LH1, HH1 and HH2 de l'image ondelette et appliquer une transformation ondelette inverse afin d'obtenir le plan Y.

La Figure 7.3 illustre le principe d'extraction de l'image couleur.

Cette approche substitutive dans le domaine ondelette est intéressante pour son potentiel de dissimulation de l'information couleur dans une image en niveaux de gris. La figure 7.4 donne quelques images clefs du processus sur l'image baboon. L'image de luminance, l'image dissimulant l'information de couleur, et la décomposition ondelette de l'image de luminance avec substitution de sous-bandes par des sous-bandes de chrominances sont données aux figures 7.4.a, 7.4.b, et 7.4.c. On remarque que l'image couleur ré-extraite (figure 7.4.e) à partir de l'image en niveaux de gris est de faible qualité visuelle en comparaison de l'image couleur originale (figure 7.4.d). L'image est légèrement floue avec des artefacts de rebond. Ces artefacts proviennent de la détérioration trop forte du plan Y due à la mise à zéro d'un grand nombre de sous-bandes. Dans le tableau 7.1 nous remarquons que les PSNRs couleur calculés entre l'image couleur d'origine et l'image reconstruite sont inférieurs à 31 dB. L'approche de [Queiroz et al. 06] est donc intéressante pour l'impression d'image couleur puisqu'elle « ajoute de la texture » à l'image de luminance, mais elle est moins intéressante pour la dissimulation puisque l'image couleur reconstruite est de qualité médiocre. Nous verrons dans la section suivante que l'approche de Campisi *et al.* [Campisi et al. 02] donne de meilleurs PSNRs et des images visuellement plus agréables.

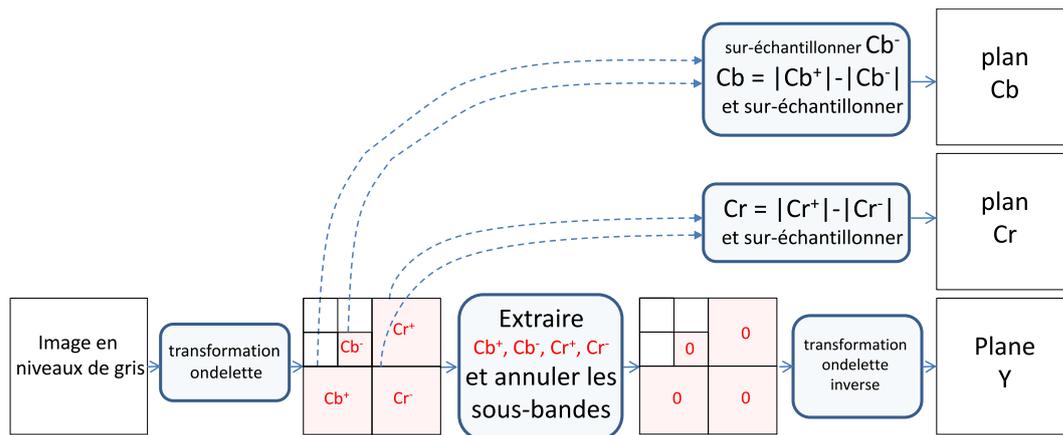


FIGURE 7.3 – Extraction de l’image couleur à partir de l’image en niveaux de gris dans l’approche de De Queiroz et Braun [Queiroz et al. 06].

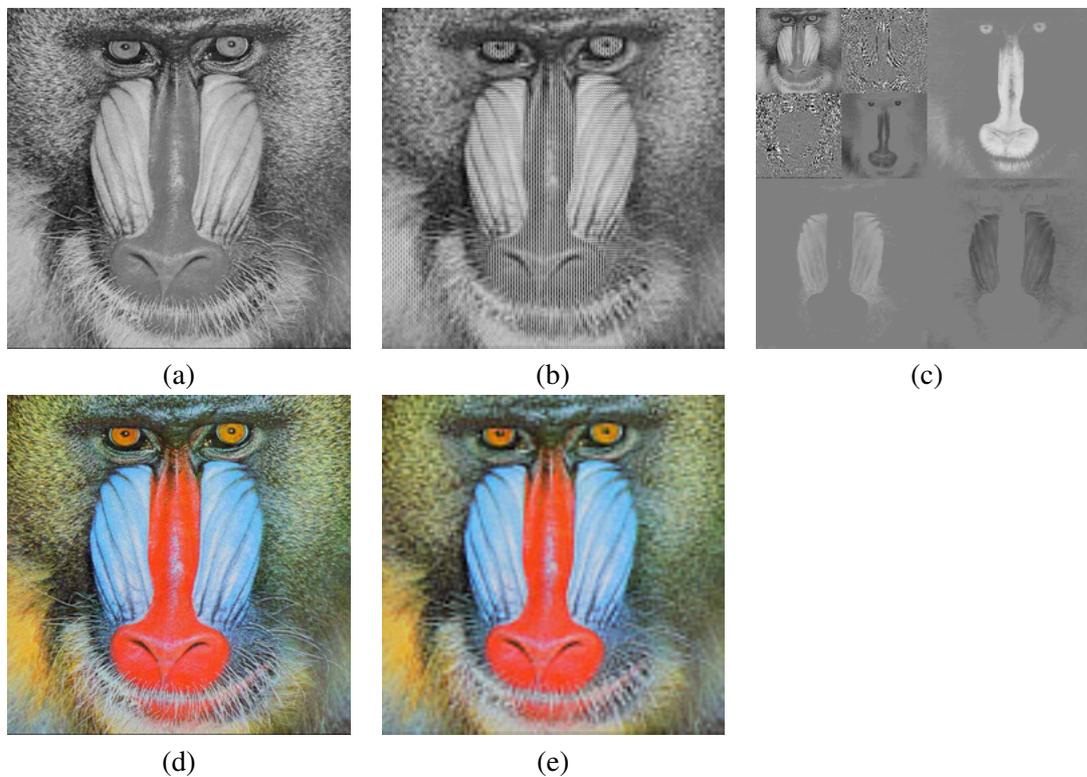


FIGURE 7.4 – Application de l’approche basée substitution de [Queiroz et al. 06] : a) Luminance de l’image originale, b) Image en niveaux de gris embarquant les plans de chrominance, c) Décomposition en ondelette de l’image en niveaux de gris, d) Image couleur originale, e) Image reconstruite à partir de l’image en niveaux de gris.

TABLE 7.1 – PSNR calculé entre l’image de luminance et l’image en niveaux de gris et PSNR calculé entre l’image couleur originale (espace RGB) et l’image reconstruite (espace RGB) en utilisant l’image en niveaux de gris embarquant la couleur pour l’approche de [Queiroz et al. 06].

Images	PSNR _(luminance, niveaux de gris)	PSNR _(couleur originale, reconstruite)
baboon	21.03 dB	23.93 dB
barbara	23.93 dB	26.33 dB
airplane	26.47 dB	28.56 dB
peppers	21.24 dB	28.82 dB
lena	21.02 dB	30.31 dB
house	25.18 dB	30.75 dB

7.3 L’approche de Campisi *et al.*

L’approche de Campisi *et al.* [Campisi et al. 02] a pour objectif d’améliorer l’efficacité de compression d’une image couleur. L’image couleur est transformée en une image en niveaux de gris qui intègre les plans de chrominances puis cette image en niveaux de gris est compressée. Le principe de dissimulation de la couleur est très similaire à l’approche de De Queiroz et Braun. La distorsion entre l’image couleur d’origine et celle reconstruite est néanmoins bien plus petite.

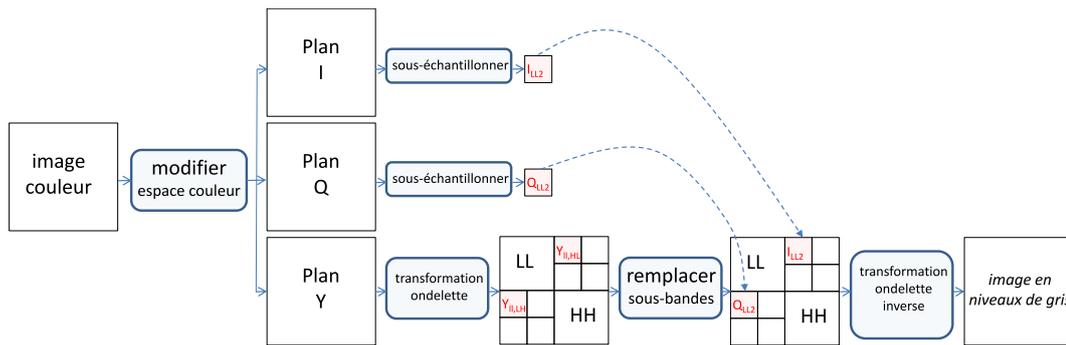


FIGURE 7.5 – Dissimulation de la couleur dans l’approche de Campisi *et al.* [Campisi et al. 02].

L’approche, illustrée sur la Figure 7.5, consiste à :

1. exprimer l’image couleur dans l’espace couleur Y, I, Q,
2. sous-échantillonner par seize les deux plans I et Q pour obtenir deux plans notés I_{LL2} et Q_{LL2} ; Ce sous-échantillonnage est réalisé en appliquant une décomposition en ondelette Daubechies 9/7 [Daubechies et al. 98] en 2-niveaux puis en conservant uniquement la sous-bande passe-bas de I et Q,
3. appliquer une décomposition ondelette Daubechies en 1-niveau sur le plan Y et réappliquer une décomposition ondelette en 1-niveau sur la sous-bande HL et LH ; La décomposition de la sous-bande HL (resp. LH) donne quatre sous-bandes et la sous bande passe-bas est notée $Y_{ll,HL}$ (resp. $Y_{ll,LH}$),

4. normaliser les deux plans I_{LL2} et Q_{LL2} avec deux valeurs réelles NI et NQ ; Ces deux valeurs seront transmises comme information adjacente,
5. substituer la sous-bande $Y_{u,HL}$ (resp. $Y_{u,LH}$) par I_{LL2} (resp. Q_{LL2}),
6. comprimer l'image en niveaux de gris.

Pour extraire l'information couleur (les plans Y, I, Q) à partir de l'image en niveaux de gris et ensuite reconstruire l'image couleur, il faut :

1. appliquer les transformées ondelette sur l'image en niveaux de gris,
2. extraire I_{LL2} et Q_{LL2} et les multiplier par les valeurs de normalisation NI et NQ,
3. sur-échantillonner I_{LL2} et Q_{LL2} pour obtenir les deux plans I et Q,
4. mettre à zéro l'ensemble des coefficients de $Y_{u,HL}$ et $Y_{u,LH}$ et appliquer les transformées ondelette inverses pour obtenir le plan Y ; La Figure 7.6 illustre le principe d'extraction des plans Y, I, Q.

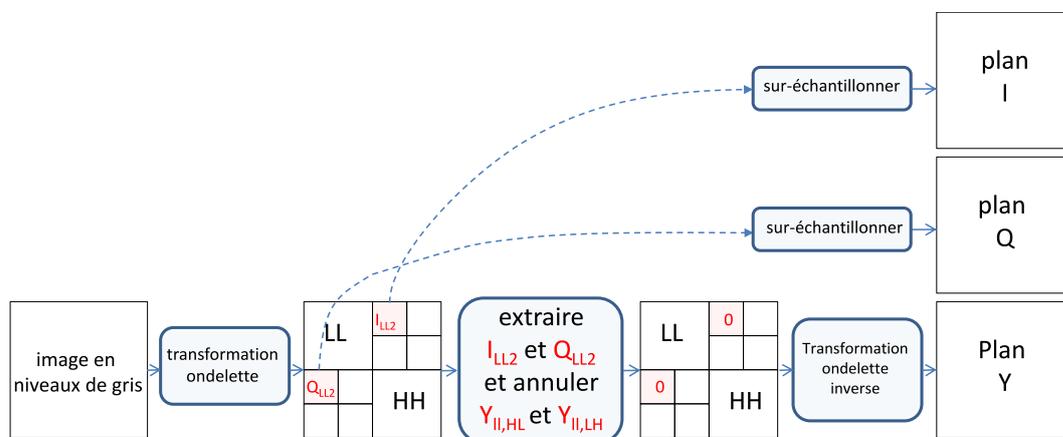


FIGURE 7.6 – Extraction de l'image couleur à partir d'une image en niveaux de gris dans l'approche de Campisi *et al.* [Campisi et al. 02].

La figure 7.7 donne les images clés du processus de dissimulation de la couleur sur l'image baboon. On peut remarquer que l'image couleur reconstruite (Figure 7.7.e) à partir de l'image en niveaux de gris est de bonne qualité. L'approche de Campisi *et al.* fonctionne bien, car elle dégrade peu la composante Y et les plans de chrominances sont de qualité suffisante. En regardant de plus près on peut tout de même observer de très faibles artefacts de rebond. Sur l'ensemble des images testées, les PSNRs calculés entre l'image couleur d'origine et l'image couleur reconstruite sont tous supérieurs à 29 dB (voir Tableau 7.2). L'approche de Campisi *et al.* permet d'obtenir des résultats objectifs et subjectifs supérieurs à l'approche de [Queiroz et al. 06]. Nous verrons dans la section suivante que l'approche basée palette permet d'obtenir des images reconstruites de meilleure qualité.

7.4 Les approches basées palettes

Afin d'obtenir une image en niveaux de gris embarquant des informations de couleur permettant de régénérer une image en couleur, nous décomposons l'image couleur en une image d'*index*

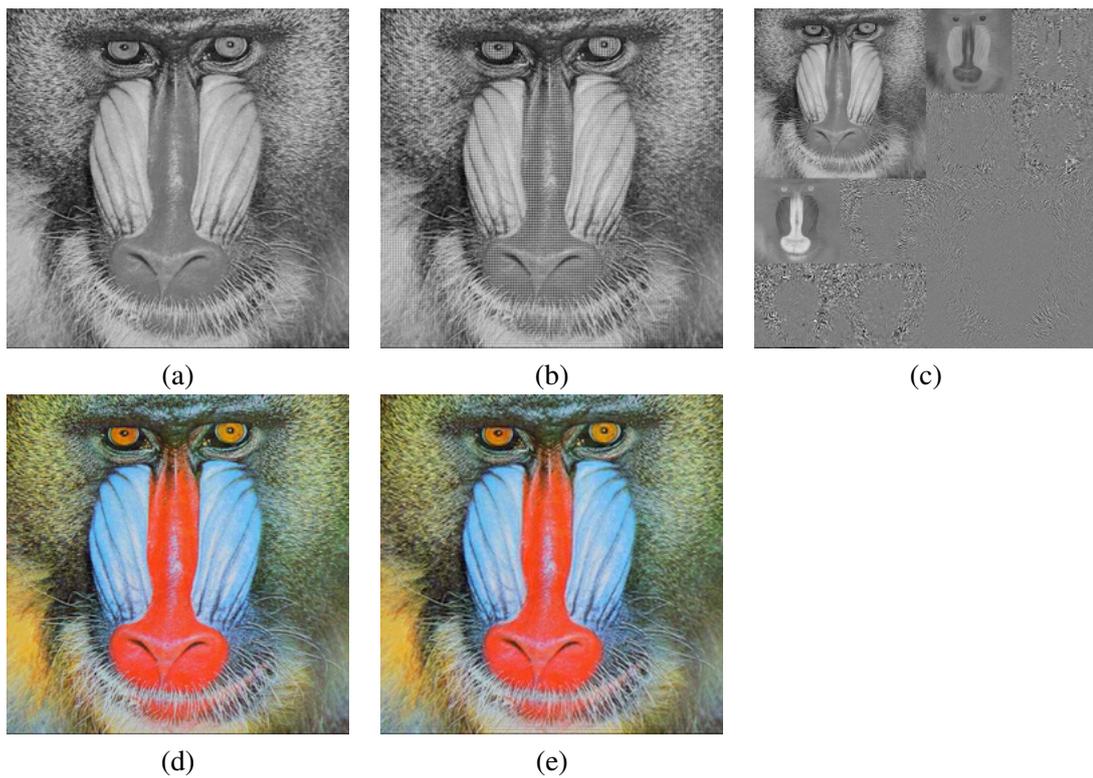


FIGURE 7.7 – .

]Application de l'approche basée substitution de [Campisi et al. 02] : a) Luminance de l'image originale, b) Image en niveaux de gris embarquant les plans de chrominance, c) Décomposition en ondelette de l'image en niveaux de gris, d) Image couleur originale, e) Image reconstruite à partir de l'image en niveaux de gris.

TABLE 7.2 – PSNR calculé entre l'image de luminance et l'image en niveaux de gris et PSNR calculé entre l'image couleur originale (espace RGB) et l'image reconstruite (espace RGB) en utilisant l'image en niveaux de gris embarquant la couleur pour l'approche de [Campisi et al. 02].

Images	PSNR _(luminance, niveaux de gris)	PSNR _(couleur originale, reconstruite)
baboon	27.37 dB	29.8 dB
barbara	30.61 dB	31.75 dB
peppers	25.82 dB	32.36 dB
airplane	34.12 dB	32.58 dB
house	30.76 dB	31.76 dB
lena	26.91 dB	36.75 dB

et une palette de couleurs. La figure 7.8 illustre la décomposition de l'image couleur Baboon en une image d'*index* (figure 7.8.c) et une palette de couleurs (figure 7.8.d). L'image d'*index* est une matrice contenant des indices. Chaque indice correspond à une case du tableau de couleurs ; chaque indice pointe donc sur une couleur. Remarquons que la décomposition d'une image couleur correspond à un processus de quantification (voir l'image quantifiée à la figure 7.8.b) avec réduction du nombre de couleurs. Lors du mécanisme de dissimulation, la palette de couleurs est cachée dans l'image d'*index*. La dissimulation de l'information couleur par dissimulation d'une palette fait apparaître trois contraintes : l'image d'*index* doit être similaire à la luminance de l'image couleur ou tout du moins compréhensible, le processus de dissimulation doit faiblement dégrader l'image d'*index* ou ne pas la dégrader du tout, et la palette de couleurs doit être soigneusement ordonnée pour, si besoin, faciliter sa compression.

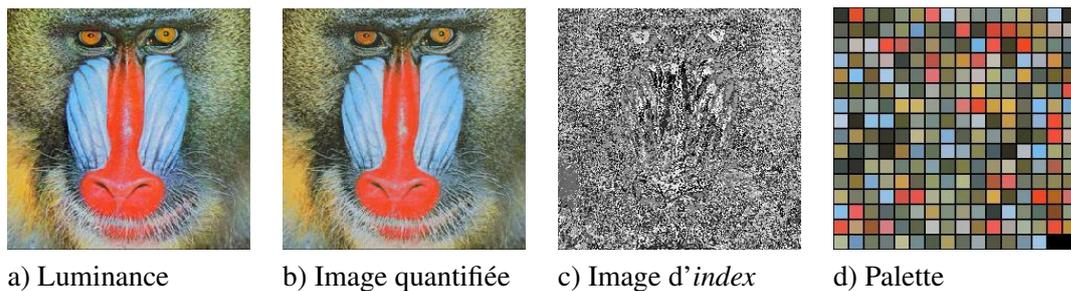


FIGURE 7.8 – Décomposition d'une image couleur en une image d'*index* et une image couleur.

De nombreux travaux de dissimulation de données proposent de cacher dans une image, des données numériques, en utilisant un domaine d'insertion issu de la décomposition d'une image couleur en une image d'*index* et une palette de couleurs. L'insertion de données est alors effectuée : soit dans l'image d'*index* [Fridrich 99], soit dans la palette de couleurs [Wu et al. 03], [Tzeng et al. 04]. Parmi les nombreux travaux du domaine, aucun ne propose de cacher la palette de couleurs dans l'image d'*index* c'est-à-dire de protéger les informations couleur.

Nous avons proposé deux solutions à ce problème : l'approche floue [Chaumont et al. 07c] que nous présentons dans le chapitre 8 et l'approche par réorganisation [Chaumont et al. 08a] que nous présentons dans le chapitre 9.

Chapitre 8

Une approche par modélisation floue et optimisation

Résumé

Dans ce chapitre, nous proposons de sécuriser les images couleurs par dissimulation de leur palette de couleurs. L'approche proposée est parue dans [Chaumont et al. 07c]. Le problème est tout d'abord modélisé par une fonctionnelle décrivant la décomposition d'une image couleur en une image d'*index* et une palette de couleurs, puis la fonctionnelle est minimisée. Le modèle proposé permet d'obtenir une décomposition en une image d'*index* semblable à l'image de luminance et une palette de couleurs adaptée pour l'insertion de données cachées. Après la décomposition de l'image couleur, la palette est insérée dans l'image d'*index*. Les images obtenues (images d'*index* marquées et images couleur reconstruites) sont de bonne qualité. Bien que la complexité de l'approche soit élevée, la proposition est élégante et permet aisément d'adapter l'influence de chacune des contraintes.

8.1 Modélisation du problème de décomposition et résolution

Dans cette section nous présentons le modèle énergétique de décomposition d'une image couleur. Nous abordons ensuite dans la section 8.2 la méthode d'insertion de la palette de couleurs dans son image d'*index*. Dans la section 8.3 nous analysons les résultats. Enfin, nous concluons en section 8.4.

Dans [Chaumont et al. 06] l'approche se décompose en deux étapes successives : une quantification et un réordonnement de la palette de couleurs. La contribution repose sur la définition d'un *algorithme de parcours en couches* dont le but est de parcourir l'espace RGB quantifié pour trouver une palette de couleurs réorganisée. L'approche est extrêmement rapide et permet d'obtenir une image d'*index* fortement contrastée.

Dans l'approche par modélisation floue (*fuzzy*) [Chaumont et al. 07c], l'image d'*index* est meilleure visuellement et nous obtenons un meilleur équilibre qualitatif entre l'image d'*index* et l'image couleur quantifiée. Le problème est également mieux modélisé : la quantification de l'image couleur et l'ordonnement de palette de couleurs sont réalisés en une seule étape, la contrainte sur la continuité des couleurs de la palette est plus faible et la solution optimale du modèle énergétique est atteinte et non plus approchée.

La décomposition d'une image couleur doit respecter les trois contraintes suivantes :

1. l'image d'*index* doit être semblable à l'image de luminance,
2. l'image couleur quantifiée doit être fidèle à l'image couleur originale,
3. la palette de couleurs doit être composée de couples de couleurs proches.

Mathématiquement cela revient à trouver les $\mathbf{C}(k), k \in \{1, \dots, K\}$ couleurs (\mathbf{C} représente la palette de couleurs) et les valeurs d'appartenance $\mathbf{P}_{i,k}, i \in \{1, \dots, N\}$ indiquant le degré d'appartenance d'un pixel i à la $k^{\text{ème}}$ couleur. Notons que les $\mathbf{P}_{i,k}$ sont des valeurs réelles qui appartiennent à $[0, 1]$ et sont appelées valeurs d'appartenance floue (*fuzzy membership values*) dans les approches de clustering c-mean flou [Dunn 74]. Notons également que l'ensemble des $\mathbf{P}_{i,k}$ donne indirectement l'image d'*index* telle que : $\forall i, \text{Index}(i) = \arg_k \max_k \mathbf{P}_{i,k}$.

Ainsi, nous cherchons à minimiser le modèle énergétique E pour obtenir $\forall i, \forall k, \mathbf{P}_{i,k}$ et $\mathbf{C}(k)$:

$$\begin{aligned}
 E = & \underbrace{\sum_{i=1}^{i=N} \sum_{k=1}^{k=K} \mathbf{P}_{i,k}^\gamma \text{dist}^2(\mathbf{C}(k), \mathbf{I}(i))}_{\text{premier terme}} \\
 & + \lambda_1 \underbrace{\sum_{i=1}^{i=N} \sum_{k=1}^{k=K} \mathbf{P}_{i,k}^\gamma (\mathbf{Y}(i) - k)^2}_{\text{deuxième terme}} \\
 & + \lambda_2 \underbrace{\sum_{k|k \in [1..K] \text{ et } k \text{ est impair}} \text{dist}^2(\mathbf{C}(k), \mathbf{C}(k+1))}_{\text{troisième terme}},
 \end{aligned} \tag{8.1}$$

avec \mathbf{I} l'image couleur, \mathbf{Y} l'image de luminance, dist la fonction distance L2 entre deux couleurs RGB, λ_1 et λ_2 deux scalaires et $\gamma \in]1, \infty[$ le coefficient flou réglant le degré d'équi-répartition¹.

Le premier terme exprime la contrainte de quantification couleur. Le but est de trouver les K couleurs les plus représentatives. Le deuxième terme force l'image d'*index* à être la plus proche possible de l'image de luminance \mathbf{Y} . Le dernier terme contraint les couples de couleurs de la palette à être proches.

La minimisation de l'équation 8.1 telle que :

$$\{\mathbf{P}_{i,k}, \mathbf{C}(k)\} = \arg \min_{\{\mathbf{P}_{i,k}, \mathbf{C}(k)\}} E, \tag{8.2}$$

est obtenue en itérant alternativement le calcul des couleurs $\mathbf{C}(k)$ et le calcul des valeurs d'appartenance $\mathbf{P}_{i,k}$. Les couleurs $\mathbf{C}(k)$ sont donc mises à jour en figeant les valeurs d'appartenance $\mathbf{P}_{i,k}$ et en résolvant le système linéaire suivant :

$\forall k \text{ impair}$:

$$\left(\lambda_2 + \sum_{i=1}^{i=N} \mathbf{P}_{i,k}^\gamma \right) \times \mathbf{C}(k) - \lambda_2 \times \mathbf{C}(k+1) = \sum_{i=1}^{i=N} \mathbf{P}_{i,k}^\gamma \mathbf{I}(i),$$

1. γ est positionné à 2 pour réduire la complexité calculatoire.

$\forall k$ pair :

$$-\lambda_2 \times \mathbf{C}(k-1) + \left(\lambda_2 + \sum_{i=1}^{i=N} \mathbf{P}_{i,k}^\gamma \right) \times \mathbf{C}(k) = \sum_{i=1}^{i=N} \mathbf{P}_{i,k}^\gamma \mathbf{I}(i). \quad (8.3)$$

Les valeurs d'appartenance $\mathbf{P}_{i,k}$ (avec $\gamma = 2$) sont mises à jour en figeant les $\mathbf{C}(k)$:

$$P_{i,k} = \frac{\left(\sum_{l=1}^{l=K} \frac{1}{2 \times (\text{dist}^2(\mathbf{C}(l), \mathbf{I}(i)) + \lambda_1(\mathbf{Y}(i)-l)^2)} \right)^{-1}}{2 \times (\text{dist}^2(\mathbf{C}(k), \mathbf{I}(i)) + \lambda_1(\mathbf{Y}(i) - k)^2)}. \quad (8.4)$$

Les détails mathématiques sont donnés en Annexe.

8.2 La méthode d'insertion de données cachées

Nous insérons la palette de couleurs dans les bits de poids faibles (*LSB : Least Significant Bit*) des pixels de l'image d'*index* (l'image est de taille N pixels). La palette de couleurs \mathbf{C} , c'est-à-dire le message à insérer, est représentée par un vecteur binaire \mathbf{m} composé de $|\mathbf{m}|$ bits². Le *payload* d'insertion relatif, en *bit/pixel*, est donc $\alpha = |\mathbf{m}|/N$. Par conséquent, le *payload* α dépend uniquement de la taille N de l'image d'*index*.

L'image d'*index* est donc divisée en régions de taille $\lfloor 1/\alpha \rfloor$ pixels. Un Générateur de Nombre Pseudo-Aléatoire (GNPA) choisit aléatoirement, pour chaque région, *un seul pixel* i comme site d'insertion. Cette procédure de partitionnement garantit une répartition homogène du message sur toute l'image d'*index*. Le LSB du pixel i est alors modifié en fonction du bit $\mathbf{m}[j]$, $j \in \{1, \dots, |\mathbf{m}|\}$ à insérer³ :

$$Index_{\mathbf{m}}(i) = Index(i) - Index(i) \bmod 2 + \mathbf{m}[j],$$

avec $Index_{\mathbf{m}}(i)$ la valeur du pixel i après insertion.

Cette technique d'insertion de la palette de couleurs garantit que chaque pixel marqué est au pire modifié d'un niveau de gris. Ainsi, le pixel couleur reconstruit est toujours proche de sa couleur originale. En effet, le troisième terme de l'équation 8.1 impose que les couples de couleur soient faiblement éloignés.

8.3 Résultats

Nous avons appliqué la méthode sur des images couleur de taille 256×256 pixels. Pour toutes les expériences, $\lambda_1 = 1$, $\lambda_2 = 0.01 \times N/(K+1)$ et $\gamma = 2$ (voir équation 8.1). Les résultats obtenus montrent que l'approche est efficace quel que soit le type d'image (voir PSNR du tableau 8.1). Les principales étapes de notre approche sont commentées ci-dessous sur l'image *baboon*⁴.

Après avoir minimisé l'équation 8.1 sur l'image *baboon* avec $K = 256$ couleurs nous obtenons une image d'*index* (figure 8.1.b) et sa palette de couleurs (figure 8.1.c). L'image de luminance est donnée figure 8.1.a. L'image d'*index* et l'image de luminance sont visuellement similaires et la valeur de PSNR de 27.90 dB confirme ce sentiment. Notons que l'image d'*index* est bien meilleure que celle obtenue dans [Chaumont et al. 06] et représentée figure 8.1.f. (PSNR = 16.32 dB).

2. $|\mathbf{m}| = 256 \text{ couleurs} \times 3 \text{ composantes} \times 8 \text{ bits} = 6144 \text{ bits}$.

3. La formule est donnée pour des valeurs d'*index* appartenant à $[0, K-1]$.

4. Notons que les résultats obtenus sur l'image *baboon* donnent la borne inférieure en terme de qualité PSNR.

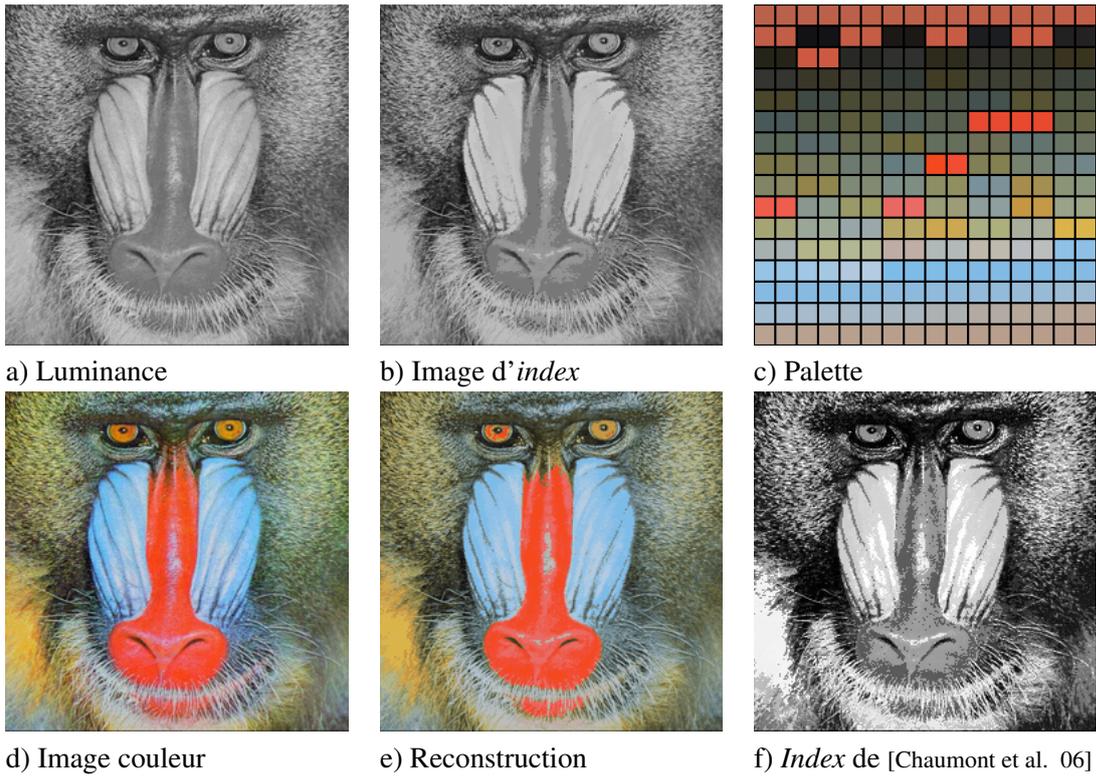


FIGURE 8.1 – Quelques étapes de la sécurisation de la couleur par l'approche par modélisation floue.

Dans la méthode proposée dans [Chaumont et al. 06], la première étape est une quantification vectorielle sur $K = 256$ couleurs. Une telle quantification tend à uniformiser la distribution des valeurs index ; l’histogramme de l’image d’index de [Chaumont et al. 06] est relativement plat (voir « Histogramme image Index avec k-mean » de la figure 8.2). La similarité avec l’histogramme de luminances est donc faible. Dans la méthode que nous proposons ici, l’intervalle de niveaux de gris et la forme de l’histogramme de l’image d’*index* (figure 8.2) sont plus proches de l’histogramme de luminances. On peut également remarquer sur l’histogramme d’*index* que de nombreuses valeurs d’index sont inutilisées ce qui se traduit par la présence de couleurs inutilisées dans la palette de couleurs (figure 8.1.c). Notons également sur la palette de couleurs de la figure 8.1.c que les couples de couleurs sont colorimétriquement proches grâce au troisième terme de l’équation 8.1. Il faut enfin préciser que le coût de calcul nécessaire à la minimisation de l’équation 8.1 est très élevé. Dans l’article [Chaumont et al. 07d] nous proposons de réduire ce coût en utilisant une version sous-échantillonnée de l’image. Le temps de calcul est effectivement moindre, mais en contrepartie les PSNRs des images sont plus faibles.

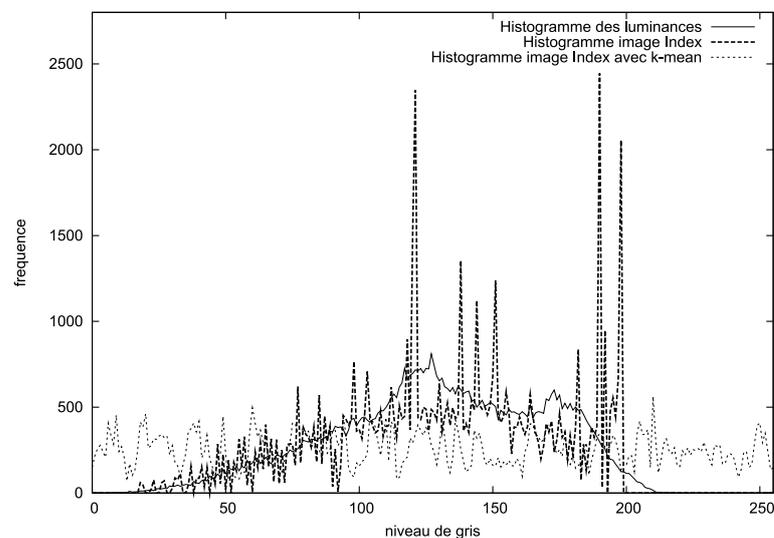


FIGURE 8.2 – Histogrammes.

La longueur du message inséré (la palette de couleurs) est de $|\mathbf{m}| = 6144$ bits ce qui donne un *payload* d’insertion pour une image de 256×256 pixels, de $\alpha = 6144 / (256 \times 256) = 0.093$ bit/pixel. L’image d’*index* est alors partitionnée en blocs de 10 pixels. Dans chaque bloc, un bit de la palette de couleurs est inséré à la position choisie par le GNPA comme expliqué dans la section 8.2. La sécurité est obtenue par l’utilisation d’une clé secrète de 128 bits comme graine pour le GNPA.

La figure 8.1.e représente l’image couleur reconstruite à partir de l’image d’*index* marquée. Cette image est visuellement proche de l’image couleur originale même si la valeur de PSNR de 27.90 dB est faible. Notons que la dégradation de l’image couleur quantifiée due à l’insertion de données cachées (dans l’image d’*index*) est faible car au pire une couleur est remplacée par une couleur colorimétriquement proche.

Les PSNRs sont donnés dans le tableau 8.1. Les images couleur reconstruites sont de qualité moyenne (plus de 27 dB), mais visuellement plaisantes. Par rapport à l’approche par substitution

de [Campisi et al. 02] (voir le tableau 9.1 au chapitre suivant), la qualité des images couleur reconstruites est parfois meilleure, parfois moins bonne. Par contre, l'image d'*index* est très proche de l'image de luminance. Nous verrons dans le chapitre suivant qu'en réduisant l'importance de la contrainte liée à la luminance, les images couleurs reconstruites sont de meilleure qualité.

Les PSNRs pour l'image d'*index* marqué sont supérieurs à 29 dB. Dans le cadre d'une application de sécurisation de la couleur par dissimulation dans une image en niveaux de gris, il peut être dommageable pour la sécurité que les images en niveaux de gris (c'est-à-dire les images d'*index* marquées) soient de bonne qualité. En effet, les images peuvent être « re-coloriées » de manière semi-automatique [Chaumont et al. 08b]. La colorisation peut être considérée comme une attaque indirecte dans le sens où il n'y a pas extraction de la palette de couleurs, mais simple estimation d'une palette menant à une image couleur agréable. Le chapitre suivant expose une solution réduisant énormément la qualité PSNR de l'image d'*index* marquée ce qui rend beaucoup plus difficile l'attaque par colorisation.

TABLE 8.1 – PSNR calculé entre l'image de luminance et l'image d'*index* marquée, et PSNR calculé entre l'image couleur originale (espace RGB) et l'image reconstruite (espace RGB) en utilisant l'image d'*index* marquée embarquant sa palette de couleurs.

images	PSNR ^{<i>luminance</i>} _(originale, <i>index</i>-marquée)	PSNR ^{<i>couleur</i>} _(originale, reconstruite)
baboon	29.74 dB	27.90 dB
airplane	35.95 dB	33.66 dB
pepper	35.03 dB	31.68 dB
house	35.40 dB	35.45 dB
barbara	34.86 dB	30.74 dB

8.4 Conclusion

Dans ce chapitre, nous proposons une méthode pour insérer de manière sécurisée dans une image en niveaux de gris ses informations couleur. Cette méthode repose sur la décomposition d'une image couleur en une image d'*index* et une palette de couleurs. L'image d'*index* joue le rôle de l'image de luminance et la palette de couleurs est cachée dans l'image d'*index*. La méthode est composée de deux étapes majeures qui sont : la décomposition de l'image couleur (en une image d'*index* et une palette de couleurs) et l'insertion de données cachées. L'originalité de l'approche est de modéliser le problème avec une fonction énergétique et ensuite de la minimiser. Les résultats obtenus montrent une réelle amélioration en comparaison de [Chaumont et al. 06]. La complexité de l'approche ainsi que sa faille de sécurité face à l'attaque par colorisation sont cependant problématiques. Le chapitre suivant propose une approche qui n'a pas ces inconvénients.

Annexe

Calcul des $\mathbf{P}_{i,k}$

Sachant que les valeurs d'appartenance doivent appartenir à l'intervalle $[0, 1]$ et que $\forall i, \sum_{k=1}^K \mathbf{P}_{i,k} = 1$, nous exprimons cette contrainte supplémentaire en ré-écrivant l'équation 8.1 :

$$E_{mod} = E + \lambda \sum_{i=1}^N \sum_{k=1}^K (1 - \mathbf{P}_{i,k}).$$

En annulant $\frac{\partial E_{mod}}{\partial \mathbf{P}_{i,k}}$ nous pouvons exprimer $\mathbf{P}_{i,k}$:

$$\mathbf{P}_{i,k} = \frac{\lambda}{2 \times (dist^2(\mathbf{C}(k), \mathbf{I}(i)) + \lambda_1(\mathbf{Y}(i) - k)^2)}. \quad (8.5)$$

Sachant que $\forall i, \sum_{k=1}^K \mathbf{P}_{i,k} = 1$, nous déduisons alors λ :

$$\lambda = \frac{1}{\sum_{l=1}^K \frac{1}{2 \times (dist^2(\mathbf{C}(l), \mathbf{I}(i)) + \lambda_1(\mathbf{Y}(i) - l)^2)}}.$$

L'équation 8.4 est alors obtenue en substituant λ dans l'équation 8.5.

Calcul des $\mathbf{C}(k)$

En annulant $\frac{\partial E}{\partial \mathbf{C}(k)}$, nous pouvons exprimer $\mathbf{C}(k)$ par un système linéaire donné à l'équation 8.3 ; La matrice \mathbf{A} et le vecteur \mathbf{B} du système linéaire $\mathbf{A} \cdot \mathbf{C} = \mathbf{B}$ sont donnés ci-dessous :

$$\mathbf{A} = \begin{pmatrix} \lambda_2 + \sum_{i=1}^{i=N} \mathbf{P}_{i,1}^\gamma & -\lambda_2 & 0 & \dots \\ -\lambda_2 & \lambda_2 + \sum_{i=1}^{i=N} \mathbf{P}_{i,2}^\gamma & 0 & \dots \\ 0 & 0 & \lambda_2 + \sum_{i=1}^{i=N} \mathbf{P}_{i,3}^\gamma & \dots \\ 0 & 0 & -\lambda_2 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \sum_{i=1}^{i=N} \mathbf{P}_{i,1}^\gamma I(i) \\ \sum_{i=1}^{i=N} \mathbf{P}_{i,2}^\gamma I(i) \\ \dots \\ \sum_{i=1}^{i=N} \mathbf{P}_{i,K-1}^\gamma I(i) \\ \sum_{i=1}^{i=N} \mathbf{P}_{i,K}^\gamma I(i) \end{pmatrix}.$$

Chapitre 9

Une approche par heuristique étendue au cas 512 couleurs

Résumé

Dans ce chapitre, nous proposons une méthode de dissimulation de la couleur dans une image en niveaux de gris reprenant le principe de réorganisation de palette [Chaumont et al. 07b] ainsi qu'une approche de tatouage réversible [Chaumont et al. 09a]. L'approche permet d'obtenir des images couleurs de grande qualité puisque lors de la quantification couleur nous utilisons une palette de 512 couleurs au lieu des 256 couleurs de [Chaumont et al. 07b] et [Chaumont et al. 07c]. La méthode est composée de deux étapes : la décomposition de l'image couleur en image d'*index* et palette de couleurs, et la dissimulation de la palette de couleurs et d'un plan de bits.

9.1 Introduction

Dans le chapitre précédent, nous avons évoqué deux approches permettant de protéger les informations couleur d'une image en dissimulant la palette de couleurs dans l'image d'*index*. Dans [Chaumont et al. 07b] nous trions les couleurs de la palette afin d'obtenir une image d'*index* proche de l'image de luminance. La palette obtenue est telle que les couleurs consécutives sont proches. Dans [Chaumont et al. 07c] l'image d'*index* et la palette sont obtenues grâce à l'optimisation d'une fonction modélisant le problème de la dissimulation de la palette dans une image d'*index*. Dans ces deux approches, la sécurité et la qualité de l'image couleur sont insuffisantes.

Dans ce chapitre nous proposons une solution qui améliore la qualité de l'image couleur et rend plus difficile l'attaque par colorisation [Chaumont et al. 08b]. Les points clés de l'approche sont :

- l'utilisation d'une palette de 512 couleurs ; les valeurs d'*index* appartiennent à $\{0, \dots, 511\}$; l'image d'*index* est donc codée sur 9 bits,
- la compression de la palette de couleurs et d'un plan de bits,
- l'utilisation d'un algorithme de tatouage réversible à fort *payload*,
- la dissimulation dans une image en niveaux de gris codée sur 8 bits.

Le schéma général d'insertion et d'extraction est donné à la figure 9.1. Les différentes étapes sont : la décomposition d'une image couleur dans une image d'*index* et une palette de couleurs (section 9.2.1 et 9.2.2), le codage de la palette de couleurs et d'un plan de bits (section 9.2.3), et le tatouage

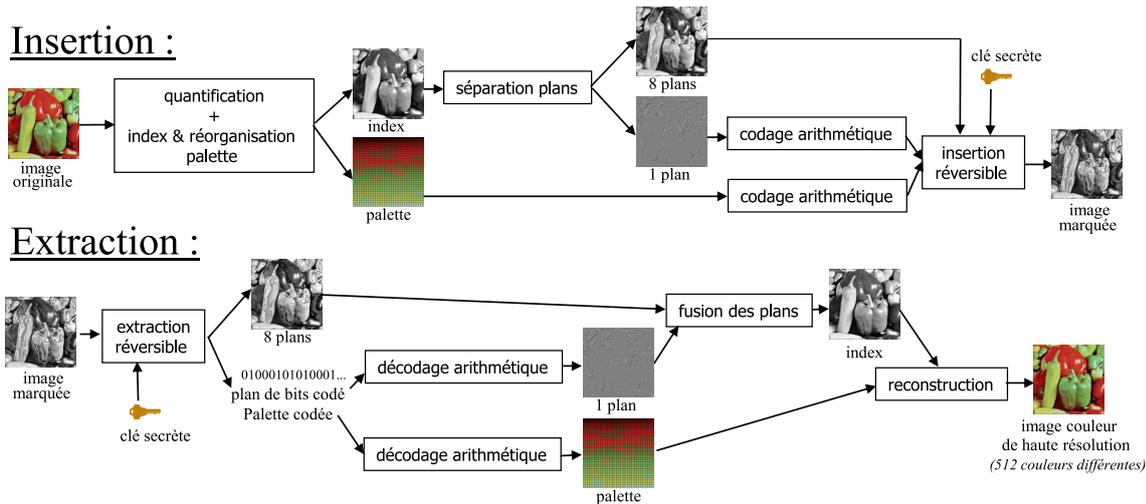


FIGURE 9.1 – Schéma général de l’insertion et de l’extraction pour l’approche de dissimulation par palette de 512 couleurs.

de l’image 8 bits construite à partir de l’image d’*index* (section 9.3). Le message à dissimuler dans l’image en niveaux de gris 8 bits est composé de la palette de couleurs et d’un plan de bits. L’image tatouée peut alors être stockée sur un site web public. Le schéma d’extraction est illustré figure 9.1. Ce schéma est l’exact inverse du schéma d’insertion.

9.2 La décomposition de l’image

9.2.1 La quantification couleur

La réduction du nombre de couleurs d’une image couleur est un problème de quantification. La solution optimale pour extraire les K couleurs est obtenue en résolvant :

$$\{\mathbf{P}_{i,k}, \mathbf{C}(k)\} = \arg \min_{\{\mathbf{P}_{i,k}, \mathbf{C}(k)\}} \sum_{i=1}^N \sum_{k=1}^K \mathbf{P}_{i,k} \cdot \text{dist}^2(\mathbf{I}(i), \mathbf{C}(k)), \quad (9.1)$$

$$\text{and } \forall i, \exists! k', [\mathbf{P}_{i,k'} = 1 \text{ and } \forall k \neq k', \mathbf{P}_{i,k} = 0],$$

où \mathbf{I} est une image couleur de dimension N pixels, $\mathbf{C}(k)$ est la k -ième couleur parmi les K couleurs recherchées, dist est une fonction de distance dans l’espace couleur (L2 dans l’espace couleur RGB) et $\mathbf{P}_{i,k} \in \{0, 1\}$ est la valeur d’appartenance du pixel i à la couleur k .

Une solution bien connue pour minimiser l’équation (9.1) et ensuite obtenir les K couleurs, est d’utiliser l’algorithme ISODATA des k -moyens [Ball et al. 66]. $\mathbf{P}_{i,k}$ est définie telle que :

$$\forall i, \forall k, \mathbf{P}_{i,k} = \begin{cases} 1 & \text{si } k = \arg \{ \min_{k' \in \{1, \dots, K\}} \text{dist}(\mathbf{I}(i), \mathbf{C}(k')) \}, \\ 0 & \text{sinon,} \end{cases}$$

$$\text{avec } \mathbf{C}(k) = \frac{\sum_{i=1}^N \mathbf{P}_{i,k} \times \mathbf{I}(i)}{\sum_{i=1}^N \mathbf{P}_{i,k}}.$$

Dans notre approche le nombre K est significatif ($K = 512$). Si nous utilisons l'algorithme classique des k-moyens, le nombre de couleurs extrait sera souvent en dessous de K . C'est le problème bien connu de « classes mortes ». De plus, l'algorithme des k-moyens est assez long en temps processeur par rapport aux approches non optimales mais rapides telles que la *quantification couleur par octree* de Gervautz et Purgathofer [Gervautz et al. 90], l'approche par *coupe médiane* de [Heckbert 82] etc. Pour pallier ces deux problèmes (« classes mortes » et complexité calculatoire), nous utilisons l'algorithme de *quantification couleur par octree* comme une initialisation à l'algorithme des k-moyens. Les valeurs de $\mathbf{P}_{i,k}$ sont initialisés à partir du résultat obtenu avec l'algorithme de *quantification couleur par octree*.

9.2.2 L'algorithme de parcours en couches

Une fois que la quantification couleur a été traitée, l'image à K couleurs obtenue peut être représentée par une image d'*index* (valeurs des $\mathbf{P}_{i,k}$) et une palette de couleurs (valeurs $\mathbf{C}(k)$). L'image d'*index* est notée *Index* et définie par :

$$\forall i \in \{1, \dots, N\}, \text{Index}(i) = \arg \max_{k \in \{1, \dots, K\}} \mathbf{P}_{i,k}.$$

Notre but est alors de résoudre deux contraintes ; la première contrainte est d'obtenir une image d'*index* où chaque niveau de gris est proche de la luminance de l'image couleur originale ; la deuxième contrainte consiste à obtenir une palette de couleurs dont les couleurs consécutives sont peu éloignées. Grâce à la quantification couleur, nous possédons déjà une image d'*index* et une palette de couleurs. Notre problème consiste alors à trouver une fonction de permutation qui permutent simultanément les valeurs de l'image d'*index* et les valeurs de la palette de couleurs. La fonction de permutation Φ est trouvée en résolvant l'équation :

$$\Phi = \arg \min_{\Phi} \sum_{i=1}^N \text{dist}^2(\mathbf{Y}(i), \Phi(\text{Index}(i))) + \lambda \sum_{k=1}^{K-1} \text{dist}^2(\mathbf{C}(\Phi^{-1}(k)), \mathbf{C}(\Phi^{-1}(k+1))), \quad (9.2)$$

où \mathbf{Y} est la luminance de l'image couleur originale et λ est la valeur du Lagrangien. La fonction de permutation Φ est une fonction bijective dans \mathbb{N} et définie telle que $\Phi : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$.

Nous approchons l'optimum de l'équation (9.2) en utilisant un algorithme heuristique : l'*algorithme de parcours en couches* [Chaumont et al. 07b]. Le but de cet algorithme est de trouver un ordonnancement pour les K couleurs tel que les couleurs consécutives soient peu éloignées et tel que la luminance des couleurs soient ordonnées des plus sombres aux plus claires. Cet ordonnancement définit pour chaque k -ième couleur une position k' qui nous donne la fonction Φ telle que $\Phi(k) = k'$.

Pour trouver un ordonnancement des K couleurs, l'algorithme parcourt l'espace des couleurs pour construire la suite ordonnée de couleurs. La figure 9.2 illustre le chemin obtenu après un parcours dans le cube RGB. Ce parcours est effectué en « sautant » de couleur en couleur, dans l'espace couleur, en choisissant la couleur la plus proche de la couleur courante. La première couleur de cette suite est choisie comme étant la couleur la plus sombre parmi les K couleurs. Une contrainte supplémentaire à ce parcours consiste à limiter la recherche de couleur aux couleurs peu éloignées en luminance. Cela signifie que le parcours dans l'espace des couleurs est limité à une fenêtre définie sur les informations de luminance. Cet *algorithme de parcours en couches* peut être vu comme une sorte « de parcours 3D en spirale » dans l'espace des couleurs.

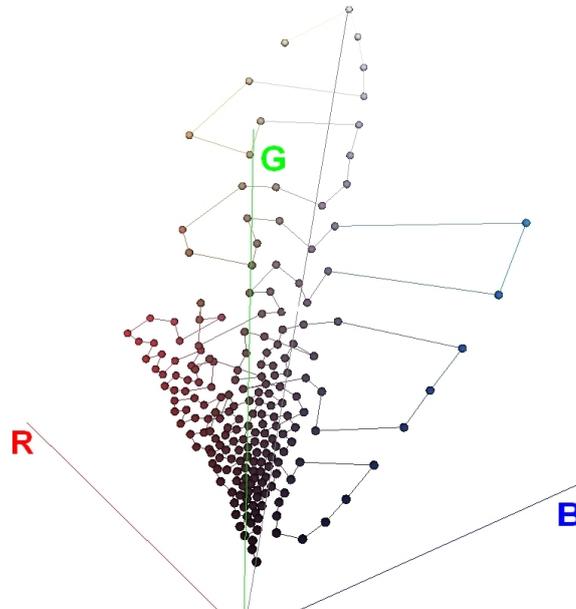


FIGURE 9.2 – Vue du parcours en couches dans le cube RGB.

Cet *algorithme de parcours en couches* possède un paramètre caché qui est la *taille de la couche* utilisée lors du parcours dans l'espace couleur. Notre but étant de minimiser l'équation 9.2, une manière satisfaisante pour régler automatiquement ce paramètre est de tester toutes les valeurs possibles de *taille de couche* et de garder la *taille de couche* minimisant l'équation 9.2. Sachant que les valeurs possibles de *taille de couche* appartiennent à $\{1, \dots, K\}$ et qu'il est très rapide de faire un parcours dans l'espace couleur, cette approche donne une solution élégante et rapide pour approximer l'équation 9.2.

Finalement, le paramètre λ de l'équation 9.2 est fixé à $\lambda = \alpha \times N / (3 \times (K - 1))$, avec α la valeur permettant de régler le compromis entre une image d'*index* proche de l'image de luminance, ou une palette de couleurs dont les couleurs sont proches deux à deux (voir équation 9.2). Par exemple, lorsque α est fixé à 1 la même importance est donnée aux deux contraintes, lorsque α est fixé à 0,5 cela signifie une image d'*index* plus proche de l'image de luminance, a contrario lorsque α est fixé à 2 cela implique une palette de couleurs plus continue.

Une fois que l'image d'*index* et la palette de couleurs ont été calculées et réorganisées, on effectue la dissimulation de données. La section suivante explique quel est le message à insérer, comment le mettre en forme et comment fonctionne l'algorithme de tatouage réversible.

9.2.3 Construction du message

Une fois que la décomposition de l'image couleur (en 512 couleurs) et son réarrangement (*algorithme de parcours en couches*) ont été effectués, un des plans de bits de l'image d'*index* est supprimé afin de créer une image de couverture codée sur 8 bits. Le plan de bits supprimé et la palette de couleurs sont ensuite compressés pour être insérés dans l'image de couverture (voir figure 9.1).

La solution retenue pour comprimer la palette de couleurs est de coder l'erreur de prédiction

(codage différentiel + codage arithmétique) de chaque couleur appartenant à la palette de couleurs. Pour comprimer le plan de bits, il est également nécessaire de passer par un codage d'erreur de prédiction afin d'obtenir un taux de compression élevé. Pour cela, le vecteur noté \mathbf{b} représentant le plan de bits est transformé en un vecteur d'erreur de prédiction, noté \mathbf{e} , tel que $\mathbf{e} \in \{-1, 0, 1\}^N$ avec N la taille de l'image, puis ce vecteur est codé par un codage entropique. Pour obtenir \mathbf{e} , nous procédons en deux étapes.

À la première étape, pour chaque pixel $Index(i)$ (codé sur 9 bits) à la position i , nous calculons la prédiction $Pred_{pixel}(i)$:

$$Pred_{pixel}(i) = \begin{cases} C & \text{si } |A - B| < |A - C| \\ B & \text{sinon,} \end{cases} \quad (9.3)$$

avec A, B et C les pixels voisins du pixel courant $Index(i)$ tel que :

ligne précédente	A	B
ligne suivante	C	$Index(i)$

Ce type de prédiction est utilisé dans l'approche « Differential Pulse Code Modulation » (DPCM). Cette technique prend essentiellement en compte les contours d'objets afin d'avoir une prédiction plus efficace qu'une simple moyenne locale. La prédiction est suffisamment efficace pour notre approche. Notons que les prédictions au bord de l'image sont calculées avec les valeurs disponibles.

La deuxième étape consiste à trouver pour chaque position i la meilleure prédiction du bit $\mathbf{b}(i)$ (provenant d'un des plans de l'image $Index$). Que cela soit du côté du codeur ou du côté du décodeur, pour cette position i , les informations des autres plans de bit sont connues. On a donc deux prédictions possibles sachant les autres plans : ou bien prédire un 0 ou bien un 1. Sachant que l'on a la valeur de $Pred_{pixel}(i)$, on doit décider quelle est l'hypothèse la plus probable entre avoir le bit $\mathbf{b}(i)$ à 0 (dans ce cas, la valeur du pixel est notée $Index_0(i) \in \{0, \dots, 511\}$) ou avoir le bit $\mathbf{b}(i)$ à 1 (dans ce cas, la valeur du pixel est notée $Index_1(i) \in \{0, \dots, 511\}$). L'hypothèse la plus probable (qui sera utilisée comme valeur de prédiction) est choisie en fonction du maximum entre les deux probabilités $prob(Pred_{pixel}(i)|Index_0(i))$ et $prob(Pred_{pixel}(i)|Index_1(i))$. Il s'agit d'un classique test à deux hypothèses et en supposant que les distributions de probabilité sont gaussiennes le meilleur choix est le suivant :

$$Pred_{bit}(i) = \begin{cases} 0 & \text{si } (Pred_{pixel}(i) - Index_0(i))^2 < (Pred_{pixel}(i) - Index_1(i))^2 \\ 1 & \text{sinon.} \end{cases} \quad (9.4)$$

Pour finir, le vecteur d'erreur de prédiction \mathbf{e} est calculé (les valeurs d'erreur sont soit -1, 0 ou 1) :

$$\mathbf{e}(i) = \mathbf{b}(i) - Pred_{bit}(i). \quad (9.5)$$

Le vecteur d'erreur de prédiction \mathbf{e} peut maintenant être codé et en raison du bon comportement de la prédiction, l'entropie est plus faible et le coût de codage petit.

9.3 Schéma de tatouage réversible

L'algorithme utilisé pour le tatouage réversible est l'un des plus efficaces en terme de quantité de bits insérés. Une brève description est donnée ci-dessous. L'algorithme est détaillé dans [Chaumont et al. 09a] et est une version améliorée de la proposition de Coltuc [Coltuc 07]. L'algorithme repose sur le calcul de congruences. Les propriétés de congruences permettent de définir **trois états possibles** pour un pixel :

- l'état *embarquant*, qui correspond à un pixel embarquant un coefficient entier appartenant à $\{1, \dots, n\}$ (section 9.3.1),
- l'état *à-corriger*, qui correspond à un pixel qui a été modifié, mais n'embarque pas d'informations ; ce pixel sera corrigé au cours du processus de réversion (section 9.3.2),
- l'état *original*, qui correspond à un pixel original, c'est-à-dire inchangé (section 9.3.3).

Soit une valeur constante entière n supérieure ou égale à 3. Définissons une transformation T qui prend deux entiers x_1 et x_2 en entrée et renvoie un entier :

$$T : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$T(x_1, x_2) = (n + 1).x_1 - n.x_2. \quad (9.6)$$

Dans la suite de ce chapitre, nous allons nommer « codage » le processus d'insertion d'un message dans une image et « décodage » le processus d'extraction du message et de reconstruction de l'image initiale. Nous allons maintenant définir les trois états possibles pour un pixel et les algorithmes de « codage » et « décodage ».

9.3.1 L'état *embarquant*

Un pixel i dans l'état *embarquant* est un pixel tel que :

$$0 \leq T(\mathbf{X}(i), \mathbf{X}(i + 1)) \text{ et } T(\mathbf{X}(i), \mathbf{X}(i + 1)) + n \leq L, \quad (9.7)$$

avec \mathbf{X} l'image d'*index* dont on a supprimé un plan de bits. Dans notre cas \mathbf{X} (de la taille N pixels) est codée sur 8 bits, les niveaux de gris appartiennent à $\{0, \dots, L\}$; donc L est égale à 255. Tous les pixels dans l'état *embarquant* :

1. sont transformés par la fonction T telle que $\mathbf{X}_T(i) = T(\mathbf{X}(i), \mathbf{X}(i + 1))$, avec $\mathbf{X}_T(i)$ le pixel résultant de la transformation,
2. **doivent** embarquer **un** coefficient $w \neq 0$ **appartenant à** $\{1, \dots, n\}$ tel que $\mathbf{X}_w(i) = \mathbf{X}_T(i) + w$ avec $\mathbf{X}_w(i)$ le pixel résultant de l'insertion.

Notons qu'après l'insertion du coefficient $w \neq 0$ appartenant à $\{1, \dots, n\}$, il est impossible de retrouver $\mathbf{X}(i)$ avec la seule connaissance de $\mathbf{X}(i + 1)$ et de n . En effet $\mathbf{X}_w(i) = (n + 1).\mathbf{X}(i) - n.\mathbf{X}(i + 1) + w$ avec $w \neq 0$, donc $\mathbf{X}(i) = \frac{\mathbf{X}_w(i) - n.\mathbf{X}(i + 1) - w}{n + 1}$. Il faut donc connaître w pour pouvoir calculer $\mathbf{X}(i)$. On a en particulier la propriété suivante :

$$(\mathbf{X}_w(i) + n.\mathbf{X}(i + 1)) \text{ mod } (n + 1) \neq 0. \quad (9.8)$$

La **propriété de congruence** de l'équation 9.8 permet la détection d'un pixel *embarquant* durant le « décodage ». Notons que durant le processus de « décodage », le pixel $\mathbf{X}_w(i + 1)$ doit avoir été préalablement rétabli à sa valeur initiale $\mathbf{X}(i + 1)$ afin de pouvoir calculer la congruence. Cela implique que l'ordre de parcours de l'image, utilisés lors du « décodage », soit l'opposé de l'ordre utilisé lors du « codage ».

9.3.2 L'état *à-corriger*

Un pixel i dans l'état *à-corriger* est un pixel tel que la négation de l'équation 9.7 est vrai c'est-à-dire tel que :

$$T(\mathbf{X}(i), \mathbf{X}(i + 1)) < 0 \text{ ou } T(\mathbf{X}(i), \mathbf{X}(i + 1)) + n > L. \quad (9.9)$$

Tous les pixels qui sont dans cet état *à-corriger* sont ensuite modifiés de telle sorte que :

$$\begin{aligned} c &= (\mathbf{X}(i) + n \cdot \mathbf{X}(i + 1)) \bmod (n + 1); \\ \text{si } (\mathbf{X}(i) - c) < 0 \text{ alors } c &= -(n + 1 - c); \\ \mathbf{X}_w(i) &= \mathbf{X}(i) - c. \end{aligned} \quad (9.10)$$

Les coefficients c appartiennent à $\{-n, \dots, n\}$ et sont insérés (dans les pixels *embarquant*) afin de permettre la réversibilité des pixels *à-corriger* durant le « décodage ». Nous appelons les coefficients c les **mots-de-correction**. Après la modification exprimée par l'équation 9.10, le pixel $\mathbf{X}_w(i)$ vérifie la propriété :

$$(\mathbf{X}_w(i) + n \cdot \mathbf{X}(i + 1)) \bmod (n + 1) = 0. \quad (9.11)$$

La **propriété de congruence** de l'équation 9.11 permet la détection d'un pixel *à-corriger* durant le « décodage ». Notons que durant le processus de « décodage », le pixel $\mathbf{X}_w(i + 1)$ doit avoir été préalablement rétabli à sa valeur initiale $\mathbf{X}(i + 1)$ afin de pouvoir calculer la congruence.

9.3.3 L'état original

Étant donné un ordre de parcours de l'image lors du « codage », un pixel dans l'état *original* (c'est-à-dire un pixel non modifié) doit toujours être présent juste avant un pixel dans l'état *à-corriger*. Pour un parcours de l'image allant de haut en bas et de gauche à droite, si un pixel à la position i est détecté dans l'état *à-corriger* (voir équation 9.9), alors le pixel à la position $i - 1$ est automatiquement désigné comme étant dans l'état *original* et ce pixel n'est pas modifié. Afin d'assurer cette contrainte forte (les pixels dans l'état *original* et dans l'état *à-corriger* vont par paires), lors du parcours de l'image, quand un pixel à la position i est détecté comme *à-corriger*, une recherche en avant est réalisée afin de trouver la prochaine position dans l'état *embarquant*. Notons cette position *next*. Les états *original* et *à-corriger* sont alors alternés entre la position i (ou $i - 1$) et la position *next* - 1.

Cette contrainte de groupement (construction de paires (*original* - *à-corriger*)) supprime les dépendances problématiques qui rendraient le « décodage » impossible. En effet, lors du « décodage », l'ordre de parcours de l'image est l'exact inverse de l'ordre utilisé lors du « codage ». Un pixel *à-corriger* à la position i ne peut donc pas être restauré immédiatement si son *mot-de-correction* associé n'a pas encore été extrait (voir équation 9.10). Néanmoins, puisque le pixel à la position $i - 1$ est un pixel dans l'état *original*, le pixel à la position $i - 2$ peut être traité immédiatement et le processus de « décodage » peut continuer (le pixel à la position i sera corrigé plus tard).

9.3.4 Les algorithmes de « codage » et de « décodage »

Le processus de « codage » est composé de deux étapes successives :

1. classer chaque pixel dans l'un des trois états suivants : *embarquant*, *à-corriger*, *original*,
2. insérer dans les pixels *embarquant*, le signal de tatouage composé des *mots-de-correction* et du message.

Lors du processus de « décodage », l'ordre de parcours est inversé. Le processus de « décodage » est également composé de deux étapes successives :

1. Extraire le signal de tatouage à partir des pixels *embarquant*, et dans le même temps, restaurer tous les pixels dans l'état *embarquant*, et localiser tous les pixels *à-corriger*,
2. À partir du signal de tatouage extrait, récupérer les *mots-de-correction* et le message, et corriger les pixels *à-corriger*.

Remarquons que la sécurité de l'insertion peut être assurée en rendant secret l'ordre de parcours. La clé secrète de l'utilisateur peut être utilisée comme graine d'initialisation d'un générateur de nombres pseudo-aléatoires. La séquence pseudo-aléatoire obtenue permet de générer un ordre de parcours adapté. Le signal de tatouage peut également être chiffré. Ainsi, aucune information sur le message (palette de couleurs + plan de bits) ne peut être extraite par un attaquant. En outre, l'attaque par colorisation [Chaumont et al. 08b], qui consiste à re-colorier semi-automatiquement l'image en niveaux de gris (une petite intervention humaine est nécessaire) est difficile, car l'image en niveaux de gris embarquant le message est de mauvaise qualité. Notons enfin que la technique de tatouage réversible n'est pas robuste et n'a pas besoin de l'être dans le cadre d'une application de sécurisation de la couleur.

9.4 Résultats

Nous avons appliqué l'« approche 512 couleurs » sur des images couleurs de taille de 256×256 pixels. Pour toutes les expériences, n est fixé à 4 pour le tatouage réversible. Les résultats obtenus montrent que l'approche est efficace quel que soit le type d'images. Les principales étapes de l'approche sont données pour l'image *peppers* sur la figure 9.3.

Après avoir réalisé la décomposition de l'image couleur *peppers* (figure 9.3.a), nous obtenons une image d'*index* dont les valeurs appartiennent à $\{0, \dots, 511\}$ (figure 9.3.e), et une palette de couleurs composée de $K = 512$ couleurs (Figure 9.3.c). L'image quantifiée, qui est reconstruite à partir de la connaissance de l'image d'*index* et de la palette de couleurs, est donnée figure 9.3.b. Le PSNR de cette image quantifiée est égal à 38,95 dB. Le meilleur PSNR obtenu avec une approche basée palette est de 36,32 dB avec l'approche par ré-ordonnancement de [Chaumont et al. 07b]. Le **gain** en qualité obtenu avec l'approche à 512 couleurs est de plus de 2 dB pour l'image couleur reconstruite.

L'image de luminance de l'image couleur d'origine est donnée figure 9.3.d. On peut observer la grande similarité visuelle entre l'image de luminance et de l'image d'*index*. L'image d'*index* est plus contrastée, mais conserve son intelligibilité sémantique.

Une fois la décomposition de l'image couleur faite, l'image d'*index* est décomposée en une image 8 bits (Figure 9.3.f) et une image binaire formée d'un seul plan de bits. Pour l'image *peppers* c'est le huitième plan de bits qui est choisi pour former l'image binaire. L'image binaire est ensuite modifiée afin d'obtenir une image d'erreur de prédiction (image ternaire) de faible entropie (voir section 9.2.3). Cette image d'erreur de prédiction est illustrée à la figure 9.3.h. Sur la figure, les pixels gris correspondent à la valeur 0, les pixels noirs correspondent à la valeur -1 , et les pixels blancs correspondent à la valeur 1. L'image d'erreur de prédiction et la palette de couleurs sont ensuite encodés avec un codeur arithmétique [Said 03]. Le signal de tatouage inséré est alors composé de la concaténation des deux flux binaires résultant du codage entropique. La figure 9.3.i représente une carte donnant la localisation des sites d'insertion (pixels blancs) avec l'approche de tatouage réversible de [Chaumont et al. 09a]. La figure 9.3.g donne l'image finale 8 bits intégrant la palette de couleurs (de 512 couleurs) et le huitième plan de bits de l'image d'*index*.

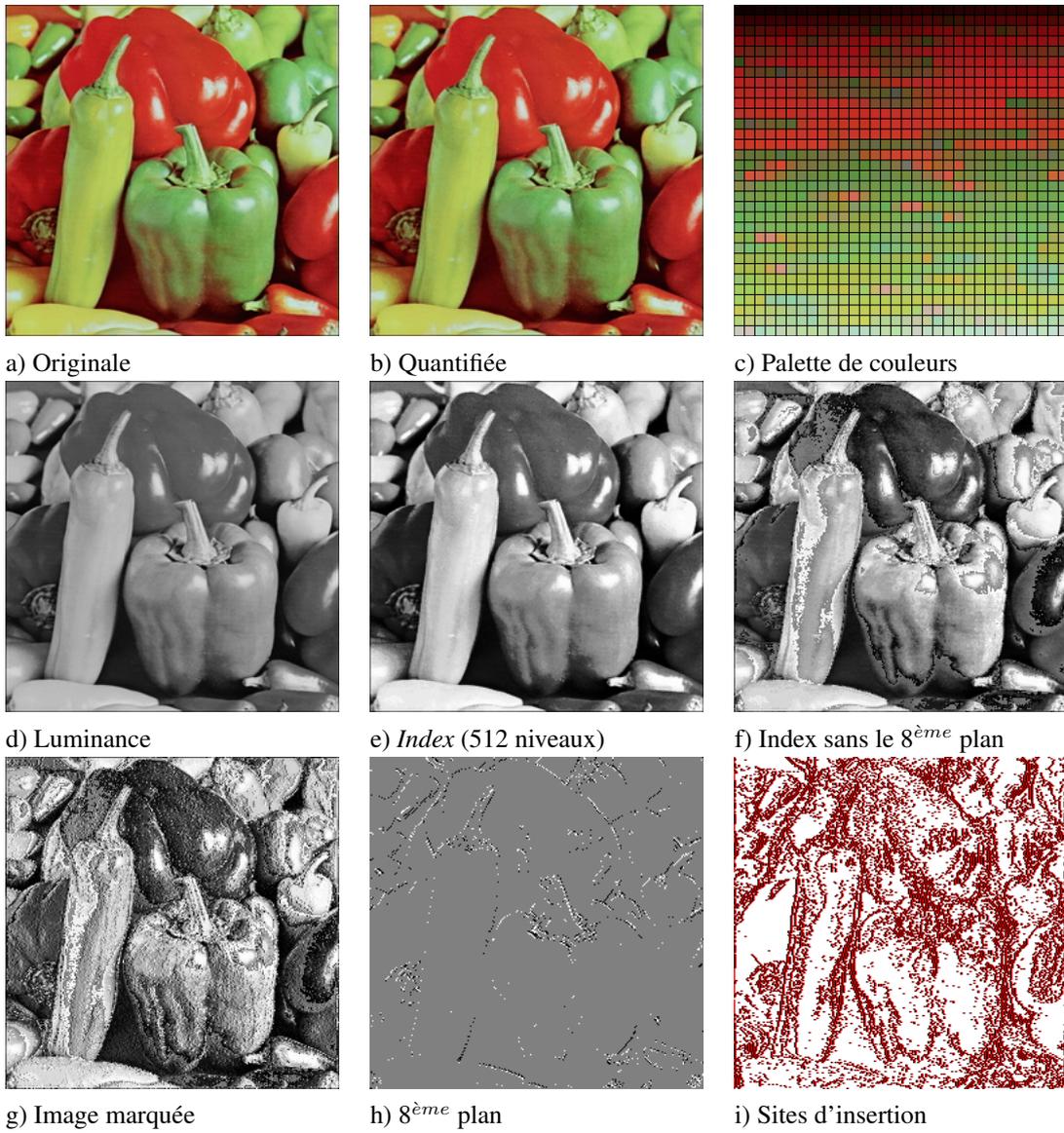


FIGURE 9.3 – Étapes de l'approche 512 couleurs.

La qualité de l'image tatouée (voir figure 9.3.g) est mauvaise, mais comme nous l'avons précédemment expliqué c'est une bonne propriété. En effet, l'image tatouée est plus difficile à « attaquer » par une attaque par coloriage [Chaumont et al. 08b]. Au contraire, l'image couleur reconstruite, sachant la clé secrète, donne une image couleur de très bonne qualité (PSNR = 38,95 dB).

L'approche à 512 couleurs est également illustrée pour l'image *barbara* 315×230 sur la figure 9.4. La figure 9.4.a représente l'image originale, la figure 9.4.c correspond à l'image tatouée et la figure 9.4.b est l'image reconstruite (c'est-à-dire quantifiée) avec un PSNR de 38,75 dB. Notons que le contenu sémantique de l'image tatouée est préservé.

Le tableau 9.1 donne sur quelques images les PSNRs obtenus entre l'image couleur d'origine et l'image reconstruite pour différentes approches. Pour toutes les images, l'approche basée palette à 512 couleurs est supérieure de 2dB à la deuxième meilleure approche de [Chaumont et al. 07b].



FIGURE 9.4 – Exemple sur l'image *barbara*.

TABLE 9.1 – Comparaison des PSNRs calculés entre l'image couleur originale et l'image couleur reconstruite pour différentes approches.

Images	Queiroz et Braun	Campisi <i>et al.</i>	Flou	Réarrangement	512 couleurs
	[Queiroz et al. 06]	[Campisi et al. 02]	[Chaumont et al. 07c]	[Chaumont et al. 07b]	[Chaumont et al. 08a]
baboon	23.93 dB	29.8 dB	27.90 dB	33.31 dB	35.86 dB
peppers	28.82 dB	32.36 dB	31.68 dB	36.32 dB	38.95 dB
lena	30.31 dB	36.75 dB	37.87 dB	38.63 dB	40.93 dB
house	30.75 dB	31.76 dB	35.45 dB	39.27 dB	41.67 dB
airplane	28.56 dB	32.58 dB	33.66 dB	39.90 dB	42.96 dB

9.5 Conclusion

Dans cette partie du manuscrit, nous avons évoqué différentes solutions afin de protéger les informations de couleur d'une image. Nous avons rappelé deux techniques utilisables pour la protection de la couleur [Queiroz et al. 06], [Campisi et al. 02]. Ces techniques sont basées sur des substitutions de sous-bandes ondelettes. Nous avons ensuite présenté une approche basée palette :

l'approche floue [Chaumont et al. 07c]. Nous avons enfin présenté une approche à 512 couleurs [Chaumont et al. 08a].

L'approche à 512 couleurs consiste à cacher une palette de 512 couleurs dans une image 8 bits en niveaux de gris. La méthode est basée sur une décomposition d'une image couleur dans une image d'*index* et une palette de couleurs de 512 couleurs. L'image d'*index* est alors divisée en une image 8 bits et une image binaire. L'image binaire et la palette de couleurs sont ensuite insérées de manière réversible dans l'image 8 bits. L'image tatouée résultante est encore sémantiquement compréhensible.

Les évaluations PSNR montrent que l'approche 512 couleurs améliore la qualité générale de l'image couleur reconstruite par rapport aux autres approches. En outre, la sécurité (confidentialité de la couleur) est renforcée par la réduction de la qualité de l'image en niveaux de gris. Enfin, la complexité calculatoire est faible.

En conclusion, il est possible de protéger la couleur d'une image en fournissant une image en niveaux de gris dégradée, mais sémantiquement compréhensible et dont l'image couleur reconstruite est de bonne qualité. Des améliorations sont sans doute possibles en mélangeant les différentes approches.

Conclusion générale

Le tatouage est une discipline jeune d'une vingtaine d'années. À ses débuts elle a très largement été financée par l'industrie de la musique et du cinéma pour produire des applications de protection des droits d'auteurs. Le tatouage était vu comme un instrument de sécurisation (droits d'auteur, suivi de transaction, authenticité, contrôle de copie). La formulation théorique de la robustesse et de la sécurité n'est pourtant apparue qu'à partir des années 2000. D'autre part, le tatouage (ou dissimulation de donnée) a aussi une dimension qui n'est pas liée à la sécurité et où la robustesse n'est pas toujours nécessaire. Le tatouage est alors généralement utilisé pour enrichir le média en y ajoutant des méta-données.

Le tatouage robuste

Durant les cinq dernières années, le tatouage robuste a été bien mieux compris et en particulier à travers deux compétitions de cassage de schéma de tatouage BOWS [Piva et al. 07] et BOWS-2 [Bas et al. 08b]. De plus, les définitions liées à la sécurité ont été formalisées et des attaques à la sécurité ont été proposées. Dans cet esprit de compréhension des mécanismes de tatouage robuste et sûr, nous avons proposé une approche de tatouage sûr par treillis ainsi qu'une approche de tatouage par quantification. L'objectif était d'étudier les mécanismes de tatouage informé multibits à forte capacité, et ceci, dans des conditions d'utilisation réalistes (dégradation réaliste de la qualité, faible complexité calculatoire, utilisation d'attaque classique). Notre souhait était de déterminer laquelle de ces deux familles de tatouage était la plus performante en pratique.

C'est cette question que nous avons abordé dans la première partie du manuscrit, où nous avons traité du tatouage robuste. Nous avons présenté les principes de tatouage informé, ainsi que les deux principales familles : le tatouage basé quantification et le tatouage basé treillis. Dans le chapitre 2, nous avons proposé une approche par rotation pour le tatouage par code à papier sale basé treillis. Dans le chapitre 3, nous avons intégré une approche par TCQ au sein d'un système de tatouage basé quantification. Dans les deux propositions, le système de codage est à fort payload et prend en compte les aspects psychovisuels.

La robustesse de ces schémas, lorsque le payload est très élevé, est sensiblement la même. Les schémas ne résistent qu'à des attaques de très faible puissance. Le schéma basé quantification est cependant le moins complexe en coût de calcul. Il est évident que le payload sur lequel nous avons expérimenté est trop élevé pour des applications pratiques où la robustesse doit être bien plus élevée. Ces approches peuvent cependant être adaptées pour des payload bien plus faibles et donc permettent d'obtenir des schémas utilisables en pratique. Il faut également noter que ces schémas ne sont pas sûrs en terme de sécurité (au sens de Kerckhoffs [Kerckhoffs 83]), [Bas et al. 08a], [PérezFreire et al. 07]. De ce point de vue, l'approche par treillis reste plus sûre puisque l'attaque

proposée [Bas et al. 08a] repose sur une version simplifiée de l'algorithme de tatouage et ne fonctionne pas sur la version non simplifiée. Quoiqu'il en soit, si l'on cherche des solutions robustes et de faibles payloads, les deux schémas sont tout à fait pertinents.

Le tatouage conjoint à la compression

Une des particularités que nous avons développée dans l'équipe ICAR est d'ajouter aux standards de compression (JPEG, JPEG2000, H.264, ...) des mécanismes de sécurisation ou d'enrichissement. Nous avons ainsi développé des mécanismes de dissimulation robuste et non robuste intégrés à la compression. L'objectif est principalement de s'intégrer finement à l'étape de codage et bien souvent de manière à obtenir une complexité de calcul faible. Par ailleurs, le tatouage peut prendre en compte la dégradation de l'étape de quantification et cela permet de réaliser un tatouage survivant à cette « attaque ».

Dans la deuxième partie du manuscrit nous avons abordé le tatouage intégré au sein d'un système de compression H.264 ou JPEG2000. Si le tatouage est effectué avant la compression, alors le schéma ne prend pas nécessairement en compte la dégradation due à la compression. Un mécanisme de tatouage intégré à l'étape de compression permet de prendre en compte la phase de dégradation du signal et donc rendre le signal de tatouage robuste à cette étape. De plus, dans un schéma conjoint à la compression, bien souvent la complexité calculatoire est plus faible.

Dans le chapitre 5 du manuscrit, nous présentons une approche utilisant la TCQ de JPEG2000 partie 2 pour effectuer la quantification et le tatouage simultanément. Pour que le schéma résiste à la phase de contrôle de taux, il faut sélectionner les coefficients et également préserver l'information de cheminement dans le treillis. Le schéma se comporte bien si l'on considère que c'est un simple schéma d'insertion de données cachées ne nécessitant pas de robustesse. Par contre, si l'on souhaite rendre le schéma robuste et également à détection aveugle (c'est-à-dire ne nécessitant pas d'information autre que la clé secrète lors de l'extraction) il est nécessaire de mettre en place des mécanismes de code correcteur gérant les effacements. Quoiqu'il en soit on constate qu'il est difficile d'intégrer un mécanisme de tatouage au sein d'un codeur car les contraintes structurelles du codeur peuvent gêner l'étape de tatouage.

On retrouve d'ailleurs ce même genre de constat dans la chapitre 6 du manuscrit où nous présentons une approche de tatouage intégrée à H.264, prenant en compte l'optimisation débit-distorsion et embarquant un code de Tardos. Le schéma se comporte bien face à des attaques par collusion. Beaucoup d'améliorations sont envisageables notamment en reprenant une insertion similaire à celle proposé dans Broken Arrows [Furon et al. 08]. Il est également nécessaire de trouver des codes de Tardos plus courts ou bien de trouver des mécanismes de traçage de traître nécessitant d'insérer peu de bits. Pour conclure, le schéma possède les avantages des meilleures approches de tatouage conjointe à la compression : ils sont simples, ils sont peu complexes calculatoirement, et ils fournissent une bonne robustesse. Cependant ce schéma possède également les défauts des approches de tatouage conjointe à la compression : les paramètres de quantification sont des informations adjacentes nécessaires lors de l'extraction, il est difficile d'assurer la robustesse aux attaques valométriques, il est difficile d'étendre le schéma à une approche robuste aux désynchronisations, enfin les schémas sont souvent plus vulnérables aux attaques malicieuses ou aux attaques à la sécurité que les schémas de tatouage dans les images.

La dissimulation de données

La dissimulation de données ne nécessite pas toujours des propriétés de robustesse : il existe de nombreuses applications de la dissimulation de données pour lesquelles il n'est pas important d'assurer qu'en cas de dégradation du média le message puisse toujours être extrait. On peut notamment citer les applications relevant de l'ajout de méta-data, du contrôle de périphériques, de l'analyse de flux, ou encore de l'amélioration des performances de compression et de transmission. Au cours du projet ANR Tsar de sécurisation de la base de données d'images du Louvre, nous avons développé une idée originale qui consistait à dissimuler l'information couleur d'une image au sein d'une représentation en niveaux de gris de cette même image. La dissimulation de la couleur via l'utilisation d'une palette de couleurs apporte une nouvelle façon de procéder. Les propositions et les idées que nous avons développées (notions d'attaque par coloriage, notion d'approche non robuste, mais dont le message est sécurisé, notion de décomposition en palette de couleur en vue du tatouage) ont apporté un autre regard sur ce domaine.

Dans la troisième et dernière partie du manuscrit, nous présentons cette approche de dissimulation de la couleur dans une image en niveaux de gris, pour laquelle la robustesse n'est pas du tout recherchée. Après avoir présenté les approches de dissimulation par substitution de sous-bandes ondelettes nous présentons deux approches utilisant la décomposition en palette de couleurs.

Nous proposons tout d'abord au chapitre 8 une formulation énergétique du problème de décomposition d'une image couleur en une image d'index et une palette couleur pour le problème de dissimulation. L'optimisation donne de bons résultats, mais la complexité calculatoire de l'approche est relativement élevée.

Nous présentons ensuite dans le chapitre 9 une approche empirique rapide permettant de réarranger une palette, préalablement générée par une approche par quantification. La palette que nous manipulons comporte 512 couleurs et nous la dissimulons dans une image en niveaux de gris par le biais d'un schéma de tatouage réversible. L'approche par palette de 512 couleurs donne de très bons résultats en comparaison de l'état de l'art. Le schéma de tatouage réversible ne fonctionne probablement pas sur toutes les images et dans les cas d'échec, il faudrait envisager une insertion non réversible (par LSB par exemple). Certains points peuvent être améliorés comme la phase de quantification couleurs, l'utilisation d'un domaine couleur plus adapté, ou encore l'utilisation de palettes annexes.

De manière générale, la dissimulation de données non robuste sert à l'enrichissement de médias. Les mécanismes d'insertion utilisent principalement de l'insertion basée LSB, des approches liées au traitement du signal et de l'image, et du codage de source et de canal. Chaque application est différente et le panel d'applications est vaste (voir tableau 1.1). Les applications qui sont le plus proches du tatouage dans le sens « communication fiable » sont celles qui cherchent à améliorer les performances de compression [Campisi et al. 02] [Thiesse et al. 10] et celles qui utilisent l'information dissimulée comme mécanismes de corrections d'erreurs après transmission [Adsumilli et al. 05]. Ce genre d'applications reste pour l'instant très peu étudié.

Projet de recherche

Le tatouage est une discipline qui a maintenant une vingtaine d'années ; ses bases théoriques ont été posées à la fin des années 90. Au cours de ces années, beaucoup de verrous théoriques ont été levés, comme la définition de la capacité théorique d'insertion, ou la définition de classes permettant de décider si un algorithme de tatouage est sûr. Les possibilités d'enrichissement des mécanismes déjà définis en tatouage peuvent être classées en quatre grands axes :

- La sécurité et le traçage de traîtres. L'insertion de données pour le traçage de traîtres est un volet du tatouage visant à dissimuler dans un media des informations identifiant l'acheteur du media, et permettant par la suite, si ce media se retrouvait redistribué illégalement, de retrouver l'auteur de cette distribution illégale. Une telle insertion de données est bien sûr extrêmement intéressante pour les vendeurs de média et les ayants-droits. Les enjeux scientifiques de cette problématique sont issus de la communauté de la cryptographie, et s'insèrent dans les problématiques de la théorie de l'information. Les enjeux ne sont pas dans les mécanismes de tatouage en eux-même mais dans les codes à ajouter aux médias.
- La robustesse aux désynchronisations. La robustesse aux désynchronisations (par exemple pour le cinéma ou les bases de données d'images) reste non résolue dans le cas général d'un tatouage à détection aveugle : il reste à découvrir des mécanismes de tatouage permettant de retrouver le message inséré dans un média qui a subi des déformations quelconques (transformation affine par morceaux, transformation perspective, warping, etc.), sans connaissance a priori de ces déformations. À la vue des récentes avancées en tatouage informé, ainsi qu'en sécurité, les schémas robustes aux désynchronisations, proposés autour de l'année 2000, peuvent être revus et étendus [Zheng et al. 07] [Chaumont 09c]. On peut penser que les espaces invariants aux désynchronisations [Schlauweg et al. 08] [Chen et al. 09] et l'intégration de codes correcteurs d'erreurs à gestion d'effacements [Solanki et al. 06], en particulier pour la vidéo, peuvent donner des résultats meilleurs que ce qui a été obtenu jusqu'ici. Les verrous majeurs à lever dans cette problématique relèvent ici plus du traitement du signal que des mécanismes d'insertion proprement dits.
- L'utilisation de codes. Les codes correcteurs peuvent être utilisés pour la gestion des effacements, c'est-à-dire pour récupérer les parties identifiées comme manquantes dans le message extrait. Ce mécanisme peut venir en soutien des systèmes de tatouage par exemple dans notre schéma de tatouage conjoint à JPEG2000 (présenté dans le chapitre 5). On peut également envisager de revoir la construction des codes de tatouage avec la vision « codes correcteurs », comme cela a été proposé pour l'algorithme DPTC (présenté dans le chapitre 2). Les approches turbo sont également un point d'amélioration des mécanismes de tatouage basé quantifications (présentés dans le chapitre 3). Cette problématique commence à être mise en place dans certains schémas de tatouage, et donne de bons résultats [Le Guelvouit 09]. La massification de cette utilisation permettra certainement d'augmenter les résultats de la ma-

porité des approches de tatouage, toutefois leur mise en œuvre relève plus d'application de mécanismes bien connus et de leur réglages fins ; cet aspect ne présente donc pas de verrou scientifique majeur.

- Les nouveaux supports et autres formats. Le tatouage est généralement beaucoup moins étudié (en termes de robustesse, de sécurité, ou de complexité) pour les nouveaux standards de compression ou bien dans des médiums différents de l'image, par exemple les modèles 3D [Wang et al. 11], ou le son [Cayre et al. 09]. Un travail d'extension vers ces nouveaux formats et standards reste donc à faire. Les mécanismes de tatouage existants pourront bien souvent être appliqués en les adaptant aux différents formats et aux différents supports.

Si le domaine du tatouage présente donc encore des enjeux importants (comme la sécurité et la robustesse), les solutions à apporter trouvent leurs sources dans d'autres domaines que celui du tatouage (e.g. la cryptographie ou le traitement du signal). En revanche, les mécanismes au cœur du tatouage sont maintenant bien installés, et peu de résultats fondamentaux restent à établir : les problèmes persistant dans les mécanismes d'insertion sont plus marginaux, et relèvent plus d'adaptations de pratiques existantes. On peut d'ailleurs noter ces dernières années dans les conférences majeures du domaine une quasi absence de solutions vraiment novatrices et fondamentales. En 2009, une table-ronde faisait suite au workshop IWDW (International Workshop on Digital Multimedia) dédié quasi exclusivement à la communauté du tatouage, et concluait à un essoufflement de la discipline.

Ce constat après cinq années de recherche sur le tatouage, m'a amené à m'intéresser au domaine de la stéganographie, porté par la même communauté de la sécurisation du multimédia. La stéganographie est l'étude de la communication secrète. Le tatouage et la stéganographie partagent le même objectif d'insertion d'un message, et à ce titre beaucoup de mécanismes communs, ainsi qu'une même culture.

En 2010, j'ai donc débuté une nouvelle activité de recherche en stéganographie, via l'encadrement de stagiaires sur ce domaine. En octobre 2010, la thèse de Sarra Kouider a débuté avec pour objectif la proposition de schémas stéganographiques sûrs, par l'étude de codes performants et l'invisibilité statistique. Nous avons mis en place un nouveau paradigme : la stéganographie par base de données d'images, et proposé l'algorithme de stéganographie adaptative par oracle (ASO : Adaptative Steganography by Oracle). Cet algorithme donne les résultats les plus sûrs de l'état de l'art en stéganographie d'image par modification.

Ces prochaines années, je concentrerai mes travaux sur la stéganographie, en développant trois axes principaux :

- à court terme, je m'intéresserai au sous-domaine de la stéganalyse, en travaillant à l'extension des hypothèses d'utilisation de la stéganographie vers des hypothèses plus réalistes que celles qui sont utilisées actuellement.
- à moyen terme, je mettrai à profit mes travaux sur la tatouage robuste en travaillant sur la stéganographie à gardien actif (ou gardien malicieux), dans laquelle le média stéganographié peut être altéré.
- à plus long terme, j'aimerais étudier l'utilisation de codes correcteurs pour l'insertion du message stéganographié.

Extension des scénarios de stéganalyse

La stéganographie/stéganalyse peut s'expliquer comme un jeu à trois participants. Les stéganographes, classiquement appelées Alice et Bob, veulent s'envoyer un message secret (une

suite binaire), sans que cela se sache. La stéganalyste, généralement appelée Eve, observe tous les échanges qui ont lieu entre Alice et Bob et peut stopper la communication entre Alice et Bob si elle découvre qu'il y a présence d'une communication secrète. Alice et Bob peuvent être des terroristes, des journalistes, des dissidents, des espions industriels, etc. Eve peut être une agence de sécurité nationale, un régime dictatorial, une société, etc.

Pour illustrer le propos, on peut imaginer qu'Alice et Bob communiquent par un réseau Internet, Intranet, GSM, etc. Tous les flux binaires envoyés d'Alice à Bob sont analysés par Eve (la stéganalyste). Si Eve détecte la présence de message secret alors elle peut décider de supprimer la communication. Alice et Bob doivent donc envoyer des messages à l'insu d'Eve. Alice et Bob ne peuvent pas s'envoyer de messages chiffrés sans qu'Eve ne soupçonne une communication secrète. Alice et Bob doivent donc communiquer à travers un canal secret en dissimulant le message secret dans un autre message qui lui n'est pas secret et qui n'est d'aucune importance pour Alice et Bob.

Alice et Bob vont donc utiliser des techniques stéganographiques pour dissimuler de manière statistiquement indécélable un message secret au sein d'un support anodin (une image, un son, une vidéo, ...). Eve va utiliser des techniques de stéganalyse pour détecter la présence d'un message secret au sein des supports qu'elle voit transiter sur le réseau. Dans ce jeu entre Alice et Bob (la stéganographie) d'un côté, et Eve (la stéganalyse) de l'autre, tout est une question de détection de présence de message caché et de communication secrète.

L'étude de la stéganographie/stéganalyse moderne a réellement débuté au début des années 2000. Actuellement, lorsque l'on effectue une stéganalyse, on définit un « scénario », c'est-à-dire un certain nombre d'hypothèses sur ce que la stéganalyste (Eve) connaît de l'environnement utilisé par le stéganographe (Alice et Bob). Nous développons ici, deux scénarios classiques pour de la stéganographie/stéganalyse d'images. Les principes peuvent être adaptés à d'autres supports numériques dégradables (vidéos, sons, etc).

Le scénario 1 « stéganalyse à clairvoyance » est similaire au scénario proposé par Kerckhoffs [Kerckhoffs 83] en cryptographie. La stéganalyste Eve connaît l'algorithme d'insertion et d'extraction du message, l'ensemble des paramètres publics de ces deux algorithmes (par exemple le nombre de bits insérés), et elle dispose d'une base d'images similaire à celle d'Alice et Bob. Le seul paramètre qu'Eve ne connaît pas est la clef secrète utilisée par Alice et Bob pour insérer/extraire le message dans les images. Actuellement, les approches de stéganalyse les plus performantes pour ce scénario consistent à utiliser des outils d'apprentissage comme les Support Vector Machine [Chang et al. 11] ou l'ensemble Classifier [Kodovský et al. 11].

Le scénario 2 « stéganalyse ciblée » est pratiquement le même que le scénario 1. La seule différence est que la stéganalyste ne connaît pas le nombre de bits insérés dans les images.

Ces deux scénarios font des hypothèses fortes quant aux informations connues par la stéganalyste. Il est donc naturel d'étudier des scénarios où la connaissance du stéganalyste est moindre. On peut par exemple énoncer deux scénarios plus réalistes en pratique :

- Le scénario 3 « stéganalyse universelle ». Dans ce scénario, la stéganalyste Eve ne connaît pas l'algorithme de stéganographie possiblement utilisé. Elle ne connaît pas non plus le nombre de bits inséré s'il y a insertion d'un message secret. Par contre, elle dispose d'une base d'images similaire à celle d'Alice. Il n'y a pas pour le moment de solution satisfaisante pour ce scénario [Pevný et al. 08], [Pevný 11].
- Le scénario 4 « stéganalyse universelle avec cover-source mismatch ». Ce scénario diffère du scénario 3 par le fait que la stéganalyste Eve dispose d'une base d'image qui est statistiquement différente de celle d'Alice. Si Eve utilise un algorithme de classification, qui

apprend avec sa propre base d'images, à distinguer les images ne contenant pas de message (images *couvertures*) des images contenant un message (images *stego*), alors son algorithme de classification se comportera de manière erronée lorsqu'il sera utilisé pour analyser les images échangées entre Alice et Bob, à cause de la non-correspondance entre les images couvertures d'Eve et celles d'Alice.

Ce sont ces scénarios et leurs variantes que j'étudierai à court terme.

Le scénario 3 « stéganalyse universelle » nécessite d'adapter les algorithmes d'apprentissage pour qu'ils soient capables de détecter l'utilisation par Alice et Bob, de nouveaux algorithmes jusque-là jamais observés par Eve. On parle de mécanismes de gestion de la nouveauté. Pour cela je compte intégrer aux classifieurs de l'état de l'art une fonctionnalité additionnelle pour gérer cette nouveauté en m'inspirant de la notion d'enveloppe que l'on retrouve dans les classifieurs *une-classe*.

Le scénario 4 « stéganalyse universelle avec cover-source mismatch » nécessite de s'intéresser au problème de non-correspondance entre les images « couverture » d'Eve et celles d'Alice. Une solution élégante consisterait à déterminer un espace, caractérisant les images, qui soit invariant aux types d'images utilisés. Les résultats préliminaires de [Gul et al. 11] ont montré que le filtrage d'image est une solution prometteuse. J'envisage de m'inspirer de l'apport de la criminalistique d'appareils photo pour attaquer le problème [Chen et al. 08], [Fridrich 09], [Cortiana et al. 11], [Kang et al. 11]. L'idée est de filtrer la base d'apprentissage, mais également les images à classer lors de la phase de stéganalyse, ceci dans l'objectif de définir un espace invariant à la distribution des images de « couverture ».

Ces scénarios visent à étudier des scénarios plausibles et dans certains cas plus « réalistes ». Un scénario extrêmement réaliste est celui de la « *pooled* stéganalyse universelle (et *batch* stéganographie) » où l'objectif est de construire un analyseur de trafic automatique qui analyse ce qui passe sur un réseau. Chaque utilisateur du réseau transmet des images et certains utilisateurs utilisent certaines images pour transmettre un message caché [Ker et al. 11]. Le stéganalyseur doit faire le tri pour détecter les utilisateurs suspects et par la suite entreprendre une stéganalyse plus poussée sur les images de ces suspects.

Ces quelques scénarios (scénario 3, 4, etc.) donnent une idée des recherches qu'il faudra mener en vue d'aller vers un stéganalyseur automatique.

Stéganographie à gardien actif

Le modèle de stéganographie couramment utilisé est celui proposé par Simmons en 1983 [Simmons 83]. Ce modèle considère que deux prisonniers enfermés dans des cellules séparées ont l'autorisation d'échanger des lettres sous la surveillance d'un gardien. Le gardien (la stéganalyste) essaye de déterminer si les prisonniers planifient une évasion. Le gardien peut analyser les lettres, modifier les lettres, falsifier frauduleusement les lettres. Les prisonniers (les stéganalyseurs) vont essayer de transmettre un message secret, à travers l'échange des lettres. Ce modèle présente trois types de comportements de la part du gardien : le comportement passif où le gardien ne modifie pas le support de communication, le comportement actif où le support est altéré, et le comportement malicieux où le gardien réussit à communiquer sur le canal secret.

Actuellement, la majeure partie des algorithmes de stéganographie par modification du support de couverture considèrent uniquement le scénario d'un gardien passif. Ainsi, un gardien prévoyant (un gardien actif) peut, dans la majeure partie des cas, empêcher que la communication secrète ait lieu, simplement en modifiant légèrement le support anodin échangé entre les deux prisonniers.

Si le support de couverture est une image et que l'algorithme stéganographique est un algorithme par modification du support alors le gardien peut, par exemple, changer le format de compression et également réencoder l'image avec un facteur de quantification légèrement plus grand. Dans ce scénario, aucun algorithme de stéganographie ne pourra assurer de communication secrète.

Les modèles théoriques de gardien actif ont été définis il y a environ cinq ans, mais il y a très peu de travaux sur le sujet. La stéganographie à gardien actif se rapproche des problématiques du tatouage robuste puisque le message doit résister à une dégradation du support. Par exemple, certains schémas de tatouage comme le natural watermarking [Bas et al. 06] essayent de préserver un modèle de distribution. Le modèle de distribution est bien entendu trop simpliste pour résister à la stéganalyse actuelle, mais l'idée de préserver un modèle tout en utilisant des codes différents de ce que l'on utilise en stéganographie à gardien passif est prometteuse. Ainsi, il est possible que cet axe de recherche puisse faire se rejoindre la discipline du tatouage et celle de la stéganographie. Je compte étudier cette problématique en abordant la littérature théorique en stéganographie puis en m'inspirant de mes connaissances en tatouage pour effectuer le lien entre les deux disciplines.

Code correcteurs

Je pense que l'aspect code et stéganographie nécessite une étude approfondie que je réaliserai à plus long terme.

Pour insérer un message dans une image, on recourt à la technique de codage matriciel. Cette approche a été introduite en stéganographie par Crandall [Crandall 98] en 1998. La première implémentation a été proposée par Westfeld avec l'algorithme de stéganographie F5 [Westfeld 01]. Le message est inséré au sein de l'image avec la contrainte de minimiser le nombre de coefficients (ou pixels) modifiés dans l'image. Pour ce faire, on a recours aux codes détecteurs et correcteurs d'erreurs en bloc. L'utilisation des codes correcteurs est cependant détournée de son utilisation usuelle. Du côté récepteur, c'est-à-dire du côté de Bob, Bob calcule les syndromes¹ de chaque bloc de coefficients à partir de la matrice de contrôle du code correcteur. Les syndromes correspondent au message qui est contenu dans l'image. Toute l'astuce consiste donc, du côté émetteur, c'est-à-dire du côté d'Alice, à modifier l'image de sorte que les syndromes calculés par Bob représentent le message et également de sorte que l'image soit la moins modifiée.

Depuis 2001, des codes plus performants ont été utilisés pour réaliser le codage matriciel : algorithme Modified Matrix Encoding : MME [Kim et al. 06], FastBCH [Zhang et al. 09], algorithme basé Reed-Solomon (RS) [Fontaine et al. 09], etc. L'insertion est toujours basée sur le calcul de syndrome, mais les codes utilisés possèdent une efficacité d'insertion meilleure que celle du code de Hamming utilisé dans F5, pour un même nombre de bits insérés.

Une évolution des codes matriciels a été également proposée à travers les « codes à papier mouillé » [Fridrich et al. 05] et consiste à sélectionner les sites d'insertion du côté émetteur, mais avec un récepteur ignorant les sites sélectionnés.

En 2010, Filler *et al.* [Filler et al. 10] ont proposé un code utilisant la notion de « stéganographie adaptative ». On suppose que l'émetteur dispose pour chaque pixel d'une information relative à l'impact sur la détectabilité de l'image si le pixel est modifié. L'émetteur possède donc une « carte de détectabilité » associant une valeur réelle sur l'intervalle $[0, \infty[$ pour chaque pixel ; plus la valeur est faible plus le pixel peut être modifié pour insérer un message. La proposition de Filler

1. Le syndrome d'un vecteur \mathbf{x} composé de coefficients appartenant à un corps fini (par exemple le corps des binaires) est le vecteur obtenu par le produit de \mathbf{x} et de la matrice de contrôle (transposée) du code associé.

consiste toujours à utiliser le principe de codage matriciel c'est-à-dire à transmettre le message secret à travers le syndrome, mais cette fois-ci en guidant le processus d'insertion du message dans l'image par la minimisation d'une métrique qui prend en compte la carte de détectabilité. Pour une même quantité de bits insérés, l'objectif n'est plus de minimiser le nombre de pixels modifiés, mais plutôt de modifier plus de pixels en choisissant les pixels impliquant une plus faible détectabilité. L'utilisation de ce code dans l'algorithme HUGO [Pevný et al. 10] a montré la supériorité d'un tel code (et du principe de stéganographie adaptative) en termes de sécurité (indétectabilité) par rapport à d'autres algorithmes de l'état de l'art.

Le code de [Filler et al. 10] est actuellement très performant. À même distorsion, la taille du message inséré dans une image est éloignée de 7% à 14% de la taille théorique simulée [Filler et al. 11]. Il est cependant envisageable de proposer des améliorations en s'inspirant des codes non binaires comme BCH [Zhang et al. 09], ou Reed Salomon [Fontaine et al. 09] ou bien en trouvant un lien avec les turbo codes. Par ailleurs, la borne théorique repose sur des hypothèses qui pourraient être révisées (hypothèse d'insertion ± 1 , hypothèse d'additivité de la carte de détectabilité), et donc influencer le mécanisme de codage.

Bien que le code de Filler soient très performants, les quelques indices listés ci-dessus laissent penser que ce point est prometteur. Je compte donc m'y intéresser dans les années à venir.

Fil conducteur : la caractérisation d'une image

La stéganalyse devra également faire face dans les années à venir à la notion de stéganographie par base que nous abordons dans la thèse de Sarra Kouider. En effet, avec cette approche un utilisateur non spécialiste peut rendre la distinction entre une image « couverture » et une image *stego* très difficile puisque le système choisit automatiquement les images *stego* dont les distributions ressemblent le plus à celle d'une image « couverture ». Que cela soit pour le stéganographe comme pour la stéganalyste, la difficulté est alors dans le choix des caractéristiques à préserver (du point de vue du stéganographe) ou à analyser (du point de vue du stéganalyste). On voit donc apparaître une course entre le stéganographe et la stéganalyste pour calculer des vecteurs de plus en plus grands de caractéristiques pertinentes. Une question qui devra être éclaircie par la communauté est de déterminer quelles sont les caractéristiques et dans quel ordre on doit les préserver. Il est également nécessaire d'évaluer si la course à la détermination de meilleures caractéristiques et donc également la sécurité n'est pas fonction de la complexité calculatoire mise en oeuvre. En fil conducteur de mes recherches en stéganographie, je chercherai à répondre à ces questions.

Bibliographie

- [Abrardo et al. 06] Abrardo (A.), Barni (M.), Perez-Gonzalez (F.) et Mosquera (C.). – Improving the Performance of RDM Watermarking by Means of Trellis Coded Quantisation. *IET Information Security, IEE Proceedings*, 153(3) :107–114, Septembre 2006.
- [Adsumilli et al. 05] Adsumilli (C.B.), Farias (M.C.Q.), Mitra (S.K.) et Carli (M.). – A Robust Error Concealment Technique Using Data Hiding for Image and Video Transmission Over Lossy Channels. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(11) :1394–1406, Novembre 2005.
- [Ball et al. 66] Ball (G. H.) et Hall (D. J.). – ISODATA, A novel Method of Data Analysis and Pattern Classification. – *Proceedings of the International Communication Conference*, Philadelphia, Juin 1966.
- [Bardyn et al. 09] Bardyn (D.), Dooms (A.), Dams (T.) et Schelkens (P.). – Comparative Study of Wavelet Based Lattice QIM Techniques and Robustness Against AWGN and JPEG Attacks. – Ho (Anthony T.S.), Shi (Yun Q.), Kim (H.J.) et Barni (Mauro) (édité par), *8th International Workshop on Digital Watermarking, IWDW'2009*, vol. 5703 of *Lecture Notes in Computer Science*, pp. 39–53, University of Surrey, Guildford, United Kingdom, Août 2009. Springer.
- [Bas et al. 06] Bas (P.) et Cayre (F.). – Achieving Subspace or Key Security for WOA using Natural or Circular Watermarking. – *ACM proceedings of the 8th workshop on Multimedia and security, MM&SEC'2006*, pp. 80–88, Geneva, Switzerland, Septembre 2006.
- [Bas et al. 08a] Bas (P.) et Doërr (G.). – Evaluation of an Optimal Watermark Tampering Attack Against Dirty Paper Trellis Schemes. – *10th ACM workshop on Multimedia and Security, MM&Sec'2008*, pp. 227–232, Oxford, United Kingdom, Septembre 2008.
- [Bas et al. 08b] Bas (P.) et Furon (T.). – BOWS-2 Contest (Break Our Watermarking System), 2008.
- [Bas et al. 09] Bas (P.) et Westfeld (A.). – Two Key Estimation Techniques for the Broken-Arrows Watermarking Scheme. – *11th ACM workshop on Multimedia and Security, MM&Sec'2009*, pp. 1–8, Princeton, USA, Septembre 2009.
- [Blayer et al. 08] Blayer (O.) et Tassa (T.). – Improved Versions of Tardos Fingerprinting Scheme. *Designs, Codes and Cryptography*, 48 :79–103, Juillet 2008.
- [Campisi et al. 02] Campisi (P.), Kundur (D.), Hatzinakos (D.) et Neri (A.). – Compressive Data Hiding : An Unconventional Approach for Improved Color Image Coding. *EURASIP Journal on Applied Signal Processing*, 2002(2) :152–163, 2002.
- [Cayre et al. 05] Cayre (F.), Fontaine (C.) et Furon (T.). – Watermarking Security : Theory and Practice. *IEEE Transactions on Signal Processing*, 53(10) :3976–3987, 2005. – special issue "Supplement on Secure Media III".

- [Cayre et al. 09] Cayre (F.), Baras (C.) et Medico (V. Del). – Compromis sécurité / robustesse en resynchronisation pour le tatouage. – *Compression et REprésentation des Signaux Audiovisuels, CORESA'2009*, Toulouse, France, Mars 2009.
- [Chang et al. 11] Chang (Chih-Chung) et Lin (Chih-Jen). – LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2 :27 :1–27 :27, 2011. – Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [Chaumont 09a] Chaumont (M.). – Fast Embedding Technique for Dirty Paper Trellis Watermarking. – Ho (A. T.S.), Shi (Y. Q.), Kim (H.J.) et Barni (M.) (édité par), *Digital Watermarking, 8th International Workshop, IWDW 2009, Guildford, UK, August 24-26, 2009. Proceedings*, vol. 5703 of *Lecture Notes in Computer Science*, pp. 110–120, University of Surrey, Guildford, United Kingdom, Août 2009. Springer.
- [Chaumont 09b] Chaumont (M.). – Psychovisual Rotation-based DPTC Watermarking Scheme. – *17th European Signal Processing Conference, EUSIPCO'2009*, Glasgow, Scotland, Août 2009.
- [Chaumont 09c] Chaumont (M.). – Tutorial état de l'art sur le "tatouage robustes aux désynchronisation", réunion du gdr-isis sur les nouvelles avancées en tatouage d'images, téléchargeable à l'adresse <http://www.lirmm.fr/~chaumont/tatouage.html>, Mars 2009.
- [Chaumont 10a] Chaumont (M.). – A Novel Embedding Technique For Dirty Paper Trellis Watermarking. – *Visual Information Processing and Communication, Part of IS&T/SPIE 22th Annual Symposium on Electronic Imaging, VPC'2010, SPIE'2010*, vol. 7543, San Jose, California, USA, Janvier 2010.
- [Chaumont 10b] Chaumont (M.). – *Invited Paper - Tutorial : H.264 Video Watermarking : Applications, Principles, Deadlocks, and Future.* – *International Conference on Image Processing Theory, Tools and Applications, IPTA'2010*, Paris, France, Juillet 2010.
- [Chaumont 11] Chaumont (M.). – *Invited Paper - Tutorial : Ensuring Security of H.264 Videos by Using Watermarking.* – *Mobile Multimedia/Image Processing, Security, and Applications, Part of SPIE Defense, Security, and Sensing, DSS'2011, SPIE'2011*, vol. 8063, Orlando, Florida, USA, Avril 2011.
- [Chaumont et al. 06] Chaumont (M.) et Puech (W.). – A Color Image in a Grey-Level Image. – *IS&T Third European Conference on Colour in Graphics, Imaging, and Vision, CGIV'2006*, pp. 226–231, Leeds, UK, Juin 2006.
- [Chaumont et al. 07a] Chaumont (M.) et Puech (W.). – 3D Face Model Tracking Based on a Multiresolution Active Search. – *IS&T/SPIE 19th Annual Symposium on Electronic Imaging, Visual Communications and Image Processing, VCIP'2007, SPIE'2007*, vol. 6508, San Jose, California, USA, Janvier 2007.
- [Chaumont et al. 07b] Chaumont (M.) et Puech (W.). – A Fast and Efficient Method to Protect Color Images. – *IS&T/SPIE 19th Annual Symposium on Electronic Imaging, Visual Communications and Image Processing, VCIP'2007, SPIE'2007*, vol. 6508, San Jose, California, USA, Janvier 2007.
- [Chaumont et al. 07c] Chaumont (M.) et Puech (W.). – A Grey-Level Image Embedding its Color Palette. – *IEEE International Conference on Image Processing, ICIP'2007*, vol. I, pp. 389–392, San Antonio, Texas, USA, Septembre 2007.

- [Chaumont et al. 07d] Chaumont (M.) et Puech (W.). – Fast Protection of the Color of High Dimension Digital Painting Images. – *Eighth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'2007*, pp. 60–63, Santorini, Greece, Juin 2007.
- [Chaumont et al. 08a] Chaumont (M.) et Puech (W.). – A 8-Bits-Grey-Level Image Embedding its 512 Color Palette. – *16th European Signal Processing Conference, EUSIPCO'2008*, Lausanne, Switzerland, Août 2008.
- [Chaumont et al. 08b] Chaumont (M.) et Puech (W.). – Attack By Colorization of a Grey-Level Image Hiding its Color Palette. – *IEEE International Conference on Multimedia & Expo, ICME'2008*, Hannover, Germany, Juin 2008.
- [Chaumont et al. 09a] Chaumont (M.) et Puech (W.). – A High Capacity Reversible Watermarking Scheme. – *IS&T/SPIE 21th Annual Symposium on Electronic Imaging, Visual Communications and Image Processing, VCIP'2009, SPIE'2009*, vol. 7257, San Jose, California, USA, Janvier 2009.
- [Chaumont et al. 09b] Chaumont (M.) et Puech (W.). – Protecting the Color Information by Hiding it. *Recent Advances in Signal Processing*, éd. par Zaher (Ashraf A), pp. 101–122. – InTech, 2009.
- [Chaumont et al. 11a] Chaumont (M.) et Goudia (D.). – TCQ Practical Evaluation in the Hyper-Cube Watermarking Framework. – *IEEE International Conference on Multimedia and Expo, ICME'2011*, Barcelona, Spain, Juillet 2011.
- [Chaumont et al. 11b] Chaumont (M.), Goudia (D.) et Puech (W.). – Hyper-Cube Watermarking Scheme. – *Visual Information Processing and Communication II, Part of IS&T/SPIE 23th Annual Symposium on Electronic Imaging, VIPC'2011, SPIE'2011*, vol. 7882, pp. 10–18, San Francisco, California, USA, Janvier 2011.
- [Chen et al. 01] Chen (B.) et Wornell (G.). – Quantization Index Modulation : A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE Transactions on Information Theory*, 47(4) :1423–1443, 2001.
- [Chen et al. 08] Chen (M.), Fridrich (J.), Goljan (M.) et Lukas (J.). – Determining Image Origin and Integrity Using Sensor Noise. *IEEE Transactions on Information Forensics and Security*, 3(1) :74–90, 2008.
- [Chen et al. 09] Chen (C.), Ni (J.) et Huang (J.). – Temporal Statistic Based Video Watermarking Scheme Robust against Geometric Attacks and Frame Dropping. – *8th International Workshop on Digital Watermarking, IWDW'2009*, pp. 81–95, Août 2009.
- [Chen et al. 10] Chen (J.), Chen (T. S.), Lin (C. N.) et Cheng (C. Y.). – A Bitrate Controlled Data Hiding Scheme for JPEG2000. *International Journal of Computers and Applications*, 2(32) :238–241, 2010.
- [Coltuc 07] Coltuc (D.). – Improved Capacity Reversible Watermarking. – *IEEE International Conference on Image Processing, ICIP'2007*, San Antonio, Texas, USA, Septembre 2007.
- [Cortiana et al. 11] Cortiana (A.), Conotter (V.), Boato (G.) et Natale (F. G. B. De). – Performance comparison of denoising filters for source camera identification. – *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7880, San Francisco, California, USA, Février 2011.
- [Costa 83] Costa (M.). – Writing on Dirty Paper. *IEEE Transactions on Information Theory*, 29(3) :439–441, 1983.

- [Cox et al. 97] Cox (I. J.), Kilian (J.), Leighton (T.) et Shamoon (T.). – Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, 6(12) :1673–1687, 1997.
- [Cox et al. 99] Cox (I. J.), Miller (M. L.) et McKellips (A. L.). – Watermarking as Communications with Side Information. *IEEE Special Issue on Identification and Protection of Multimedia Information*, 87 :1127–1141, Juillet 1999.
- [Cox et al. 07a] Cox (I.), Miller (M.), Bloom (J.), Fridrich (J.) et Kalker (T.). – *Digital Watermarking and Steganography*. – Morgan Kaufmann, Novembre 2007, 2nd édition.
- [Cox et al. 07b] Cox (I.), Miller (M.), Bloom (J.), Fridrich (J.) et Kalker (T.). – *Digital Watermarking and Steganography*, chap. 5, pp. 147–152. – Morgan Kaufmann, Novembre 2007, 2nd édition.
- [Crandall 98] Crandall (R.). – Notes on Steganography, 1998.
- [Craver et al. 07] Craver (S.), Atakli (I.) et Yua (J.). – How We Broke the BOWS Watermark. – *IS&T/SPIE 19th Annual Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX edited by Edward J. Delp III, Ping Wah Wong, SPIE'2007*, vol. 6505, pp. 1–8, San Jose, California, USA, Janvier 2007.
- [Cérou et al. 08] Cérou (F.), Furon (T.) et Guyader (A.). – Experimental Assessment of the Reliability for Watermarking and Fingerprinting Schemes. *EURASIP Journal on Information Security*, pp. 1–12, 2008.
- [Daubechies et al. 98] Daubechies (I.) et Sweldens (W.). – Factoring Wavelet Transforms into Lifting Steps. *Journal of Fourier Analysis Applications*, 4(3) :247–269, 1998.
- [Deguillaume et al. 99] Deguillaume (F.), Csurka (G.), O'Ruanaidh (J.) et Pun (T.). – Robust 3D DFT Video Watermarking. – *IS&T/SPIE Security and Watermarking of Multimedia Contents, SPIE'1999*, vol. 3657, San Jose, California, USA, Janvier 1999.
- [Doërr 05] Doërr (G.). – *Security Issue and Collusion Attacks in Video Watermarking*. – PhD. Thesis, University of Nice-Sophia Antipolis, France, Juin 2005. Supervisé par J.-L. Dugelay.
- [Dunn 74] Dunn (J. C.). – A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3 :32–57, 1974.
- [Eggers et al. 03] Eggers (J. J.), Bäuml (R.), Tzschoppe (R.) et Girod (B.). – Scalar Costa Scheme for Information Embedding. *IEEE Transactions on Signal Processing*, 51(4) :1003–1019, 2003.
- [Filler et al. 10] Filler (T.), Judas (J.) et Fridrich (J.). – Minimizing Embedding Impact in Steganography using Trellis-Coded Quantization. – *Media Watermarking, Security, and Forensics XII, Part of IS&T/SPIE 20th Annual Symposium on Electronic Imaging, SPIE'2010*, vol. 7541, San Francisco, California, USA, Janvier 2010.
- [Filler et al. 11] Filler (T.) et and (J. Fridrich). – Design of Adaptive Steganographic Schemes for Digital Images. – *Media Watermarking, Security, and Forensics XIII, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7880, San Francisco, California, USA, Février 2011.
- [Fontaine et al. 09] Fontaine (C.) et Galand (F.). – How Reed-Solomon Codes Can Improve Steganographic Schemes. *EURASIP Journal on Information Security*, 2009 :10, 2009.
- [Fridrich 99] Fridrich (J.). – A New Steganographic Method for Palette-Based Images. – *Proceedings of the IS&T PICS conference*, Savannah Georgia, Avril 1999.

- [Fridrich 09] Fridrich (J.). – Digital Image Forensic Using Sensor Noise. *IEEE Signal Processing Magazine*, 26(2) :26–37, 2009.
- [Fridrich et al. 05] Fridrich (J.), Goljan (M.) et Soukal (D.). – Efficient wet paper codes. – *Proceedings of the 7th international conference on Information Hiding*, Information Hiding 2005, pp. 204–218, Berlin, Heidelberg, Juin 2005. Springer-Verlag.
- [Furon 09] Furon (T.). – Le traçage de traîtres. – *7ème symposium sur la sécurité des technologies de l’information et des communications, SSTIC’09*, vol. 2009, pp. 283–296, Rennes, France, Juin 2009.
- [Furon et al. 08] Furon (T.) et Bas (P.). – Broken Arrows. *EURASIP Journal on Information Security*, 2008 :3 :1–3 :13, 2008.
- [Gervautz et al. 90] Gervautz (M.) et Purgathofer (W.). – A Simple Method for Color Quantization : Octree Quantization. *Graphics Gems, A.S. Glassner*, pp. 287–293, 1990.
- [Golikeri et al. 07] Golikeri (A.), Nasiopoulos (P.) et Wang (Z. J.). – Robust Digital Video Watermarking Scheme for H.264 Advanced Video Coding Standard. *Journal of Electronic Imaging*, 16(4), 2007.
- [Gong et al. 08] Gong (X.) et Lu (H.-M.). – Towards Fast and Robust Watermarking Scheme for H.264 Video. – *Proceedings of the 2008 Tenth IEEE International Symposium on Multimedia, ISM ’08*, pp. 649–653, Washington, DC, USA, 2008. IEEE Computer Society.
- [Goudia 11] Goudia (Dalila). – *Tatouage conjoint à la compression d’images fixes avec Jpeg2000*. – PhD. Thesis, Université de Montpellier, France, Décembre 2011. Supervisé par W. Puech et M. Chaumont.
- [Goudia et al. 11a] Goudia (D.), Chaumont (M.), Puech (W.) et Said (N. Hadj). – A Joint Trellis Coded Quantization (TCQ) Data Hiding Scheme in the JPEG2000 Part 2 Coding Framework. – *The 19th European Signal Processing Conference, EUSIPCO’2011*, Barcelona, Spain, Septembre 2011.
- [Goudia et al. 11b] Goudia (D.), Chaumont (M.), Puech (W.) et Said (N. Hadj). – Joint Trellis Coded Quantization Watermarking for JPEG2000 images. *Annals of telecommunications*, pp. 1–15, Décembre 2011.
- [Goudia et al. 12] Goudia (D.), Chaumont (M.), Puech (W.) et Said (N. Hadj). – Quantization Watermarking for Joint Compression and Data Hiding Schemes. *Watermarking Book 1*, éd. par Gupta (Dr. Mithun Das), p. 26. – InTech, Mars 2012.
- [Gul et al. 11] Gul (G.) et Kurugollu (F.). – A New Methodology in Steganalysis : Breaking Highly Undetectable Steganography (HUGO). – *Information Hiding, 13th International Conference, IH’2011*, vol. 6958 of *Lecture Notes in Computer Science*, pp. 71–84, Prague, Czech Republic, Mai 2011. Springer.
- [H2603] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T REC. H.264 ISO/IEC 14496-10 AVC)*. – Rapport de recherche, Joint Video Team (JVT), 2003.
- [Haitsma et al. 01] Haitsma (L.) et Kalker (T.). – A Watermarking Scheme for Digital Cinema. – *IEEE International Conference on Image Processing, ICIP’2001*, vol. 1, pp. 587–489, Thessaloniki, Greece, Octobre 2001.
- [Hartung et al. 98] Hartung (F.) et Girod (B.). – Watermarking of Uncompressed and Compressed Video. *Signal Processing*, 66 :283–301, May 1998.

- [Heckbert 82] Heckbert (P.). – Color Image Quantization for Frame Buffer Display. *Computer Graphics*, 16(3) :297–303, 1982.
- [Horiuchi et al. 10] Horiuchi (T.), Noharaa (F.) et Tominaga (S.). – Accurate Reversible Color-to-Gray Mapping Algorithm Without Distortion Conditions. *Pattern Recognition Letters*, 31(15), Novembre 2010.
- [JPE91] *ISO/IEC IS 10918-1 | ITU-T Recommendation T.81*. – Rapport de recherche, ISO and ITU-T, 1991.
- [JPE00] *ISO/IEC JTC1/SC29 WG1 JPEG 2000 Image Coding System Final Committee Draft Version 1.0*. – Rapport de recherche, ISO and ITU-T, 2000.
- [JPE04] *ISO/IEC 15444-2 :2004 Information technology – JPEG 2000 image coding system : Extensions*. – Rapport de recherche, ISO and ITU-T, 2004.
- [Kang et al. 11] Kang (X.), Li (Y.), Qu (Z.) et Huang (J.). – Enhancing ROC performance of trustworthy camera source identification. – *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7880, San Francisco, California, USA, Février 2011.
- [Ker et al. 11] Ker (A. D.) et Pevný (T.). – A New Paradigm for Steganalysis via Clustering. – *Media Watermarking, Security, and Forensics XIII, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7880, pp. 0U01–0U13, San Francisco, California, USA, Février 2011.
- [Kerckhoffs 83] Kerckhoffs (A.). – La Cryptographie Militaire. *Journal des Sciences Militaires*, IX, pp. 5-38 Jan. 1883, pp. 161-191, Feb. 1883.
- [Kim et al. 06] Kim (Y.), Duric (Z.) et Richards (D.). – Modified Matrix Encoding Technique for Minimal Distortion Steganography. – *Information Hiding*, vol. 4437 of *Lecture Notes in Computer Science*, pp. 314–327. Springer, 2006.
- [Ko et al. 08] Ko (K.-W.), Kwon (O.-S.), Son (C.-H.) et Ha (Y.-H.). – Color Embedding and Recovery Based on Wavelet Packet Transform. *Journal of Imaging Science and Technology*, 52(1), Février 2008.
- [Kodovský et al. 11] Kodovský (J.) et Fridrich (J.). – Steganalysis in high dimensions : fusing classifiers built on random subspaces. – *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7880, San Francisco, California, USA, Février 2011.
- [KTA11] KTA (Key Technical Areas) Software, téléchargeable à l'adresse <http://www.tnt.uni-hannover.de/~vatis/kta/> et à l'adresse <http://iphome.hhi.de/suehring/tml/>, 2011.
- [Le Guelvouit 09] Le Guelvouit (G.). – Tatouage Robuste d'Images par Turbo TCQ. *Traitement du Signal*, 25(6), Avril 2009. – sources téléchargeables à l'adresse <http://www.gleguelv.org/wt/ttcq/>.
- [Li et al. 03] Li (K.) et Zhang (X.-P.). – Reliable Adaptive Watermarking Scheme Integrated with JPEG2000. – *International Symposium on Image and Signal Processing and Analysis ISPA'2003*, vol. 1, pp. 117 – 122, Septembre 2003.
- [Li et al. 07] Li (Q.) et Cox (I. J.). – Using Perceptual Models to Improve Fidelity and Provide Resistance to Valumetric Scaling for Quantization Index Modulation Watermarking. *IEEE Transactions on Information Forensics and Security*, 2(2) :127–139, 2007.

- [Lin et al. 05] Lin (L.), Cox (I. J.), Doërr (G.) et Miller (M. L.). – An Efficient Algorithm for Informed Embedding of Dirty Paper Trellis Codes for Watermarking. – *IEEE International Conference on Image Processing, ICIP'2005*, vol. 1, pp. 697–700, Genova, Italy, Septembre 2005.
- [Linnartz et al. 98] Linnartz (J. P. M. G.) et Talstra (J. C.). – MPEG PTY-Marks : Cheap Detection of embedded Copyright Data in DVD-Video. – *Proceedings 5th European Symposium on Research in Computer Security, ESORICS'1998*, vol. 1485, pp. 221–240, Louvain la Neuve, Belgium, Septembre 1998.
- [Marcellin et al. 90] Marcellin (M. W.) et Fischer (T. R.). – Trellis Coded Quantization of Memoryless and Gauss-Markov Sources. *IEEE Transaction on communication, TC'1990*, 38(1) :82–93, Janvier 1990.
- [Meerwald 01] Meerwald (P.). – Quantization Watermarking in the JPEG2000 Coding Pipeline. – *International Conference on Communications and Multimedia Security Issues of the New Century IFIP TC6/TC11*, pp. 69–79, Deventer, The Netherlands, The Netherlands, 2001. Kluwer, B.V.
- [Miller et al. 99] Miller (M. L.) et Bloom (J. A.). – Computing the Probability of False Watermark Detection. – *Proceedings of the Third International Workshop on Information Hiding, IH'1999*, pp. 146–158, Dresden, Germany, Septembre 1999.
- [Miller et al. 04] Miller (M. L.), Doërr (G.) et Cox (I. J.). – Applying Informed Coding and Informed Embedding to Design a Robust, High Capacity Watermark. *IEEE Transactions on Image Processing*, 13(6) :792–807, 2004.
- [Mobasserri et al. 07] Mobasserri (B. G.) et Raikar (Y. N.). – Authentication of H.264 streams by direct watermarking of CAVLC blocks. – *Security, Steganography, and Watermarking of Multimedia Contents IX, Part of IS&T/SPIE 19th Annual Symposium on Electronic Imaging, SSWMC'2007, SPIE'2007*, vol. 6505, San Jose, California, USA, Février 2007.
- [Noorkami et al. 08] Noorkami (M.) et Mersereau (R. M.). – Digital Video Watermarking in P-Frames With Controlled Video Bit-Rate Increase. *IEEE Transactions on Information Forensics and Security, TIFS'2008*, 3(3) :441–455, Septembre 2008.
- [Ope11] OpenJPEG (open-source JPEG 2000 codec), téléchargeable à l'adresse <http://www.openjpeg.org/>, 2011.
- [OuledZaid et al. 09] Ouled Zaid (A.), Makhoulfi (A.), Bouallegue (A.) et Olivier (C.). – Improved QIM-based Watermarking Integrated to JPEG2000 Coding Scheme. *Signal, Image and Video Processing*, 3 :197–207, 2009.
- [Patrizio 99] Patrizio (A.). – Why the Dvd Hack Was a Cinch. *Wired*, Novembre 1999.
- [PérezFreire et al. 06] Pérez-Freire (L.), Comesana (P.), Troncoso-Pastoriza (J. Ramón) et Pérez-González (F.). – Watermarking Security : a Survey. *IEEE Transactions on Data Hiding and Multimedia Security*, 1(4300) :41–72, 2006.
- [PérezFreire et al. 07] Pérez-Freire (L.) et Pérez-González (F.). – Exploiting Security Holes in Lattice Data Hiding. – *Information Hiding, IH'2007, Lecture Notes in Computer Science*, pp. 159–173. Springer-Verlag, Juin 2007.
- [PérezGonzález et al. 04] Pérez-González (F.), Barni (M.), Abrardo (A.) et Mosquera (C.). – Rational Dither Modulation : A Novel Data-hiding Method Robust to Valuemetric Scaling Attacks. – *IEEE International Workshop on Multimedia Signal Processing, IWMSp'2004*, pp. 139–142, Siena, Italy, Septembre 2004.

- [Pevný 11] Pevný (T.). – Detecting Messages of Unknown Length. – *Media Watermarking, Security, and Forensics III, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, SPIE'2011*, vol. 7880, San Francisco, California, USA, Février 2011.
- [Pevný et al. 08] Pevný (T.) et Fridrich (J.). – Novelty Detection in Blind Steganalysis. – *Proceedings of the 10th ACM workshop on Multimedia and security, MM&Sec'2008*, pp. 167–176. ACM, Septembre 2008.
- [Pevný et al. 10] Pevný (T.), Filler (T.) et Bas (P.). – Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. – *Information Hiding, 12th International Conference, IH'2010*, vol. 6387 of *Lecture Notes in Computer Science*, pp. 161–177, Calgary, Alberta, Canada, Juin 2010. Springer.
- [Piva et al. 07] Piva (A.) et Barni (M.). – The first BOWS Contest (Break Our Watermarking System). – *IS&T/SPIE 19th Annual Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX edited by Edward J. Delp III, Ping Wah Wong, SPIE'2007*, vol. 6505, pp. 1–10, San Jose, California, USA, Janvier 2007.
- [Queiroz 10] Queiroz (R. De). – Reversible Color-to-Gray Mapping Using Subband Domain Texturization. *Pattern Recognition Letters, 20th SIBGRAPI : Advances in Image Processing and Computer Vision*, 31(4) :269–276, Mars 2010.
- [Queiroz et al. 06] Queiroz (R. De) et Braun (K.). – Color to Gray and Back : Color Embedding Into Textured Gray Images. *IEEE Transaction on Image Processing*, 15(6) :1464–1470, 2006.
- [Richardson 10] Richardson (I.). – *The H.264 Advanced Video Compression Standard*. – Wiley, Août 2010, 2nd édition.
- [Said 03] Said (A.). – *Arithmetic Coding, In : Lossless Compression Handbook, K. Sayood (Ed.)*, chap. 5, pp. 101–152. – Academic Press, Janvier 2003, 1st édition.
- [Schlauweg et al. 08] Schlauweg (Mathias), Pröfrock (Dima), Zeibich (Benedikt) et Müller (Erika). – Self-Synchronizing Robust Texel Watermarking in Gaussian Scale-Space. – *Proceedings of the 10th ACM workshop on Multimedia and security, MM&Sec'2008*, pp. 53–62, New York, NY, USA, Septembre 2008. ACM.
- [Shahid 10] Shahid (Z.). – *Protection of Scalable Video by Encryption and Watermarking*. – PhD. Thesis, Université de Montellier 2, France, Août 2010. Supervisé par W. Puech et M. Chaumont.
- [Shahid et al. 09] Shahid (Z.), Chaumont (M.) et Puech (W.). – Fast Protection of H.264/AVC by Selective Encryption of CABAC. – *IEEE International Conference on Multimedia and Expo, ICME'2009*, New York City, Juin 2009.
- [Shahid et al. 10a] Shahid (Z.), Chaumont (M.) et Puech (W.). – Selective Encryption of C2DVLC of AVS Video Coding Standard for I & P Frames. – *IEEE International Conference on Multimedia & Expo, ICME'2010*, Singapore, Juillet 2010.
- [Shahid et al. 10b] Shahid (Z.), Chaumont (M.) et Puech (W.). – Spread Spectrum-Based Watermarking for Tardos Code-Based Fingerprinting of H.264/AVC Video. – *IEEE International Conference on Image Processing, ICIP'2010*, Hong-Kong, China, Septembre 2010.
- [Shahid et al. 11a] Shahid (Z.), Chaumont (M.) et Puech (W.). – Considering the Reconstruction Loop for Data Hiding of Intra- and Inter-Frames of H.264/AVC. *Signal, Image and Video Processing*, pp. 1–19, Avril 2011.

- [Shahid et al. 11b] Shahid (Z.), Chaumont (M.) et Puech (W.). – Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I & P Frames. *IEEE Transactions on Circuits and Systems for Video Technology*, 9, Mars 2011.
- [Shahid et al. 11c] Shahid (Z.), Chaumont (M.) et Puech (W.). – Scalable Video Coding. *Effective Video Coding for Multimedia Applications*, éd. par by Sudhakar Radhakrishnan (Edited), pp. 3–20. – InTech, 2011.
- [Shahid et al. 11d] Shahid (Z.), Chaumont (M.) et Puech (W.). – Spread Spectrum-Based Watermarking for Tardos Code-Based Fingerprinting for H.264/AVC Video. – 2011. Unpublished.
- [Simmons 83] Simmons (G. J.). – The subliminal channel and digital signatures. – by D. Chaum (Edited) (édité par), *Proceeding of of Crypto'83*, pp. 51–67. New York, Plenum Press, Août 1983.
- [Solanki et al. 06] Solanki (K.), Madhow (U), Manjunath (B. S.), Chandrasekaran (S.) et El-khalil (I.). – 'Print and Scan' Resilient Data Hiding in Images. *IEEE Transactions on Information Security and Forensics, TISF'2006*, 1(4) :464–478, Décembre 2006.
- [Su et al. 03] Su (P. C.) et Kuo (C. J.). – Steganography in JPEG2000 Compressed Images. *IEEE Transaction on Consumer Electronics*, 49 :824–832, Novembre 2003.
- [Tanaka et al. 90] Tanaka (K.), Nakamura (Y.) et Matsui (K.). – Embedding Secret Information Into a Dithered Multi-Level Image. – *Military Communications Conference, 1990. MILCOM '90, Conference Record, A New Era, IEEE*, vol. 1, pp. 216–220, Monterey, CA , USA, Septembre 1990.
- [Tardos 03] Tardos (G.). – Optimal Probabilistic Fingerprint Codes. – *ACM symposium on Theory of computing*, pp. 116–125, New York, NY, USA, 2003.
- [Taubman 00] Taubman (D.). – High performance scalable image compression with EBCOT. – *IEEE Transactions On Image Processing*, vol. 9, pp. 1158–1170, Juillet 2000.
- [Thiesse et al. 10] Thiesse (J.-M.), Jung (J.) et Antonini (M.). – Data Hiding of Intra Prediction Information in Chroma Samples for Video Compression . – *IEEE International Conference on Image Processing, ICIP'2010*, pp. 2861 – 2864, Hong Kong, Septembre 2010.
- [Tzeng et al. 04] Tzeng (C.H.), Yang (Z.F.) et Tsai (W.H.). – Adaptative Data Hiding in Palette Images by Color Ordering and Mapping With Security Protection. *IEEE Transaction on Communications*, 52(5) :791–800, 2004.
- [Ungerboeck. 82] Ungerboeck. (G.). – Channel Coding with Multilevel/Phase Signals. *IEEE Transaction on Information Theory*, 28 :55–67, 1982.
- [Viterbi 95] Viterbi (Andrew J.). – *CDMA : Principles of Spread Spectrum Communication*. – Addison-Wesley Wireless Communications, 1995.
- [Škorić et al. 08a] Škorić (B.), Katzenbeisser (S.) et Celik (M. U.). – Symmetric Tardos Fingerprinting Codes for Arbitrary Alphabet Sizes. *Designs, Codes and Cryptography*, 46 :137–166, Février 2008.
- [Škorić et al. 08b] Škorić (B.), Vladimirova (T. U.), Celik (M. U.) et Talstra (J.). – Tardos Fingerprinting is Better Than we Thought. *IEEE Transactions on Information Theory, TIT'2008*, 54(8) :3663 – 3676, 2008.
- [Wang et al. 04] Wang (Z.), Bovik (A. C.), Sheikh (H. R.) et Simoncelli (E. P.). – Image Quality Assessment : From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4) :600–612, 2004.

- [Wang et al. 06] Wang (C.), Doërr (G.) et Cox (I. J.). – Toward a Better Understanding of Dirty Paper Trellis Codes. – *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'2006*, vol. 2, pp. 233–236, Toulouse, France, Mai 2006.
- [Wang et al. 07] Wang (C.), Doërr (G.) et Cox (I. J.). – Trellis Coded Modulation to Improve Dirty Paper Trellis Watermarking. – *IS&T/SPIE 19th Annual Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX* edited by Edward J. Delp III, Ping Wah Wong, *SPIE'2007*, vol. 6505, pp. 0G1–0G10, San-Jose, USA, Janvier 2007.
- [Wang et al. 11] Wang (K.), Lavoué (G.), Denis (F.) et Baskurt (A.). – Robust and Blind Mesh Watermarking Based on Volume Moments. *Computers & Graphics*, 35(1) :1–19, Février 2011.
- [Watson 93] Watson (A. B.). – DCT Quantization Matrices Optimized for Individual Images. – *Human Vision, Visual Processing, and Digital Display IV, SPIE'1993*, vol. 1913, pp. 202–216, 1993.
- [Westfeld 01] Westfeld (A.). – F5 - A Steganographic Algorithm : High Capacity Despite Better Steganalysis. – *4th International Workshop on Information Hiding*, vol. 2137, pp. 289–302. Springer-Verlag, 2001.
- [Wu et al. 03] Wu (M.-Y.), Ho (Y.-K.) et Lee (J.-H.). – An Iterative Method of Palette-Based Image Steganography. *Pattern Recognition Letters*, 25 :301–309, 2003.
- [Xie et al. 08] Xie (F.), Furon (T.) et Fontaine (C.). – On-Off Keying Modulation and Tardos Fingerprinting. – *10th ACM Multimedia and Security Workshop, MM&Sec'2008*, pp. 101–106, Septembre 2008.
- [Xie et al. 10a] Xie (F.), Furon (T.) et Fontaine (C.). – Better Security Levels for 'Broken Arrows'. – *IS&T/SPIE 22th Annual Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents XII*, vol. 7541, Janvier 2010.
- [Xie et al. 10b] Xie (F.), Furon (T.) et Fontaine (C.). – Towards Robust and Secure Watermarking. – *12th ACM Multimedia and Security Workshop, MM&SEC'2010*, pp. 153–160. ACM, Septembre 2010.
- [Zhang et al. 01] Zhang (Jun), Li (Jie-gu) et Zhang (Ling). – Video Watermark Technique in Motion Vector. – *Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI'2001*, SIBGRAPI'2001, pp. 179–182, Washington, DC, USA, Octobre 2001. IEEE Computer Society.
- [Zhang et al. 09] Zhang (R.), Sachnev (V.) et Kim (H.). – Fast BCH Syndrome Coding for Steganography. *Information Hiding*, éd. par Katzenbeisser (Stefan) et Sadeghi (Ahmad-Reza), pp. 48–58. – Springer Berlin / Heidelberg, 2009.
- [Zheng et al. 07] Zheng (D.), Liu (Y.), Zhao (J.) et Saddik (A. El). – A Survey of RST Invariant Image Watermarking Algorithms. *ACM Computing Surveys*, 39 :70 pages, Juillet 2007.
- [Zhu et al. 08] Zhu (X.) et Tang (Z.). – Improved Quantization Index Modulation Watermarking Robust Against Amplitude Scaling Distortions. – *IEEE International Conference on Multimedia & Expo, ICME'2008*, pp. 237–240, Hannover, Germany, Juin 2008.
- [Zou et al. 09] Zou (D.) et Bloom (J. A.). – H.264/AVC Substitution Watermarking : a CAVLC Example. – *Media Forensics and Security I, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, MFS'2009, SPIE'2009*, p. 7254, Janvier 2009.

Quatrième partie

Quelques publications

FAST PROTECTION OF H.264/AVC BY SELECTIVE ENCRYPTION OF CAVLC AND CABAC FOR I & P FRAMES

Z. Shahid, M. Chaumont and W. Puech

Abstract—This paper presents a novel method for the protection of bitstreams of state of the art video codec H.264/AVC. The problem of selective encryption (SE) is addressed along with the compression in the entropy coding modules. H.264/AVC supports two types of entropy coding modules. Context-adaptive variable length coding (CAVLC) is supported in H.264/AVC baseline profile and context-adaptive binary arithmetic coding (CABAC) is supported in H.264/AVC main profile. SE is performed in both types of entropy coding modules of this video codec. For this purpose, in this paper the encryption step is done simultaneously with the entropy coding CAVLC or CABAC. SE is performed by using the Advanced Encryption Standard (AES) algorithm with the Cipher Feedback (CFB) mode on a subset of *codewords/binstrings*. For CAVLC, SE is performed on equal length *codewords* from a specific variable length coding (VLC) table. In case of CABAC, it is done on equal length *binstrings*. In our scheme, entropy coding module serves the purpose of encryption cipher without affecting the coding efficiency of H.264/AVC by keeping exactly the same bitrate, generating completely compliant bitstream and utilizing negligible computational power. Owing to no escalation in bitrate, our encryption algorithm is better suited for real-time multimedia streaming over heterogeneous networks. It is perfect for playback on hand-held devices because of negligible increase in processing power. Nine different benchmark video sequences containing different combinations of motion, texture and objects are used for experimental evaluation of the proposed algorithm.

Index Terms—Selective Encryption, CABAC, CAVLC, Video Security, AES Algorithm, Stream Cipher

I. INTRODUCTION

With the rapid growth of processing power and network bandwidth, many multimedia applications have emerged in the recent past. As digital data can easily be copied and modified, the concern about its protection and authentication have surfaced. Digital

rights management (DRM) has emerged as an important research field to protect the copyrighted multimedia data. DRM systems enforce the rights of the multimedia property owners while ensuring the efficient rightful usage of such property.

Multimedia data requires either full encryption or selective encryption (SE) depending on the application requirements. For example military and law enforcement applications require full encryption. Nevertheless, there is a large spectrum of applications that demands security on a lower level, as for example that ensured by SE. SE encrypts part of the plaintext and has two main advantages. First, it reduces the computational requirements, since only a part of plaintext is encrypted [6]. Second, encrypted bitstream maintains the essential properties of the original bitstream [3]. SE just prevents abuse of the data. In the context of video, it refers to destroying the commercial value of video to a degree which prevents a pleasant viewing experience.

SE schemes based on H.264/AVC have been already presented on CAVLC [29] and CABAC [30]. These two previous methods fulfill real-time constraints by keeping the same bitrate and by generating completely compliant bitstream. In this paper, we have enhanced the previous proposed approaches by encryption of more syntax elements for CAVLC and extending it for P frames. Here we have also used AES [7] in the Cipher Feedback (CFB) mode which is a stream cipher algorithm. Security of the proposed schemes has also been analyzed in detail. The rest of the paper is organized as follows. In Section II, overview of H.264/AVC and AES algorithm is presented. We explain the whole system architecture of the proposed methods in Section III. Section IV contains experimental evaluation and security analysis. In Section V, we present the concluding remarks about the proposed schemes.

Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II, 161, rue Ada, 34392 MONTPELLIER Cedex 05, FRANCE
zafar.shahid@lirmm.fr, marc.chaumont@lirmm.fr,
william.puech@lirmm.fr

II. DESCRIPTION OF THE H.264/AVC-BASED VIDEO ENCRYPTION SYSTEM

A. Overview of H.264/AVC

H.264/AVC (also known as MPEG4 Part 10) [1] is state of the art video coding standard of ITU-T and ISO/IEC. In H.264/AVC, an input video frame is divided into macro-blocks (MB) of 16x16 pixels and each of them is encoded separately. Each video frame can be encoded as *intra* (I frame) or *inter* (P and B frames). In I frame, the current MB is predicted spatially from MBs which have been previously encoded and reconstructed (MB at top and left). In P frame, motion compensated prediction is done from the previous reference frames, while bidirectional prediction from both previous and next reference frames is performed for B frames. The purpose of the reconstruction in the encoder is to ensure that both the encoder and decoder use identical reference frames to create the predictions. If this is not the case, then the predictions in encoder and decoder will not be identical, leading to an increasing error or "drift" between the encoder and decoder. The difference between original and predicted frame is call residual. This residual is coded using transform coding followed by quantization and zigzag scan. In the last step, entropy coding comes into action. Quantized transform coefficients are then coded using either CAVLC [4] or CABAC [20]. The block diagram of H.264/AVC is shown in Fig. 1. On the decoding side, compressed bitstream is decoded by entropy decoding module, followed by inverse-zigzag scan. These coefficients are then rescaled and inverse transformed to get the residual signal which is added to the predicted signal to get the decoded video frame. H.264/AVC has some additional features as compared to

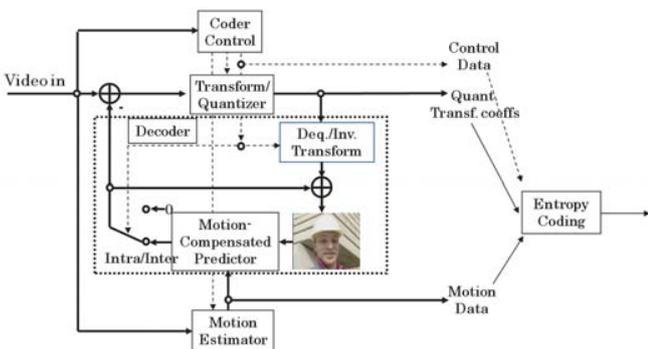


Fig. 1: Block diagram of H.264/AVC video encoder.

previous video standards including MPEG2 and MPEG4 Part II. In *baseline* profile of H.264/AVC, it has 4×4 integer transform (IT) in contrast to 8×8 transform of previous standards. In higher profiles, it offers transform

coding for adaptive size. DCT transform has been replaced by IT which does not need any multiplication operation and can be implemented by only additions and shifts and thus requires lesser number of computations. For *Inter* frame, H.264/AVC supports variable block size motion estimation, quarter pixel accuracy, multiple reference frames, improved skipped and direct motion inference. For *Intra* frame, it offers additional spatial prediction modes. All these additional features of H.264/AVC are aimed to outperform previous video coding standards [35].

H.264 scans the non-zero coefficients (NZs) in inverse zigzag order (from high frequency NZs to low frequency NZs) which are then passed to entropy coding module. We review the basic working of CAVLC in Section II-A1 and of CABAC in Section II-A2.

1) *Context-Adaptive Variable Length Coding*: In CAVLC, Run-length coding is performed first as it encodes levels and runs separately. CAVLC is designed to exploit the characteristics of NZs and works in several steps.

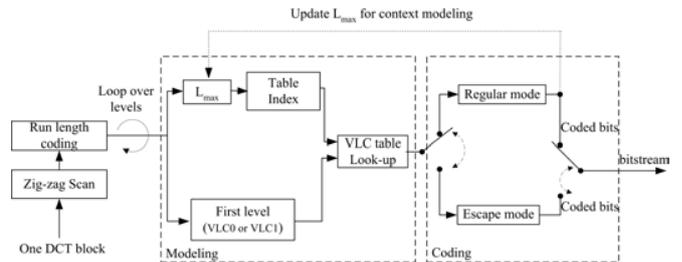


Fig. 2: Block diagram of level coding in CAVLC of H.264/AVC.

To adapt to the local statistical features of DCT coefficients, CAVLC uses seven fixed variable length coding (VLC) tables. For example, '2' will be coded as '010' using VLC1 table, while it will be coded as '1010' using VLC3 table. If magnitude of NZ lies within the range of that VLC table, it is coded by regular mode, otherwise escape mode is used. Adaptive nature is introduced by changing the table for the next NZ based on the magnitude of the current NZ as shown in Fig. 2. For the first NZ, VLC0 table is used unless there are more than 10 NZs and less than 3 trailing ones, in which case it is coded with VLC1 table. The tree representation of first four VLC tables is shown in Fig. 3.

2) *Context-Adaptive Binary Arithmetic Coding*: CABAC is designed to better exploit the characteristics of NZs as compared to CAVLC, consumes more processing and offers about 10% better compression than CAVLC on average [22]. Run-length coding has been replaced by *significant map* coding which specifies the

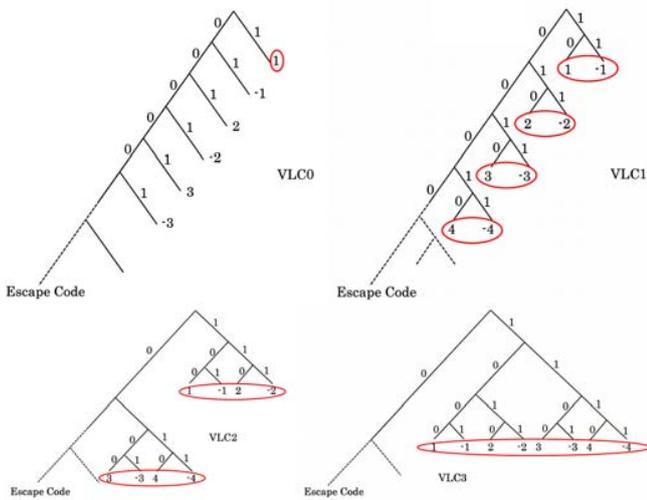


Fig. 3: Tree representation of first four VLC tables used in CAVLC. In each VLC table, VLC codes for encircled coefficients have same code length.

position of NZs in the 4x4 block. Binary arithmetic coding module (BAC) of CABAC uses many context models to encode NZs and context model for a specific NZ depends on recently coded NZs.

CABAC consists of multiple stages as shown in Fig. 4.a. First of all, *binarization* is done in which, non-binary syntax elements are converted to binary form called *binstrings* which are more amenable to compression by BAC. Binary representation for a non-binary syntax element is done in such a way that it is close to minimum redundancy code. In CABAC, there are four basic code trees for *binarization* step, namely the *unary* code, the *truncated unary* code, the *kth order Exp-Golomb* code (EGk) and the *fixed length* code as shown in Fig. 4.b.

For an unsigned integer value $x \geq 0$, the *unary* code consists of x 1's plus a terminating 0 bit. The *truncated unary* code is only defined for x with $0 \leq x \leq s$. For $x < s$ the code is given by the *unary* code, whereas for $x = s$ the terminating "0" bit is neglected. EGk is constructed by a concatenation of a prefix and a suffix parts and is suitable for binarization of syntax elements that represent prediction residuals. For a given unsigned integer value $x > 0$, the prefix part of the EGk *binstring* consists of a unary code corresponding to the length $l(x) = \lceil \log_2(\frac{x}{2^k} + 1) \rceil$. The EGk suffix part is computed as the binary representation of $x + 2^k(1 - 2^{l(x)})$ using $k + l(x)$ significant bits. Consequently for EGk binarization, the code length is $2l(x) + k + 1$. When $k = 0$, $2l(x) + k + 1 = 2l(x) + 1$.

The *fixed length* code is applied to syntax elements with a nearly uniform distribution or to syntax elements, for which each bit in the *fixed length* code

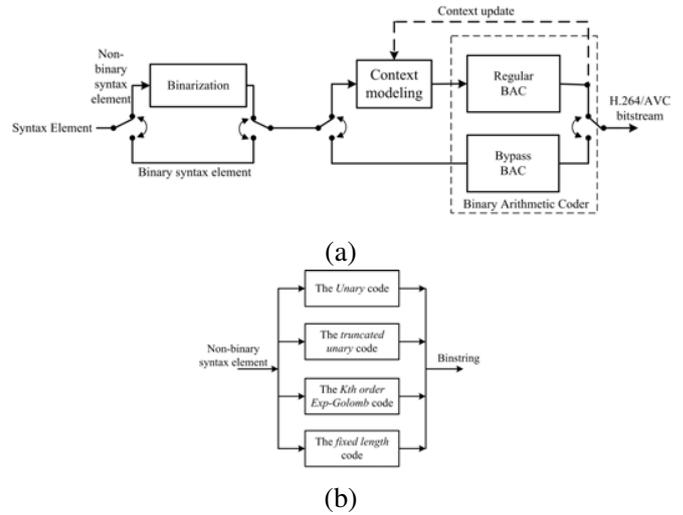


Fig. 4: (a) Block diagram of CABAC of H.264/AVC, (b) Binarization stage.

binstring represents a specific coding decision e.g., *coded block flag*. Three syntax elements are binarized by concatenation of the basic code trees, namely *coded block pattern*, NZ and the motion vector difference (MVD). Binarization of absolute level of NZs is done by concatenation of *truncated unary* code and EG0. The *truncated unary* code constitutes the prefix part with cutoff value $S = 14$. Binarization and subsequent arithmetic coding process is applied to the syntax element $coeff_abs_value_minus1 = abs_level - 1$, since quantized transformed coefficients with zero magnitude are encoded using *significant map*. For MVD, *binstring* is constructed by concatenation of the *truncated unary* code and EG3. The *truncated unary* constitutes the prefix part with cutoff value $S = 9$. Suffix part of MVDs contains EG3 of $|MVD| - 9$ for $|MVD| > 9$ and sign bit.

B. The AES encryption algorithm

The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps repeated for a number of iterations called rounds [7]. The number of rounds depends on the size of the key and the size of the data block. The number of rounds is 9 for example, if both the block and the key are 128 bits long. Given a sequence $\{X_1, X_2, \dots, X_n\}$ of bit plaintext blocks, each X_i is encrypted with the same secret key k producing the ciphertext blocks $\{Y_1, Y_2, \dots, Y_n\}$. To encipher a data block X_i in AES you first perform an AddRoundKey step by XORing a subkey with the block. The incoming data and the key are added together in the first AddRoundKey step. Afterward, it follows the

round operation. Each regular round operation involves four steps which are SubBytes, ShiftRows, MixColumns and AddRoundKey. Before producing the final ciphered data Y_i , the AES performs an extra final routine that is composed of SubBytes, ShiftRows and AddRoundKey steps.

The AES algorithm can support several cipher modes: ECB (Electronic Code Book), CBC (Cipher Block Chaining), OFB (Output Feedback), CFB (Cipher Feedback) and CTR (Counter) [31]. The ECB mode is actually the basic AES algorithm. With the ECB mode, each plaintext block X_i is encrypted with the same secret key k producing the ciphertext block $Y_i = E_k(X_i)$. The CBC mode adds a feedback mechanism to a block cipher. Each ciphertext block Y_i is XORed with the incoming plaintext block X_{i+1} before being encrypted with the key k . An initialization vector (IV) is used for the first iteration. In fact, all modes (except the ECB mode) require the use of an IV. In CFB mode, as shown in Fig. 5, the keystream element Z_i is generated and the ciphertext block Y_i is produced as:

$$\begin{cases} Z_i = E_k(Y_{i-1}), \text{ for } i \geq 1 \\ Y_i = X_i \oplus Z_i \end{cases}, \quad (1)$$

where \oplus is the XOR operator.

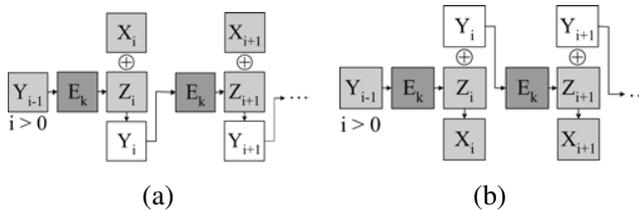


Fig. 5: CFB stream cipher: (a) Encryption, (b) Decryption.

In the OFB mode, Z_0 is substituted by the IV and the input data is encrypted by XORing it with the output Z_i . The CTR mode has very similar characteristics to OFB, but in addition it allows pseudo random access for decryption. It generates the next keystream block by encrypting successive values of a counter.

Although AES is a block cipher, in the OFB, CFB and CTR modes it operates as a stream cipher. These modes do not require any special measures to handle messages whose lengths are not multiples of the block size since they all work by XORing the plaintext with the output of the block cipher. Each mode has its advantages and disadvantages. For example in ECB and OFB modes, any modification in the plaintext block X_i causes the corresponding ciphered block Y_i to be altered, but other ciphered blocks are not affected. On the other hand, if a

plaintext block X_i is changed in CBC and CFB modes, then Y_i and all subsequent ciphered blocks will be affected. These properties mean that CBC and CFB modes are useful for the purpose of authentication while ECB and OFB modes treat separately each block. Therefore, we can notice that OFB does not spread noise, while the CFB does that.

C. SE of image and video

Selective encryption (SE) is a technique aiming to save computational time or to enable new system functionalities by only encrypting a portion of a compressed bitstream while still achieving adequate security [18]. SE as well as partial encryption (PE) are applied only on certain parts of the bitstream. In the decoding stage, both the encrypted and the non-encrypted information should be appropriately identified and displayed [6], [21], [26]. The copyright protection of the multimedia content is a required feature for DRM systems. The technical challenges posed by such systems are high and previous approaches have not entirely succeeded in tackling them [17].

In [32], Tang proposed a technique called zigzag permutation applicable to DCT-based image and video codecs. On one hand this method provides a certain level of confidentiality, while on the other hand it increases the overall bitrate. For image, several SE techniques have been proposed in literature. In [8], Droogenbroeck and Benedett proposed a technique for encryption of JPEG images. It encrypts a selected number of AC coefficients. The DC coefficients are not ciphered since they carry important visual information and they are highly predictable. In spite of the constancy in the bitrate while preserving the bitstream compliance, the compression and the encryption process are separated and consequently the computational complexity is increased.

The Advanced Encryption Standard (AES) [7] has been used for SE of image and video in literature. The AES was applied on the Haar discrete wavelet transform compressed images in [23]. The encryption of color images in the wavelet transform has been addressed in [21]. In this approach the encryption is performed on the resulting wavelet code bits. In [25], SE was performed on color JPEG images by selectively encrypting only *luma* component using AES cipher. The protection rights of individuals and the privacy of certain moving objects in the context of security surveillance systems using viewer generated masking and the AES encryption standard has been addressed in [37].

Combining PE and image/video compression using the set partitioning in hierarchical trees was used in [6].

Nevertheless, this approach requires a significant computational complexity. A method that does not require significant processing time and which operates directly on the bit planes of the image was proposed in [19]. The robustness of partially encrypted videos to attacks which exploit the information from non-encrypted bits together with the availability of side information was studied in [27]. Fisch *et al.* [10] proposed a scalable encryption method for a DCT-coded visual data wherein the data are organized in a scalable bitstream form. These bitstreams are constructed with the DC and some AC coefficients of each block which are then arranged in layers according to their visual importance and PE process is applied over these layers.

For video, there are several SE techniques for different video codecs presented in literature. SE of MPEG4 video standard was studied in [34] wherein Data Encryption Standard (DES) was used to encrypt fixed length and variable length codes. In this approach, the encrypted bitstream is completely compliant with MPEG4 bitstream format but it increases the bitrate. A trade off has to be made among complexity, security and the bit overhead. In [38], SE of MPEG4 video standard is proposed by doing frequency domain selective scrambling, DCT block shuffling and rotation. This scheme is very easy to perform but its limitation is its bitrate overhead. SE of ROI of MPEG4 video has been presented in [9]. It performs SE by pseudo randomly inverting sign of DCT coefficients in ROI. SE of H.264/AVC has been studied in [15] wherein encryption has been carried out in some fields like intra-prediction mode, residual data, inter-prediction mode and motion vectors. A scheme for commutative encryption and watermarking of H.264/AVC is presented in [16]. Here SE of some MB header fields is combined with watermarking of magnitude of DCT coefficients. This scheme presents a watermarking solution in encrypted domain without exposing video content. The limitation of techniques proposed in [15], [16] is that they are not format compliant. Encryption for H.264/AVC has been discussed in [5] wherein they do permutations of the pixels of MBs which are in *region of interest* (ROI). The drawback of this scheme is that bitrate increases as the size of ROI increases. This is due to change in the statistics of ROI as it is no more a slow varying region which is the basic assumption for video signals.

SE of H.264/AVC at network abstraction layer (NAL) has been proposed by [14]. Important NAL units namely instantaneous decoding refresh (IDR) picture, sequence parameter set (SPS), and picture parameter set (PPS) are encrypted with a stream cipher. The limitation of this scheme is that it is not format compliant and cannot be

parsed even at frame level. SE of H.264/AVC using AES has been proposed in [2]. In this scheme, encryption of I frame is performed, since P and B frame are not significant without I frames.. This scheme is not format compliant.

The use of general entropy coder as encryption cipher using statistical models has been studied in the literature in [36]. It encrypts by using different Huffman tables for different input symbols. The tables, as well as the order in which they are used, are kept secret. This technique is vulnerable to known plaintext attacks as explained in [12]. Key-based interval splitting of arithmetic coding (KSAC) has used an approach [13] wherein intervals are partitioned in each iteration of arithmetic coding. Secret key is used to decide how the interval will be partitioned. Number of sub intervals in which an interval is divided should be kept small as it increases the bitrate of bitstream. Randomized arithmetic coding [11] is aimed at arithmetic coding but instead of partitioning of intervals like in KSAC, secret key is used to scramble the order of intervals. The limitation of these entropy coding based techniques is that encrypted bitstream is not format compliant. Moreover, these techniques require lot of processing power.

In the context of DRM systems, our study addresses the simultaneous SE and compression for state of the art H.264/AVC. The encrypted bitstream is format compliant with absolutely no escalation in bitrate. Furthermore, it does not require lot of processing power for encryption and decryption. In Section III we describe our proposed approaches to apply simultaneously SE and H.264/AVC compression in video sequences.

III. THE PROPOSED SELECTIVE ENCRYPTION SCHEMES

Our approach consists of SE during the entropy coding stage of H.264/AVC as shown in Fig. 6. In baseline profile, SE is performed in CAVLC entropy coding stage (SE-CAVLC). While in main profile, it is performed in CABAC entropy coding stage (SE-CABAC). In SE of video, encrypted bitstream compliance is a required feature for some direct operations such displaying, time seeking and browsing. Encrypted bitstream will be compliant and fulfills real-time constraints if the following three conditions are fulfilled:

- To keep the bitrate of encrypted bitstream same as the original bitstream, encrypted *codewords/binstrings* must have the same size as the original *codewords/binstrings*.
- The encrypted *codewords/binstrings* must be valid so that they may be decoded by entropy decoder.

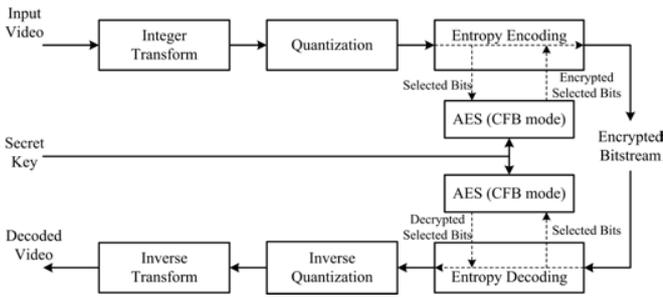


Fig. 6: Block diagram of encryption and decryption process in H.264/AVC.

- The decoded value of syntax element from encrypted *codewords/binstrings* must stay in the valid range for that syntax element. Any syntax element which is used for prediction of neighboring MBs should not be encrypted. Otherwise the drift in the value of syntax element will keep on increasing and after a few iterations, value of syntax element will fall outside the valid range and bitstream will be no more decodable.

In each MB, header information is encoded first, which is followed by the encoding of MB data. To keep the bitstream compliant, we cannot encrypt MB header, since it is used for prediction of future MBs. MB data contains NZs and can be encrypted. A MB is further divided into 16 blocks of 4x4 pixels to be processed by IT module. The *coded block pattern* is a syntax element used to indicate which 8x8 blocks within a MB contain NZs. The *macroblock mode* (MBmode) is used to indicate whether a MB is *skipped* or not. If MB is not *skipped*, then MBmode indicates the prediction method for a specific MB. For a 4x4 block inside MB, if *coded block pattern* and MBmode are set, it indicates that this block is encoded. Inside 4x4 block, *coded block flag* is the syntax element used to indicate whether it contains NZs or not. It is encoded first. If it is zero, no further data is transmitted; otherwise, it is followed by encoding of *significant map* in case of CABAC. Finally, the absolute value of each NZ and its sign are encoded. Similar to MB header, header of 4x4 block which includes *coded block flag* and *significant map*, should not be encrypted for the sake of bitstream compliance.

Available encryption space (ES) which fulfills the above mentioned conditions for SE-CAVLC and SE-CABAC is presented in Section III-A and III-B respectively. Encryption and decryption of the protected bitstream are presented in Section III-C and III-D respectively.

A. Encryption space (ES) for SE-CAVLC

In CAVLC, five syntax elements are used to code levels and runs as shown in Fig. 7. NZs are coded by three syntax elements namely *coeff_token*, signs of trailing ones and remaining non-zero levels. Zeros are coded by two syntax elements namely total no. of zeros and runs of zeros. A single syntax element namely *coeff_token* is used to code total NZs and number of trailing ones. It is followed by coding of signs of trailing ones (T1's). Remaining NZs are then coded using seven VLC look-up tables either by regular mode or by escape mode as explained in Section II-A1. They are mapped to some code from a specific VLC look-up table.

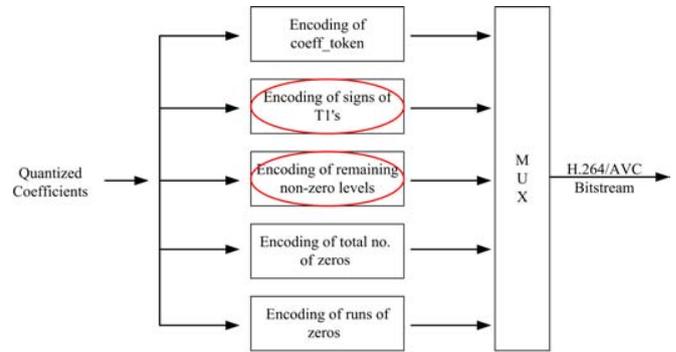


Fig. 7: Block diagram of CAVLC of H.264/AVC. Encircled syntax elements are used for SE-CAVLC.

To keep the bitstream compliant, we cannot encrypt *coeff_token*, total number of zeros and runs of zeros. Two syntax elements fulfill the above mentioned conditions for encryptions. First is signs of trailing ones. Second is sign and magnitude of remaining NZs, both in regular and escape mode. For the sake of same bitrate, ES of SE-CAVLC consists of only those NZs whose VLC *codewords* have the same length. CAVLC uses multiple VLC tables with some threshold for incrementing the table as given in equation (2). Since the threshold for a specific table is highest possible value possible with that *codeword* length (this is the case when all the suffix bits of the *codeword* are 1), magnitude of encrypted NZ is such that VLC table transition is not affected. VLC codes, having same code length, constitute the ES. For VLC_n table, ES is 2^n as given in equation (3). For table VLC₀, every NZ has different *codeword* length, consequently we cannot encrypt the NZs in this table:

$$TH[0 \dots 6] = (0, 2, 3, 6, 12, 24, 48, \infty). \quad (2)$$

$$ES[0 \dots 6] = (1, 2, 4, 8, 16, 32, 64, \infty). \quad (3)$$

B. Encryption space (ES) for SE-CABAC

The main difference between SE-CAVLC and SE-CABAC is that in SE-CABAC, SE is not performed on CABAC bitstream. Rather it is performed on *binstrings* which are input to BAC as shown in Fig. 8. Among all the four *binarization* techniques, the *unary* and *truncated unary* codes have different code lengths for each input value as explained in Section II-A2. They do not fulfill the first condition and their encryption will change the bitrate of bitstream. Suffix of EGk and the *fixed length* code can be encrypted while keeping the bitrate unchanged. EGk is used for binarization of absolute value of levels and MVDs. Number of MVD *binstrings* have the same length and hence, first and second conditions are fulfilled. But owing to the fact that MVDs are part of MB header and are used for prediction of future motion vectors, their encryption does not fulfill third condition and their encryption makes the bitstream non-compliant. To conclude, the syntax elements which fulfill the criteria for encryption of H.264/AVC compliant bitstream are suffix of EG0 and sign bits of levels. Hence for each NZ with $|NZ| > 14$, encryption is performed on $l(x)$ of EG0. It is followed by encryption of syntax element *coeff_sign_flag* which represents sign of levels of all non-zero levels. The *fixed length* code is used for binarization of syntax elements which belong to MB header and cannot be encrypted.

To keep the bitrate intact, ES for SE-CABAC consists of only those NZs whose EG0 *binstrings* have the same length as shown in Fig. 9. EG0 codes, having same code length, constitute the ES and it depends upon $\|NZ\|$. The ES is $2^{\log_2(n+1)}$ where n is the maximum possible value by suffix bits of EG0 *i.e.* when all the bits in suffix are 1.

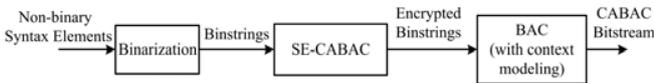


Fig. 8: SE of *binstrings* in SE-CABAC.

C. SE of NZs in the entropy coding stage of H.264/AVC

Let us consider $Y_i = X_i \oplus E_k(Y_{i-1})$ as the notation for the encryption of a n bit block X_i , using the secret key k with the AES cipher in CFB mode as given by equation (1), and performed as described in the scheme from Fig. 5. We have chosen to use this mode in order to keep the original compression rate. Indeed, with the CFB mode for each block, the size of the encrypted data Y_i can be exactly the same one as the size of the plaintext X_i . In

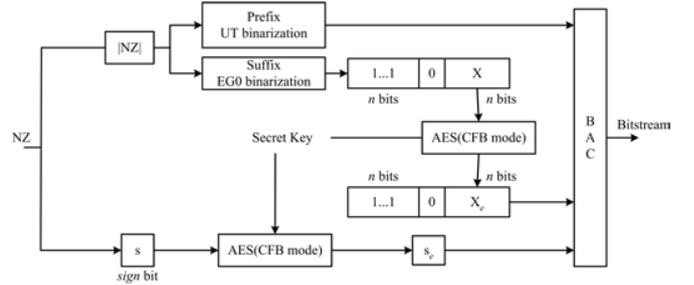


Fig. 9: Encryption process for NZs and their signs in CABAC of H.264/AVC.

this mode, the code from the previously encrypted block is used to encrypt the current one as shown in Fig. 5. The three stages of the proposed algorithm are: the construction of the plaintext X_i , described in Section III-C1, the encryption of X_i to create Y_i which is provided in Section III-C2 and the substitution of the original *code-word/binstring* with the encrypted information, which is explained in Section III-C3. The overview of the proposed SE method is provided in Fig. 10.

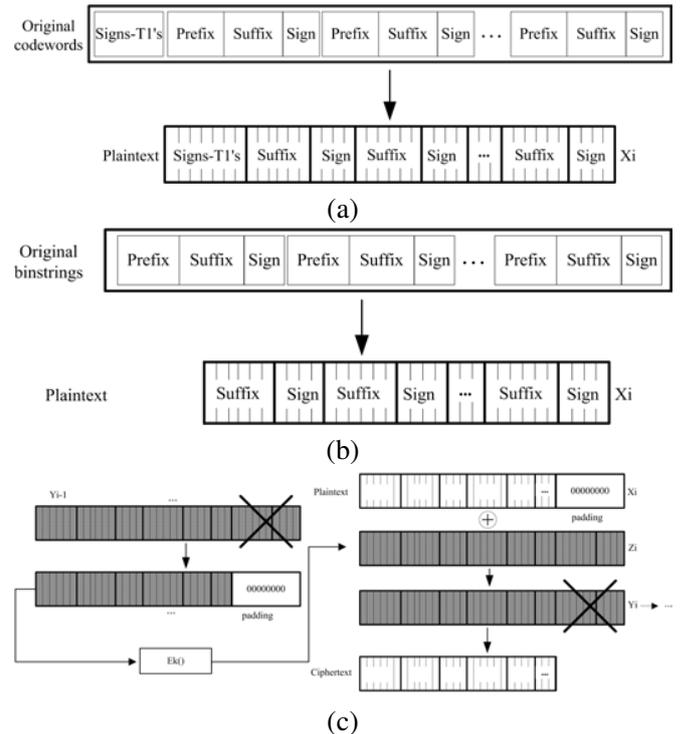


Fig. 10: Global overview of the proposed SE method (a) Preparation of plaintext for CAVLC, (b) Preparation of plaintext for CABAC, (c) Proposed SE scheme.

1) *The construction of plaintext:* As slices are independent coding units, SE should be performed on them independently. In case of SE-CAVLC, the plaintext is

created by copying the encrypt-able bits from CAVLC bitstream to the vector X_i until either X_i is completely filled or slice-boundary comes as shown in Fig. 10.a. Let C , the length of the vector X_i , is 128. In case of SE-CABAC, we perform SE before BAC as shown in Fig. 10.b. In that case, we transform the non-binary syntax elements to *binstrings* through process of binarization and at the same time we fill the X_i with encrypted bits until either the vector X_i is completely filled or the slice boundary comes. The binarization of many syntax elements at the same time also makes the CABAC coding faster and increases its throughput [39].

Let $L(X_i)$ be the length up to which vector X_i is filled. In case of slice boundary, if $L(X_i) < C$, we apply a padding function $p(j) = 0$, where $j \in \{L(X_i) + 1, \dots, C\}$, to fill in the vector X_i with zeros up to C bits. Historically, padding was used to increase the security of the encryption, but in here it is used for rather technical reasons [28].

2) *Encryption of the plaintext with AES in the CFB mode:* In the encryption step with AES in the CFB mode, the previous encrypted block Y_{i-1} is used as the input of the AES algorithm in order to create Z_i . Then, the current plaintext X_i is XORed with Z_i in order to generate the encrypted text Y_i as given by equation (1). For the initialization, the IV is created from the secret key k according to the following strategy. The secret key k is used as the seed of the pseudo random number generator (PRNG). Firstly, the secret key k is divided into 8 bits (byte) sequences. The PRNG produces a random number for each byte component of the key that defines the order of IV formation. Then, we substitute Y_0 with the IV, and Y_0 is used in AES to produce Z_1 . As illustrated in Fig. 10.c, with the CFB mode of the AES algorithm, the generation of the keystream Z_i depends on the previous encrypted block Y_{i-1} . Consequently, if two plaintexts are identical $X_i = X_j$ in the CFB mode, then always the two corresponding encrypted blocks are different, $Y_i \neq Y_j$.

3) *Substitution of the original bitstream:* The third step is the substitution of the original Y_i by the encrypted Y_i . For SE-CAVLC, CAVLC bitstream is accessed in sequential order as in the first step (construction of the plaintext X_i). Given the length in bits of each amplitude $(S_n, S_{n-1}, \dots, S_1)$, we start substituting the original bits in the bitstream by the corresponding parts of Y_i as shown in Fig. 10. For SE-CABAC, *binstrings* are accessed in sequential order and we start substituting the original bits in them by the corresponding parts of Y_i as shown in Fig. 10. In case of slice boundaries, the total quantity of replaced bits is $L(X_i)$ and consequently we do not necessarily use all the bits of Y_i .

D. Decryption process

The decryption process in the CFB mode works as follows. The previous block Y_{i-1} is used as the input to the AES algorithm in order to generate Z_i . By knowing the secret key k , we apply the same function $E_k(\cdot)$ as that used in the encryption stage. The difference is that the input of this process is now the ciphered vector. In case of SE-CAVLC, the ciphered vector is accessed in the sequential way in order to construct the plaintext Y_{i-1} which is then used in the AES to generate the keystream Z_i . The keystream Z_i is then XORed with the current block Y_i to generate X_i , as shown in Fig. 5.b. For SE-CAVLC, the resulting plaintext vector is split into segments in order to substitute the signs of trailing ones and suffixes $(S_n, S_{n-1} \dots S_1)$ in the ciphered bitstream and to generate the original CAVLC bitstream. Afterward, we apply the entropy decoding and retrieve the quantized DCT coefficients. After the inverse quantization and the inverse DCT we get the decrypted and decoded video frame.

In case of SE-CABAC, the difference is that binary arithmetic decoder is used to transform the SE-CABAC bitstream to encrypted *binstrings* which are then accessed to make the plaintext Y_{i-1} . The plaintext is decrypted and substituted back to generate original *binstrings*. They are then passed through inverse binarization, inverse quantization and inverse DCT steps to get the decrypted and decoded video frame.

IV. EXPERIMENTAL RESULTS

In this section we analyze the results for SE-CAVLC and SE-CABAC. We have used the reference implementation of H.264 JSVM 10.2 in AVC mode for video sequences in QCIF and SD resolution. For the experimental results, nine benchmark video sequences have been used for the analysis in QCIF format. Each of them represents different combinations of motion (fast/slow, pan/zoom/rotation), color (bright/dull), contrast (high/low) and objects (vehicle, buildings, people). The video sequences 'bus', 'city' and 'foreman' contain camera motion while 'football' and 'soccer' contain camera panning and zooming along with object motion and texture in background. The video sequences 'harbour' and 'ice' contain high luminance images with smooth motion. 'Mobile' sequence contains a complex still background and foreground motion.

In Section IV-A we present an analysis of joint SE and H.264/AVC compression while in Section IV-B we compare PSNR and quality when applying SE only on I frames and on I+P frames. In Section IV-C, security

analysis, showing the efficiency of the proposed method, is developed¹.

A. Analysis of joint SE and H.264/AVC compression

We have applied simultaneously our SE and H.264/AVC compression as described in Section III, on all the benchmark video sequences. SE-CAVLC and SE-CABAC impart some characteristics to the bitstream. In spatial domain, SE video gets flat regions and change in pixel values mostly occur on MB boundaries. In temporal domain, *luma* and *chroma* values rise up to maximum limit and then come back to minimum values. This cycle keeps on repeating. Owing to this phenomenon, the pixel values change drastically in temporal domain. Lot of transitions are observed in values of color and brightness. This phenomenon can be observed for SE-CAVLC and SE-CABAC in Fig. 11 and Fig. 12 respectively for QP value 18 for *foreman* video sequence.

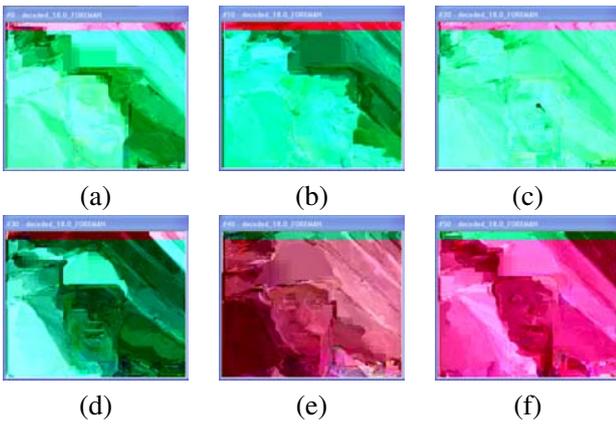


Fig. 11: Six frames of *foreman* video sequence for SE-CAVLC for QP value 18 with frame: a) #0, b) #10, c) #20, d) #30, e) #40, f) #50.

In a first set of experiments, we have analyzed the available encryption space (ES) in H.264/AVC bitstreams for both of SE-CAVLC and SE-CABAC. ES is defined as percentage of total bitstream size. MBs that contain many details and texture will have lot of NZs and consequently, will be strongly encrypted. On the other hand, the homogeneous MBs, *i.e.* blocks that contain series of identical pixels, are less ciphered because they contain a lot of null coefficients which are represented by runs in CAVLC and by significant map in CABAC. In Table I, we provide ES for SE-CAVLC and SE-CABAC for different benchmark video sequences for QP value 18. While in Table II, ES for various QP

¹Encrypted video bitstreams are available on <http://www.lirmm.fr/~shahid/>.

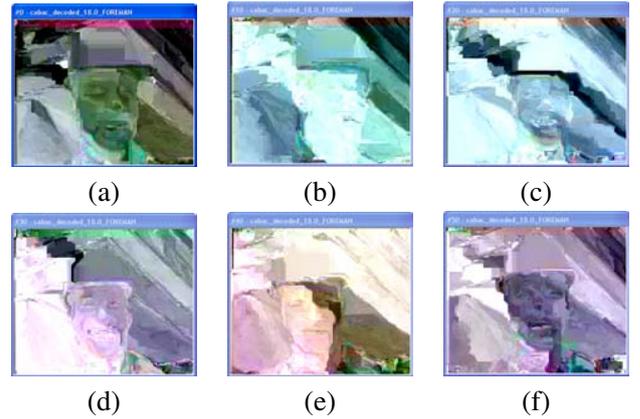


Fig. 12: Six frames of *foreman* video sequence for SE-CABAC for QP value 18 with frame: a) #0, b) #10, c) #20, d) #30, e) #40, f) #50.

values is shown for *foreman* video sequence. Here the average number of bits available for SE per MB are also provided. One can note that ES is inversely proportional to QP value. When QP value is higher and implicitly the video compression is higher, we are able to encrypt fewer bits in the compressed frame. This is due to the fact that H.264/AVC has lesser number of NZs at higher QP values. From both these tables, it is evident that more ES is available for SE-CAVLC as compared to SE-CABAC. But ES is more affected by change in QP values for SE-CAVLC as compared to SE-CABAC. For example, for *foreman* video sequence, ES varies from 28.55% to 6.70% for SE-CAVLC when QP varies from 12 to 42. For the same QP range, the change in ES for SE-CABAC is from 19.97% to 9.46% as shown in Table II. From Table I and II, since PSNR of original H.264/AVC are very similar for both CAVLC and CABAC, in the rest of this section for the sake of comparison, we list only PSNR of CAVLC bitstreams.

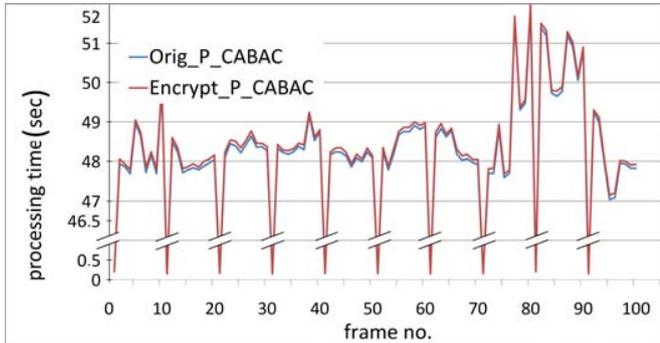
Seq.	SE-CAVLC				SE-CABAC			
	encoder		decoder		encoder		decoder	
	I (%)	I+P (%)	I (%)	I+P (%)	I (%)	I+P (%)	I (%)	I+P (%)
bus	0.69	0.31	3.77	2.7	0.57	0.25	3.37	2.3
city	0.5	0.26	3.36	2.4	0.44	0.23	3.06	2.1
crew	0.31	0.15	2.52	1.5	0.29	0.14	2.22	1.2
football	0.41	0.23	3.46	2.4	0.31	0.18	3.26	2.2
foreman	0.47	0.23	3.19	2.2	0.41	0.20	2.99	2.0
harbour	0.55	0.30	3.65	2.7	0.47	0.26	3.25	2.3
ice	0.41	0.21	3.16	2.1	0.33	0.17	2.96	1.9
mobile	0.76	0.35	4.33	3.3	0.72	0.33	4.03	3.0
soccer	0.44	0.21	3.17	2.2	0.38	0.18	2.87	1.9

TABLE III: Analysis of increase in processing power for SE-CAVLC and SE-CABAC at QP value 18.

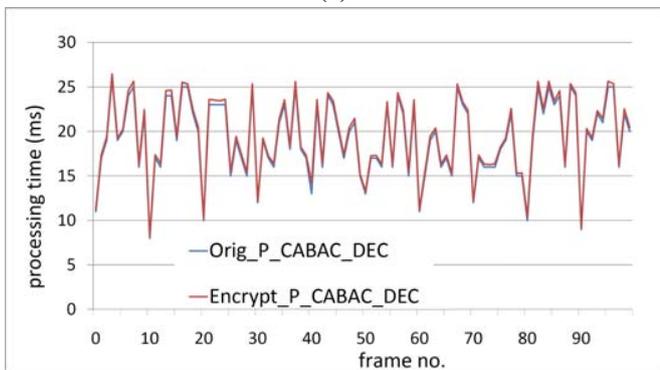
Seq.	SE-CAVLC				SE-CABAC			
	PSNR (dB)	Total Size (Bytes)	ES (%)	Avg. ES Bits/MB.	PSNR (dB)	Total Size Bytes	ES (%)	Avg. ES Bits/MB.
bus	44.25	1254523	31.05	39	44.24	1255497	19.93	25
city	44.29	1022852	26.41	27	44.27	1024053	19.79	20
crew	44.82	779480	20.66	16	44.81	777037	18.97	15
football	44.61	997640	25.33	26	44.59	987936	19.45	19
foreman	44.38	813195	22.76	19	44.36	806063	18.72	15
harbour	44.10	1279309	30.49	39	44.09	1268153	20.01	26
ice	46.47	472573	24.64	12	46.46	469323	17.72	8
mobile	44.44	1768771	36.17	65	44.43	1753381	19.80	35
soccer	44.27	922527	23.42	22	44.21	902847	19.94	18

TABLE I: Analysis of ES for SE for different benchmark video sequences at QP value 18.

QP	SE-CAVLC				SE-CABAC			
	PSNR (dB)	Total Size (Bytes)	ES (%)	Avg. ES Bits/MB.	PSNR (dB)	Total Size Bytes	ES (%)	Avg. ES Bits/MB.
12	50.07	1260001	28.55	36	50.05	1257024	19.97	25
18	44.38	813195	22.76	19	44.36	806063	18.72	15
24	39.43	478496	17.13	8	39.42	464794	17.61	8
30	35.08	268012	13.24	4	35.08	255287	15.65	4
36	31.04	145736	9.88	1	31.06	134143	12.22	2
42	27.23	88333	6.70	1	27.35	70616	9.46	1

 TABLE II: Analysis of ES for SE over whole range of QP values for *foreman* video sequence.


(a)



(b)

 Fig. 13: Framewise time taken by SE-CABAC of *foreman* video sequence for I+P frames at QP value 18 with *intra period* 10 during: a) Encoding, b) Decoding.

Table III gives a detailed overview of the required processing power for I and I+P video sequences at QP value 18. *intra period* has been set 10 for I+P video sequences. One can observe that increase in computation time for encoder is less than 0.4% for both of SE-CAVLC and SE-CABAC while it is below than 3% for decoder for I+P sequence.

Fig. 13.a and 13.b show the framewise analysis of increase in processing power for SE-CABAC at QP value 18 for *foreman*. For experimentation, 2.1 GHz Intel Core 2 Duo T8100 machine with 3072 MB RAM has been used. For I+P sequence encoding of 100 frames with *intra period* 10, it took 4372.5 seconds and 4381.3 seconds for CABAC and SE-CABAC respectively. While it took 2.005 seconds and 2.045 seconds for CABAC and SE-CABAC decoding. It is a negligible increase in processing power and can be managed well even by hand-held devices. It is important to note that increase in processing power of SE-CABAC is less than SE-CAVLC owing to two reasons. First, ES of SE-CABAC is lesser than that of SE-CAVLC as shown in Table I and Table II. Second, CABAC takes lot more processing power than CAVLC. So increase in processing power because of encryption will be lower in terms of percentage. Thus, SE-CAVLC and SE-CABAC is possible in real-time along with compression.

B. PSNR and Quality of SE-CAVLC & SE-CABAC for I Frames and I+P Frames

Peak signal to noise ratio (PSNR) is widely used objective video quality metric. However, it does not perfectly correlate with a perceived visual quality due to non-linear behavior of human visual system. Structural similarity index (SSIM) [33] takes into account the structural distortion measurement, since human vision system is highly specialized in extracting structural information from the viewing field. SSIM has a better correlation to the subjective impression. SSIM ranges from -1 to 1 . SSIM is 1 when both the images are the same.

To present the visual protection of encrypted video sequences, PSNR and SSIM of I and I+P frames are presented.

1) *I Frames*: To demonstrate the efficiency of our proposed scheme, we have compressed 100 I frames of each sequence at 30 fps. Fig. 14 and Fig. 15 show the encrypted first frame of *foreman* video sequence at different QP values for SE-CAVLC and SE-CABAC respectively. In H.264/AVC, blocks on the top array are predicted only from left while blocks on left are always predicted from top. Owing to this prediction, a band having width of 8 pixels at top of video frames can be observed for both of SE-CAVLC and SE-CABAC while this band has width of 4 pixels on left of video frames as shown in Fig. 14 and Fig. 15. The average PSNR values of *foreman* is given in Table IV over whole QP range. It is also compared with the PSNR obtained for the same video sequence without encryption. In Table IV we present PSNR of original video only for CAVLC. PSNR for CABAC is very much similar as presented in Table I. One can note that whatever is the QP value, the quality of the encrypted video remains in the same lower range.

Table V compares the average PSNR of 100 I frames of all benchmark video sequences at QP value 18 without encryption and with SE. Average PSNR value of *luma* for all the sequences at QP value 18 is 9.49 dB for SE-CAVLC and 9.80 dB for SE-CABAC. It confirms that this algorithm works well for various combinations of motion, texture and objects for I frames. It is also evident in frame-wise PSNR of *luma* of I frames of *foreman* video sequence as shown in Fig. 16.

Table VI contains the experimental results of SE of 100 I frames for SD resolution. Here, Average PSNR value of *luma* is 9.82 dB for SE-CAVLC and 9.83 dB for SE-CABAC, which is almost the same as that of QCIF resolution. It is evident that this algorithm is capable to encrypt high quality information at all resolutions. For the rest of the section, we present analysis for QCIF

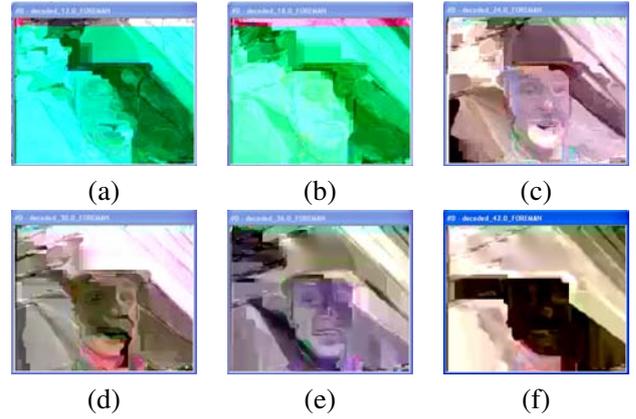


Fig. 14: Decoding of SE-CAVLC frame #1 of *foreman* video sequence with QP value equal to: a) 12, b) 18, c) 24, d) 30 e) 36, f) 42.

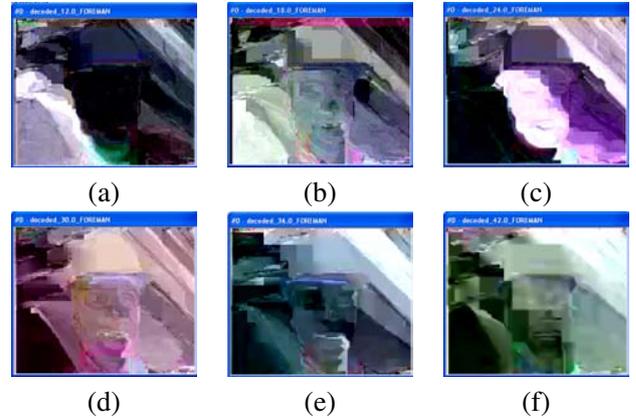


Fig. 15: Decoding of SE-CABAC frame #1 of *foreman* video sequence with QP value equal to: a) 12, b) 18, c) 24, d) 30, e) 36, f) 42.

resolution only, since more benchmark video sequences are available in this resolution.

Table VII shows the SSIM values of *luma* of benchmark video sequences without encryption and with SE. Results verify the proposed scheme has distorted the structural information present in the original video. Average SSIM value of video sequences without encryption is 0.993 , while it is 0.164 and 0.180 for SE-CAVLC and SE-CABAC respectively. Fig. 17 shows the frame-wise SSIM of *luma* of *foreman* video sequence for I frames. It is important to note SSIM value of complex video sequences is less than that of simple video sequences.

2) *I+P Frames*: Video data normally consists of an I frame and a trail of P frames. I frames are inserted periodically to restrict the drift because of lossy compression and rounding errors. In these experiments, *intra period* is set at 10 in a sequence of 100 frames. Results shown in Table VIII verify the effectiveness of our scheme over the

QP	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC
12	50.07	8.61	8.43	50.00	19.78	24.09	50.79	9.57	22.58
18	44.38	8.67	8.58	45.68	24.14	24.40	47.57	10.16	22.10
24	39.43	8.71	8.72	41.93	26.39	24.35	44.19	24.91	22.84
30	35.08	9.43	8.69	39.75	27.45	24.58	41.41	25.36	23.64
36	31.04	9.37	8.53	37.74	28.12	24.93	38.62	24.78	23.16
42	27.23	9.45	8.67	36.23	25.51	24.91	36.86	24.59	24.02

TABLE IV: PSNR comparison for I frames without encryption and with SE for *foreman* at different QP values.

Seq.	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC
bus	44.25	7.90	8.18	45.22	26.82	24.95	46.55	26.65	27.25
city	44.29	10.90	11.23	45.84	31.89	30.27	46.83	33.47	31.80
crew	44.82	8.96	9.90	45.84	23.99	23.45	45.70	19.74	19.79
football	44.61	11.48	11.49	45.77	14.85	14.39	46.05	24.28	23.59
foreman	44.38	8.67	8.58	45.68	24.14	24.40	47.57	10.16	22.10
harbour	44.10	9.25	9.50	45.61	27.07	24.61	46.67	33.25	31.31
ice	46.47	10.59	10.40	48.81	24.26	25.58	49.28	16.86	20.39
mobile	44.44	8.32	8.29	44.15	10.44	13.08	44.06	9.58	10.97
soccer	44.27	9.34	10.61	46.62	22.10	19.73	47.93	28.21	24.41
avg.	44.63	9.49	9.80	45.95	22.84	22.27	46.74	22.47	23.51

TABLE V: PSNR comparison for I frames without encryption and with SE at QP value 18.

Seq.	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC
city	44.65	9.94	10.12	47.82	27.34	26.20	49.07	31.37	29.92
crew	45.15	9.16	9.08	46.56	24.52	22.80	47.74	20.14	19.97
harbour	44.54	9.35	9.37	47.48	22.91	22.92	48.73	28.79	26.81
ice	46.17	10.67	10.38	51.50	27.79	27.72	52.01	25.04	26.09
soccer	45.12	9.96	10.19	47.68	18.36	18.02	49.21	26.68	24.08
avg.	45.13	9.82	9.83	48.21	24.18	23.53	49.35	26.40	25.37

TABLE VI: PSNR comparison for I frames without encryption and with SE at QP value 18 (SD resolution).

Seq.	ORIG-CAVLC	SE-CAVLC	ORIG-CABAC	SE-CABAC
bus	0.995	0.069	0.994	0.064
city	0.994	0.115	0.994	0.093
crew	0.991	0.184	0.991	0.153
football	0.991	0.219	0.991	0.184
foreman	0.990	0.198	0.990	0.165
harbour	0.998	0.047	0.998	0.038
ice	0.990	0.419	0.990	0.398
mobile	0.998	0.040	0.998	0.356
soccer	0.988	0.185	0.988	0.171
avg	0.993	0.164	0.993	0.180

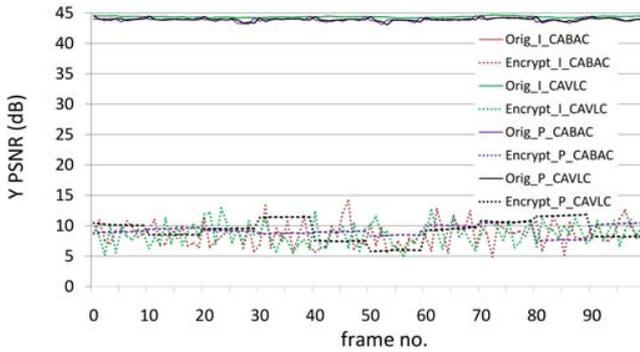
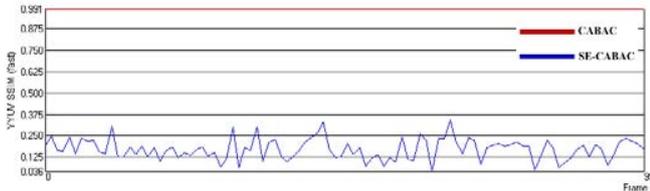
TABLE VII: SSIM comparison of *luma* of I frames without encryption and with SE at QP value 18.

QP	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC
12	49.55	8.73	8.11	49.89	18.35	22.98	50.63	10.41	21.63
18	43.93	9.14	10.44	45.53	23.56	23.87	47.56	8.03	23.19
24	38.92	9.60	9.72	42.04	26.93	24.87	44.27	25.77	24.98
30	34.60	9.24	9.25	39.84	28.61	24.95	41.54	26.63	24.03
36	30.72	10.09	8.19	37.91	28.45	24.28	38.75	22.78	23.36
42	26.95	9.44	8.64	36.30	26.46	26.82	36.92	25.60	24.65

TABLE VIII: PSNR comparison for I+P frames without encryption and with SE for *foreman* at different QP values.

Seq.	PSNR (Y) (dB)			PSNR (U) (dB)			PSNR (V) (dB)		
	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC	ORIG	SE-CAVLC	SE-CABAC
bus	43.73	7.58	7.72	45.10	27.15	25.42	46.42	24.73	27.01
city	43.81	11.42	11.14	45.73	32.47	30.16	46.76	32.53	31.66
crew	44.46	8.97	10.00	45.81	25.09	21.98	45.73	19.63	20.18
football	44.16	12.13	11.28	45.72	14.31	14.58	46.06	24.77	24.27
foreman	43.93	9.14	10.44	45.53	23.56	23.87	47.56	8.03	23.19
harbour	43.71	9.46	9.78	45.45	24.53	22.93	46.58	33.87	31.67
ice	46.14	10.93	10.38	48.61	23.63	25.29	49.14	19.17	19.71
mobile	43.85	8.44	8.84	44.16	10.09	12.48	44.07	9.61	11.85
soccer	43.56	9.65	10.56	46.47	21.83	20.76	47.76	27.40	22.24
avg.	44.15	9.75	10.02	45.84	22.52	21.94	46.68	22.19	23.53

TABLE IX: Comparison of PSNR without encryption and with SE for I+P frames at QP value 18.


 Fig. 16: Framewise PSNR of I and I+P frames for *foreman* for SE-CAVLC and SE-CABAC at QP value 18.

 Fig. 17: Framewise SSIM of I frames for *foreman* for SE-CABAC at QP value 18.

whole range of QP values for *foreman* video sequence. Table IX verifies the performance of our algorithm for all video sequences for I+P frames at QP value 18. Average PSNR of *luma* for all the sequences is 9.75 dB and 10.02 dB for SE-CAVLC and SE-CABAC respectively. Fig. 16 shows the frame-wise PSNR of *luma* of *foreman* video sequence for I+P. Here PSNR of SE-CAVLC and SE-CABAC remains almost the same for sequence of P frames and changes at every I frame, thus producing a staircase graph. SSIM quality metric has very low values and is not given here for the sake of brevity.

C. Security Analysis

1) *Analysis of entropy and local standard deviation:*
The security of the encrypted image can be measured by considering the variations (local or global) in the protected image. Entropy is a statistical measure of randomness or disorder of a system which is mostly used to characterize the texture in the input images. Considering this, the information content of image can be measured with the entropy $H(X)$ and local standard deviation $\sigma(j)$. If an image has 2^k gray levels α_i with $0 \leq i \leq 2^k$ and the probability of gray level α_i is $P(\alpha_i)$, and without considering the correlation of gray levels, the 1st order entropy $H(X)$ is defined as:

$$H(X) = - \sum_{i=0}^{2^k-1} P(\alpha_i) \log_2(P(\alpha_i)). \quad (4)$$

If the probability of each gray level in the image is $P(\alpha_i) = \frac{1}{2^k}$, then the encryption of such image is robust against statistical attacks of 1st order, and thus $H(X) = \log_2(2^k) = k$ bits/pixel. In the image the information redundancy r is defined as:

$$r = k - H(X). \quad (5)$$

Similarly the local standard deviation $\sigma(j)$ for each pixel $p(j)$ taking account of its neighbors to calculate the local mean $\overline{p(j)}$, is given as:

$$\sigma(j) = \sqrt{\frac{1}{m} \sum_{i=1}^m (p(i) - \overline{p(j)})^2}, \quad (6)$$

where m is the size of the pixel block to calculate the local mean and standard deviation, and $0 \leq j < M$, if M is the image size.

In case of full encryption, entropy $H(X)$ is maximized with high values of local standard deviation. But in case of SE-CAVLC and SE-CABAC, the video frame is transformed to flat regions with blocking artifacts as depicted in Fig. 14 and Fig. 15. It is generally owing to variation in pixel values at MB boundaries. For

all the benchmark sequences, the average information redundancy r for SE-CAVLC and SE-CABAC sequences is 0.94 and 0.55 respectively, while it is 1.11 for all the original sequences. Despite the fact that SE-CAVLC and SE-CABAC transform the video frames into flat region, the entropy of the encrypted video sequences from equation (4) is higher as compared to original sequences.

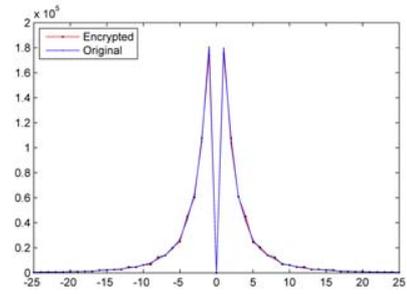
These flat regions are because of two reasons. Firstly, flat regions are due to the fact that prediction is performed from edge pixels of neighboring MBs. Secondly, pixels have either with very high value (bright video frame) or very low value (dark video frame) in SE video frame. This is owing to the fact that during reconstruction pixel value are clipped to 255 if they are greater than it and to 0 if they are below this lower range. So if many pixels have value beyond the upper or lower range, all of them will be clipped to the same value, thus creating a flat region which is either dark or bright. Based on this analysis, the statistical characteristics of SE-CAVLC and SE-CABAC bitstreams vary from full encryption systems. Fig. 18.a, 18.b and 18.c show the histogram of the original and the encrypted NZs for the *foreman* video sequence using for SE-CAVLC. Here we can see that SE has given an effect of staircase because of treating the coefficients with equal probability in the same interval.

From equation (6), we also analyzed the local standard deviation σ for each pixel while taking into account its neighbors. In Table X, the mean local standard deviation for *foreman* sequence at different QP values is given. For all benchmark video sequences, the mean local standard deviation of *luma* equals to 69.15 and 61.48 for the SE-CAVLC and SE-CABAC bitstreams respectively, where the mean local standard deviation is less than 10 gray levels for the original benchmark sequences. One can note that local standard deviation of encrypted sequences is higher than original sequences.

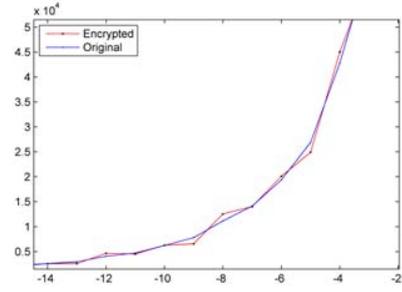
QP	CAVLC		CABAC	
	ORIG	SE-CAVLC	ORIG	SE-CABAC
12	6.75	71.49	7.02	69.69
18	7.21	73.23	7.53	59.97
24	8.57	91.98	8.63	84.55
30	6.35	35.99	6.71	57.87
36	6.90	47.42	6.93	68.04
42	7.91	75.26	8.11	71.17

TABLE X: Standard deviation for SE of *foreman* video sequence at different QP values.

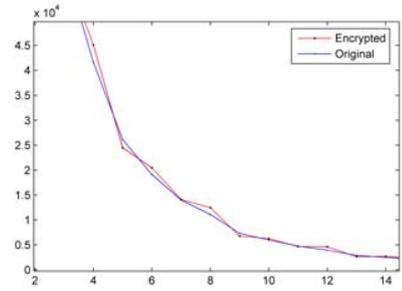
2) *Correlation of adjacent pixels*: Visual data is highly correlated i.e. pixels values are highly probable to repeat in horizontal, vertical and diagonal directions. A correlation of a pixel with its neighboring pixel is then



(a)



(b)



(c)

Fig. 18: Histograms of original and SE-CAVLC NZs: a) Complete graph, b) Zoomed graph of negative x-axis, c) Zoomed graph of positive x-axis.

given by a tuple (x_i, y_i) where y_i is the adjacent pixel of x_i . Since there is always three directions in images i.e. horizontal, vertical and diagonal, so we can define correlation direction between any two adjacent pixels as:

$$corr_{(x,y)} = \frac{1}{n-1} \sum_0^n \left(\frac{x_i - \bar{x}_i}{\sigma_x} \right) \left(\frac{y_i - \bar{y}_i}{\sigma_y} \right), \quad (7)$$

where n represents the total number of tuples (x_i, y_i) , \bar{x}_i and \bar{y}_i represent the local mean and σ_x and σ_y represent the local standard deviation respectively.

Owing to the flat regions in SE-CAVLC and SE-CABAC video sequences, the correlation values in these sequences will be higher as compared to original image which contain texture and edges. For all the benchmark sequences, the average horizontal correlation coefficient

is 0.88 and 0.87 for the SE-CAVLC and SE-CABAC respectively, while it is 0.80 for the original sequences.

3) *Key sensitivity test:* Robustness against cryptanalyst can be improved if the cryptosystem is highly sensitive towards the key. The more the visual data is sensitive towards the key, the more we would have data randomness. For this purpose, a key sensitivity test is assumed where we pick one key and then apply the proposed technique for encryption and then make a one bit change in the key and decode the bitstream. Numerical results show that the proposed technique is highly sensitive towards the key change, that is, a different version of encrypted video sequence is produced when the keys are changed, as shown in Fig. 19. PSNR of *luma* of decrypted frames with 1-bit different key is 10.39 dB and 8.31 dB for SE-CAVLC and SE-CABAC as shown in Table XI. It lies in the same lower range as decoded frames without decryption.

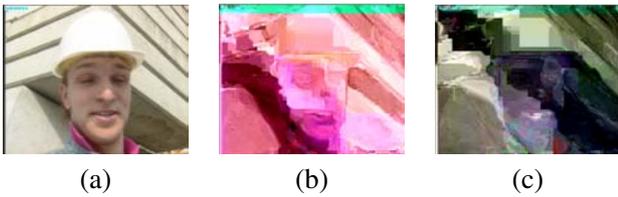


Fig. 19: Key sensitivity test for encrypted frame #1 of *foreman* video sequence for QP value 18. Encrypted frames are decrypted and decoded with: a) original key, b) 1-bit different key(SE-CAVLC), c) 1-bit different key(SE-CABAC).

4) *Removal of encrypted data attack:* In another experiment we have replaced the encrypted bits with constant values in order to measure the strength of SE-CAVLC and SE-CABAC proposed method as described in [27]. Here we have used frame #1 of *foreman* video sequence with QP value 24. Fig. 20 shows both encrypted and attacked video frames for SE-CAVLC and SE-CABAC. For example, Fig. 20.a shows SE-CAVLC video frame with $PSNR = 10.01\text{ dB}$ for *luma*. If we set the encrypted bits of all NZs to zero, we get the video frame illustrated in Fig. 20.b with *luma* $PSNR = 8.87\text{ dB}$. Similarly, Fig. 20.c shows SE-CABAC video frame having $PSNR = 8.20\text{ dB}$ while the attacked SE-CABAC video frame has $PSNR = 7.72\text{ dB}$ as shown in Fig. 20.d.

D. Comparative evaluation

For the sake of comparative evaluation of our scheme, we have compared it with six other recent techniques, which include scrambling [9], NAL unit encryption [14],

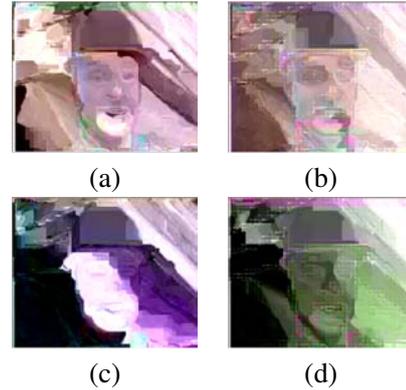


Fig. 20: Attack in the selectively encrypted image by removing the encrypted data: a) SE-CAVLC encrypted image $\{Y, U, V\} = \{10.01, 26.86, 25.24\}\text{ dB}$, b) SE-CAVLC attacked image $\{Y, U, V\} = \{8.87, 27.3, 26.3\}\text{ dB}$, c) SE-CABAC encrypted image $\{Y, U, V\} = \{8.20, 17.95, 24.53\}\text{ dB}$, d) SE-CABAC attacked image $\{Y, U, V\} = \{7.72, 28.6, 24.6\}\text{ dB}$.

MB header encryption [16], reversible ROI encryption [5], I frame encryption [2] and multiple Huffman table permutation [36]. These techniques are different from each other in several aspects e.g. working domain (pixel, transform or bitstream) and encryption algorithm (pseud orandom permutation, stream cipher or AES). The comparison has been made based on several important characteristics of SE systems and is summarized in Table XII.

Encryption algorithm used in SE scheme is of vital importance for the security level. AES has the highest security among all the known ciphers and our proposed scheme utilizes AES. Among the recent techniques, AES has been used only in [2] but their SE scheme is very naive and encrypts only I frames.

Selective encryption should not result in increase of bitrate. For example, if a video for 3G wireless connection has bitrate of 384 kbps. Its encrypted version should have the same bitrate. Otherwise it cannot be played back on 3G connection. Our scheme keeps the bitrate intact. It is in contrast to other schemes which either allow increase in bitrate [9], [5], [36], or use stream cipher for the sake of same bitrate [14], [16], thus compromising on the security of the system.

Format compliance is another important aspect for encrypted video data. Most of the schemes are not format compliant and their encrypted bitstreams cannot be decoded by reference decoder except SE schemes which work in pixel domain [5] and transform domain [9].

Our SE-CABAC scheme is the first format compliant technique which is for arithmetic coding based entropy

	PSNR (Y) (dB)	PSNR (U) (dB)	PSNR (V) (dB)
Original key	44.60	45.73	47.35
SE-CAVLC (1-bit different key)	10.39	24.46	14.02
SE-CABAC (1-bit different key)	8.31	25.13	24.82

TABLE XI: Key sensitivity test of SE-CAVLC and SE-CABAC encrypted video for frame #1 *foreman* video sequence for QP value 18.

coding module, while keeping the bitrate unchanged. Recent encryption techniques for arithmetic coding [13], [11] are not format compliant and require lot of processing power.

To summarize, our proposed schemes (SE-CAVLC and SE-CABAC) meet all the requirements of an integrated compression-encryption systems. Our proposed system is fully compliant to H.264/AVC decoder, with no change in bitrate and has the security of AES cipher.

V. CONCLUSION

In this paper, an efficient SE system has been proposed for H.264/AVC video codec for CAVLC and CABAC. The SE is performed in the entropy coding stage of the H.264/AVC using the AES encryption algorithm in the CFB mode. In this way the proposed encryption method does not affect the bitrate and the H.264/AVC bitstream compliance. The SE is performed in CAVLC *codewords* and CABAC *binstrings* such that they remain a valid *codewords/binstrings* thereafter having exactly the same length. Experimental analysis has been presented for I and P frames. The proposed scheme can be used for B frames without any modification, since B frames are also *inter* frames but have bidirectional prediction. The proposed method has the advantage of being suitable for streaming over heterogeneous networks because of no change in bitrate. The experiments have shown that we can achieve the desired level of encryption, while maintaining the full bitstream compliance, under a minimal set of computational requirements. The presented security analysis confirms a sufficient security level for multimedia applications in the context of SE. The proposed system can be extended for ROI specific video protection [26] for video surveillance and can be applied to medical video transmission [24].

VI. ACKNOWLEDGMENT

This work is in part supported by the VOODOO project (2008-2011) which is a French national project of ANR *Agence Nationale de la Recherche* and the region of Languedoc Roussillon, France.

REFERENCES

- [1] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 / ISO/IEC 14496-10 AVC)," Joint Video Team (JVT), Doc. JVT-G050, Tech. Rep., March 2003.
- [2] M. Abomhara, O. Zakaria, O. Khalifa, A. Zaiden, and B. Zaiden, "Enhancing Selective Encryption for H.264/AVC Using Advanced Encryption Standard," *International Journal of Computer and Electrical Engineering*, vol. 2, no. 2, pp. 223–229, 2010.
- [3] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-Preserving Encryption," in *Proc. 16th Annual International Workshop on Selected Areas in Cryptography*, Calgary, Canada, 2009, pp. 295–312.
- [4] G. Bjontegaard and K. Lillevold, "Context-Adaptive VLC Coding of Coefficients," in *JVT Document JVT-C028*, Fairfax, VA, May 2002.
- [5] P. Carrillo, H. Kalva, and S. Magliveras, "Compression Independent Reversible Encryption for Privacy in Video Surveillance," *EURASIP Journal on Information Security*, vol. 2009, p. 13, 2009.
- [6] H. Cheng and X. Li, "Partial Encryption of Compressed Images and Videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2445, Aug. 2000.
- [7] J. Daemen and V. Rijmen, "AES Proposal: The Rijndael Block Cipher," Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, Tech. Rep., 2002.
- [8] M. V. Droogenbroeck and R. Benedett, "Techniques for a Selective Encryption of Uncompressed and Compressed Images," in *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, Sept. 2002, pp. 90–97.
- [9] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1168–1174, aug. 2008.
- [10] M. M. Fisch, H. Stgner, and A. Uhl, "Layered Encryption Techniques for DCT-Coded Visual Data," in *Proc. 12th European Signal Processing Conference (EUSIPCO'04)*, Vienna, Austria, Sep. 2004, pp. 821–824.
- [11] M. Grangetto, E. Magli, and G. Olmo, "Multimedia Selective Encryption by Means of Randomized Arithmetic Coding," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 905–917, Oct. 2006.
- [12] G. Jakimoski and K. Subbalakshmi, "Cryptanalysis of Some Multimedia Encryption Schemes," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 330–338, April 2008. [Online]. Available: <http://dx.doi.org/10.1109/TMM.2008.917355>
- [13] W. Jiangtao, K. Hyungjin, and J. Villasenor, "Binary arithmetic coding with key-based interval splitting," *IEEE Signal Processing Letters*, vol. 13, no. 2, pp. 69–72, Feb. 2006.
- [14] C. Li, X. Zhou, and Y. Zong, "NAL Level Encryption for Scalable Video Coding," *Lecture notes in Computer Science, Springer*, no. 5353, pp. 496–505, 2008.
- [15] S. Lian, Z. Liu, Z. Ren, and Z. Wang, "Selective Video Encryption Based on Advanced Video Coding," *Lecture notes*

Video Selective Encryption Scheme	Format compliant	Robust to transcoding	Domain	Bitrate increase	Compression independent	Encryption algorithm
Scrambling for privacy protection [9]	Yes	No	Transform	Yes	Yes	Pseudo random sign inversion
NAL unit encryption [14]	No	No	Bitstream	No	No	Stream Cipher
MB header data encryption [16]	No	No	Transform	No	No	Stream Cipher
Reversible encryption of ROI [5]	Yes	Yes	Pixel	Yes	Yes	Pseudo random pixel permutations
I frame encryption [2]	No	No	Bitstream	No	No	AES
Multiple Huffman tables [36]	No	No	Bitstream	Yes	No	Huffman Table permutations
Our scheme	Yes	No	Bitstream*	No	No	AES (CFB mode)

* For SE-CAVLC, bitstream is encrypted, while for SE-CABAC, binstrings are encrypted as explained in Section III-B.

TABLE XII: Comparison of proposed scheme with other recent methods.

in Computer Science, Springer-verlag, no. 3768, pp. 281–290, 2005.

[16] S. Lian, Z. Liu, Z. Ren, and H. Wang, “Commutative Encryption and Watermarking in Video Compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 6, pp. 774–778, June 2007.

[17] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp, “Advances in Digital Video Content Protection,” *Proc. of the IEEE*, vol. 93, no. 1, pp. 171–183, Jan. 2005.

[18] T. Lookabaugh and D. Sicker, “Selective Encryption for Consumer Applications,” *IEEE Communications Magazine*, vol. 42, no. 5, pp. 124–129, May 2004.

[19] R. Lukac and K. Plataniotis, “Bit-Level Based Secret Sharing for Image Encryption,” *Pattern Recognition*, vol. 38, no. 5, pp. 767–772, May 2005.

[20] D. Marpe, H. Schwarz, and T. Wiegand, “Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, July 2003.

[21] K. Martin, R. Lukac, and K. Plataniotis, “Efficient Encryption of Wavelet-Based Coded Color Images,” *Pattern Recognition*, vol. 38, no. 7, pp. 1111–1115, Jul. 2005.

[22] I. Moccagatta and K. Ratakonda, “A Performance Comparison of CABAC and VCL-Based Entropy Coders for SD and HD Sequences,” Joint Video Team (JVT), Doc. JVT-E079r2, Tech. Rep., Oct. 2002.

[23] S. Ou, H. Chung, and W. Sung, “Improving the Compression and Encryption of Images Using FPGA-Based Cryptosystems,” *Multimedia Tools and Applications*, vol. 28, no. 1, pp. 5–22, Jan 2006.

[24] W. Puech and J. Rodrigues, “A New Crypto-Watermarking Method for Medical Images Safe Transfer,” in *Proc. 12th European Signal Processing Conference (EUSIPCO’04)*, Vienna, Austria, 2004.

[25] J.-M. Rodrigues, W. Puech, and A. Bors, “A Selective Encryption for Heterogenous Color JPEG Images Based on VLC and AES Stream Cipher,” in *Proc. European Conference on Colour in Graphics, Imaging and Vision (CGIV’06)*, Leeds, UK, Jun. 2006, pp. 34–39.

[26] —, “Selective Encryption of Human Skin in JPEG Images,” in *Proc. IEEE Int. Conf. on Image Processing, Atlanta, USA*, Oct. 2006, pp. 1981–1984.

[27] A. Said, “Measuring the Strength of Partial Encryption Scheme,” in *Proc. IEEE Int. Conf. on Image Processing, Genova, Italy*, vol. 2, 2005, pp. 1126–1129.

[28] B. Schneier, *Applied cryptography*. Wiley, New-York, USA, 1995.

[29] Z. Shahid, M. Chaumont, and W. Puech, “Fast Protection of H.264/AVC by Selective Encryption,” in *SinFra 2009, Singaporean-French IPAL Symposium, Fusionopolis*, Singapore, 18-20 Feb. 2009.

[30] —, “Fast Protection of H.264/AVC by Selective Encryption of CABAC for I & P frames,” in *Proc. 17th European Signal Processing Conference (EUSIPCO’09)*, Glasgow, Scotland, Aug. 2009, pp. 2201–2205.

[31] D. R. Stinson, *Cryptography: Theory and Practice, (Discrete Mathematics and Its Applications)*. New York: Chapman & Hall/CRC Press, November 2005.

[32] L. Tang, “Methods for Encrypting and Decrypting MPEG Video Data Efficiently,” in *Proc. ACM Multimedia*, vol. 3, New York, NY, USA, 1996, pp. 219–229.

[33] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.

[34] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin, “A Format-Compliant Configurable Encryption Framework for Access Control of Video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 545–557, Jun. 2002.

[35] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h.264/AVC video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[36] C.-P. Wu and C.-C. Kuo, “Design of Integrated Multimedia Compression and Encryption Systems,” *IEEE Transactions on Multimedia*, vol. 7, pp. 828–839, 2005.

[37] K. Yabuta, H. Kitazawa, and T. Tanaka, “A New Concept of Security Camera Monitoring with Privacy Protection by Masking Moving Objects,” in *Proc. Advances in Multimedia Information Processing*, vol. 1, no. LNCS 3767, 2005, pp. 831–842.

[38] W. Zeng and S. Lei, “Efficient Frequency Domain Selective Scrambling of Digital Video,” *IEEE Transactions on Multimedia*, vol. 5, pp. 118–129, 2003.

[39] S. Ziauddin, I. U. Haq, and M. A. Khan, “Method and System for Fast Context based Adaptive Binary Arithmetic Coding,” Patent US7 221 296, 2007.

Considering the reconstruction loop for data hiding of intra- and inter-frames of H.264/AVC

Zafar Shahid · Marc Chaumont · William Puech

Received: 7 April 2010 / Revised: 26 February 2011 / Accepted: 8 March 2011
© Springer-Verlag London Limited 2011

Abstract This paper presents and analyzes a new approach to data hiding that embeds in both the *intra*- and *inter*-frames from the H.264/AVC video codec. Most of the current video data hiding algorithms take into account only the *intra*-frames for message embedding. This may be attributed to the perception that *inter*-frames are highly compressed due to the motion compensation, and any embedding message inside these may adversely affect the compression efficiency significantly. Payload of the *inter*-frames is also thought to be less, compared with the *intra*-frames, because of the lesser residual data. We analyze data hiding in both *intra*- and *inter*-frames over a wide range of QP values and observe that the payload of the *inter* is comparable with that of the *intra*-frames. Message embedding, in only those non-zero quantized transform coefficients (QTCs) which are above a specific threshold, enables us to detect and extract the message on the decoding side. There is no significant effect on the overall bitrate and PSNR of the video bitstream because instead of embedding message in the compressed bitstream, we have embedded it during the encoding process by taking into account the reconstruction loop. For the non-zero QTCs, in the case of *intra*-frames, we benefit from the spatial masking, while in the case of *inter*-frames, we exploit the motion and texture masking. We can notice that the data hiding is done during the compression process and the proposed scheme takes into account the reconstruction loop. The proposed scheme does not target robustness and the obtained

payload is higher, with a better trade-off in terms of quality and bitrate, as compared with previous works.

Keywords Video data hiding · Non-zero quantized transform coefficients · H.264/AVC · Reconstruction loop · Intra- and inter-frames

1 Introduction

Many multimedia applications have emerged in the last decade thanks to the rapid growth in processing powers and network bandwidths. The relative ease, with which digital data can be copied or modified, necessitates its proper protection and authentication. Digital video watermarking has emerged as an important research field to protect the copyrighted multimedia data. Watermarking, steganography, and more generally data hiding are used in many applications for owner identification, copyright protection, integrity, and metadata embedding. For a video codec, data hiding can be carried out in either spatial or frequency domain. Data embedded in spatial domain can be lost because of the lossy stage of quantization. In the frequency domain, data hiding is done normally in the QTCs. Normally, for large videos with real-time constraints, the data hiding process is made part of the video encoder. In this context, few specific methods have been developed for the MPEG video standards [1, 13]. The purpose of this paper is to investigate the payload capacity of *intra*- and *inter*-frames, since a typical video consists of an *intra* followed by a trail of *inters*. Challenge lies in the fact that the bitrate may rise significantly because of the message embedding. To overcome this limitation, the message has been embedded in only those QTCs which have a magnitude beyond a certain threshold.

Z. Shahid · M. Chaumont · W. Puech (✉)
LIRMM Laboratory, UMR 5506 CNRS, University of Montpellier II,
161, rue Ada, 34392 Montpellier Cedex 05, France
e-mail: william.puech@lirmm.fr

Z. Shahid
e-mail: zafar.shahid@lirmm.fr

M. Chaumont
e-mail: marc.chaumont@lirmm.fr

In H.264/AVC video codec, both the intra- and inter-predictions should be taken into account by embedding the hidden message inside the reconstruction loop. It is crucial, especially while streaming over heterogeneous networks, to keep the bitrate intact. Hence, bitrate escalation, due to the message embedding, must be taken into account by the rate distortion module.

The rest of the paper is organized as follows. In Sect. 2, first, we present the H.264/AVC video codec, with its integer transform (IT) and the quantization process, followed by an overview of the previous watermarking and data hiding techniques related to this video standard. We present, in Sect. 3, the proposed method by elaborating its embedding and extraction steps while taking into account the reconstruction loop. Sect. 4 contains the experimental results and a performance analysis of both *intra*- and *inter*-frames after embedding in more than one least significant bits (LSBs). In the said section, we also present a comparison of message embedding in a given video bitstream inside and outside the reconstruction loop. Finally, in Sect. 5, we present some concluding remarks about the proposed method.

2 H.264/AVC data hiding, challenges and prospects

Since significant changes have been incorporated in the H.264/AVC standard as compared to the previous video coding standards. An overview of H.264/AVC, with an emphasis on transform and quantization, is presented in Sect. 2.1. It is followed, in Sect. 2.2, by an overview of the previous watermarking and data hiding techniques already proposed in the literature for H.264/AVC. We have used capital letters to represent matrices e.g., A , Y , W and small letters along with index to represent the elements of matrices e.g., $x(i, j)$ represents j th element in i th row of matrix X .

2.1 Overview of H.264/AVC

The H.264/AVC standard [12] has some additional features as compared to previous video standards. The *baseline* standard has a 4×4 transform in contrast to 8×8 transform of the previous standards. DCT transform has been replaced by the integer transform (IT), which can be implemented by just additions and shifts in 16-bit arithmetic without any multiplication and hence requires lesser number of computations. The H.264/AVC codec uses a uniform scalar quantization. For *inter*-frame, H.264/AVC supports variable block size motion estimation, quarter pixel accuracy, multiple reference frames, improved skipped, and direct motion inference. For *intra*-frame, it offers additional spatial prediction modes. All these additional features of H.264/AVC are aimed at outperforming the previous video coding standards [36]. The block diagram of H.264/AVC is shown in Fig. 1.

The 4×4 IT has two main advantages. Firstly, it can be exhaustively implemented with simple 16-bit additions and shifts. Secondly, in contrast to floating point arithmetic, which gives different results on different platforms, there is no problem of mismatch on the encoder and decoder side for the integer arithmetic. A macro-block (MB) is divided into 16 blocks of 4×4 pixels which are processed one by one.

In the *intra*-mode, H.264/AVC has three alternatives, namely, *Intra* $_4 \times 4$, *Intra* $_{16} \times 16$, and *I_PCM*. In *Intra* $_{16} \times 16$ mode, Hadamard transform is additionally employed to encode the DC coefficients. In the *Intra* $_{16} \times 16$ mode, the entire MB is predicted from top and left neighboring pixels and has 4 modes namely *horizontal*, *vertical*, *DC*, and *plane* modes. In the *Intra* $_4 \times 4$ mode, each 4×4 luma block is predicted from top and left pixels of the reconstructed 4×4 neighbors. This alternative has nine different prediction modes. The *I_PCM* mode is used to limit the maximum size of the encoded block and is directly entropy encoded by skipping the transform and quantization stages. The scanning of these 4×4 blocks, inside MB, is not in a raster scan fashion, as illustrated with the help of numbers in Fig. 2. In the case of *Intra* $_{16} \times 16$ mode, Hadamard transform coefficients are sent first.

Transform and quantization process are embedded with each other to save the processing power and to avoid multiplications. Let a 4×4 block is defined as $X = \{x(i, j) | i, j \in \{0, 3\}\}$ as shown in Fig. 1. $x(i, j)$ is predicted from its neighboring blocks, and we get the residual block:

$$e(i, j) = P(x(i, j), b_1(i, j), b_2(i, j), b_3(i, j), b_4(i, j)), \quad (1)$$

where $b_k(i, j)$ are the pixels from the reconstructed top and left blocks from intra-prediction, and $P(\cdot)$ is the prediction function. For example, for a vertical prediction mode, the prediction will be performed from top block as $P(x, a, b, c, d) = x - a$, where a is the reconstructed block at top. From [25], the forward and inverse IT 4×4 matrices (A, A_{inv}) are as follows:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} A_{inv} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 1 & -1/2 \end{bmatrix}. \quad (2)$$

The residual block E is then transformed using the following equation:

$$Y = AEA^T, \quad (3)$$

where $E = \{e(i, j) | i, j \in \{0, 3\}\}$ is in the spatial domain, and $Y = \{y(i, j) | i, j \in \{0, 3\}\}$ is in the frequency domain. Scalar multiplication and quantization are defined as:

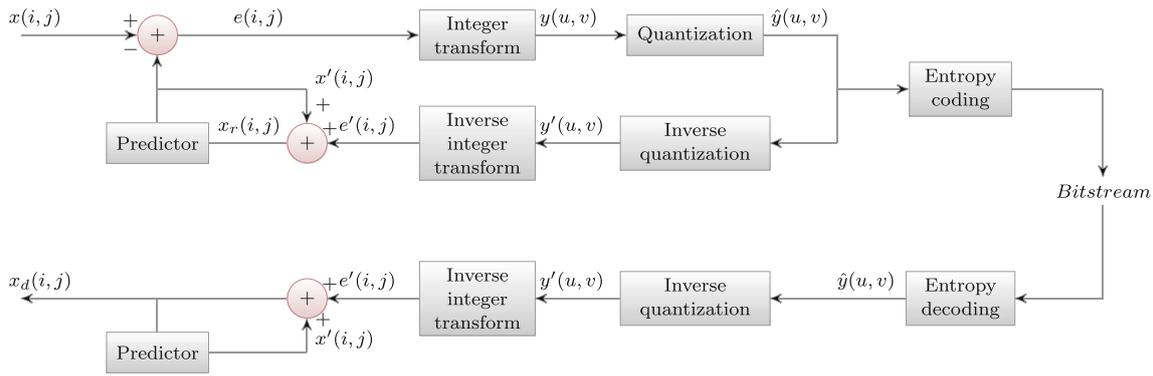


Fig. 1 Detailed block diagram explaining the prediction, the transform and the quantization steps in the H.264/AVC

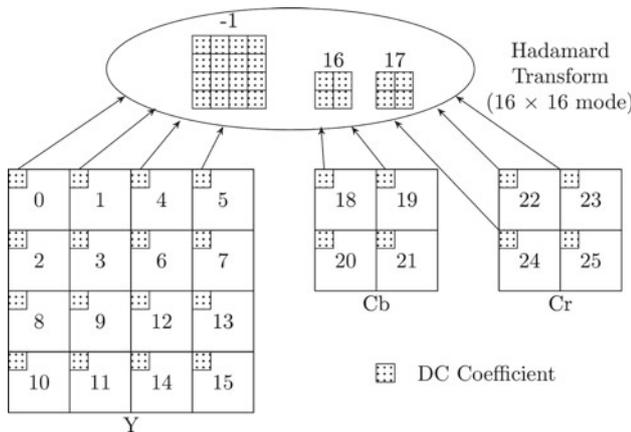


Fig. 2 Order of transmission of the luma and the chroma *Intra*_{4 × 4} blocks inside MB

$$\hat{y}(u, v) = \text{sign}\{y(u, v)\} [(|y(u, v)| \times Aq(u, v) + Fq(u, v) \times 2^{15+Eq(u,v)}) / 2^{(15+Eq(u,v))}], \quad (4)$$

where $\hat{y}(u, v)$ is a QTC, $Aq(u, v)$ is the value from the 4×4 quantization matrix, and $Eq(u, v)$ is the shifting value from the shifting matrix. Both $Aq(u, v)$ and $Eq(u, v)$ are indexed by QP. $Fq(u, v)$ is the rounding off factor from the quantization rounding of factor matrix. This $\hat{y}(u, v)$ is entropy coded and sent to the decoder side.

On the decoder side, inverse quantization is carried out according to the expression:

$$y'(u, v) = \{ [(\hat{y}(u, v) \times (Bq(u, v) \times 2^4)) \times 2^{Eq(u,v)}] + 2^3 \} / 2^4,$$

where $Bq(u, v)$ and $Eq(u, v)$ are the values from the inverse 4×4 quantization matrix and the shifting factor, respectively. $y'(u, v)$ is then inverse transformed to get $E' = (A_{inv} Y' A_{inv}^T + 2^5) / 2^6$. The decoded residual signal $e'(i, j)$ is then added to the predicted signal to reconstruct the original signal back.

2.2 Previous work on video watermarking and data hiding

Many digital watermarking and data hiding schemes have been proposed in the literature for both image and video

data. Data hiding methods can be classified into three broad categories namely robust [6, 20, 33, 38, 39], semi-fragile [14] and fragile [22]. Different watermarking and data hiding techniques offer various combinations of rate, distortion, and robustness. For each application, a particular watermark algorithm can be selected depending on its requirements. For example, applications for copyright protection would require using a robust watermark, while applications for proving integrity would employ a fragile or semi-fragile watermark.

As far as standard video codecs are concerned, five encoding stages can be identified where embedding can take place, namely the pre-compression stage, the video codec structure, the transform stage, the quantization step, and the bitstream as illustrated in Fig. 3. The embedding is primarily motivated by the goal of integrating data hiding and compression to reduce overall real-time video processing complexity.

In the pre-compression stage, marked as stage 1 in Fig. 3, message embedding is performed before the compression process [5, 8, 9, 15, 23, 26]. Embedding can be performed either in the pixel domain or in some transform domain e.g., DCT, DFT, DWT. Temporal aspect of the video can also be exploited for watermark embedding by taking into account multiple frames at the same time [3, 30]. In spatial domain, LSB modification is a very simple method to embed hidden message into the cover object [17]. This method may survive against attacks such as cropping but any addition of noise or lossy compression is likely to defeat the message extraction. In [17], a method has been proposed to use PRNG to decide the pixels for LSB substitution. This may improve the security but still vulnerable to the substitution of the LSB(s) with a constant value. LSB modification may be imperceptible but statistically still discernible. In [2], an upper bound for the LSB payload has been defined so that it remains statistically invisible.

In [4], the transform domain has been employed for LSB embedding. Here, the signature image is first quantized using vector quantization in order to hide larger message. The bit-by-bit message is then directly embedded into LSB of DCT coefficients of the image. Using the embedding replicate of

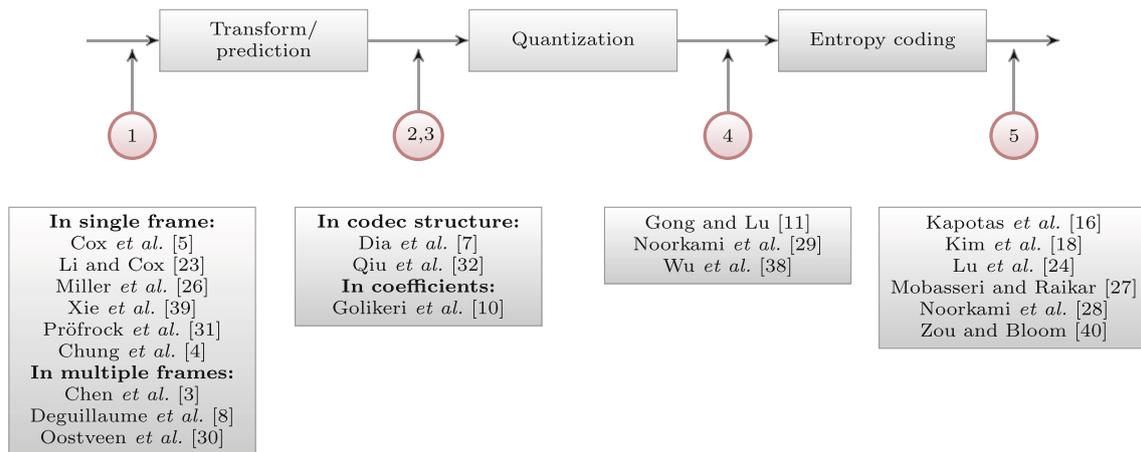


Fig. 3 Classification of the watermarking schemes on the basis of the working domain: 1 pre-compression, 2 video codec structure, 3 transform domain, 4 Quantized transform domain, 5 bitstream

the message provides further robustness against the signal processing attacks.

In [31], Pröfrock *et al.* have presented a watermarking technique in the spatial domain, which is robust to H.264/AVC video compression. Hidden message is embedded in the perceptually significant parts of the video in an imperceptible manner, by changing the spatial position of the object borders. Borders are defined by new normed center of gravity (NCG). Influence of lossy compression on NCGs is predicted, and watermark is embedded with enough robustness to compensate the lossy compression. A geometric warping process is proposed to quantize the NCG and embeds the watermark payload with a defined robustness. Xie *et al.* [39] have also proposed robust watermarking based on the on-off keying scheme which uses the DWT transform to embed the watermark.

The video codec structure is the second candidate domain for data hiding. Some researchers have proposed to embed the message in motion vectors [7, 32]. In [21], Li *et al.* propose to perform robust watermarking for H.264/AVC by embedding the hidden message in a new syntax element named reference index. They also modify the current block to improve the robustness of the scheme by a geometric method with the least degradation in the video quality. These video watermarking techniques are vulnerable to re-encoding and conversion to other video codecs.

Hidden message can also be embedded in the transformed coefficients before quantization, as proposed by Golikeri *et al.* [10]. This kind of approach is illustrated by stage 3 in Fig. 3. They have used the visual models, developed by Watson [35], to choose the coefficients to be watermarked based on their frequency sensitivity, luminance masking, and contrast masking.

Some researchers have proposed algorithms to embed hidden message in the QTCs of H.264/AVC, as shown by stage 4

in Fig. 3. For example, Noorkami and Merserau [29] have presented a technique to embed message in both *intra*- and *inter*-frames in all the non-zero QTCs. They claim that visual quality of *inter*-frames is not compromised even if we embed message in all the non-zero QTCs. Owing to the embedding of the message, only in the non-zero QTCs, their method does not affect the compression efficiency of the run-length coding. The performance of context-based adaptive variable length coding (CAVLC), however, gets affected, and as a result, a controlled increase in the bitrate is eventually observed, since there are a lot of QTCs whose magnitude is 1 and CAVLC encodes *trailing ones* (T1's) separately. In [11], Gong and Lu embedded watermarks in the H.264/AVC video by modifying the quantized DC coefficients in the *luma* residual blocks. To increase the robustness while maintaining the perceptual quality of the video, a texture-masking-based perceptual model is used to adaptively choose the watermark strength for each block. To eliminate the effects of drift, a drift compensation algorithm is proposed which adds the drift compensation signal before embedding the watermark bit.

To avoid processing intensive decoding followed by re-encoding along with watermarking, some methods have suggested embedding the message into the compressed bitstream [18, 24, 27, 28, 40]. Kim *et al.* [18] suggest to hide the message in the sign bit of the trailing ones in CAVLC of H.264/AVC. Bitrate of the watermarked video remains exactly the same with the resultant PSNR greater than 43 dB. In [27], authentication of the H.264/AVC is performed by the direct watermarking of CAVLC codes. Zou and Bloom [40] have proposed to perform direct replacement of CAVLC. Kapotas *et al.* [16] have presented a data hiding method in H.264 streams for fragile watermarking. It takes advantage of the different block sizes used by the H.264 encoder during the inter-prediction stage, in order to hide the desired data.

The message can be extracted directly from the encoded stream without any need of the original host video. This approach can be mainly used for content-based authentication. Such algorithms face two major limitations. First, payload of such algorithms is very low—of the order of a few bytes per second [13]. Second, there is a continuous drift that degrades the visual quality significantly.

3 The proposed algorithm

In this paper, we have used LSB modification approach in the DCT domain and the hidden message is not embedded in all the non-zero QTCs. Rather, we have embedded the message in only those QTCs which are above a certain threshold. The threshold value depends on the number of message bits being embedded. This offers two advantages. First, it makes it possible to extract the message on the decoder side. Second, it does not affect the compression efficiency of the entropy coding engine significantly. We have not targeted robustness here. Rather, we have demonstrated the high payload capability of the proposed scheme which is very high as compared with other schemes. Hence, the proposed scheme can be used in application where robustness is not required, e.g., broadcasting and hiding of metadata.

In H.264/AVC, intra-prediction is performed in the spatial domain. Hence, even for the *intra*-mode, the transform is performed on prediction residuals. In contrast to previous methods, which embed hidden message in the DC coefficients, we have embedded the message in all those non-zeros QTCs having magnitude above a certain threshold, with the exception of the DC coefficients. Our algorithm is not robust but has a very high payload. For *Intra*₄ × 4 mode, we have not embedded message in the DC QTCs, while for *Intra*₁₆ × 16 mode, we have not modified the Hadamard transform coefficients either, since DC QTCs contain most of the energy and embedding message in these may affect the video quality and the bitrate significantly. We have embedded the message in the LSBs of QTCs keeping in view the following points:

- QTC, which we want to use for data hiding, should be non-zero. If a QTC with zero magnitude becomes non-zero in the course of embedding, it will highly affect the compression efficiency of run-length encoding.
- QTC to be used for data hiding should be preferably greater than 1 because there are many QTC, with magnitude ‘1’ and in CAVLC, they are also encoded as T1’s. Thus, changing of number of T1’s will affect the compression efficiency of CAVLC.
- Finally, the message is embedded in such a fashion that it can be completely extracted on the decoder side.

3.1 The watermark embedding

The embedding process is performed on QTCs of Eq. 4 as:

$$\hat{y}_w(u, v) = f(\hat{y}(u, v), M, [K]), \quad (5)$$

where $f()$ is the data hiding process, M is the hidden message and K is an optional key.

3.1.1 Analysis of embedding after the encoding loop

Message embedding can be done in QTC before the entropy coding, as shown in Fig. 4. It is analogous to embedding the message in a compressed bitstream. This includes two data hiding approaches. The first approach embeds the message in the VLC domain, and the bitstream needs only be entropy decoded to use this approach e.g., as proposed by Lu et al. [24]. Another approach embeds the message in DCT domain, and for this approach, bitstream has to be entropy decoded and inverse quantized. An example of this approach is differential energy watermarking scheme proposed by Langelaar et al. [19].

Embedding the message after the reconstruction loop creates two problems. Firstly, we start reconstruction on the encoder side with QTC $\hat{y}(u, v)$, while on the decoder side we start decoding with watermarked QTC $\hat{y}_w(u, v)$. This may result in a mismatch on the decoder side, which may keep on increasing because of the prediction process. Because of this mismatch, the difference in PSNR may be considerable, even for *intra*-frames, let alone the *inter*-frames. Secondly, the rate distortion (RD) bit allocation algorithm works in the quantization module, and any change in bitrate/quality trade-off, because of the watermarking of QTCs, is not taken into account.

3.1.2 Embedding within the encoding loop

To solve both the aforementioned problems, message embedding should be performed inside the reconstruction loop as shown in Fig. 5. In this case, we have the same watermarked QTC $\hat{y}_w(u, v)$ on both encoder and decoder side for prediction, and the RD bit allocation algorithm is working on $\hat{y}_w(u, v)$ for both *intra*- and *inter*-frames. In the next section, we present the data embedding while taking into account the reconstruction loop.

3.2 Hidden message—aware rate distortion

Many encoding parameters, like the prediction modes, the quantization parameter (QP) value and the motion vectors are adjusted in the encoding process based on the video content and the required quality. Since a video data are very diverse in nature, both spatially and temporally, these parameters vary from scene to scene. The bit allocation algorithms are used to

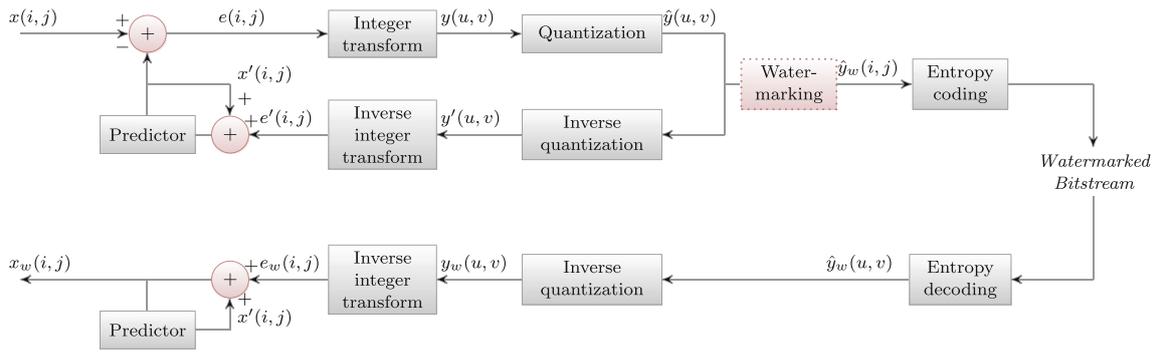


Fig. 4 Block diagram of the H.264/AVC along with the data hiding module outside the reconstruction loop

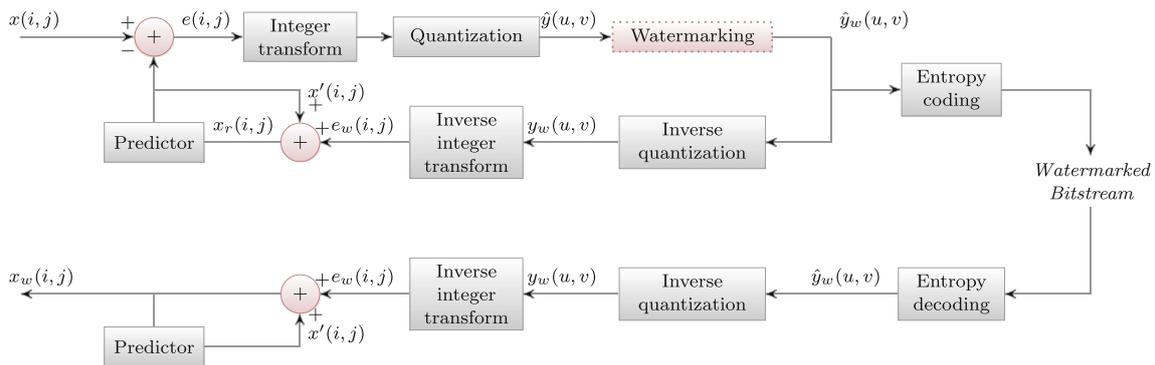


Fig. 5 The proposed data hiding method inside the reconstruction loop

find the best-suited values of these parameters to achieve the trade-off between bitrate and quality. For RD, the Lagrangian bit allocation is widely used owing to its simplicity and effectiveness. The simplified Lagrangian cost function is $J = D + \lambda R$, where J represents the cost of encoding for a given MB, D is the distortion, λ is the Lagrangian parameter, which depends on the QP value, and R is the number of bits to encode a MB. In H.264/AVC, several modes are supported to encode a MB as *intra* or *inter*, as shown in Fig. 6. To obtain the cost J , for a specific prediction mode P , we first predict the MB for that mode to get the residual E . We then apply IT on the residual E , followed by quantization with some QP value, to get the QTCs which are then entropy coded. The number of bits, R , consists of MB header bits and data bits. The residual is, then, reconstructed by performing inverse quantization and inverse IT to give the reconstructed residual E' . Generally, the distortion, D , may be the sum of absolute differences (SAD), the sum of squared differences (SSD), or the sum of absolute transformed differences (SATD) between E and E' . Thus, we end up with the cost J for encoding this MB in the prediction mode, P . In a similar fashion, we find the cost J for all other prediction modes. The mode which yields the minimum cost is selected as the RD optimized mode for this MB.

Embedding a message in a video bitstream affects quality of the picture. It also affects the bitrate because this frame is

used for the prediction after the reconstruction. Hence, RD optimization should take into account the embedding of the hidden message in QTCs in order to select the best prediction mode. In this case, simplified Lagrangian cost function is $J_w = D_w + \lambda R_w$, for finding the cost for a specific prediction mode. QTCs are first watermarked to get QTC_w , which are then entropy coded to find the number of bits R_w to encode MB and reconstructed to measure the distortion D_w . By moving the message-embedding process to the inside of the reconstruction loop constitutes the best suitable mode for the watermarked blocks.

3.3 The embedding strategy

For message embedding in the video bitstream, we developed a strategy to embed message in the 1 LSB, 2 LSBs, or '1 & 2' LSBs together. For the n LSB mode, the hidden message is embedded in a QTC in n LSBs if its magnitude is greater than $2^n - 1$. Owing to this threshold, detection and extraction of the message are performed on the decoder side. Algorithm 1 describes the embedding of 1 watermark bit (WMBit) in the LSB of $|QTC|$. Here, the threshold value is 1. If $|QTC|$ is less than or equal to 1, it will remain unchanged. For $|QTC| \geq 2$, output will either be the same or will get modified by ± 1 , depending on whether the WMBit is '0' or '1'.

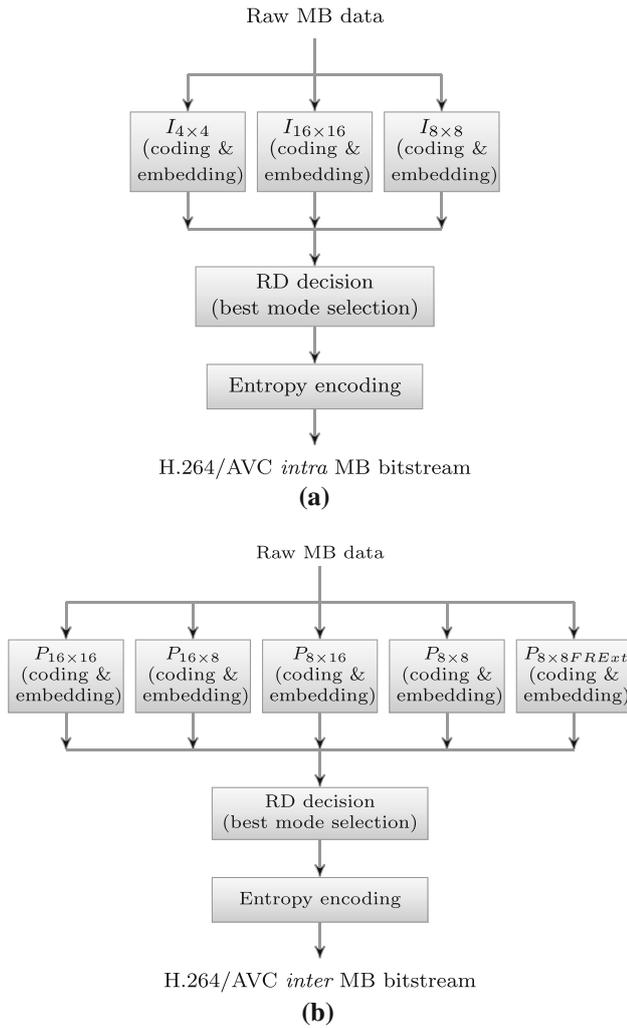


Fig. 6 Different prediction modes in the H.264/AVC for: **a** the *intra*-MBs **b** the *inter*-MBs

In this case, we have a 50% probability that the coefficient will remain unchanged even after being watermarked.

Algorithm 1 The embedding strategy in the 1 LSB.

```

1: if  $|QTC| > 1$  then
2:    $|QTC_w| \leftarrow |QTC| - |QTC| \bmod 2 + WMBit$ 
3: end if
4: end

```

Similarly, Algorithm 2 outlines the embedding of 2 bits in 2 LSBs of a QTC. By keeping the threshold more than ‘3’, we can extract the hidden message on the decoder side successfully. QTC will remain unchanged if $|QTC| < 4$, otherwise it will get modified depending on whether WMBits are ‘00’, ‘01’, ‘10’, or ‘11’. In this case, we have only 0.25 probability that the coefficient will remain unchanged even after being watermarked.

Algorithm 2 The embedding strategy in the 2 LSBs.

```

1: if  $|QTC| > 3$  then
2:    $|QTC_w| \leftarrow |QTC| - |QTC| \bmod 4 + WMBits$ 
3: end if
4: end

```

We can also perform a ‘1 & 2’ LSBs embedding together. In this case, we embed message in 0, ‘1 & 2’ LSBs depending on value of $|QTC|$, as shown in Algorithm 3. So we embed 2 WMBits if $|QTC|$ is high enough or 1 WMBit if $|QTC| > 1$.

Algorithm 3 The embedding strategy in the ‘1 & 2’ LSBs.

```

1: if  $|QTC| > 3$  then
2:    $|QTC_w| \leftarrow |QTC| - |QTC| \bmod 4 + WMBits$ 
3: else
4:   if  $|QTC| > 1$  then
5:      $|QTC_w| \leftarrow |QTC| - |QTC| \bmod 2 + WMBit$ 
6:   end if
7: end if
8: end

```

3.4 The hidden message extraction

During the extraction process, we can extract the message from the watermarked QTCs as follows:

$$M = g(\hat{y}_w(u, v), [K]), \quad (6)$$

where $g()$ is the data hiding detection/extraction process, $\hat{y}_w(u, v)$ is the watermarked QTC, and K an optional key required for extraction. When using ‘1 & 2’ LSBs watermark extraction, $g()$ can be given as shown in Algorithm 4.

Algorithm 4 The extraction strategy using the ‘1 & 2’ LSBs.

```

1: if  $|QTC| > 3$  then
2:    $WMBits \leftarrow |QTC_w| \bmod 4$ 
3: else
4:   if  $|QTC| > 1$  then
5:      $WMBit \leftarrow |QTC_w| \bmod 2$ 
6:   end if
7: end if
8: end

```

4 Experimental results

For experimental simulations, we have used the reference implementation of H.264¹ and applied our method on 150 frames of each of the nine selected standard video sequences

¹ We have used reference software JSVM 10.2 in AVC mode.

in the CIF format. Each of them represents different combinations of motion (fast/slow and pan/zoom/rotation), color (bright/dull), contrast (high/low), and objects (vehicle, buildings, people). The video sequences ‘bus’, ‘city’, and ‘foreman’ contain camera motion, while ‘football’ and ‘soccer’ contain camera panning and zooming along with object motion and texture in the background. The video sequences ‘harbor’ and ‘ice’ contain high luminance images with smooth motion. ‘Mobile’ sequence contains a still complex background and motion in the foreground. In case of the *intra*- and *inter*-sequences, *intra-period* has been set to 15 for all the simulations. In these simulations, the hidden message, being embedded, is of noise type and has been generated using a pseudorandom number generator.

Peak signal to noise ratio (PSNR) is a widely used objective video quality metric. However, it does not perfectly correlate with a perceived visual quality due to the non-linear behavior of the human visual system (HVS). Structural similarity index (SSIM) [34] takes into account the structural distortion measurement, since the HVS is highly specialized in extracting the structural information from the viewing field. SSIM has a better correlation to the subjective impression. SSIM ranges from -1 to 1 . SSIM is 1 when both the images are the same. To present the visual comparison of original and watermarked video sequences, both PSNR and SSIM values are presented.

The detailed results for both *intra*- and *inter*-frames are explained in Sects. 4.1 and 4.2, respectively. They also contain the comparison of our scheme with: (1) the outside loop embedding and (2) the message embedding in LSBs of all the QTCs. A comparative analysis of our proposed method with other recent techniques has been presented in Sect. 4.3.

4.1 Analysis of *intra*-frames

In *intra*-frames, non-zero QTCs are present in those parts which contain texture and edges. These are the spatial masking areas, and the hidden message is naturally embedded in these areas of *intra*-frames. To analyze the effect of message embedding on payload, bitrate and PSNR, the video sequences have been encoded at QP values of 18 and 36.

Figure 7a, b illustrate the payload for each *intra*-frame in the *foreman* for the QP values of 18 and 36 for all of the three data hiding modes, namely the 1 LSB, the 2 LSBs and the 1 & 2 LSBs. At a QP value of 18, we have a large number of QTCs which can be watermarked and hence payload is high for all the modes. At a QP value of 36, we have an adequate number of QTCs for the 1 LSB mode and hence enough number of WMBits to embed. But the payload for the 2 LSBs mode is, however, lower since fewer QTCs have magnitudes above the threshold for this mode. Figure 8a, b show the effect of message embedding on the bitrate. Owing to the fact that we have neither modified the QTCs with magnitude ‘0’ and T1’s,

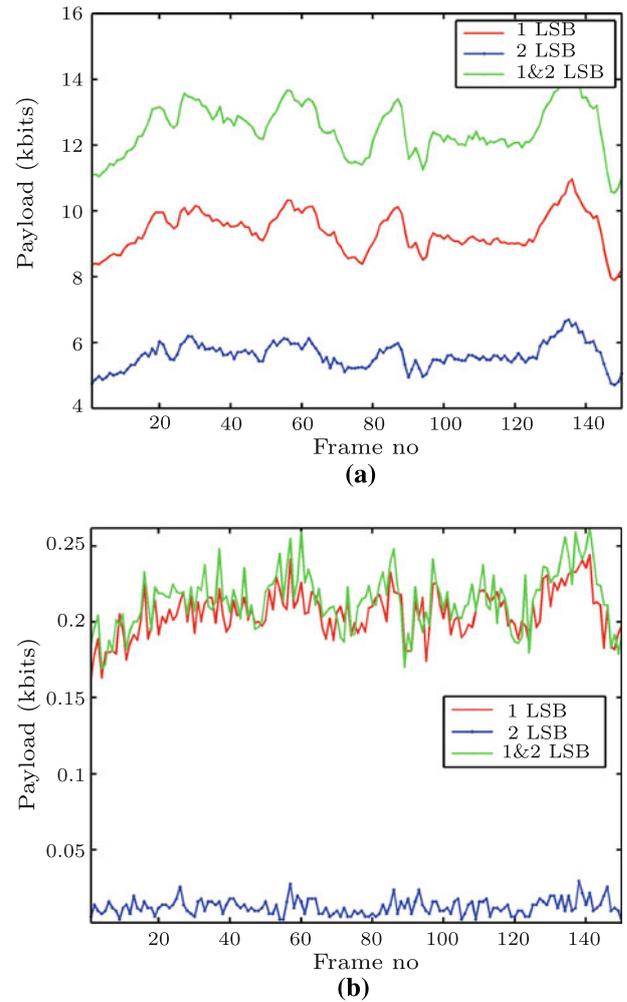


Fig. 7 Analysis of the payload capability for the hidden message embedding of the *intra*-frames in *foreman* for the QP value of: **a** 18 **b** 36

the bitrate has increased only slightly. This increase in bitrate is due to two reasons. One, watermarked reconstructed QTCs are used for the prediction of the future MBs, which results in more residuals and hence increase in the bitrate. Two, the absolute value of QTCs increases gradually in the inverse scan order, and the entropy coding is designed for this distribution. After the WMBit embedding, this order may get disturbed and depends on the WMBits being embedded. With the embedding of the WMBits, the QTCs are modified and hence there is a decrease in the PSNR as shown in Fig. 9a, b. At the QP value of 18, higher number of coefficients are watermarked and hence a greater reduction in the PSNR is observed. While at the QP value of ‘36’, we have lesser QTCs to be watermarked, hence less degradation in the quality is observed. At the QP value of 36, few QTCs have magnitude above threshold, for the 2 LSBs embedding, and very few WMBits can be embedded in this mode. But we have adequate number of QTCs with magnitude greater than 1,

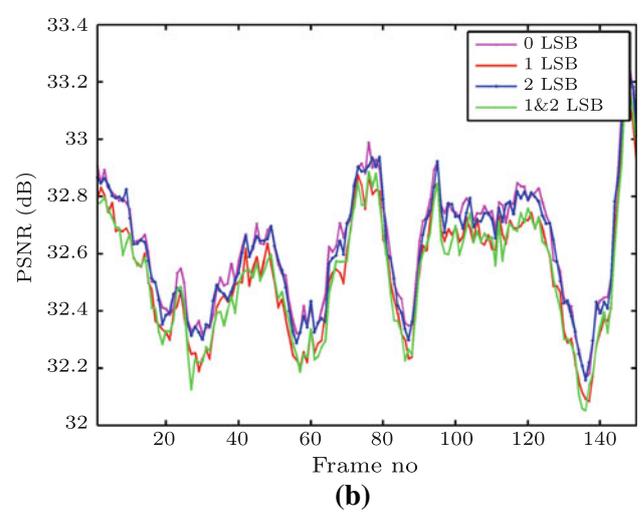
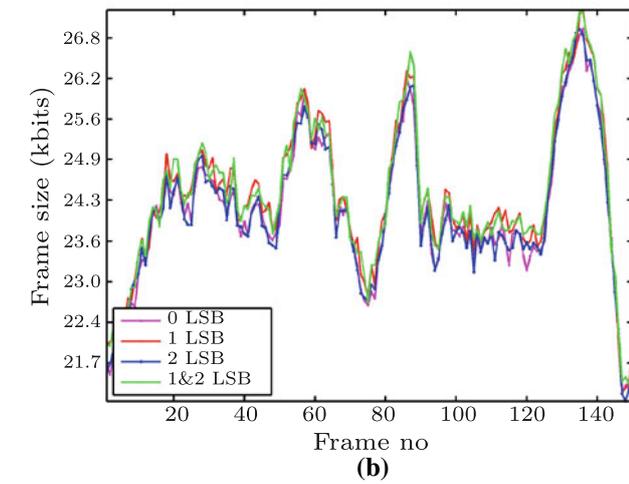
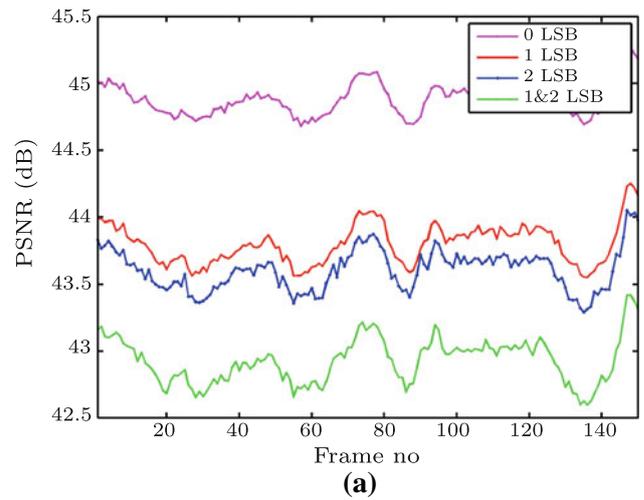
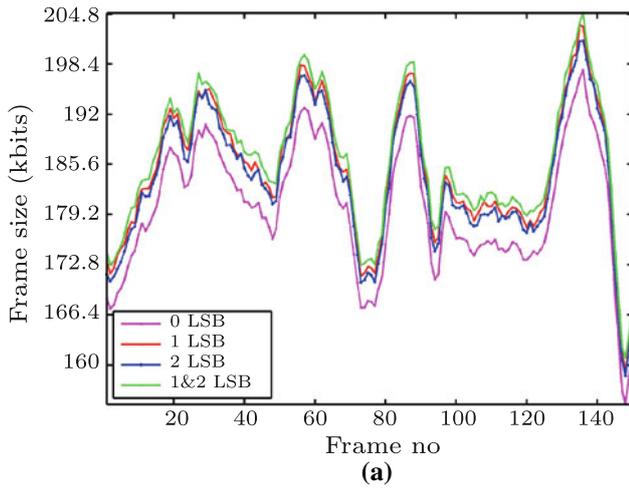


Fig. 8 Analysis of the bitrate variation for the hidden message embedding of the *intra*-frames in *foreman* for the QP values of: **a** 18 **b** 36

Fig. 9 Analysis of the change in PSNR for the hidden message embedding of the *intra*-frames *foreman* for the QP values of: **a** 18 **b** 36

resulting in enough number of WMBits embedded for this mode. Table 1 contains the payload, bitrate, and PSNR analysis for the *foreman* and the *football* sequences at the QP values of 18 and 36. For the 1 & 2 LSBs mode, with the QP

value of 18, the increase in the bitrate is 3.34 and 5.68%, the payload equals to 312.10 and 396.80 kbps, and the decrease in the PSNR is 1.95 and 2.03 dB for the *foreman* and the *football*, respectively.

Table 1 Results for the *intra* for the *foreman* and the *football* sequences

QP	Hiding mode (LSBs)	<i>Foreman</i>			<i>Football</i>		
		Payload (kbps)	Bitrate (kbps)	PSNR (dB)	Payload (kbps)	Bitrate (kbps)	PSNR (dB)
18	0	0	4504	44.88	0	4730	45.32
	1	234.38	4622	43.80	293.88	4928	44.13
	2	140.13	4600	43.61	188.23	4920	43.99
	1&2	312.10	4654	42.93	396.80	4999	43.29
36	0	0.00	603.2	32.63	0.00	750.1	32.52
	1	5.15	609.6	32.54	8.26	759.9	32.40
	2	0.30	603.2	32.61	0.84	752.5	32.50
	1&2	5.35	609.6	32.53	8.73	760.4	32.39

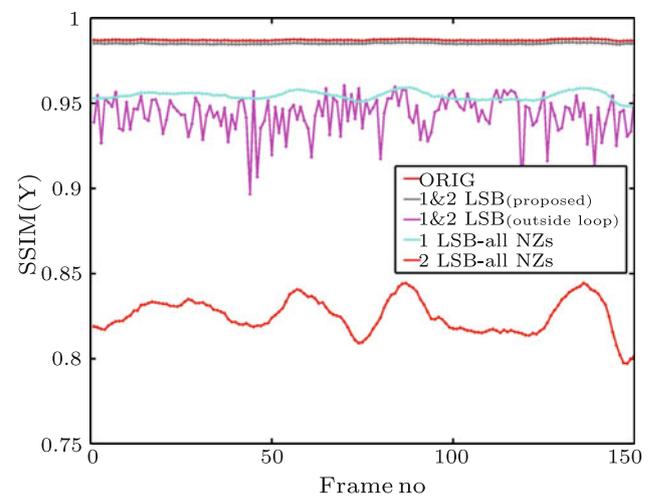
Table 2 Comparison of the bitrate, the payload and the PSNR at the QP 18 for the data hiding embedding inside and outside the reconstruction loop for the *intra* sequence with the 1&2 LSBs mode

Seq.	Orig		LSB-inside Loop (Proposed method)			LSB-outside Loop		
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)
Bus	44.52 (0.9932)	6.90	40.43 (0.9897)	7.32	856.00	24.35 (0.9232)	6.93	891.93
City	44.52 (0.9921)	6.07	41.66 (0.9881)	6.33	611.36	24.91 (0.9306)	6.10	640.82
Crew	45.22 (0.9883)	4.34	43.80 (0.9864)	4.48	290.64	27.72 (0.9412)	4.37	305.74
Football	45.32 (0.9901)	4.62	43.29 (0.9877)	4.88	396.80	27.54 (0.9464)	4.64	435.15
Foreman	44.88 (0.9872)	4.40	42.93 (0.9851)	4.55	312.10	24.54 (0.9432)	4.43	315.49
Harbour	44.30 (0.9967)	7.20	40.38 (0.9935)	7.70	895.74	22.22 (0.9386)	7.22	937.46
Ice	47.29 (0.9898)	2.25	45.08 (0.9886)	2.34	197.28	29.34 (0.9588)	2.26	194.80
Mobile	44.59 (0.9958)	10.38	41.07 (0.9930)	10.63	895.74	22.85 (0.9369)	10.38	1640.64
Soccer	45.19 (0.9889)	4.42	43.29 (0.9860)	4.68	373.39	26.81 (0.9434)	4.45	410.98
Avg.	45.09 (0.9913)	5.62	42.44 (0.9887)	5.88	536.56	25.59 (0.9403)	5.64	641.45

Table 2 shows the SSIM/PSNR, bitrate, and payload trade-off of our scheme for the *intra*-sequences of all the benchmark video sequences at the QP value of 18. It also contains a comparison with the embedding after the reconstruction loop. For our algorithm, decrease in the SSIM/PSNR is 0.0026/2.65 dB, whereas it is 0.0510/19.5 dB for embedding after the encoding loop, on the average (Fig 10).

For a subjective quality comparison, Fig. 11 contains frame # 0 of the *foreman* and the *football* sequences. Artifacts can be observed in the *intra*-frame because of the message embedding after the encoding loop. Ghost artifacts are encircled red in these video frames. For flat regions, a change in luminance can also be observed in the video frames. These artifacts are because of the spatial prediction from the top and the left blocks. If the encoding loop is not taken into account, during the embedding process, a drift or ‘increasing error’ will be created between the encoder and the decoder and different values will be used for the prediction on the encoder and the decoder side. Hence, distortion will increase gradually from the top-left corner to the bottom-right corner of the image.

To show the efficiency of the QTC selection criteria of our scheme, we have compared it with a naive data hiding in LSBs of all the QTCs, while taking into account the recon-

**Fig. 10** SSIM of proposed scheme with: 1 the embedding outside the reconstruction loop, 2 the naive embedding in all the QTCs for the *intra*-frames for the *foreman* sequence at the QP value 18

struction loop. Figure 12 shows the frame # 0 of the *foreman* and the *football*. In contrast to our algorithm, one can note the noise artifacts in the frames in which message embedding is performed in all the QTCs using the naive 1 LSB and 2 LSBs

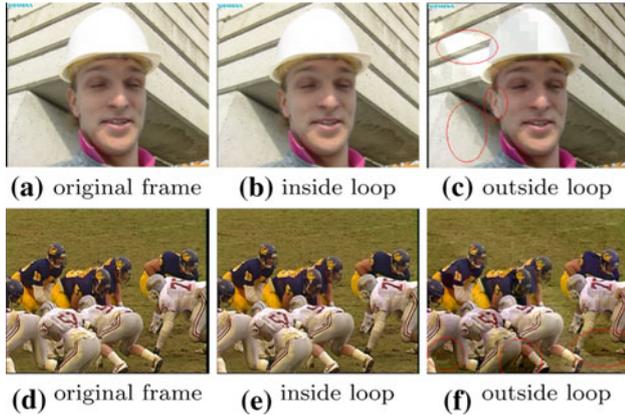


Fig. 11 Artifacts created in *intra* due to outside loop data hiding embedding with the 1&2 LSBs mode with the QP 18 for the frame # 0 of *foreman* and *football*

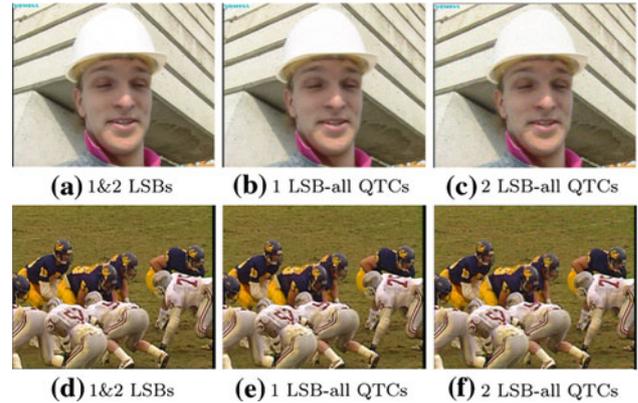


Fig. 12 Visual comparison of the 1&2 LSBs mode with the 1 LSB and the 2 LSBs embedding in all the QTCs for the *intra*-frame # 0 for the QP value of 18

embeddings. This noise is due to the introduction of new frequencies as a result of the conversion of the zero QTCs to non-zeros. Table 3 shows the SSIM/PSNR, bitrate, and payload analysis for the naive 1 LSB and 2 LSBs embeddings. We have compared it with the 1&2 LSBs mode of our algorithm,

which has the highest trade-offs in terms of the SSIM/PSNR and the bitrate. When the *skip-mode* is off, the payload for the CIF resolution at 25 fps will be 3712.5 and 7425 kbps for naive 1 LSB and 2 LSBs embeddings, respectively. One can note that the decrease in the SSIM/PSNR for the 1&2

Table 3 Comparison of the bitrate and the PSNR of our scheme with the data hiding embedding in all the QTCs for the *intra* sequence at the QP 18

Seq.	Orig		1&2 LSBs mode (Our method)			1 LSB-all NZs		2 LSBs-all NZs	
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB)	Bitrate (mbps)
Bus	44.52 (0.9932)	6.90	40.43 (0.9897)	7.32	856.00	38.87 (0.9760)	9.55	32.71 (0.9087)	12.67
City	44.52 (0.9921)	6.07	41.66 (0.9881)	6.33	611.36	38.99 (0.9723)	8.74	32.74 (0.8935)	11.92
Crew	45.22 (0.9883)	4.34	43.8 (0.9864)	4.48	290.64	39.53 (0.9564)	7.61	32.73 (0.8257)	11.34
Football	45.32 (0.9901)	4.62	43.29 (0.9877)	4.88	396.80	39.44 (0.9615)	8.01	32.69 (0.8432)	11.61
Foreman	44.88 (0.9872)	4.40	42.93 (0.9852)	4.55	312.10	39.39 (0.9545)	7.59	32.76 (0.8253)	11.41
Harbour	44.3 (0.9967)	7.20	40.38 (0.9935)	7.70	895.74	38.68 (0.9880)	9.81	32.72 (0.9531)	12.94
Ice	47.29 (0.9898)	2.25	45.08 (0.9886)	2.34	197.28	39.78 (0.9312)	6.71	32.53 (0.7258)	11.00
Mobile	44.59 (0.9958)	10.38	41.07 (0.9930)	10.63	895.74	38.63 (0.9820)	12.20	32.47 (0.9302)	14.31
Soccer	45.19 (0.9890)	4.42	43.29 (0.9860)	4.68	373.39	39.3 (0.9564)	7.85	32.70 (0.8305)	11.41
Avg.	45.09 (0.9913)	5.62	42.44 (0.9887)	5.88	536.56	39.18 (0.9642)	8.68	32.67 (0.8596)	12.07

The reconstruction loop has been taken into account for the data hiding embedding in all the QTCs

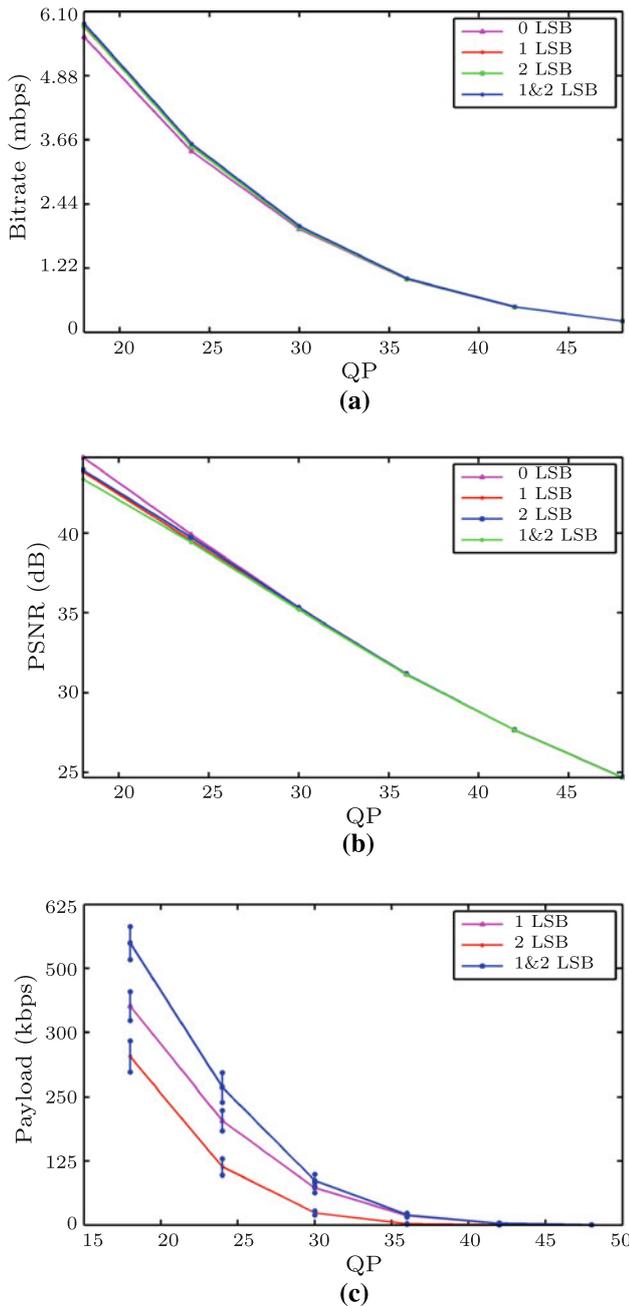


Fig. 13 Analysis of the data hiding for the *intra*-frames for all the nine benchmark video sequences over the whole range of the QP values for: **a** the bitrate, **b** the PSNR, **c** the payload. The standard deviation of the payload for all the QP values is also shown

LSBs mode of our algorithm is 0.0026/2.65 dB, in contrast to the naive versions where it is 0.0271/5.91 dB for the naive 1 LSB embedding and 0.1307/12.42 dB for the naive 2 LSBs embedding for all the benchmark sequences. Increase in the bitrate is 4.6% for our algorithm, in comparison with 54.44 and 114.76% for the naive 1 LSB and the 2 LSBs modifications, respectively. Hence, the trade-offs for the SSIM/PSNR and the bitrate are so high that the naive LSB embeddings

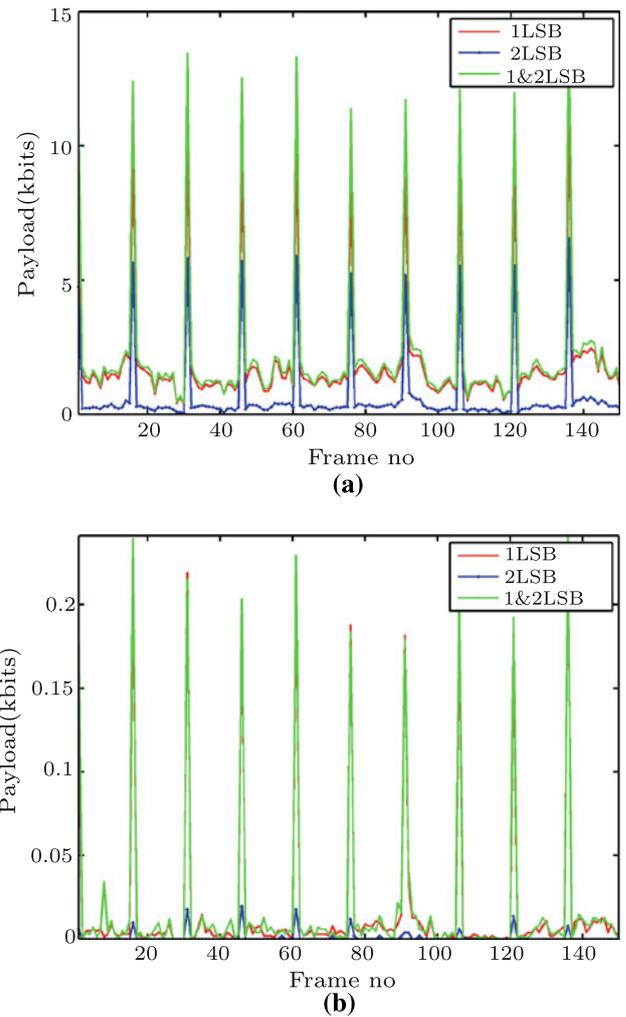


Fig. 14 Analysis of the payload capability for the message embedding of the *intra*- & *inter*-frames for *foreman* for the QP values: **a** 18 **b** 36

cannot be used in practical applications. A framewise SSIM comparison of our scheme has been presented in Fig. 10 with the outside loop embedding and embedding in all the QTCs. One can note that visual quality of the proposed scheme is consistently preserved for the *intra*-sequence.

4.1.1 Analysis over a whole range of QP values for the *intra*-frames

The results of our data hiding algorithm over a whole range of QP values—18, 24, 30, 36, 42, and 48—have been demonstrated in Fig. 13 for all the nine video sequences. Figure 13a illustrates the change in the bitrate over the whole range for 1 LSB, 2 LSBs, and 1 & 2 LSBs. Figure 13b shows the PSNR while Fig. 13c illustrates the payload capacity at various QP values. The PSNR graph is linear, and from the QP value of 36 onward, there is no considerable degradation in the quality. It is important to note that unlike the PSNR, bitrate and

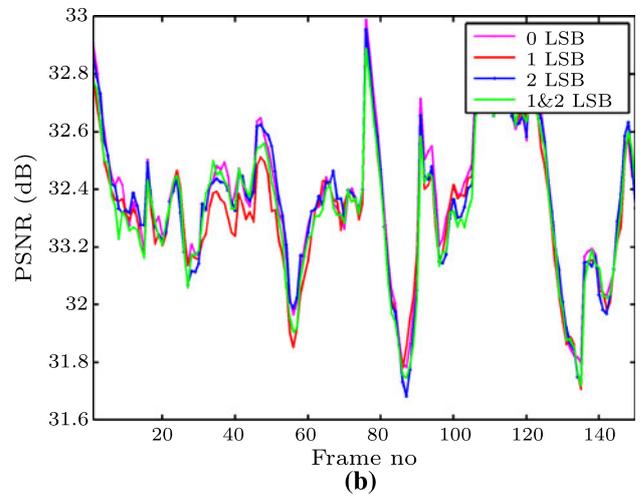
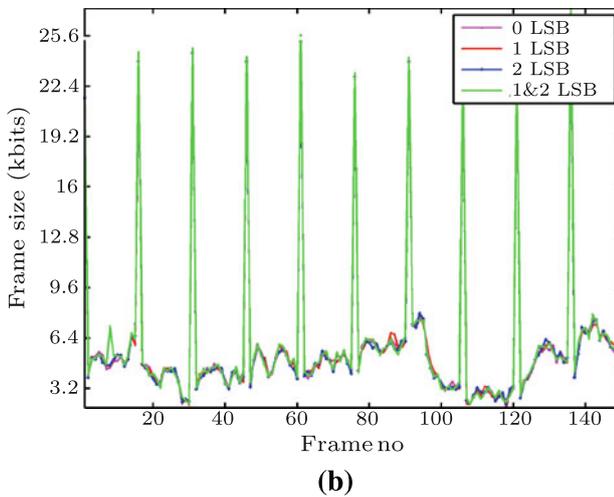
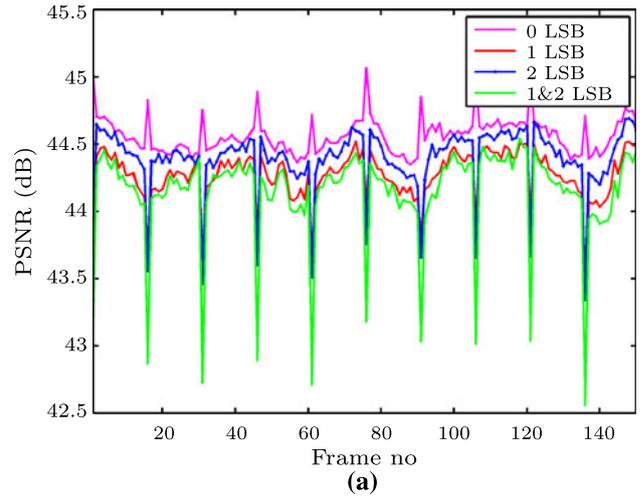
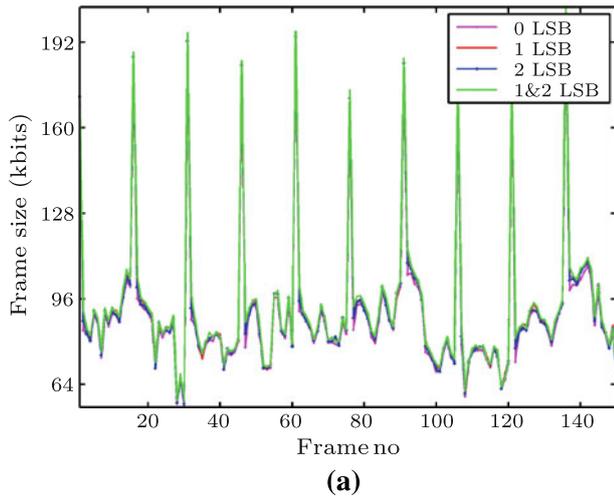


Fig. 15 Analysis of the change in bitrate for the message embedding of the *intra*- & *inter*-frames for *foreman* for the QP values: **a** 18 **b** 36

Fig. 16 Analysis of the change in PSNR for the message embedding of the *intra*- & *inter*-frames for *foreman* for the QP values: **a** 18, **b** 36

payload graphs are non-linear. The reason is that the quantization value is an exponential function of the QP value and PSNR is a logarithmic measure.

4.2 Analysis of the *inter*-frames

The *inter*-frames contain both *intra*- and *inter*-MBs. The prediction is performed from the preceding video frames in the case of *inter*-MBs, and from the top and the left blocks in the case of *intra*-MBs. The non-zero QTCs are found in the parts of frames containing motion and texture. The message is naturally embedded in these temporal masking areas of the *inter*-frames. In a video sequence, after every *intra*-frame, first few *inter*-frames are better predicted and contain lesser residual errors. Hence, any message embedding affects more the quality and the compression ratio. But as we go away from *intra*-frames, accumulated errors appear and message embedding does not affect much the quality of the *inter*-frames. On the

average, after 5 *inter*-frames followed by *intra*-frame, the ratio of the payload to the size of the *inter*-frames is comparable with that of the *intra*-frames, especially at the lower QP values.

Figures 14, 15, and 16 have been used for the payload, the bitrate, and the PSNR analysis, respectively at the QP values of 18 and 36 for the *foreman* sequence. One can note that the payload is adequate at the QP value of 18, and it decreases sharply. In fact, at the QP value of 36, we have very few QTCs with magnitudes above the threshold for the 1 LSB mode, let alone the 2 LSBs mode. Tables 4, 5 show the payload, the bitrate, and the PSNR analysis for the *foreman* and the *football* sequences at the QP value of 18 and 36 for the *intra*- & *inter*-frames. For the *foreman* sequence with the 1 & 2 LSBs mode at the QP value of 18, the increase in the bitrate is 2.81 and 2.89%, the payload is 38.25 and 56.45 kbps, and the PSNR decrease is 0.22 and 0.20 dB for the *inter* and the *intra* & *inter*, respectively. In contrast, for the *football*

Table 4 Data hiding results with the *intra* and the *inter* frames of the *foreman* and the *football* video sequences for the QP value of 18

QP	Hiding mode (LSBs)	<i>Foreman</i>			<i>Football</i>		
		Payload (kbps)	Bitrate (kbps)	PSNR (dB)	Payload (kbps)	Bitrate (kbps)	PSNR (dB)
I	0	0	4.40	44.88	0	4.67	45.29
	1	233.80	4.51	43.80	295.62	4.86	44.10
	2	139.65	4.49	43.61	196.25	4.85	43.90
	1&2	311.30	4.55	42.92	402.42	4.93	43.20
P	0	0.00	2.05	44.54	0	3.47	44.76
	1	34.45	2.09	44.30	172.91	3.58	43.65
	2	7.00	2.08	44.45	82.01	3.64	43.68
	1&2	38.25	2.11	44.23	207.43	3.70	43.05
I+	0	0	2.21	44.56	0	3.55	44.79
	1	47.73	2.25	44.27	181.09	3.67	43.69
	2	15.83	2.24	44.39	89.62	3.72	43.70
P+	1&2	56.45	2.27	44.14	220.43	3.70	43.06

Table 5 Data hiding results with the *intra* and the *inter* frames of the *foreman* and the *football* video sequences for the QP value of 36

QP	Hiding mode (LSBs)	<i>foreman</i>			<i>football</i>		
		Payload (kbps)	Bitrate (kbps)	PSNR (dB)	Payload (kbps)	Bitrate (kbps)	PSNR (dB)
I	0	0	601.6	32.61	0	781.3	32.42
	1	4.96	608.5	32.52	9.08	791.2	32.30
	2	0.28	601.4	32.59	0.97	782.4	32.38
	1&2	5.19	609.4	32.52	9.70	792.6	32.28
P	0	0	117.8	32.35	0	441.9	31.62
	1	0.11	117.9	32.31	2.91	448.0	31.55
	2	0.003	117.1	32.34	0.12	443.0	31.62
	1&2	0.13	118.7	32.32	2.93	448.2	31.55
I+	0	0	149.9	32.37	0	464.5	31.67
	1	0.44	150.6	32.33	3.32	470.9	31.60
	2	0.02	149.4	32.35	0.18	465.6	31.67
P+	1&2	0.47	151.3	32.33	3.38	471.0	31.60

sequence, the increase in the bitrate is 6.73 and 6.62%, the payload is 207.43 and 220.43 kbps, and the PSNR decrease is 1.71 and 1.73 dB for the *inter* and the *intra& inter*, respectively. The *football* sequence has greater payload capacity than the *foreman*, especially in the P frames. It is because of the texture and the high amount of motion in the *football* sequence.

The overall analysis of all the benchmark video sequences is given in Table 6. It also contains a comparison with the message embedding after the encoding loop. Decrease in the SSIM/PSNR for our scheme is 0.0106/1.38 dB, while it is 0.0687/19.2 dB for the embedding after the encoding loop. For a subjective quality comparison with the outside loop watermark embedding, Fig. 18 contains frame # 89 of the

foreman and the *football* sequences and shows the artifacts in the *inter*-frame because of the watermark embedding after the encoding loop. One can note that the *inter*-frame is heavily distorted in this case. The results show that any data hiding after the encoding loop distorts the video frame in the case of *inter*. Hence, message embedding after the encoding loop is not a workable solution for state of the art video codecs because of the spatial and the temporal prediction.

For comparison, with the naive message embedding in the LSBs of all the QTCs, Fig. 19 shows frame # 28 of the *foreman* and the *football*. Just like the *intra*-frames, one can only note the noise artifacts in those frames in which message embedding is performed in all the QTCs using the 1 LSB and the 2 LSBs embeddings, owing to the introduc-

Table 6 Comparison of the bitrate, the payload and the PSNR for the message embedding inside and outside the reconstruction loop for the *intra* & *inter* sequence with the 1&2 LSBs mode

Seq.	Orig		LSB-inside Loop (Proposed method)			LSB-outside Loop		
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)
Bus	44.27 (0.9923)	4.00	42.21 (0.9916)	4.22	278.38	24.12 (0.9101)	4.34	356.86
City	44.36 (0.9916)	2.57	43.80 (0.9912)	2.66	87.14	25.16 (0.9304)	2.76	115.12
Crew	44.80 (0.9859)	3.12	44.00 (0.9854)	3.24	111.90	28.48 (0.9020)	3.50	164.94
Football	44.79 (0.9876)	3.56	43.06 (0.9867)	3.79	220.43	26.94 (0.8898)	3.99	320.76
Foreman	44.56 (0.9856)	2.22	44.14 (0.9856)	2.28	56.46	25.04 (0.9323)	2.53	92.18
Harbour	44.18 (0.9963)	4.45	42.05 (0.9950)	4.72	330.15	22.46 (0.9331)	4.70	380.64
Ice	46.93 (0.9884)	1.06	46.27 (0.9886)	1.11	46.03	29.14 (0.9475)	1.29	89.51
Mobile	44.18 (0.9951)	5.70	41.00 (0.9937)	5.96	526.03	22.97 (0.9285)	5.98	581.07
Soccer	44.82 (0.9873)	2.35	43.99 (0.9870)	2.46	100.12	25.86 (0.9178)	2.64	153.12
Avg.	44.77 (0.9900)	3.23	43.39 (0.9894)	3.38	195.18	25.57 (0.9213)	3.52	250.47

The QP value is 18 and the *Intra period* is 15.

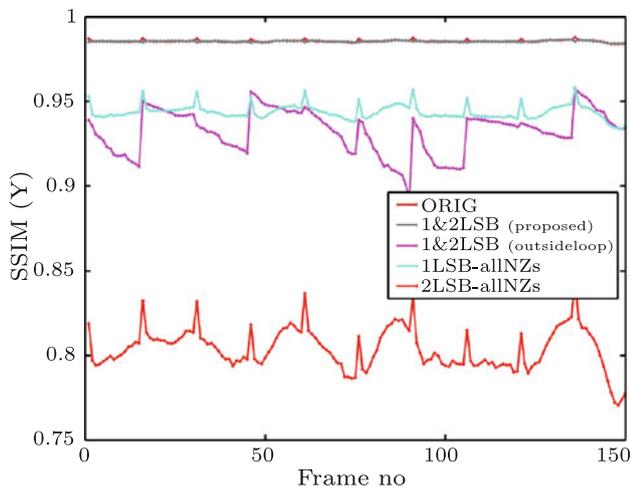


Fig. 17 SSIM of the proposed scheme with: 1 The embedding outside reconstruction loop, 2 The naive embedding in all the QTCs for the *intra*- & *inter*-frames for the *foreman* sequence at the QP value 18

reduction of new frequencies. Table 7 shows the SSIM/PSNR, the bitrate, and the payload analysis at the QP value of 18. The payload, for the naive 1 LSB and 2 LSBs embedding, will be

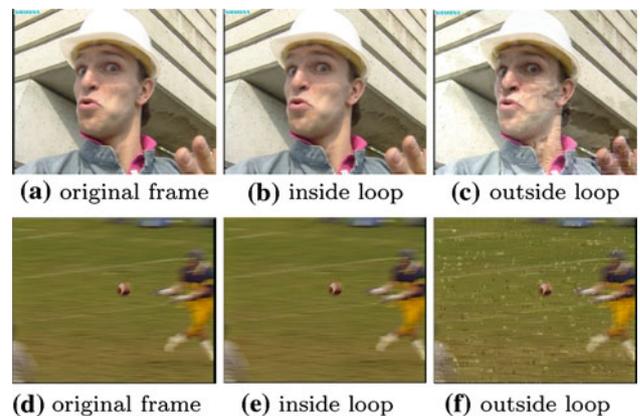


Fig. 18 Artifacts created in the *inter* due to the outside loop message embedding with the 1&2 LSBs mode with the QP 18 for the frame # 89 of *foreman* and *football*

similar to the *intra*-frames, *i.e.* 3712.5 and 7425 kbps, respectively. Average decrease in the SSIM/PSNR for our algorithm is 0.0106/1.38 dB, while it is 0.0345/6.62 dB for the naive 1 LSB embedding and 0.1498/12.93 dB for the naive 2 LSBs

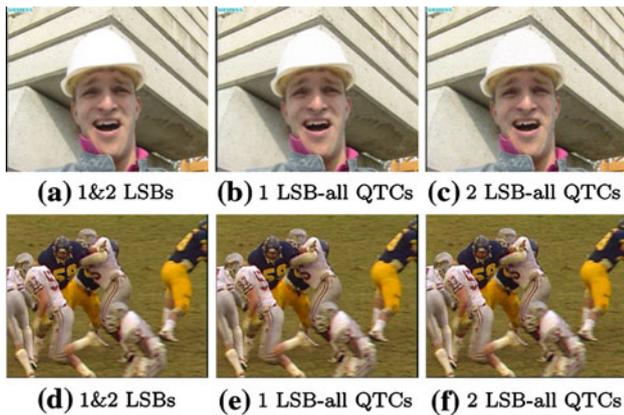


Fig. 19 Visual comparison of the 1&2 LSBs mode with the naive 1 LSB and 2 LSBs embeddings in all the QTCs. for the *inter*-frame # 28 for the QP value of 18

embedding. Increase in the bitrate is 4.6% for our algorithm, while it is 101.23 and 216.09% for the naive 1 LSB and 2 LSB modifications. Such high trade-offs for SSIM/PSNR and bitrate make it inappropriate for practical applications. Framewise SSIM comparison of our scheme in Fig. 17 for the *foreman* sequence at the QP value 18 verifies that the visual quality of the proposed scheme is consistently preserved for the *intra* & *intra*-sequence.

4.2.1 Analysis over whole range of QP values for *inter*-frames

The overall performance analysis of all the nine video sequences with an *intra*-period of ‘15’ is shown in the form of graphs in Fig. 20. Figure 20a illustrates the effect of data hiding on the bitrate, while Fig. 20b shows the change in the PSNR for different message embedding modes. Figure 20c illustrates the payload capability of our algorithm along with the standard deviation of the payload at different QP values. Among the 1 LSB and 2 LSBs embedding modes, 1 LSB performs better, having higher payloads and minimum increase in the bitrate. In the 2 LSBs mode, 2 bits are embedded in the same coefficient and thus the magnitude of compromise is higher. The 2 LSBs should be used in combination with the 1 LSB mode (1 & 2 LSBs embedding mode) when higher payload is required. One can note that just like the bitrate, the payload varies with the QP both in the case of *intra* and *inter*.

4.3 Comparative evaluation

For the sake of comparative evaluation of our scheme, we have compared it with seven other recent techniques

Table 7 Comparison of the bitrate and the PSNR of our scheme with the message embedding in all the QTCs for the *intra* & *inter* sequence with the QP 18

Seq.	Orig		1&2 LSBs mode (Proposed method)			1 LSB-all NZs		2 LSBs-all NZs	
	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	Payload (kbps)	PSNR (dB) (SSIM)	Bitrate (mbps)	PSNR (dB) (SSIM)	Bitrate (mbps)
Bus	44.27 (0.9923)	4.00	42.21 (0.9916)	4.22	278.38	37.86 (0.9703)	6.87	31.79 (0.8948)	10.36
City	44.36 (0.9916)	2.57	43.80 (0.9912)	2.66	87.14	38.28 (0.9677)	5.82	32.07 (0.8807)	9.82
Crew	44.80 (0.9859)	3.12	44.00 (0.9854)	3.24	111.90	38.33 (0.9431)	6.36	31.97 (0.8009)	10.16
Football	44.79 (0.9876)	3.56	43.06 (0.9867)	3.79	220.43	38.16 (0.9499)	6.87	31.77 (0.8187)	10.54
Foreman	44.56 (0.9856)	2.22	44.14 (0.9856)	2.28	56.46	38.45 (0.9436)	5.76	32.11 (0.8040)	9.85
Harbour	44.18 (0.9963)	4.45	42.05 (0.9951)	4.72	330.15	37.65 (0.9845)	7.02	31.79 (0.9442)	10.26
Ice	46.93 (0.9884)	1.06	46.27 (0.9886)	1.11	46.03	38.70 (0.9145)	6.08	31.51 (0.6863)	10.31
Mobile	44.18 (0.9951)	5.70	41.00 (0.9937)	5.96	526.03	37.50 (0.9777)	7.74	31.60 (0.9218)	10.57
Soccer	44.82 (0.9873)	2.35	43.99 (0.9870)	2.46	100.12	38.41 (0.9481)	5.98	31.95 (0.8100)	10.04
Avg.	44.77 (0.9900)	3.23	43.39 (0.9894)	3.38	195.18	38.15 (0.9555)	6.50	31.84 (0.8402)	10.21

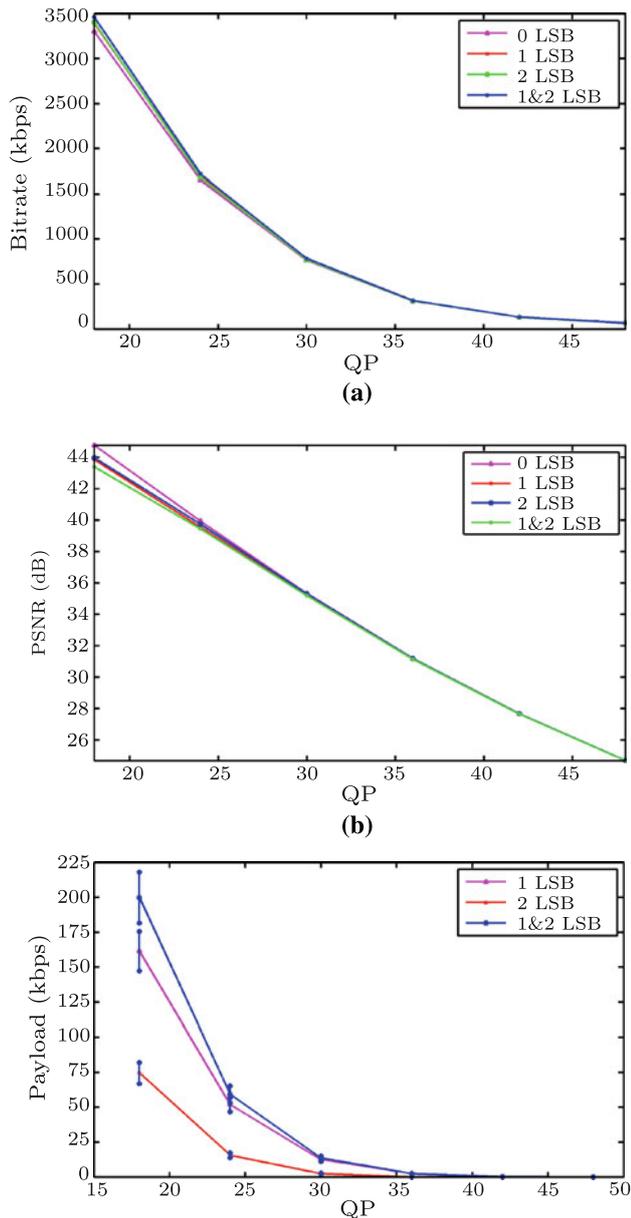


Fig. 20 Analysis of the data hiding for the *intra*- & *inter*-frames for all the nine benchmark video sequences over the whole range of the QP values for: **a** the bitrate, **b** the PSNR, **c** the payload. The standard deviation of the payload for all the the QP values is also shown

which are as follows: the particle swarm optimization-based dither modulation scheme [37], the bitstream replacement of CAVLC bitstream [40], the hybrid watermarking scheme [32], the data hiding in the H.264/AVC compressed video [18], the reference index-based watermarking of the H.264/AVC [21], the spread transform scalar Costa scheme [10], and the robust watermarking for P frames [29]. These techniques are different from each other in several aspects e.g., the working domain (pixel, transform, or bitstream) and the embedding algorithm (dither modulation, spread transform, or quantiza-

tion based). The comparison has been made based on several important characteristics of the data hiding systems and summarized in Table 8.

The embedding domain is of vital importance for a watermarking system. The watermarking algorithms of Zou and Bloom [40], and Kim et al. work in the bitstream domain (outside the reconstruction loop). These algorithms can work on compressed video, without decoding it. These schemes, however, have lower payload and can lead to a considerable decrease in the PSNR because of the drift error. For [18], the PSNR drops to 43 dB even for the I frame, let alone the P frames. Moreover, a flickering effect is visible along the temporal dimension of the video. The video quality decreases with the increase in the encoded spatial resolution because of this drift error. The bitrate change is also another important aspect of these data hiding algorithms. For the algorithms that work in bitstream domain [18], there is no change in bitrate. For our algorithm, the increase in the bitrate is 4.6% with a decrease in the PSNR of 1.38 dB for the *intra*- & *inter*-sequence.

Different watermarking algorithms have different embedding space. Some of them work on the video data (pixels or coefficients), while other embed hidden message in the video header. Our scheme works inside the reconstruction loop, while utilizing the largest possible embedding space with minimal trade-off in terms of the bitrate and the SSIM/PSNR. It is in contrast to other schemes which either use the video header fields [21, 40] or some part of the video data e.g., the luma [10, 37], or the I frames only [18, 32, 37]; hence, not fully utilizing the embedding capacity. In [29], the watermark is embedded in all the non-zero QTCs, (even having the magnitude of 1). Let the *watermark cost* be the increase in the bitrate (in bits) per watermark bit. In the case of the *intra*, the *watermark cost* is 0.06 and 0.15 at the QP values of 18 and 36, respectively. These results are far better than 1.54, the result presented in [29]. In the case of *intra* and *inter*, we get the *watermark costs* of 0.15 and 0.42 at the QP values of 18 and 36, respectively, which is far better than 1.50, the result of the work presented in [29]. At the higher QP values, we do not have lot of non-zero QTCs which can be watermarked but the ratio between the payload and the bitrate is still conserved.

To summarize, our proposed scheme presents a good payload capability with a minimal trade-off for the PSNR and the bitrate. This trade-off is possible because of the selection of the AC QTCs with magnitudes above a certain threshold and embedding inside the reconstruction loop.

5 Conclusion

In this paper, we have designed and analyzed a new video data hiding scheme with a high payload for the H.264/AVC.

Table 8 Comparison of the proposed scheme with other recent watermarking techniques for the H.264/AVC video codec.

Wm Scheme	Working domain	Insertion technique	Payload	Bitrate increase	PSNR (dB)	Embedding space
Wu et al. [37]	Transform	Dither modulation	1 bit/4x4 block	Yes	Yes	DCT coeff. of <i>luma</i> (I frames only)
Zou and Bloom [40]	Bitstream	Replacement		Yes	Yes	Intra prediction (mode change)
Qiu et al. [32]	Transform/MB header	Replacement	1 bit/MB	Yes	Yes	AC coeffs (I frames) MVs (P frames)
Kim et al. [18]	Bitstream	Replacement	1 bit/MB	No	(up to 43 dB)	Signs-TI's (only I frames)
Li et al. [21]	Transform	Spread spectrum		3.22%	-0.75 dB	reference index (P frames) except 1st P frame)
Golikeri et al. [10]	Transform (before quantization)	Spread transform scalar Costa scheme	1 bit/MB	Yes	Yes	DC coeff. of <i>luma</i> (I and P frames, HVS based)
Noorkami et al. [29]	Transform	Spread spectrum	1 bit/MB	Yes	Yes	All non-zero QTCs (I and P frames)
Our scheme	Transform	Quantization based	195.18 kbps (20.19 bits/MB)	4.6%	-1.38 dB	AC coefficients (above threshold)

Our scheme embeds the RD optimized hidden message in the QTCs for both the *intra*- and *inter*-frames. Having a regard for the reconstruction loop, the proposed scheme offers consistent payload capability to the H.264/AVC standard at different bitrates without adversely affecting the overall bitrate and the SSIM/PSNR of the video bitstream. The watermark is embedded in those regions of the *intra*-frames, which contain edges and texture, and for the *inter*-frames, message embedding is naturally done in the temporal masking regions, which contains motion and texture. The experimental results have demonstrated that the *inter*-frames can be equally good for the message embedding owing to its motion and texture masking.

Acknowledgments This work is in part supported by the project VOODOO (2008–2011) of the french ANR and the region of Languedoc Roussillon, France.

References

- Alattar, A.M., Lin, E.T., Celik, M.U.: Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video. *IEEE Trans. Circuits Syst. Video Technol.* **13**, 787–800 (2003)
- Chandramouli, R., Memon, N.: Analysis of LSB based image steganography Techniques. In: *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 1019–1022. (2001)
- Chen, C., Ni, J., Huang, J.: Temporal statistic based video watermarking scheme robust against geometric attacks and frame dropping. In: *IWDW '09: Proceedings of the 8th International Workshop on Digital Watermarking*, pp. 81–95. Springer, Heidelberg (2009)
- Chung, Y., Wang, P., Chen, X., Bae, C., Otoom, A., Tran, T.: A performance comparison of high capacity digital watermarking systems. In: *KES*, vol. 1, pp. 1193–1198. (2005)
- Cox, I., Kilian, J., Leighton, F., Shamoon, T.: Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* **6**, 1673–1687 (1997)
- Cox, R.D., Henri, C.J., Bret, P.M.: CD-based image archival and management on a hybrid radiology intranet. *Can. J. Med. Radiat. Technol.* **28**(3), (1997)
- Dai, Y., Zhang, L., Yang, Y.: A new method of MPEG video watermarking technology. In: *Proceedings of International Conference on Communication Technology*, vol. 2, pp. 1845–1847. (2003)
- Deguillaume, F., Csurka, G., O'Ruanaidh, J., Pun, T.: Robust 3D DFT video watermarking. *SPIE* **3657**, 113–124 (1999)
- Ejima, M., Miyazaki, A.: A wavelet-based watermarking for digital images and video. In: *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 678–681. (2000)
- Golikeri, A., Nasiopoulos, P., Wang, Z.: Robust digital video watermarking scheme for H.264 advanced video coding standard. *J. Electron. Imaging* **16**(4), 043008 (2007)
- Gong, X., Lu, H.: Towards fast and robust watermarking scheme for H.264 video. In: *Proceedings of IEEE International Symposium on Multimedia*, pp. 649–653. (2008)
- H.264. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC). Technical report, Joint Video Team (JVT), Doc. JVT-G050, March (2003)
- Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. *Signal Process* **66**, 283–333 (1998)
- He, D., Sun, Q., Tian, Q.: A semi-fragile object based video authentication system. In: *Proceedings of International Symposium on Circuits and Systems*, **3**, pp. 814–817 (2003)
- Kang, X., Huang, J., Shi, Y., Lin, Y.: A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. *IEEE Trans. Circuits Syst. Video Technol.* **13**(8), 776–786 (2003)
- Kapotas, S., Varsaki, E., Skodras, A.: Data hiding in H.264 encoded video sequences. In: *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, (2007)
- Katzenbeisser, S., Petitcolas, F. (eds.): *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House Inc., Norwood (2000)

18. Kim, S., Kim, S., Hong, Y., Won, C.: Data hiding on H.264/AVC compressed video. In: *Image Analysis and Recognition Lecture Notes in Computer Science*, chapter 62, pp. 698–707 (2007)
19. Langelaar, G., Lagendijk, R.: Differential energy watermarking of DCT encoded images and video. *IEEE Trans. Image Process.* **10**, 148–158 (2001)
20. Lee, P., Chen, M.: Robust error concealment algorithm for video decoder. *IEEE Trans. Consumer Electron.* **45**(3), 851–859 (1999)
21. Li, J., Liu, H., Huang, J., Zhang, Y.: A robust watermarking scheme for H.264. In: *Digital Watermarking volume 5450 of Lecture Notes in Computer Science*, pp. 1–15. Springer, Heidelberg (2009)
22. Li, K., Zhang X.: Reliable adaptive watermarking scheme integrated with JPEG2000. In: *Proceedings of 3rd International Symposium on Image and Signal Processing and Analysis (ISPA 2003)*. Rome, Italy (2003)
23. Li, Q., Cox, I.: Using perceptual models to improve fidelity and provide resistance to volumetric scaling for quantization index modulation watermarking. *IEEE Trans. Inf. Forensics Security* **2**(2), 127–139 (2007)
24. Lu, C., Chen, J., Fan, K.: Real-time frame-dependent video watermarking in VLC domain. *Signal Process. Image Commun.* **20**(7), 624–642 (2005)
25. Malvar, H., Hallapuro, A., Karczewicz, M., Kerofsky, L.: Low-complexity transform and quantization in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 598–603 (2003)
26. Miller, M., Doerr, G., Cox, I.: Applying informed coding and embedding to design a robust high capacity watermark. *IEEE Trans. Image Process.* **13**(2), 792–807 (2004)
27. Mobasser, B., Raikar, Y.: Authentication of H.264 streams by direct watermarking of CAVLC blocks. In: *Proceedings of Security, Steganography, and Watermarking of Multimedia Contents IX volume 6505 SPIE*, San Jose, CA (2007)
28. Noorkami, M., Mersereau, R.: Compressed-domain video watermarking for h.264. In: *Proceedings of IEEE International Conference on Image Processing*, pp. 890–893. Genoa, Italy (2005)
29. Noorkami, M., Mersereau, R.: Digital video watermarking in P-frames with controlled video bit-rate increase. *IEEE Trans. Inf. Forensics Security* **3**(3), 441–455 (2008)
30. Oostveen, J., Kalker, T., Haitsma J.: Visual hashing of digital video: applications and techniques. In: *Proceedings of SPIE, Applications of Digital Image Processing XXIV*. vol. 4472, pp. 121–131. San Diego CA, USA (2009)
31. Pröfrock, D., Schlauweg, M., Müller, E.: A new uncompressed-domain video watermarking approach robust to H.264/AVC compression. In: *Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pp. 99–104. Anaheim, CA, USA (2006)
32. Qiu, G., Marziliano, P., Ho, A.T.S., He, D., Sun, Q.: A hybrid watermarking scheme for h.264/avc video. *Pattern Recognition, International Conference on* **4**, pp. 865–869 (2004)
33. Ramkumar, M., Akansu, A.: Robust protocols for proving ownership of images. In: *Proceedings of International Conference on Information Technology: Coding and Computing*, pp. 22–27 (2000)
34. Wang, Q., Zhao, D., Ma, S., Lu, Y., Huang, Q., Ga, W.: Context-based 2D-VLC for video coding. In: *Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 89–92 (2004)
35. Watson, A.: DCT quantization matrices visually optimized for individual images. In: *Proceedings of SPIE Society of Photo-Optical Instrumentation Engineers*, vol. 1913, pp. 202–216 (1993)
36. Wiegand, T., Sullivan, G., Bjntegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **13**, 560–576 (2003)
37. Wu, C., Zheng, Y., Ip, W., Chan, C., Yung, K., Lu, Z.: A flexible H.264/AVC compressed video watermarking scheme using particle swarm optimization based Dither modulation. *AEU Int. J. Electron. Commun.* (2010)
38. Wu, G., Wang, Y., Hsu, W.: Robust watermark embedding/detection algorithm for H.264 video. *J. Electron. Imaging* **14**(1), (2005)
39. Xie, F., Furon, T., Fontaine, C.: On-Off keying modulation and Tardos fingerprinting. In: *Proceedings of ACM Workshop on Multimedia and security*, pp. 101–106. New York, NY, USA (2008)
40. Zou, D., Bloom, J.: H.264/AVC stream replacement technique for video watermarking. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1749–1752 (2008)

Joint trellis-coded quantization watermarking for JPEG2000 images

Dalila Goudia · Marc Chaumont · William Puech ·
Naima Hadj Said

Received: 7 March 2011 / Accepted: 22 November 2011
© Institut Télécom and Springer-Verlag 2011

Abstract Watermarking in the Joint Photographic Experts Group (JPEG)2000 coding pipeline is investigated in this paper. A joint quantization and watermarking method based on trellis-coded quantization (TCQ) is proposed to reliably embed data during the quantization stage of the JPEG2000 part 2 codec. The central contribution of this work is the use of a single quantization module to jointly perform quantization and watermark embedding at the same time. The TCQ-based watermarking technique allows embedding the watermark in the detail sub-bands of one or more resolution levels except the first one. Watermark recovery is performed after image decompression. The performance of this joint scheme in terms of image quality and robustness against common image attacks was estimated on real images.

Keywords Image compression · Digital watermarking · JPEG2000 · TCQ

1 Introduction

Digital watermarking is a relatively young research field. Watermarking techniques embed an invisible message within a multimedia content by modifying the media data. This process is done in such a way that the hidden data are not perceptible to an observer and should be resistant to a variety of manipulations. Modeling watermarking as communications with side information has led to the design of very efficient algorithms. This generation of watermarking schemes is called informed schemes [1–3]. One of the two main informed watermarking categories is the quantized-based watermarking schemes. In 1999, Chen and Wornell [1] have introduced the quantization index modulation (QIM), where the host signal is considered as the side information of Costa's scheme [6]. A practical and efficient implementation of the Costa's ideas is the scalar Costa scheme proposed by Eggers et al. [2] which is quiet similar to the DC-QIM watermarking [1].

In many applications, watermarked images are usually compressed in a specific image format before transmission or storage. The simplest way of watermarking is to embed data first with a known watermarking encoder and then compress the watermarked images via a standard compression encoder to get compressed watermarked images. However, the drawback of this approach employing separated watermarking and compression methods is that the compression stage could remove some embedded data from the watermarked images and thus degrade or damage the robustness of the watermark. The compression is considered as an attack. Therefore, instead of treating watermarking and compression separately, it is interesting and beneficial to look at the joint design of watermarking

D. Goudia (✉) · M. Chaumont · W. Puech
Laboratory LIRMM, UMR CNRS 5506,
University of Montpellier II, 161, rue Ada,
34095 Montpellier cedex 05, France
e-mail: dalila.goudia@lirimm.fr

D. Goudia · N. Hadj Said
University of Science and Technologies of Oran (USTO),
BP 1505 El Mnaouer, Oran, Algeria

M. Chaumont
University of Nîmes, rue du Docteur Georges Salan, 30021,
Nîmes cedex 1, France

and compression system. The joint approach consists to directly embed the binary message during the compression process. The watermarking process is adapted and integrated into the compression coding framework. The main constraints that must be considered are trade-offs between compression bitrate, computational complexity, watermarking payload, distortion induced by the insertion of the watermark, and robustness of watermarked images. As many images are usually compressed by Joint Photographic Experts Group (JPEG)2000 in several applications such as medical imaging, digital cinema, or 3-D imaging, it is worthwhile to investigate how to embed data in JPEG2000 compressed images efficiently. This is the main objective of our research.

We address the problem of combination of informed watermarking and source coding within a single system in order to design a joint JPEG2000 coding and watermarking scheme. We propose to use trellis-coded quantization (TCQ) properties to perform at the same time, quantization of the wavelet coefficients (source coding), and watermarking (channel coding). We chose the TCQ approach because it combines quantization and trellis coding. The other reason is that the TCQ quantization option is provided in part 2 [5] of the JPEG2000 standard.

This paper is organized as follows: We first present the JPEG2000 standard and watermarking methods in the JPEG2000 domain in Section 2. Then, we provide a detailed review of the TCQ quantization in Section 3. We present the proposed joint JPEG2000 coding and watermarking TCQ-based scheme in Section 4. Experimental results are shown in Section 5 to demonstrate the feasibility of the proposed method. Finally, concluding remarks are given in Section 6.

2 JPEG2000

2.1 The JPEG2000 coder

JPEG2000 [4] is a compression standard developed by the Joint Photographic Experts Group (JPEG). It is based on discrete wavelet transformation, and it provides several important features such as progressive transmission by resolution or quality, better resilience to bit errors, and region of interest (ROI) coding. The main encoding procedures of JPEG2000 (part 1) [4] are illustrated in Fig. 1a. Firstly, the original image undergoes some pre-processing operations (level shifting and color transformation). The image is partitioned into rectangular and non-overlapping tiles of equal size. Each tile is compressed independently using its own

set of specified compression parameters. The processed tile is decomposed by the discrete wavelet transform (DWT) into a collection of sub-bands (LL,¹ HL,² LH³, and HH)⁴ which may be organized into increasing resolution levels. The wavelet coefficients are afterward quantized by a dead-zone uniform scalar quantizer. For some applications, ROI coding may be applied. The quantized wavelet coefficients in each sub-band are partitioned into small rectangular blocks which are called *code-blocks*. Each *code-block* is encoded independently during the tier 1 encoding stage by using a bit-plane coder called embedded block coding with optimal truncation (EBCOT). So, each coding block has an independent bitstream. These bitstreams are combined into a single bitstream using tier 2 coding based on the result of rate control stage. An efficient rate-distortion algorithm provides possible truncation points of the bitstreams in an optimal way to minimize distortion according to any given target bitrate. The coded data are outputted to the code-stream in packets, and the JPEG2000 file stream is finally formed.

The following steps ensure the decompression of the compressed image as depicted in Fig. 1b: After the tier 2 decoding, the image bitstream is decoded by the EBCOT decoder. Then, the wavelet coefficients are reconstructed during the inverse scalar quantization stage. Afterward, the inverse DWT and the post-processing operations are performed to reconstruct the image.

2.2 Watermarking in the JPEG2000 domain

The JPEG2000 Secured (JPSEC) [7] or part 8 of the standard provides a framework for secure imaging. Watermarking is one of the security tools used in the data integrity service provided by JPSEC. Watermarking is considered here as a post-compression process and is used, along with other cryptographic methods, for image content integrity applications.

Another distinct approach is to insert the watermark during the JPEG2000 compression process. Several watermarking techniques integrated into the JPEG2000 coding scheme have been proposed [8–13]. Few among those methods are quantization-based watermarking schemes combined to JPEG2000 [11–13]. Meerwald [11] developed a watermarking process based on QIM

¹Horizontal and vertical low frequency sub-band.

²Horizontal high-frequency and vertical low-frequency sub-band.

³Horizontal low-frequency and vertical high-frequency sub-band

⁴Horizontal high-frequency and vertical high-frequency sub-band.

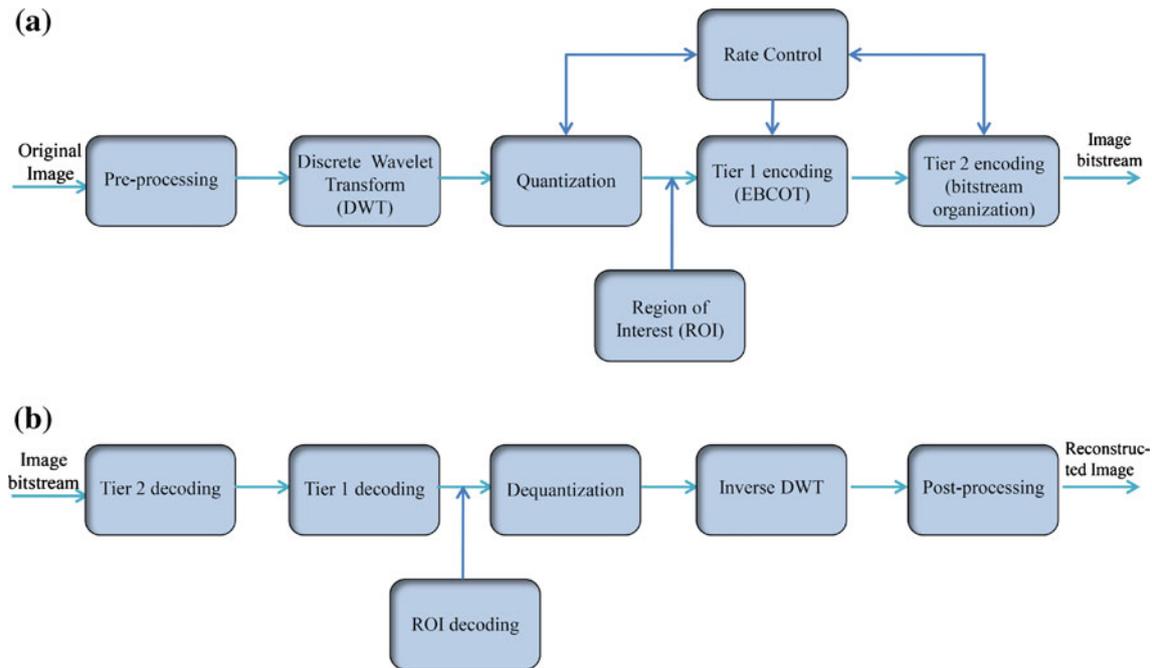


Fig. 1 **a** Block diagram of the JPEG2000 encoder algorithm. **b** Block diagram of the JPEG2000 decoder algorithm

integrated into the JPEG2000 coding chain. Despite its robustness, this method does not fulfill the visual quality requirement. Li and Zhang [10] proposed an adaptive watermarking method integrated in the JPEG2000 coding framework. Wavelet coefficients included in the watermarking process are modified depending on the target bitrate such that the embedded watermark can survive the rate allocation procedure of JPEG2000 without degrading the image quality. Fan and Tsao [8] proposed hiding two kinds of watermarks: a fragile one and a robust one by using a dual pyramid watermarking scheme. The robust pyramid watermark is designed to conceal secret information inside the image so as to attest to the origin of the host image. The fragile pyramid watermark is designed to detect any modification of the host image. Schlaueg et al. [12] have developed a semi-fragile authentication watermarking scheme by using an extended scalar quantization and hashing scheme in the JPEG2000 coding pipeline. This authentication scheme is secure, but the embedding of the watermark induces poor quality performances. Fan et al. [9] proposed ROI-based watermarking scheme. The embedded watermark can survive ROI processing, progressive transmission, and rate-distortion optimization. The only drawback of this method is that it works only when the ROI coding functionality of JPEG2000 is activated. Ouled-Zaid et al. [13] have proposed to integrate a modified QIM scheme in JPEG2000 part 2.

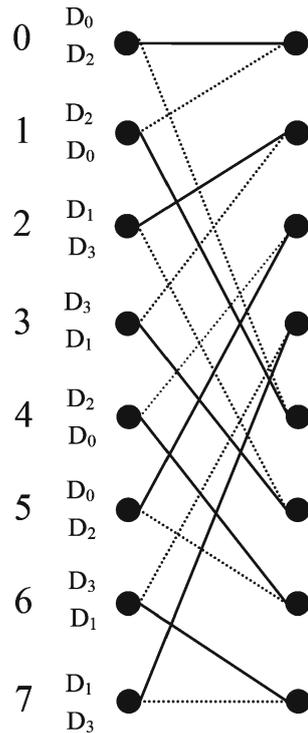
The QIM modification consists of reducing the distortion caused during quantization-based watermarking by using a non-linear scaling.

We can notice that two watermarking approaches have been commonly used in the state-of-the-art. One is to embed data into the wavelet coefficients before the quantization stage [13]. The other is to operate directly on the quantization indices after the quantization process [8–12]. In contrast of this conventional approaches, our proposed method embeds the watermark during the quantization process of JPEG2000.

3 Trellis-coded quantization in JPEG2000

Several quantization options are provided within JPEG2000. Part 2 [5] allows for the use of TCQ as a replacement for scalar quantization. TCQ is a particular kind of vector quantization proposed by Marcellin and Fischer [14]. The variant of TCQ used in the part 2 of the JPEG2000 standard is the entropy-coded TCQ [22]. A uniform scalar quantizer with step size Δ is partitioned into four subsets called D_0 , D_1 , D_2 , and D_3 . Subsets D_j , $j = 0, 1, 2, 3$ are used to label the branches of a trellis as shown in Fig. 2. The two scalar quantizers associated with each state in the trellis are combined into *union quantizers* A_0 and A_1 . The two union quantizers $A_0 = D_0 \cup D_2$, $A_1 = D_1 \cup D_3$ are illustrated in

Fig. 2 A single stage of an eight-state trellis with branch labeling used in JPEG2000



quantizer index $q(A_i)$ corresponds to the path since there are two possible codewords for each index (D_0 or D_2 and D_1 or D_3). The least significant bit determines the path through the trellis. Given the initial state and by construction of the trellis, the TCQ indices from union quantizers A_0 and A_1 provide all information necessary to reconstruct the wavelet coefficients.

4 The proposed joint JPEG2000 and watermarking scheme

TCQ have already been used in data hiding [18–21] and watermarking [16, 17]. These methods rely on the following principle: The paths in the trellis are forced by the values of the message, and the samples of the host signal are quantized with the subset corresponding to the trellis path. We propose another TCQ-based technique which is independent of the path. The proposed joint system allows both quantization of wavelet coefficients and watermark embedding without integrating an additional stage for watermarking in the JPEG2000 coding/decoding chain.

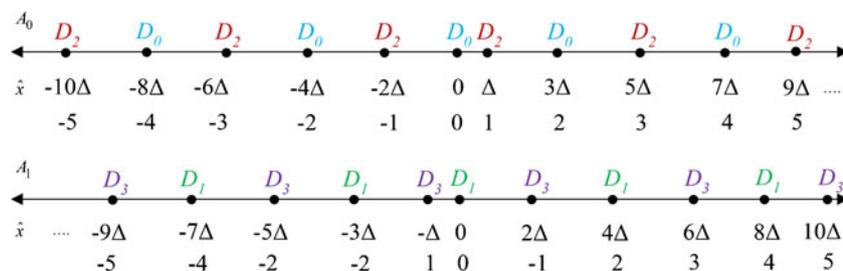
Fig. 3. This figure shows the reconstruction values \hat{x} corresponding to each union quantizer along with the corresponding union quantizer indices $q(A_i)$. Notable features of this figure are that the codebook is uniform with step size Δ and that the zero codeword appears in two subsets, D_0 and D_1 .

At each state in the trellis, we can choose between one of the two quantizers belonging to the union quantizer to quantize the input sequence \mathbf{x} . Quantization proceeds via the Viterbi algorithm [15] to determine the trellis path that minimizes the mean-squared error between the input sequence and output codewords. The Viterbi Algorithm produces two sequences: The first one is a binary sequence defining the minimum distortion path. The second one is the sequence of corresponding quantization indices. Note that by construction of the trellis, the least significant bit of each union

4.1 Overview of the proposed joint scheme

The block diagram of the joint JPEG2000 part 2 and watermark embedding scheme is illustrated in Fig. 4. The classical TCQ quantization component of the JPEG2000 encoder and decoder is replaced by a hybrid TCQ module which can perform at the same time quantization and watermark embedding. The quantization algorithm employs a modified version of the TCQ described in Section 3. One of the most important parameter to consider is the selection of the wavelet coefficients that must be included in the watermarking process. According to the wavelet decomposition, the LL sub-band carries out the low frequencies which represents the most significant data of the transformed image. In order to avoid considerable quality degradation in the reconstructed image, the wavelet coefficients

Fig. 3 Union quantizers A_0 and A_1 for TCQ in JPEG2000. \hat{x} is the reconstruction value and $q(A_i)$ is the union quantizer index corresponding to this reconstruction value



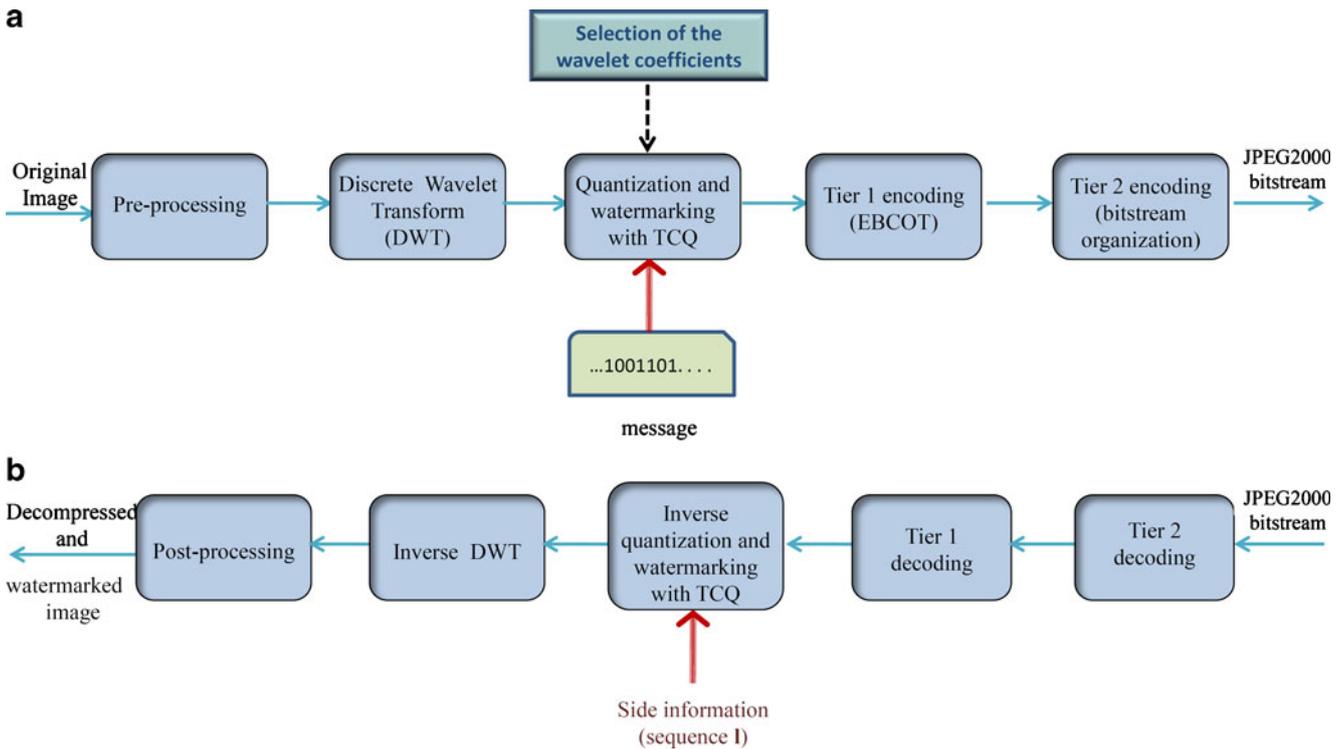


Fig. 4 The joint JPEG2000/watermarking scheme: **a** compression process, **b** decompression process

of this sub-band are not selected by the watermarking process. Therefore, we choose to embed the watermark in the HL, LH, and HH detail sub-bands of selected resolution levels. Wavelet coefficients of the other sub-bands are quantized with the classical TCQ algorithm of JPEG2000 part 2. The watermarking payload is determined by the number of detail sub-bands included in the watermarking process. The payload increases when we add more detail sub-bands from a new selection of resolution levels.

4.2 Description of the TCQ-based watermarking method

We aim to design a watermarking algorithm to quantize and watermark wavelet coefficients at the same time by using a TCQ-based quantization module. Our method is based on the principles of the QIM [1] approach associated with a trellis. We replace the uniform scalar quantizers used in JPEG2000 part 2 by shifted scalar quantizers with the same step size Δ as for the original ones. We can also use a higher step size by multiplying the original step size by a constant. These quantizers differ from the previous quantizers by the introduction of a shift d which is randomly obtained with a uniform

distribution over $[-\Delta/2, \Delta/2]$.⁵ We propose the following principle: If the bit to embed is the bit 0, then the quantizer D_j^0 , $j = 0, 1, 2, 3$ with the shift d_0 is used. If it is the bit 1, then we employ the quantizer D_j^1 with the shift d_1 satisfying the condition: $|d_0 - d_1| = \Delta/2$.

For each transition i in the trellis, two shifts $\mathbf{d}_0[i]$ and $\mathbf{d}_1[i]$ and four union quantizers $A_{0,i}^0 = D_{0,i}^0 \cup D_{2,i}^0$, $A_{1,i}^0 = D_{1,i}^0 \cup D_{3,i}^0$, $A_{0,i}^1 = D_{0,i}^1 \cup D_{2,i}^1$, $A_{1,i}^1 = D_{1,i}^1 \cup D_{3,i}^1$ are constructed. Thus, we will have two groups of union quantizers for the trellis structure used in our approach: the group 0, which consists of all shifted union quantizers corresponding to the watermark embedded bit 0, and the group 1, which incorporates shifted union quantizers corresponding to the embedded bit 1. Two dithered vectors \mathbf{d}_0 and \mathbf{d}_1 are constructed: Group 0 is associated to \mathbf{d}_0 and group 1 is associated to \mathbf{d}_1 . The trellis structure used in the proposed method has four branches leaving each state

⁵Shuchman [23] showed that the subtractive dithered quantization error does not depend on the quantizer input when the dither signal \mathbf{d} has a uniform distribution within the range of one quantization bin ($d \in [-\Delta/2, \Delta/2]$) leading to an expected squared error of $E^2 = \Delta^2/12$.

(Figs. 5 and 6a). For each state, two union quantizers instead of one are associated with branches exiting this state.

Let us consider a binary message \mathbf{m} to embed and a host signal \mathbf{x} . The embedding function $\text{Emb}(\mathbf{x}, \mathbf{m}) = \hat{\mathbf{x}}$ incorporates the message \mathbf{m} into \mathbf{x} , yielding the watermarked signal $\hat{\mathbf{x}}$. For a given transition i , $\hat{\mathbf{x}}[i]$ is obtained by quantization of the host component $\mathbf{x}[i]$ and Emb can be expressed as follows:

$$\text{Emb}(\mathbf{x}[i], \mathbf{m}[i]) = \left(\left\lfloor \frac{|\mathbf{x}[i] - \mathbf{d}_{\mathbf{m}[i]}|}{\Delta} \right\rfloor \right) \times \Delta + \mathbf{d}_{\mathbf{m}[i]}, \quad (1)$$

where $\mathbf{d}_{\mathbf{m}[i]}$ is the shift introduced in the selected quantizer of step size Δ . The watermark embedding process is split into two steps to perform watermarking within JPEG2000. The first step is achieved during the quantization stage of the JPEG2000 compression process. The quantization stage produces the sequence of TCQ quantization indices \mathbf{q} . For each transition i in the trellis, the union quantizers are selected according to the value $\mathbf{m}[i]$. The trellis is thus modified in order to remove all the branches that are not labeled with the

union quantizers that encode the message as illustrated in Fig. 6b. The subsets $D_{j,i}^{\mathbf{m}[i]}$, $j = 0, 1, 2, 3$ are associated to the branches of the modified trellis. The quantization index $\mathbf{q}[i]$ is given by:

$$\mathbf{q}[i] = Q_{D_{j,i}^{\mathbf{m}[i]}}, \quad (2)$$

where Q is the quantization function of JPEG2000, $\mathbf{m}[i]$ is the bit to embed at transition i , and $D_{j,i}^{\mathbf{m}[i]}$ is the shifted quantizer. For a given step size Δ , $\mathbf{q}[i]$ can be computed as:

$$\mathbf{q}[i] = \text{sign}(\mathbf{x}[i] - \mathbf{d}_{\mathbf{m}[i]}[i]) \left\lfloor \frac{|\mathbf{x}[i] - \mathbf{d}_{\mathbf{m}[i]}[i]|}{\Delta} \right\rfloor, \quad (3)$$

where $\mathbf{d}_{\mathbf{m}[i]}[i]$ is the shifting of the shifted quantizer $D_{j,i}^{\mathbf{m}[i]}$.

The second step is performed during the inverse quantization stage of the JPEG2000 decompression process. The trellis must be pruned in order to obtain the same trellis employed during the first step of the watermarking process. In addition to \mathbf{q} , the sequence \mathbf{l} is necessary to retrieve the modified trellis structure that have been used during the quantization stage. The reconstructed values $\hat{\mathbf{x}}$ are produced as:

$$\hat{\mathbf{x}}[i] = \bar{Q}_{D_{j,i}^{\mathbf{m}[i]}}^{-1}(\mathbf{q}[i]), \quad (4)$$

where \bar{Q}^{-1} is the inverse quantization function of JPEG2000. For a given step size Δ , the reconstructed value $\hat{\mathbf{x}}$ can be computed as:

$$\hat{\mathbf{x}}[i] = \text{sign}(\mathbf{q}[i])(|\mathbf{q}[i]| + \delta)\Delta + \mathbf{d}_{\mathbf{m}[i]}[i], \quad (5)$$

where δ is a user selectable parameter within the range $0 < \delta < 1$ (typically $\delta = 0.5$).

The proposed watermarking method have similarities with the dirty paper trellis codes (DPTC) [3]. Both methods rely on the use of a modified trellis associated to a codebook. However, we use a quantization codebook partitioned into subsets while Miller et al. use a pseudo-random code [3]. Moreover, the embedding of the watermark is done in a different way. Our joint scheme integrates a quantization-based method instead of DPTC codes, which optimally embed a watermark by applying an iterative embedding procedure with the constraint of minimizing the perceptual distance and maintaining constant robustness. The codeword is determined by using a correlation instead of a quantization.

4.3 Watermark embedding

The watermark embedding process is performed independently into each *code-block*. In order to add more

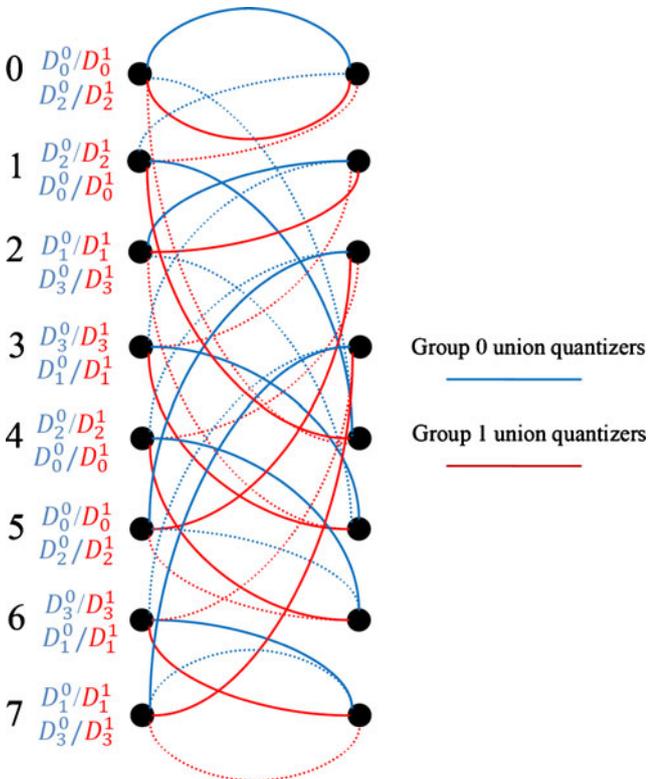


Fig. 5 A single stage of the trellis structure used our joint watermarking/JPEG2000 scheme with branch labeling

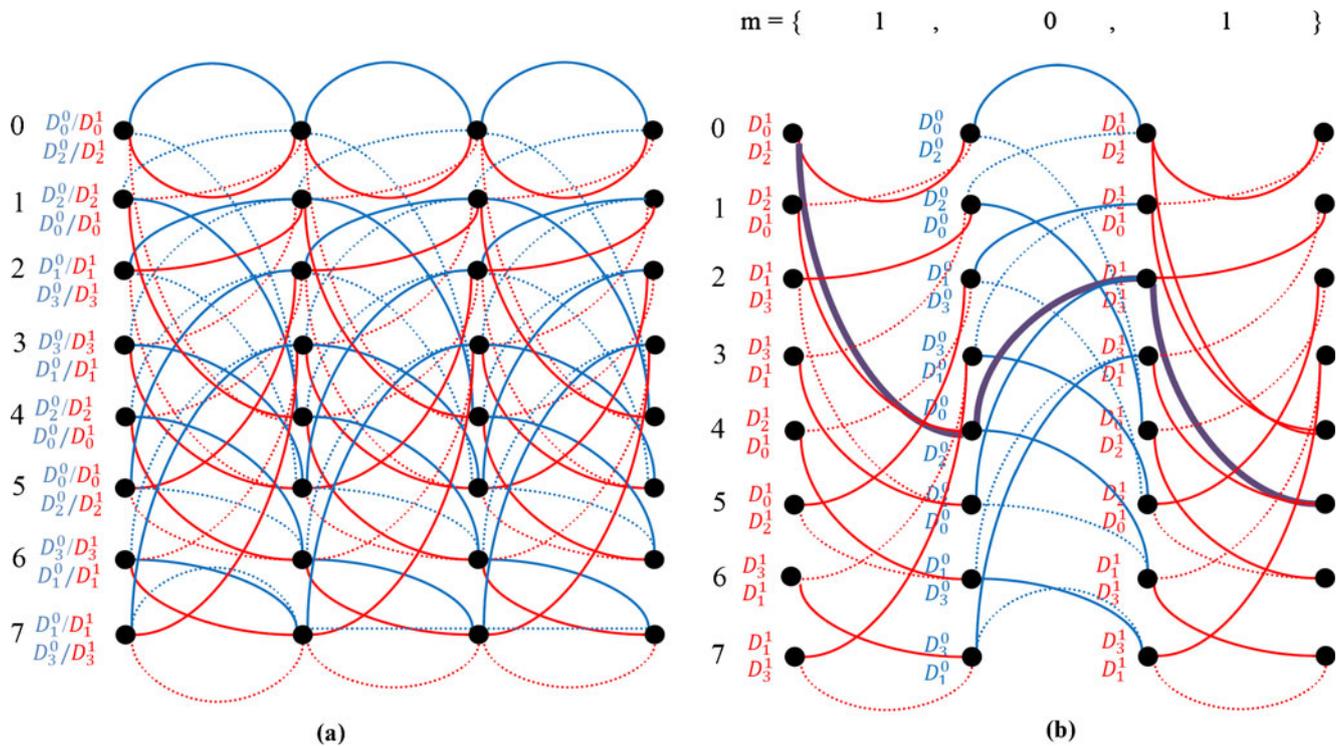


Fig. 6 **a** A three-stage modified trellis structure, **b** insertion of the message $m = \{1,0,1\}$: All the branches that are not labeled with the union quantizers that encode the message are removed.

The bold branches represent the optimal path calculated by the Viterbi algorithm

robustness to the message, we encode it with an error correcting code. After that, we shuffle (scatter) pseudo-randomly the bits of the coded message with a secret key.

Quantization For each *code-block C*, the quantization/watermark embedding procedures are:

- *Computation of the shiftings \mathbf{d}_0 and \mathbf{d}_1* : We use a pseudo-random generator initialized by the secret key k to compute the shiftings.
- *Generation of the group 0 and group 1 union quantizers*: For each transition i , we design shifted scalar quantizers. We label the branches of the trellis with these quantizers. Figure 6a shows a three-stage trellis structure used in our joint scheme. All the branches in the trellis are also labeled with reference numbers. At each transition, the straight branches are referenced by the value 0 and the dotted branches by value 1.
- *Trellis pruning*: The trellis is simplified so that all the branches through the trellis, and thus all the associated union quantizers encode the message \mathbf{m} as illustrated in Fig. 6b. For each transition, we save

the reference number of the surviving branches. We obtain the sequence \mathbf{l}_C .

- *Finding the optimal path*: The initial state of the given trellis structure is set to 0. The Viterbi Algorithm [15] is applied in order to find the minimum distortion path (Fig. 6b). The TCQ indices are produced (Eq. 3).

The sequences \mathbf{l}_C are combined to form the sequence \mathbf{l} . The obtained sequence \mathbf{l} is afterward crypted and stored in a file which is transmitted to the joint decoder as side information.

Inverse quantization The watermark embedding is completed during the inverse quantization of the JPEG2000 decompression stage. The image bitstream is decoded by the EBCOT decoder (tier 2 and tier 1 decoding) to obtain the sequence of decoded TCQ indices. For each *code-block C*, the inverse quantization steps are the following:

- *Computation of the shiftings \mathbf{d}_0 and \mathbf{d}_1*
- *Generation of the group 0 and group 1 union quantizers*

- *Retrieval of the trellis structure used during the quantization stage:* The trellis structure with four branches leaving each state is generated. Each branch of the trellis is afterward labeled with the shifted quantizers and with the reference numbers. The sequence \mathbf{I} enables us to retrieve the pruned trellis used during the quantization stage. For each transition i in the trellis, the pruning is done by removing the branches that have their reference number not equal to $\mathbf{I}_C[i]$
- *Inverse quantization:* The pruned trellis is used to reconstruct the wavelet coefficients. Given the TCQ indices, the embedding of the watermark is achieved during the computation of the reconstructed wavelet coefficients (Eq. 5).

4.4 Watermark extraction

The watermark recovery from the watermarked/decompressed image is a blind watermarking extraction process. The following operations are performed:

- *Apply the DWT:* we apply the DWT on the decompressed watermarked image. Each sub-band included in the watermarking process is partitioned into blocks of same size as the JPEG2000 codeblocks. The coefficients belonging to the current block are stored in the vector \mathbf{y} . The following steps are repeated for each processed block.
- *Retrieve the shiftings \mathbf{d}_0 and \mathbf{d}_1 :* We retrieve the shiftings by using the secret key k , and we set the union quantizers group 0 and group 1.
- *Perform the TCQ quantization:* The decoder applies the Viterbi algorithm to the entire trellis (Fig. 6a). This identifies the path that yields the minimum quantization distortion between \mathbf{y} and the output codewords. The coded message is then recovered by looking at the TCQ codebook labeling represented by the branches in that path. After that, we invert the shuffle and apply the decoding of the error correcting code to retrieve the message.

4.5 Discussion on the security of the proposed watermarking method

In this section, the security of the proposed joint scheme is discussed. The security of a watermarking system concerns its capability to resist to intentional attacks. All the parameters of the watermarking scheme are assumed to be public except the secret key used for embedding. The objective of an intentional attacker is to estimate the secret parameters used during embed-

ding. An accurate estimate of the secret key allows to implement a number of different attacks such as message modification, message copy, or message erasure while keeping a very low distortion. Different classes of security attacks can be considered [25]. We place this study under the general watermark only attack (WOA) scenario where the attacker has only access to several watermarked contents. He knows the step size values of the JPEG2000 TCQ quantizers, the trellis structure, and the repetition code rate whereas he ignore the selected resolution levels (the selected wavelet coefficients), the embedded message, the shuffle parameters, and the secret key needed to compute the dither vectors. He has access to a collection of images watermarked with the same secret key. These images are called observations.

The security of the proposed watermarking scheme relies on the dither signal and the secret pseudo-random seed used for shuffling the bits of the coded message (secret permutation). The TCQ codebooks are randomized by means of a pseudo-random dither signal. The dither vectors \mathbf{d}_0 and \mathbf{d}_1 introduce a secret shift in the TCQ quantizers. If the same dither signal is reused, the observation of several watermarked images can provide sufficient information for an attacker to estimate the dither signal. This can be done by exploiting the information leakage between several watermarked contents by means of information theoretic measures such as equivocation or mutual information between the dither signal and the observations [24]. The use of a trellis adds more complexity to the estimation problem because the attacker has to estimate all the components of the dither vectors in one time by testing all possible trellis paths [16]. It is thus more difficult for the attacker to estimate the secret key, and he needs to have a large amount of observations to be able to entirely estimate the dither signal. Once the attacker has accurately estimate the dither vectors, he can obtain the shuffled coded message. He cannot obtain the embedded message because he does not have the secret key used for the shuffle. However, it is still possible for the attacker to invert the shuffle. In this case, the problem of inverting the shuffle relies on the security of the shuffle itself. So, the shuffle of the coded message adds more security to the watermarking method by keeping the embedded message hard to read for an unauthorized user.

The proposed watermarking scheme is vulnerable to security attacks in the WOA scenario as it is the case for quantization watermarking schemes [24]. The use of a trellis and the shuffle of the embedded message allows to enhance the security level of the proposed scheme. It is also possible to increase the security of

the watermarking system by proceeding to the shuffle of the wavelet coefficients of the selected code-blocks before embedding.

5 Experimental results

To implement our joint JPEG2000 and watermarking scheme, we chose to use the OpenJPEG library⁶ which is a JPEG2000 part 1 open-source codec written in C language. We replaced the scalar uniform quantization component by a JPEG2000 part 2 compliant TCQ quantization module. The following set of compression and watermark parameters were fixed: irreversible DWT 9-7, five levels of wavelet decomposition, one tile, no ROI coding, size of *code-block*: 64×64 for the second and third levels of resolution, 32×32 for the fourth level, and 16×16 for the fifth level. A message of 1,020 bits length is inserted in the detail sub-bands of the second to the fifth resolution level. The joint scheme embed 1 bit of the (non-coded) message for every 257 pixels in an image of size 512×512 . The message is encoded with a very simple repetition code of 1/64 rate. The choice of using this low rate is dictated by the fact that the higher frequency sub-bands have low energy. Δ_{TCQ} is the step size of the shifted TCQ quantizers. We use different values of Δ_{TCQ} in our experiments: $\Delta/4$, $\Delta/2$, Δ , and 2Δ where Δ is the JPEG2000 original step size value used in JPEG2000 part 1. $\Delta/4$ is the TCQ quantizer step size value used in JPEG2000 part 2.

The evaluation of the performances of the proposed joint scheme covers two aspects: On one hand, the compression performances are studied. On the other hand, watermarking performances are investigated. Therefore, we use three kinds of experimental protocols: The first one studies the compression performances of the proposed joint scheme under various compression bitrates. The second one examines the impact of JPEG2000 compression on the watermark and the compression rate/imperceptibility trade-off. The last one studies the robustness of watermarked images against four attacks.

5.1 First experimental protocol: compression performances

Series of experiments on different JPEG2000 grayscale test images of size 512×512 have been performed

to evaluate the compression performances of the proposed joint scheme. We set the compression ratio from 2.5 to 0.2 bpp. Quality assessment was carried out using two objective evaluation criteria, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)⁷

Table 1 shows the PSNR and the SSIM results obtained for four well-known test images: Bike, Clown, Lena, and Peppers with $\Delta_{TCQ} = \Delta/4$ (the TCQ quantizer step size value used in JPEG2000 part 2). We notice that the joint scheme exhibits very good quality performances in terms of PSNR. For all the tested bitrates, the obtained values are greater than 30 dB except for Peppers image at 0.2 bpp (29.22 dB). The SSIM values are good at high bitrates and prove that our joint scheme provides good perceptual quality. We note that the quality degradation resulting from watermark embedding is very small when we compare, respectively, between the PSNR (and the SSIM) computed for the JPEG2000 compressed image and those computed for the compressed and watermarked image. Moreover, the PSNR obtained at some bitrates are slightly greater than those obtained with the classical JPEG2000 part 2 coder. This is due to the use of shifted TCQ quantizers during inverse quantization of the decoded wavelet coefficients. This sometimes provides reconstructed values closer to the original ones than those obtained with the JPEG2000 TCQ quantizers. It means that, compared to JPEG2000 compression, the visual quality obtained with our joint scheme is similar and sometimes better than with the conventional JPEG2000 coder. An example of watermarked and compressed image at different bitrates is presented in Fig. 7 for the Lena image.

The bitstream produced by the proposed joint scheme is compatible with the JPEG2000 part 2 image coding system. Conventional JPEG2000 part 2 decoders can therefore decode the watermarked bitstream and produce a decompressed image. In this case, the two union quantizers A_0 and A_1 are used to dequantize the decoded wavelet coefficients instead of group 0 and 1 dithered union quantizers. However, the JPEG2000 decoder produces an image which is close in quality to the one decoded with our joint scheme as shown in Table 2. The PSNR and SSIM results are similar and sometimes better than those obtained with the joint decoder ($\Delta_{TCQ} = \Delta/4$).

⁶The openjpeg library is available for download at <http://www.openjpeg.org>.

⁷SSIM is a perceptual measure exploiting human visual system properties. The SSIM values are real positive numbers lower or equal to 1. Stronger is the degradation and lower is the SSIM measure.

Table 1 PSNR (decibel) and SSIM for compressed image tests obtained with the joint scheme and comparison with the conventional JPEG2000 part 2 coder ($\Delta_{TCQ} = \Delta/4$)

Image test	bitrate (bpp)	PSNR (dB)		SSIM	
		JPEG2000	Our joint scheme	JPEG2000	Our joint scheme
Bike	2.5	43.23	42.99	0.9775	0.9785
	2	39.64	39.66	0.9547	0.9546
	1.6	39.33	39.40	0.9343	0.9342
	1	38.11	38.17	0.8852	0.8853
	0.5	36.51	36.49	0.8031	0.8017
Clown	0.2	33.52	33.50	0.6737	0.6814
	2.5	44.08	44.70	0.9885	0.9886
	2	42.78	42.83	0.9817	0.9815
	1.6	40.77	41.04	0.9741	0.9740
	1	38.71	38.93	0.9596	0.9593
Lena	0.5	35.76	35.85	0.9249	0.9247
	0.2	31.09	31.20	0.8384	0.8393
	2.5	47.47	47.55	0.9836	0.9834
	2	45.33	45.27	0.9749	0.9749
	1.6	43.38	43.09	0.9661	0.9658
Peppers	1	41.55	41.64	0.9490	0.9491
	0.5	40.03	39.90	0.9226	0.9227
	0.2	36.56	36.55	0.8700	0.8711
	2.5	43.13	43.43	0.9800	0.9802
	2	39.69	39.75	0.9649	0.9651
	1.6	39.20	39.16	0.9531	0.9528
	1	39.03	39.08	0.9273	0.9273
	0.5	36.50	36.53	0.8878	0.8875
	0.2	29.35	29.22	0.8378	0.8376

5.2 Second experimental protocol: impact of the compression process on the watermarking performances

**Fig. 7** Lena image watermarked and compressed with our joint scheme at different bitrates ($\Delta_{TCQ} = \Delta/4$): **a** original image, **b** 2 bpp, **c** 1 bpp, **d** 0.2 bpp

We propose to study the impact of JPEG2000 compression on the watermark extraction. We have performed our experiments on 200 grayscale images of size 512×512 .⁸ We compute the average percentage of correct embedded bits extracted at different bitrates. We note that the embedded watermark information is completely recovered for all compression bitrates for $\Delta/2$, Δ , and 2Δ . For $\Delta/4$, the percentage of correct extraction of the embedded watermark is equal to 99% at 2.5 and 2 bpp, 98% at 1.6, 1, and 0.5 bpp, and 96% at 0.2 bpp.

Figures 8 and 9, respectively, show the average PSNR and the average SSIM curves at different bitrates and different step size values for the considered image database. We should notice that there is a trade off between the quantizer step size needed for a correct extraction of the watermark and the expected quality of the decompressed/watermarked image. The step size

⁸These images are from the BOWS2 database which is located at <http://bows2.gipsa-lab.inpg.fr>.

Table 2 Comparison between the PSNR (decibel) and SSIM of the images obtained from the watermarked bitstream with the proposed joint JPEG2000/watermarking decoder and the JPEG2000 part 2 decoder ($\Delta_{TCQ} = \Delta/4$)

Image test	bitrate (bpp)	PSNR (dB)		SSIM	
		Joint decoder	JPEG2000 decoder	Joint decoder	JPEG2000 decoder
Bike	2.5	42.99	43.00	0.9785	0.9778
	2	39.66	39.67	0.9546	0.9546
	1.6	39.40	39.39	0.9342	0.9342
	1	38.17	38.22	0.8853	0.8853
	0.5	36.49	36.47	0.8017	0.8018
	0.2	33.50	33.48	0.6737	0.6815
Clown	2.5	44.70	44.57	0.9886	0.9885
	2	42.83	42.71	0.9815	0.9814
	1.6	41.04	40.99	0.9740	0.9739
	1	38.71	38.89	0.9593	0.9593
	0.5	35.85	35.83	0.9247	0.9248
	0.2	31.20	31.19	0.8393	0.8393
Lena	2.5	47.55	47.36	0.9834	0.9833
	2	45.27	45.19	0.9749	0.9748
	1.6	43.09	43.24	0.9658	0.9657
	1	41.64	41.70	0.9491	0.9491
	0.5	39.90	39.89	0.9227	0.9228
	0.2	36.55	36.49	0.8711	0.8712
Peppers	2.5	43.43	43.26	0.9802	0.9800
	2	39.75	39.69	0.9651	0.9649
	1.6	39.16	39.12	0.9528	0.9528
	1	39.08	39.07	0.9273	0.9273
	0.5	36.53	36.52	0.8875	0.8877
	0.2	29.22	29.23	0.8376	0.8377

used in JPEG2000 is small. It is big enough to ensure a correct extraction of the watermark. Nevertheless, the watermarks will not survive in case of strong power attacks. When using a larger Δ_{TCQ} , the fidelity is deteriorated because the distance between the quantization points grows. However, the advantage of using a larger Δ_{TCQ} value is that an improved robustness is obtained. The PSNR values obtained with a higher step size value are still acceptable in the context of a joint scheme as shown in Fig. 8. We note that the image fidelity decreases as the step size value increases. It is even more apparent when analyzing the SSIM values (Fig. 9). We note that the perceptual quality of the compressed and watermarked images at $\Delta_{TCQ} = 2\Delta$ decreases drastically in comparison with those obtained with other step size values. Figure 10 shows the watermarked and compressed Lena image at 1.6 bpp with step size values in the range of $\Delta/4$ and 2Δ .

5.3 Third experimental protocol: watermark robustness study

In a third round of experiments, the same database of 200 images has been considered to evaluate the robustness of the joint scheme. Four kinds of attacks have been performed: Gaussian filtering attack, Gaussian

noise attack, volumetric attack, and JPEG attack similarly to Miller et al. [3]. The bit error rate (BER) is computed for each attack. The BER is the number of erroneous extracted bits divided by the total number of embedded bits. When analyzing the results, the BER values lower than 0.1 are considered. The BER results

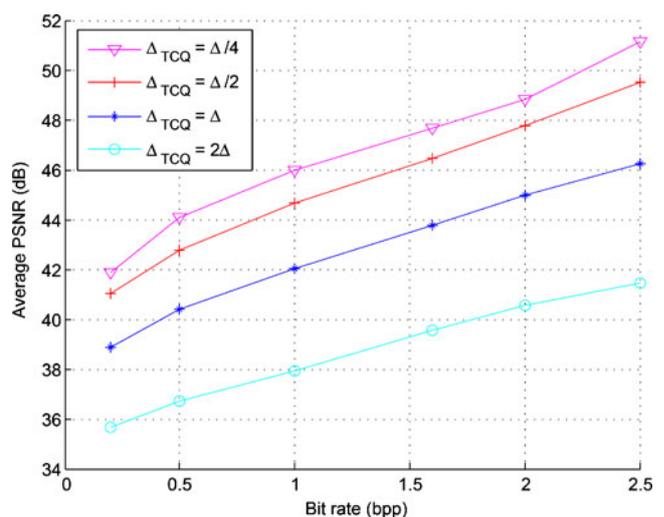


Fig. 8 Average PSNR results obtained by the proposed joint scheme for different step size values on 200 images of size 512×512

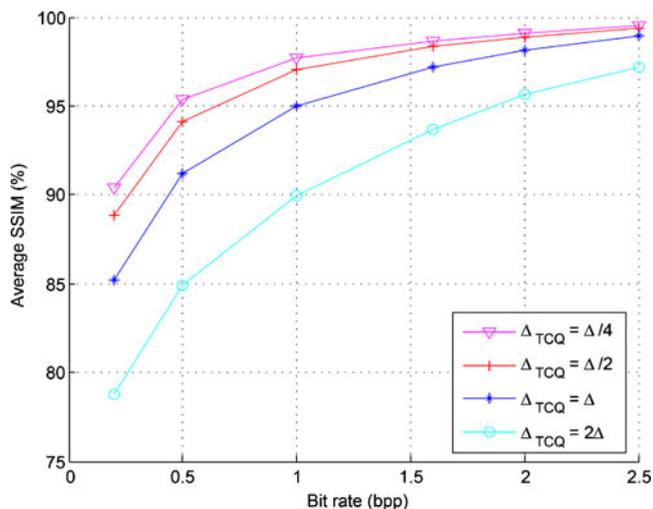


Fig. 9 Average SSIM results obtained by the proposed joint scheme for different step size values on 200 images of size 512×512

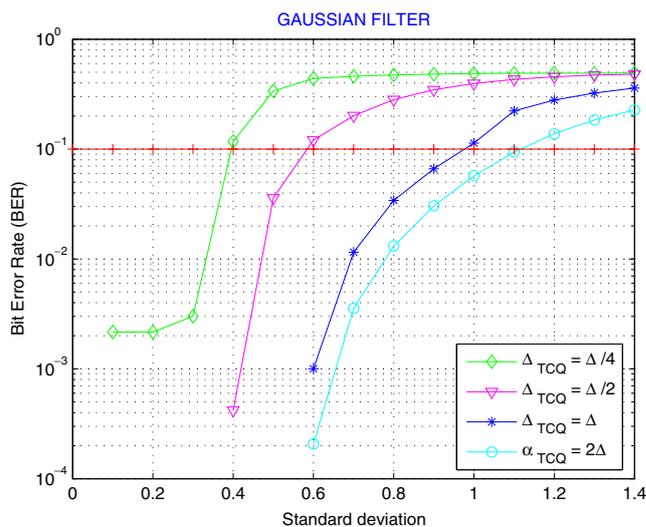


Fig. 11 BER results for filtering attack

for the four attacks are presented in Figs. 11, 12, 13, and 14. The logarithmic (base 10) scale is used for the Y-axis (BER results).



Fig. 10 Lena image watermarked and compressed with our joint scheme at 1.6 bpp with different step size values: **a** $\Delta_{TCQ} = \Delta/4$ (PSNR = 43.09 dB, SSIM = 0.9658), **b** $\Delta_{TCQ} = \Delta/2$ (PSNR = 41.86 dB, SSIM = 0.96021), **c** watermarking and JPEG2000 compression with $\Delta_{TCQ} = \Delta$ (PSNR = 39.53 dB, SSIM = 0.9412), **d** watermarking and JPEG2000 compression with $\Delta_{TCQ} = 2\Delta$ (PSNR = 36.28 dB, SSIM = 0.9010)

The watermarked images are filtered by Gaussian filter of width σ . The experiment was repeated for different values of σ , and the BER has been computed. The obtained results are reported in Fig. 11. The watermarks survive filtering effects up to $\sigma = 0.3$ for $\Delta_{TCQ} = \Delta/4$. As expected, the robustness is improved when the step size value increases. The BER obtained is lower than 0.1 when the joint scheme undergo Gaussian filtering up to: $\sigma = 0.5$ for $\Delta_{TCQ} = \Delta/2$, $\sigma = 0.9$ for $\Delta_{TCQ} = \Delta$, and $\sigma = 1.1$ for $\Delta_{TCQ} = 2\Delta$. The BERs for the four values of Δ_{TCQ} after an additive white Gaussian noise attack have been measured for different watermark-to-noise ratio as shown in Fig. 12. The joint scheme is not

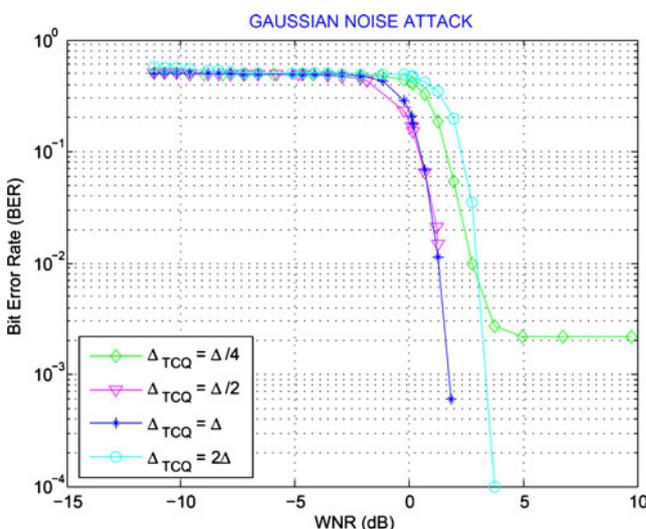


Fig. 12 BER results for Gaussian attack

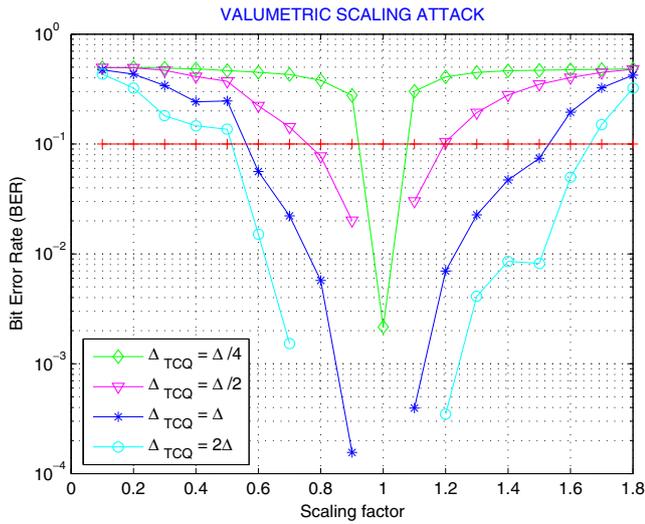


Fig. 13 BER results for scaling attack

very robust to this kind of attack. For $\Delta/4$ and $\Delta/2$, the BERs are high but it is more robust for Δ (up to $\sigma = 1$) and 2Δ (up to $\sigma = 2$).

The results against the volumetric scaling attack are summarized in Fig. 13. The quantized-based watermarking algorithms are recognized to be very sensitive to volumetric scaling (each pixel is multiplied by a constant). As usual, the robustness is better for $\Delta_{TCQ} = 2\Delta$: The BER results are under the 0.1 limit when there are changes in scaling with a scaling factor in the range of 0.6 and 1.6. Figure 14 shows the BER results against JPEG attack. We observe that the watermarks are only able to cope with JPEG quality factor up to 80 for $\Delta_{TCQ} = 2\Delta$. The weak robustness to JPEG attack is inherent to the approach since the coefficients included in the watermarking process are partly high-frequency wavelet coefficients.

In order to analyze the performance of the proposed joint system in terms of robustness, we compare its robustness with that of a conventional watermarking scheme. We use the dirty paper trellis codes [3], which have been proven to achieve high performances with respect to robustness and payload. We use a specific protocol for the DPTC code to be able to make a valid comparison: We perform a JPEG2000 compression attack after watermark embedding and before performing robustness attacks. We fixed the degradation to an average PSNR value of 45 dB for the two schemes. The payload is fixed to 1 bit embedded in 256 pixels (1,024 bits for 512×512 images). We use the step size $\Delta_{TCQ} = \Delta$ when comparing our results with those obtained by DPTC scheme because it provides the best trade off between robustness and visual

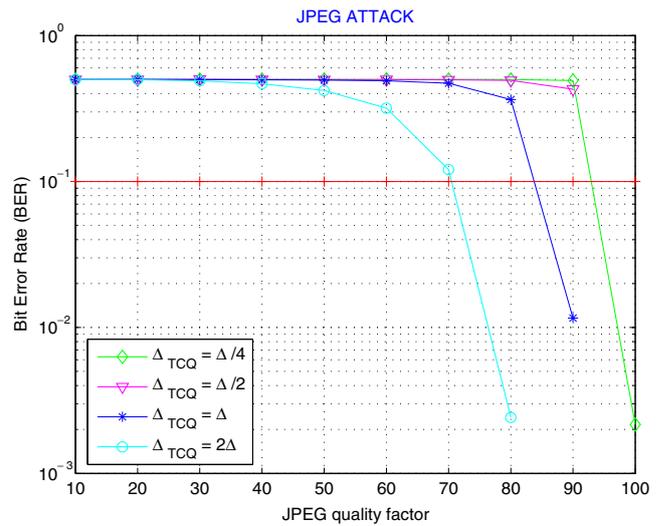


Fig. 14 BER results for JPEG attack

quality. We embed the message in the detail sub-bands of the second to the fourth resolution levels and the rate of the repetition code is $1/63$. The compression bitrate is fixed to 2 bpp. The experimental results show that DPTC outperforms our joint scheme in terms of robustness for Gaussian noise and JPEG attack. For Gaussian filtering, the robustness is relatively the same as ours, but the BERs obtained by DPTC are smaller than our joint scheme as shown in Fig. 15a. We notice that from $\sigma = 0.8$, we obtain better BER values than the DPTC algorithm. For the volumetric scaling, the DPTC results are better than ours for volumetric scaling down. However, we obtain better results for volumetric scaling up than DPTC as shown in Fig. 15b. To sum up, the DPTC scheme provides better robustness results than our joint scheme. The main reason is that DPTC watermarking makes use of both informed coding and informed embedding while our method uses only informed coding. However, DPTC scheme suffers from its CPU computation complexity. Three hours are necessary to watermark an 512×512 image with the DPTC algorithm on an Intel dual core 2-GHZ processor while it requires only 2 s to watermark and compress the same image with our joint scheme.

5.4 Comparison with other quantization-based joint schemes

We compare our results with those obtained by the three quantization-based watermarking schemes [11–13] of the state-of-the-art to evaluate the performances. The step size value used in the joint scheme is equal

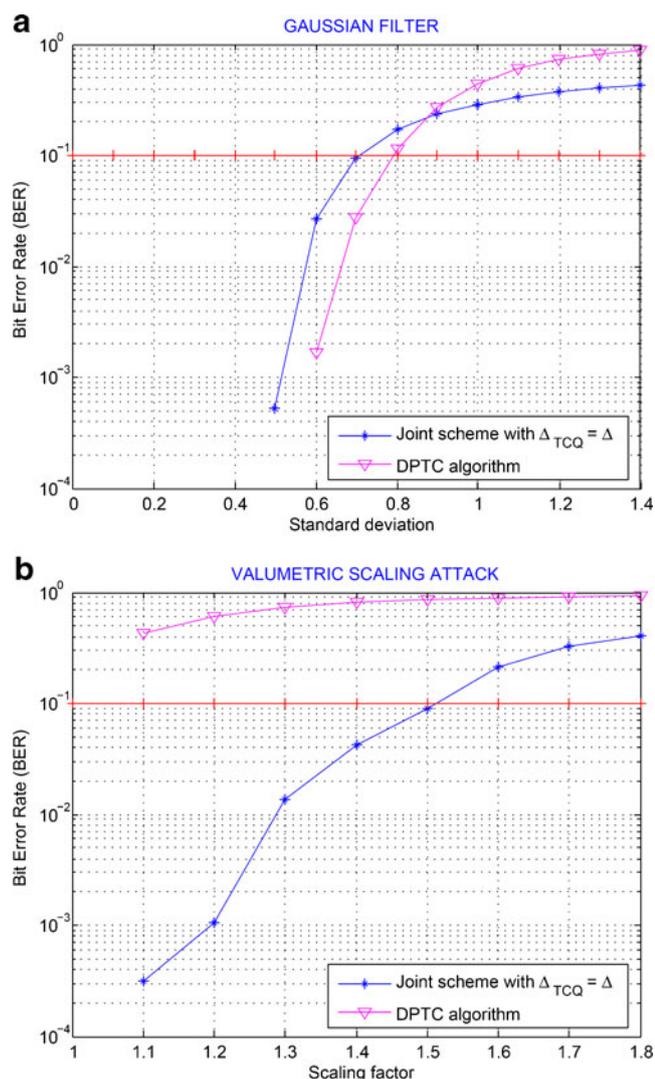


Fig. 15 Robustness performance comparison between our joint scheme and the DPTC scheme for: **a** Gaussian filter, **b** volumetric scaling up

to Δ . The first comparison is made between our work and Meerwald algorithm [11]. Meerwald gives PSNR values for the three test images, Lena, Goldhill, and Fishing boat without specifying the target bitrate. The watermark message length is relatively short, about 85 bits for Lena image, 194 bits for Goldhill image, and 383 bits for fishing boat image. For the same payloads and for different bitrates, our joint scheme gives better PSNR values. For Goldhill image, we obtain a PSNR greater than 36 dB for all the tested bitrates (40.78 dB at 2 bpp, 39.40 dB at 1 bpp, and 36.24 dB at 0.2 bpp) while Meerwald method gives a PSNR of 32.09 dB. For fishing boat image, the PSNR values are greater than 37 dB (41.07 dB at 2 bpp, 41.61 dB at 1 bpp, and 37.43 dB at 0.2 bpp) in comparison with 31.45 dB for Meerwald proposition. The second

Table 3 Comparison of image quality (in terms of PSNR) with Ouled-Zaid et al. scheme [13]

Image test	bitrate (bpp)	Our proposition PSNR (dB)	Makhloufi et al. PSNR (dB)
Lena	0.6	37.80	36.55
	0.4	37.30	35.23
	0.2	35.30	32.39
Bike	0.6	35.46	33.88
	0.4	33.15	31.58
	0.2	30.90	28.11

comparison is made between our work and Schlawweg et al. [12] scheme. Schlawweg et al. proposed a secure authentication scheme based on cryptographic tools. They use a four levels of wavelet decomposition. They mentioned in their paper that the PSNR obtained for Clown and Goldhill images are not good, and they only give the bitrate–PSNR curves for the Clown image. When we compared those curves with ours (for the same payload and the same bitrates), we find that our joint scheme gives similar PSNR values for low bitrates (<1 bpp) and better PSNR values for high bitrates. The last comparison is made between our proposition and Ouled-Zaid et al. [13] scheme. The results of the comparison in terms of PSNR are reported in Table 3 for a payload of 4,096 bits and three levels of wavelet decomposition. We can notice that the PSNR obtained with our joint system are better than their results at different bitrates. When considering the robustness of the watermark against attacks, we cannot perform a valid comparison because none of the other schemes has made extensive experimentations on an image database.

6 Conclusion and perspectives

In this work, we propose a joint JPEG2000 coding and informed watermarking scheme based on TCQ which uses duality between the source and the channel coding with side information. The properties of our joint scheme are the following:

- Quantization and watermarking are performed simultaneously. This is the main contribution of this work. In this way, the distortion induced by the insertion of the watermark is minimized.
- Error correcting coding is employed in order to spread the watermark signal on the higher-

frequency sub-bands because the energy is much lower in comparison with the LL sub-band. It is also used to add redundancy and thus increase robustness.

- High watermarking payloads can be achieved by including as many detail sub-bands as necessary and by adjusting the rate of the error correcting code.
- The step size value of the TCQ quantizers can be set either according to the robustness to be achieved, either in terms of visual quality requirements.

One drawback of the proposed scheme is that some side information is needed at the decoder during JPEG2000 decompression to perform inverse quantization and complete watermark embedding. Experimental investigations demonstrate that this joint scheme is able to achieve good visual quality in terms of PSNR and SSIM. The proposed embedding technique can survive JPEG2000 compression at low bitrates. The watermark robustness against common image attacks have also been studied. It has been noticed that the robustness is improved when a higher quantizer step size value is used. Thus, the selection of the quantizer step size must be done optimally so that the best trade off between robustness and minimum quality degradation should be achieved. The proposed joint scheme can realize high watermarking payloads and can therefore be used in content description and management applications, or in information hiding.

References

1. Chen B, Wornell G (2001) Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Trans Inf Theory* 47:1423–1443
2. Eggers JJ, Bäuml R, Tzschoppe R, Girod B (2003) Scalar costea scheme for information embedding. *IEEE Trans Signal Process* 51:1003–1019
3. Miller ML, Doerr GJ, Cox IJ (2004) Applying informed coding and informed embedding to design a robust, high capacity watermark. *IEEE Trans Signal Process* 13(6):792–807
4. Taubman DS, Marcellin MW (2002) *JPEG2000 image compression fundamentals standards and practice*. Kluwer Academic, Boston
5. ISO/IEC JTC1/SC29 WG1 (2000) *JPEG2000 part II final committee draft version 1.0*
6. Costa M (1983) Writing on dirty paper. *IEEE Trans Inf Theory* 29:439–441
7. Dufaux F, Wee S, Apostolopoulos J, Ebrahimi T (2004) JPSEC for secure imaging in JPEG2000. *SPIE application of digital image processing*, Proc. SPIE. San Jose, USA, pp 319–330
8. Fan Y-C, Tsao HW (2007) A dual pyramid watermarking for JPEG2000. *Int J High Perform Comput Netw* 5:84–96
9. Fan YC, Chiang A, Shen JH (2008) ROI-based watermarking scheme for JPEG 2000. *J Circuits Syst Signal Process* 27(5):763–774
10. Li K, Zhang X-P (2003) Reliable adaptive watermarking scheme integrated with JPEG2000. *International Symposium on Image and Signal Processing and Analysis (ISPA)*, Proc. ISPA 2003, vol 1. Rome, Italy, pp 117–122
11. Meerwald P (2001) Quantization watermarking in the JPEG2000 coding pipeline. *5th joint working conference on communications and multimedia security, communications and multimedia security issues of the new century*, Proc. IFIP TC6/TC11. Darmstadt, Germany, pp 69–79
12. Schlaueg M, Profrock D, Muller E (2006) JPEG2000-based secure image authentication. *8th ACM Multimedia and Security Workshop*, Proc. MM&Sec 2006. Geneva, Switzerland, pp 62–67
13. Ouled-Zaid A, Makhloufi A, Olivier C (2009) Improved QIM-based watermarking integrated to JPEG2000 coding scheme. *J Signal Image Video Process* 3:197–207
14. Marcellin MW, Fischer TR (1990) Trellis coded quantization of memoryless and Gauss–Markov source. *IEEE Trans Commun* 38:82–93
15. Forney GD Jr (1973) The Viterbi algorithm. *IEEE Trans Inf Theory* 61:268–278
16. Braci S, Boyer R, Delpha C (2009) Security evaluation of informed watermarking schemes. *16th IEEE international conference on image processing (ICIP)*, Proc. ICIP 2009. Cairo, Egypt, pp 117–120
17. Ouled Zaid A, Makhloufi A, Bouallegue A (2007) Wavelet domain watermark embedding strategy using TTCQ quantization. *Int J Comput Sci Netw Secur (IJCSNS)* 7(6):165–170
18. Le-Guelvouit G (2005) Trellis-coded quantization for public-key watermarking. *IEEE international conference on acoustics, speech and signal processing (ICASSP 2005)*, see the website <http://www.gleguelv.org/pub/index.html>
19. Chou J, Pradhan SS, Ramchandran K (1999) On the duality between data hiding and distributed source coding. *Annual Asilomar conference on signals systems and computers*, Proc. AACSSC 1999, vol 2. Pacific Grove, USA, pp 1503–1507
20. Esen E, Alatan AA (2004) Data hiding using trellis coded quantization. *IEEE international conference on image processing (ICIP)*, Proc. ICIP 2004, vol 1. Singapore, pp 59–62
21. Wang X, Zhang X-P (2007) Generalized trellis coded quantization for data hiding. *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Proc. ICASSP 2007, vol 2. Honolulu, Hawaii, USA, pp 269–272
22. Fischer TR, Wang M (1992) Entropy-constrained trellis-coded quantization. *IEEE Trans Inf Theory* 38:415–426
23. Schuchman L (1994) Dither signals and their effect on quantization noise. *IEEE Trans Commun Technol (COM)* 12:162–165
24. Pérez-Freire L, Pérez-Gonzalez F (2008) Security of Lattice-based data hiding against the watermarked-only attack. *IEEE Trans Inf Forensics Security* 3(4):593–610
25. Cayre F, Fontaine C, Furon T (2005) Watermarking security: theory and practice. *IEEE Trans Signal Process*, Special Issue on Content Protection 53:3976–3987

A High Capacity Reversible Watermarking Scheme

Marc Chaumont^{a,b} and William Puech^{a,b}

^a Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II,
161, rue Ada, 34392 Montpellier cedex 05, France.

^b University of Nîmes, Place Gabriel Péri, 30000 Nîmes, France.

ABSTRACT

Many **reversible** watermarking solutions have been proposed since 1996: spread-spectrum approaches, circular interpretation on histogram approaches, lossless compression approaches, expansion approaches and histogram approaches. In this paper, we propose a solution whose embedding capacity goes beyond all those reversible schemes. For certain images, the reach payload of the proposed method is over **2 bpp**. This solution is an improvement of the Coltuc reversible watermarking approach published in ICIP'2007.

Keywords: Reversible watermarking, high-capacity, congruence-based watermarking.

1. INTRODUCTION

Reversible watermarking methods for images own three major requirements: a high embedding payload, a small complexity and a good visual quality. In this paper we mostly look at a high payload scheme with a small complexity. Our algorithm does not care about the quality criteria. It produce a kind of "salt-and-pepper" noise.

"Although salt-and-pepper artifacts might appear ugly, it must be remembered that they will be removed when the original Work is recovered. ... For example, suppose the watermarked Works are used only for browsing. ... In such a scenario, the watermarked Works need only to be recognizable, and salt-and-pepper artifacts might not be a serious problem. (Digital Watermarking and Steganography, 2007, p. 382)¹".

We then believe that this approach might been used for medical or satellite images. Moreover, it has already been successfully used for color information protection in.²

Since 1996, lots of reversible watermarking solutions have been proposed: spread-spectrum approaches,³ circular interpretation on histogram approaches,⁴ lossless compression approaches,^{5,6} expansion approaches⁷ and histogram approaches.^{8,9} Our proposed solution allows to obtain an embedding-payload going beyond all the previous reversible schemes. Moreover, the algorithm complexity is very small. The proposed solution in this paper is an improvement of the Coltuc reversible watermarking approach.¹⁰ More precisely, our objective is to correct the Coltuc scheme. Indeed, in some cases the process of¹⁰ can not be reverted due to a problem of dependencies during the decoding process. We have then re-formulated its scheme in order to clarified it. We also provide lots of experimental results. In Section 2 we deal with the new watermarking scheme. In Section 3 we treat more precisely few technical points and have a discussion about the scheme. Finally, in Section 4 and 5, we give results and conclude.

Send correspondence to Marc.Chaumont@lirmm.fr.

2. ALGORITHM PRINCIPLE

In the same way as the method proposed by,¹⁰ our algorithm lies on congruence computations. But in,¹⁰ Coltuc proposes only two possible states for each pixel of an image. In our approach, we define **three possible states** for a pixel:

- the state *embedding* which corresponds to a pixel *embedding* an integer coefficient belonging to $[1, n]$,
- the state *to-correct* which corresponds to a pixel that has been modified but *does not embed* any information; this pixel will be *corrected* during the reverting process,
- the state *original* which corresponds to an *original* pixel (i.e unchanged). This state is new compared to those defined in.¹⁰

Lets define a constant integer value n greater or equals to 3. Lets also define the T transform which takes two integers x_1 and x_2 as input and return an integer:

$$T : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$T(x_1, x_2) = (n + 1).x_1 - n.x_2.$$

In order to clarify the paper, we will name the coding process, the act of embedding a message into an image and the decoding process, the act of extracting the message and rebuilding the original image. Lets now define the three possible states and the coding-decoding algorithms.

2.1 Embedding state

A pixel i in the *embedding* state is a pixel such that:

$$0 \leq T(I(i), I(i + 1)) \text{ and } T(I(i), I(i + 1)) + n \leq L, \quad (1)$$

with I the original image (of N pixel size) whose grey-levels belongs to $[0, L]$. In the case of 8-bits images, L equals to 255. All pixels i which are in the *embedding* state:

1. are T transformed such that $I_T(i) = T(I(i), I(i + 1))$, with $I_T(i)$ the resulting transformed pixel,
2. **must** then embed a coefficient w **belonging to** $[1, n]$ such that $I_w(i) = I_T(i) + w$.

Note that after the embedding of a coefficient w belonging to $[1, n]$, it is impossible to recover $I(i)$ with the only knowledge of $I(i + 1)$. Indeed $I_w(i) = (n + 1).I(i) - n.I(i + 1) + w$ with $w \neq 0$, thus $I(i) = \frac{I_w(i) + n.I(i + 1) - w}{n + 1}$ and so:

$$(I_w(i) + n.I(i + 1)) \text{ mod } (n + 1) \neq 0. \quad (2)$$

This **congruence property** (2) allows to detect an *embedding* pixel during the decoding process. Note that during the decoding process, the $I_w(i + 1)$ pixel should have been previously reverted to $I(i + 1)$ in order to compute this congruence. This implies that the image order scan used during the decoding process is the opposite of the one used during the coding process.

2.2 To-correct state

A pixel i in the *to-correct* state is a pixel such that the negation of equation (1) is true:

$$T(I(i), I(i + 1)) < 0 \text{ or } T(I(i), I(i + 1)) + n > L.$$

All pixels that are in this *to-correct* state are then modified such that:

$$\begin{aligned} c &\leftarrow (I(i) + n.I(i + 1)) \text{ mod } (n + 1); \\ \text{if } (I(i) - c) < 0 \text{ then } c &\leftarrow -(n + 1 - c); \\ I_w(i) &\leftarrow I(i) - c. \end{aligned} \quad (3)$$

The c coefficients belong to $[-n, n]^*$ and are embedded (into the *embedding* pixels) in order to enable the reversibility of the *to-correct* pixels during the decoding process. We name the c coefficients the **corrective codes**. Note that after the modification expressed by equation (3), pixel $I_w(i)$ checks the property:

$$(I_w(i) + n.I(i + 1)) \bmod (n + 1) = 0. \quad (4)$$

This **congruence property** (4) allows to detect a *to-correct* pixel at the reverting process. Note that at the decoding process the $I_w(i + 1)$ pixels should have been previously reverted to $I(i + 1)$ in order to compute this congruence.

2.3 Original state

Given an image order scan for the coding, a pixel in the *original* state (i.e. unmodified pixel) must always be present just before a pixel in the *to-correct* state. For a top-bottom-left-right scan order, if a pixel i is in the *to-correct* state, then the pixel $i - 1$ **must be** in the *original* state. In order to ensure this strong property (*original* and *to-correct* pixels go by pairs), during the scan, when a pixel i is detected as a *to-correct* one, a forward research is proceeded in order to find the next *embedding* one (noted *next*). *Original* and *to-correct* states are then alternates between the i (or $i - 1$) position and the *next* - 1 position. See 3.1 for more details and Appendix section for the algorithms.

This grouping constraint (additional state to Coltuc scheme^{10†}) breaks the problematic dependencies during the decoding process. Remember that during the decoding, the image scan order is inverted. A *to-correct* pixel i may not be reverted immediately if its associated corrective code is still not extracted. Nevertheless, because the pixel $i - 1$ is an *original* pixel, the pixel $i - 2$ may be treated immediately and the decoding process may continue (pixel i will be corrected later, in a second pass).

2.4 Coding and decoding algorithms

The **coding** process is composed of two steps (see Appendix section for the coding algorithm):

1. classify each pixel in one of the three states: *embedding*, *to-correct*, *original*,
2. embed into the *embedding* pixels, the watermark made of corrective codes plus the message.

Note that the image scan order has been chosen to be from top to bottom and from left to right.

For the **decoding** process, the image scan order is inverted; it is perform from bottom to top and from right to left. The decoding process is also composed of two steps (see Appendix section for the decoding algorithm):

1. extract the watermark from the *embedding* pixels, revert (during the scan) all those pixels and localize the *to-correct* pixels,
2. from the extracted watermark retrieve the corrective codes and the message, and correct the *to-correct* pixels.

3. FEW TECHNICAL DETAILS

3.1 About pairs of to-correct and original pixels

During the first step of the coding process the image is scan from top to bottom and from left to right in order to classify each pixel in one of the three states: *embedding*, *to-correct* and *original*. A pixel is classified in the *embedding* state if equation (1) is verified. When equation (1) is not verified, one have to alternate *original* and *to-correct* sites until reaching an *embedding* site. In this case, **we have** then to respect the constraint of producing **pairs** of *original* and *to-correct* sites.

Lets note i the position of the encountered pixel which does not verify equation (1). Lets note j the position of the first encountered pixel starting from i which verifies equation (1). If $j - i$ is odd (respectively even) then alternate *original* and *to-correct* sites from $i - 1$ (respectively i) to $j - 1$ (respectively $j - 1$).

*In,¹⁰ the author forget the case $I(i) - c < 0$; this mistake implies a wrong range $c \in [0, n]$.

†Without this additional constraint, the scheme proposed in¹⁰ does not work in lots of cases.

3.2 About the embedding of corrective codes

During the first step of the **coding process**, the image is scanned from top to bottom and from left to right, and when a *to-correct* site is found, a corrective code is built. At the end of the first step of the coding, the coder own a list of corrective codes which are arranged in the find order. This list is then concatenated to the message before embedding. During the **decoding process**, after the first step, the corrective codes list is extracted and *to-correct* pixels are then corrected.

Another important point is that corrective codes belong to $[-n, n]$ and embedding is possible only with coefficients belonging to $[1, n]$. One can entropically encode (Huffman or Arithmetic coding) the corrective codes and then embed the obtained binary vector with respect to the embedding sites payload (i.e $\log_2(n)$ bits per pixel (bpp)). In a first approach and for simplicity we do not have chosen this solution. Our solution consists in embedding:

- either directly the corrective code **plus one**, if the corrective code belongs to $[0, n-3]$;
- either two coefficients w_1 and w_2 , if the corrective code c belongs to $[-n, -1]$ or $[n-2, n]$. The first coefficient w_1 is a special code used to communicate to the decoder. If $w_1 = n-1$, this signify to the decoder that w_2 represents a corrective code belonging to $[n-2, n]$. If $w_1 = n$, this signify to the decoder that w_2 represents a corrective code belonging to $[-n, -1]$. Thus, the coefficient w_2 allows to recover the corrective code.

Equations below resume the method used to code a *corrective code* c belonging to $[-n, n]$ in order to obtain coefficient(s) belonging to $[1, n]$:

$$\begin{cases} \text{if } c \in [0, n-3] & \text{embed } w_1 = c + 1 \\ \text{if } c \in [n-2, n] & \text{embed } w_1 = n - 1 \quad \text{and } w_2 = c - n + 3 \\ \text{if } c \in [-n, -1] & \text{embed } w_1 = n \quad \text{and } w_2 = -c \end{cases}$$

At the decoding step, the decoder extract a coefficient w_1 and, depending of the case, extract coefficient w_2 :

$$\begin{cases} \text{if } w_1 \in [1, n-2] & \text{decode } c = w_1 - 1 \\ \text{if } w_1 = n - 1 & \text{decode } c = w_2 + n - 3 \\ \text{if } w_1 = n & \text{decode } c = -w_2 \end{cases}$$

3.3 Hiding payload

Lets k be the number of *embedding* pixels. Each *embedding* pixel allows the insertion of **one** coefficient belonging to the range $[1, n]$ which corresponds to an embedding-payload of $\log_2(n)$ bits. The $N - k$ remaining pixels[‡] (*to-correct* and *original* pixels) represent $(N - k)/2$ **corrective codes**. Indeed, the *to-correct* and *original* pixels go by pairs and there is corrective codes only for the *to-correct* pixels. A **corrective code** belongs to the range $[-n, n]$ and is thus corresponding to a rate of $\log_2(2n+1)$ bits. The theoretical **real** payload of the watermarking scheme equals to the embedding payload **minus** the **corrective codes** bitrate[§] i.e:

$$\frac{k}{N} \log_2(n) - \frac{N-k}{2N} \log_2(2n+1) \text{ bits per pixel.} \quad (5)$$

When the number k of *embedding* pixels is close to N the payload is approximately of $\log_2(n)$ bpp. For example, if $k \approx N$, the payload can be close to 2 bpp with $n = 4$.

[‡] N is the image size.

[§]In¹⁰ Coltuc gives a theoretical payload of $\frac{k}{N} \log_2(n) - \frac{N-k}{N} \log_2(n+1)$ bpp which is in practical impossible.

3.4 Discussions

In the Coltuc scheme¹⁰ the decoding process necessitates for each watermarked pixel $I_w(i)$ the knowledge of the original pixel $I(i+1)$. There is thus a dependency between each pixel going from the last one to the first one. Unfortunately, only two states are used in its scheme (named in our paper the *to-correct* state and the *embedding* state). During the decoding process (i.e message extraction + reversibility), for any watermarked pixel $I_w(i)$ all the pixels $I_w(i+1), I_w(i+2), \dots, I_w(N)$ should have been recovered. This strong dependency yield to a **not always possible scheme**.

Indeed, in its scheme, some pixels are modified by adding them a coefficient (named in our paper **corrective code**). Those corrective codes must be embedded additionally to the user message. At the decoding process, if a pixel $I_w(i)$ has to be corrected and if the associated corrective code has not still been extracted, then the decoding process is impossible and the message extraction fails. For example, take the two last coefficients $I_w(N-1)$ and $I(N)$ and suppose that the pixel $I_w(N-1)$ should be corrected thanks to a corrective code. This correcting code is supposed to be embedded in the image but is still not extracted. It is thus impossible to revert the pixel $I_w(N-1)$ and thus impossible to continue the decoding process. Other cases of impossibility may occur during the decoding process.

In order to break this dependency problem, we introduce the supplementary *original* state. *Original* pixels and *to-correct* pixels go by couple and are alternate if a suite of non-embedding pixels occurs. Those improvement enable to have a valid scheme and to proceed to multiple watermarking.

4. RESULTS

In this section, few results are given for classical grey-level images, 512×512 , 8 bits per pixel. Table 1 gives the payload results with parameter $\mathbf{n}=4$. The obtained payload is really impressive. For example, for the Lena image the embedding payload is 457 256 bits. To the author knowledge, the best reachable payload is with an expansion approach proposed in.⁷ In,⁷ Tian obtains a payload of 1.97 bpp \equiv 516 794 bits for Lena image; In our proposed approach we reach a maximum payload of 2.06 bpp \equiv 541 464 bits with $\mathbf{n}=9$. We could moreover remark that the scheme own a very small computational complexity and that the payload may still be improved thanks to the corrective codes entropic coding.

Image	payload		PSNR
Airplane	457 256 bits	1.74 bpp	20.90 dB
Lena	445 294 bits	1.70 bpp	19.60 dB
Godhill	435 040 bits	1.66 bpp	18.56 dB
Peppers	418 710 bits	1.60 bpp	19.56 dB
Boat	409 444 bits	1.56 bpp	19.52 dB
Barbara	285 620 bits	1.09 bpp	18.82 dB
Baboon	195 608 bits	0.75 bpp	16.10 dB

Table 1. Payload on few classical grey-level images, 512×512 , 8-bits per pixel with $\mathbf{n} = 4$.

In theory the algorithm may be run several times. In practice, with $\mathbf{n} = 4$, the algorithm succeed in embedding bits in a second algorithm execution only with Airplane and Lena images and the payload is very low (0.4 bpp for Lena and 0.03 bpp for Airplane). With Godhill, Peppers, Boat, Barbara and Baboon there were not enough payload for a second pass embedding. From equation (5), the first term is smaller than the second one.

As a consequence, for an higher embedding payload, it is most of the time more interesting to use the algorithm just with one pass execution and with a value for parameter \mathbf{n} higher than 4. Indeed, as it could be observed in Figure 1, excepted for Baboon[¶] image, a higher payload is reached when $n > 4$.

Figures 2 and 3 give results of the embedding of a pseudo-random binary vector with parameter $\mathbf{n}=4$ on the grey-level image Lena 512×512 shown in Figure 2(a) and on the grey-level image *head* 228×256 shown in Figure 3(a). Figure 2(b) and 3(b) show the resulting watermarked image. Figure 2(c) and 3(c) show the three

[¶]In general, the payload is weak for strongly textured images and one should choose a small value for \mathbf{n} .

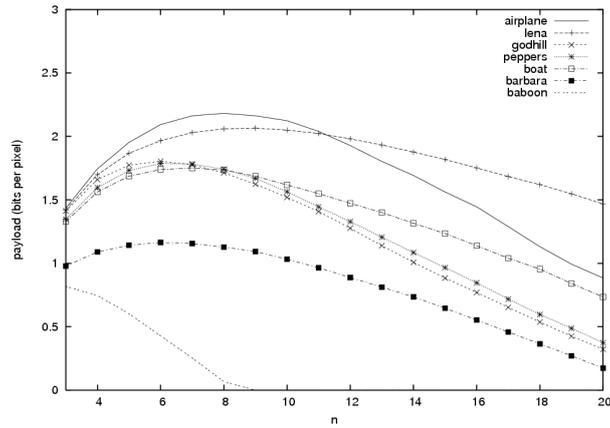


Figure 1. Real payload variation in function of the parameter n for few classical 512×512 , 8 bits images.



(a) 512×512 Lena image.



(b) Watermarked image, 1^{st} pass, $n=4$, PSNR=19.6 dB, payload=1.7 bpp.



(c) States image (1^{st} pass),
White: embedding,
Red: to-correct,
Black: original.

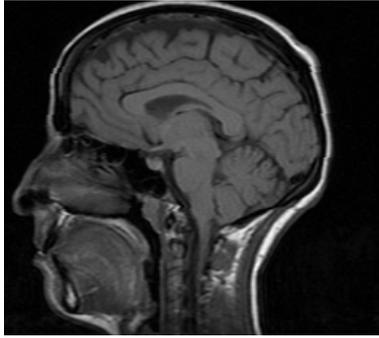


(d) Watermarked image, 2^{nd} pass, $n=4$, PSNR=14 dB, combined payload=1.7+0.4 bpp.

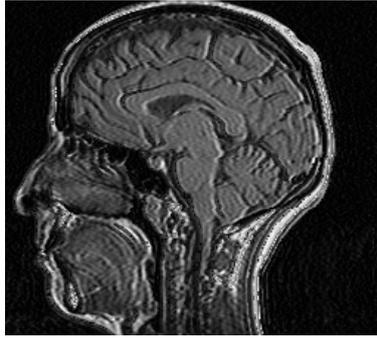


(e) States image (2^{nd} pass).

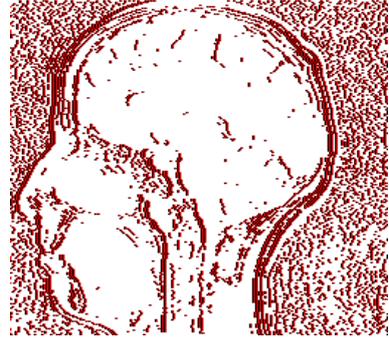
Figure 2. Illustration of the algorithm on Lena 512×512 image, with $n=4$ and 2 passes. The total embedding **real** payload equals to **2.1 bpp** i.e **550 940 bits**.



(a) 228×256 head image.



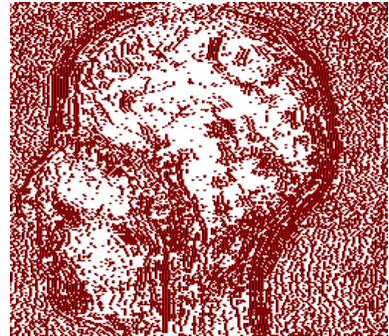
(b) Watermarked image,
1st pass, $n=4$, PSNR=19.8 dB,
payload=0.99 bpp.



(c) States image (1st pass),
White: embedding,
Red: to-correct,
Black: original.



(d) Watermarked image,
2nd pass, $n=4$, PSNR=14.3 dB,
combined payload=0.99+0.04 bpp.



(e) States image (2nd pass).

Figure 3. Illustration of the algorithm on $head 228 \times 256$ medical image with $n=4$ and 2 passes. The total embedding **real** payload equals to **1.03 bpp** i.e **60 478 bits**.

different sites: in white, there are the pixels embedding a coefficient belonging to $[1, n]$, in red there are the pixels to-correct (those pixels produce a corrective code belonging to $[-n, n]$) and in black there are the non-modified original pixels. The watermarked image obtained after two consecutive embeddings is shown in Figure 2(d) and 3(d). One can remark, on Figure 2(e) and 3(e), that in the second algorithm execution, there is lots of non-embedding sites (*to-correct* and *original* pixels).

5. CONCLUSION

To conclude, the proposed scheme in this paper improves the previous Coltuc scheme¹⁰ by breaking dependencies and thus ensuring that watermarking may always be reverted. Moreover, those dependencies broken allows multiple consecutive watermarks. Thus, embedding payload reach amazing values **around 2 bpp**. Such high capacities were never previously reached (see circular interpretation on histogram approaches,⁴ lossless compression approaches,^{5,6} expansion approaches,⁷ and histogram approaches^{8,9}). Future work should deal with the tradeoff between distortion and payload; indeed the current work always reach the maximum payload without taking into account distortion. Future work should also deal with the corrective codes compression and the used of a secret-key (modifying the order-scan) for the scheme' security.

ACKNOWLEDGMENTS

This investigation was supported in part by the TSAR project which is a french national project ANR-05-SSIA-0017-05 of the ANR ARA SSIA (*Agence Nationale de la Recherche, Action de Recherche Amont, Sécurité Systèmes Embarqués et Intelligence Ambiante*) and in part by the VOODOO project which is a french national project of the ANR Contenu et Interaction. We would also like to thank the Languedoc-Roussillon Region.

APPENDIX

Listing 1. *Embedding algorithm*

```

const Integer n;                                // n ≥ 3
Procedure embed(Image:  $I$ , Message:  $m$ ): $I_w$     // m coefficients ∈ [0, n-1]
begin

  List embedding;                                // Declaration of the embedding list
  List codes;                                    // Declaration of the corrective codes list

  // FIRST STEP
  for  $i$  from 1 to  $N-1$ 
  begin

     $t \leftarrow (n+1) \cdot I(i) - n \cdot I(i+1)$ ;    // T transform

    if (( $0 \leq t$ ) and ( $t+n \leq L$ ))
    then
       $I_w(i) \leftarrow t$ ;                            // Apply T transformation to the  $i^{th}$  pixel
       $embedding \leftarrow embedding \oplus i$ ;        // Add pixel  $i$  to the embedding list
    else
       $next \leftarrow \text{look\_forward\_for\_next\_embedding\_pixel}(i)$ ; //  $next$  is the next embedding pixel

      if ( $next - i$  is odd)
      then
         $k \leftarrow i - 1$ ;
         $I_w(i-1) \leftarrow I(i-1)$ ;                // Pixel  $i-1$  come back to its original value
         $embedding \leftarrow embedding \ominus i$ ;    // Remove pixel  $i$  from the embedding list
      else
         $k \leftarrow i$ ;
      end-if

      for  $j$  from  $k$  to  $next - 1$  alternate original and to-correct pixels
      begin
        if (it is to-correct turn)
        then
           $c \leftarrow (I(j) + n \cdot I(j+1)) \bmod (n+1)$ ; // Positive code
          if ( $(I(j) - c) < 0$ ) then  $c \leftarrow -(n+1 - c)$ ; // Negative code
           $I_w(j) \leftarrow I(j) - c$ ;
           $codes \leftarrow codes \oplus c$ ;            // Add corrective code  $c$  to the codes list
        end-if
      end-for
       $i \leftarrow next - 1$ ;                        // Next iteration, goes to the  $next$  embedding pixel
    end-if
  end-for

  Generate a coefficient sequence  $w$  with the corrective code list  $codes$  and the message  $m$ ;

  // SECOND STEP : embedding OF THE WATERMARK  $w$ 
   $\forall i \in embedding \ I_w(i) \leftarrow I_w(i) + w(i)$ ; //  $w(i) \in [1, n]$ 
end

```

Listing 2. *Extraction algorithm*

```

Procedure extract(Image:  $I_w$ ): $I$ ,  $m$ 
begin

  List to_correct;           // Declaration of the to_correct list

   $I \leftarrow I_w$ ;           // copy  $I_w$  into  $I$ 

  // FIRST STEP: EXTRACTION OF THE WATERMARK  $w$ 
  for  $i$  from  $N-1$  to 1
  begin

     $v \leftarrow (I_w(i) + n \cdot I(i+1)) \bmod (n+1)$ ;

    if ( $v = 0$ )
    then
      to_correct  $\leftarrow$  to_correct  $\oplus$   $i$ ; // Congruence property test
       $i \leftarrow i - 1$ ; // Case: pixel  $i$  is a to_correct pixel
    else // Add pixel  $i$  to the to_correct list
       $w \leftarrow w \oplus v$ ; // Important lign: pixel  $i-1$  is original and is not to treat
       $I_w(i) \leftarrow I_w(i) - v$ ; // Case: pixel  $i$  is an embedding pixel
       $I(i) \leftarrow \frac{I_w(i) + n \cdot I(i+1)}{n+1}$ ; // Concatenate coefficient  $v$  to the watermark  $w$ 
    end-if // Remove coefficient  $v$ 
    // Invert T transformation
  end

  Retrieve from the coefficient sequence  $w$  the corrective codes list  $codes$  and the message  $m$ ;

  // SECOND STEP: CORRECTION OF THE to_correct PIXELS
   $\forall i \in$  to_correct  $I(i) = I_w(i) + codes(i)$ ;
end

```

REFERENCES

- [1] Cox, I., Miller, M., Bloom, J., Fridrich, J., and Kalker, T., [*Digital Watermarking and Steganography*], ch. 11, 382, in *Multimedia Information and Systems*, Morgan Kaufmann, 2nd ed. (Nov. 2007).
- [2] Chaumont, M. and Puech, W., "A 8-Bit-Grey-Level Image Embedding its 512 Color Palette," in [*The 16th European Signal Processing Conference, EUSIPCO'2008*], 5 pages (Aug. 2008).
- [3] Bender, W., Gruhl, D., Morimoto, N., and Lu, A., "Techniques for data-hiding," in [*IBM Syst. J.*], **35**(3), 313–336 (1996).
- [4] Vleeschouwer, C. D., Delaigle, J., and Macq, B., "Circular Interpretation on Histogram for Reversible Watermarking," in [*IEEE International Multimedia Signal Processing Workshop, IMSPW'2001*], 345–350 (Oct. 2001).
- [5] Fridrich, J., Goljan, M., and Du, R., "Invertible Authentication," in [*IS&T/SPIE Annual Symposium on Electronic Imaging, Security Watermarking Multimedia Contents, SPIE'2001*], **4314**, 197–208 (Jan. 2001).
- [6] Celik, M. U., Sharma, G., and Tekalp, A. M., "Lossless Watermarking for Image Authentication: A New Framework and an Implementation," *IEEE Transactions on Image Processing* **15** (Apr. 2006).
- [7] Tian, J., "Reversible Data Embedding Using a Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology* **13**, 890–896 (Aug. 2003).
- [8] van Leest, A., van der Veen, M., and Bruekers, A., "Reversible Watermarking for Images," in [*IS&T/SPIE Annual Symposium on Electronic Imaging, Security Watermarking Multimedia Contents, SPIE'2004*], **5306** (Jan. 2004).
- [9] Ni, Z., Shi, Y.-Q., Ansari, N., and Su, W., "Reversible Data Hiding," *IEEE Transactions on Circuits and Systems for Video Technology* **16** (Mar. 2006).
- [10] Coltuc, D., "Improved Capacity Reversible Watermarking," in [*IEEE International Conference on Image Processing, ICIP'2007*], (Sept. 2007).

ATTACK BY COLORIZATION OF A GREY-LEVEL IMAGE HIDING ITS COLOR PALETTE

Chaumont M. and Puech W.

Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE

ABSTRACT

In this paper, we present a novel attack named *colorization attack*. This attack is specific to color-hiding watermarking schemes. The objective of this work is to demonstrate the feasibility of such an attack and thus to take it into account for the future color-hiding watermarking schemes.

Index Terms— watermarking, color-hiding, colorization attack, cost-function optimization

1. INTRODUCTION

The colorization is the act of adding color to a monochrome image or a movie [1]. This term was introduced by Wilson Markle in 1970 to describe its method for adding color to black and white movies or TV programs. Various techniques have been proposed [2, 3, 4], most of them based on an image segmentation into regions. Those techniques necessitate lots of manual intervention in order: to attribute a color to each region and to correct the wrongly segmented regions. Recently, the approach of Levin *et al.* [5] based on a very simple energetic model and its optimization leads to amazing good colorization results. Moreover this approach necessitates a very small manual intervention. In this paper, we propose to study the *colorization attack*. This attack is specific to watermarking schemes **hiding the color information** and is an indirect attack in order to retrieve a color image closed to the original one.

Few solutions have been recently proposed in order to **protect the color information** with the **data-hiding** paradigm. The grey-level image (embedding the color information) is freely accessible but its color version necessitates to own a key. The first group of solutions is based on wavelet decomposition and sub-band substitution and embed the chroma information in a grey-level image [6, 7, 8]. The purpose is not specifically to protect the color information but more classically to propose perceptive compression and image authentication for Campisi *et al.* [6] and Zhao *et al.* [7] and printing solution for Queiroz and Braun [8]. The second group of solutions is based on the decomposition of a color image into an *index* image and its color palette then the embedding of the color palette image into the *index* image [9, 10].

All those color-hiding solutions produce a grey-level image embedding the color information. The **hidden** color information is used to rebuild the color image. For all those schemes, the **enriched** grey-level **image** is very close to the original luminance image. A **colorization attack** may then be an **indirect** attack to retrieve a color image visually pleasant. Even if the colorized image is far from the original color one, it could give an attractive color version and then yield null and void the color-hiding watermarking schemes. Indeed, the added value of the color protection schemes is the color information; if a colorized and nice version is easily generated, the watermarking schemes are no more reliable because, for example, an illegal color printing may be sold.

In this paper we specially treat of colorization attack for the palette-based watermarking scheme [9, 10]. The subband wavelet substitution watermarking scheme [6, 7, 8] may them been attacked thanks to the colorization exposed in [5] or with an adaptation of our proposed method. Comparing to [5], we express analytically the solution to the problem (authors of [5] use a full "matrix inversion" and do not describe explicitly initials conditions) which involves, for the colorization process, a CPU complexity reduction. Moreover, we adapt the *colorization attack* to the palette-based watermarking schemes which implies an easier initialization: user gives generally less than 7 points with their associated colors.

2. THE COLORIZATION ATTACK

2.1. General colorization formulation

As shown in [5] a grey-level image may most of the time be nicely colorized with small human intervention and in low CPU complexity. In this section, we propose a scheme in order to colorize grey-level images that are in the same time *index* image ("colorization attack" of palette-based schemes). The palette-based schemes own a strong property: there is a bijection between grey-level values and a colors. It is thus easier to found just K colors compared to [5] where the number of unknowns equals to the size of the image.

As we already own an intensity information (the *index* image) we just need to extract the two chrominances planes. Thus, we decide to work in the YUV color space where the luminance plane Y is known (it is the *index* image) and where

U and V are the **two unknown chrominance planes**. Moreover, finding this two unknown planes is equivalent (in the palette-based watermarking scheme) to find the K colors of the palette C . Our unknown is thus the color palette C .

The model for the colorization problem is very simple: two spatially close pixels i and j should own a similar color if their intensity are similar. Mathematically, this may be expressed as minimizing for a pixel i all **weighted neighborhood differences** between the color $C(Index(i))$ of pixel i and the color $C(Index(j))$ of the pixel j belonging to the neighborhood of i which is noted $\mathcal{N}(i)$:

$$\forall i, \sum_{j \in \mathcal{N}(i)} w_{i,j} \cdot (C(Index(i)) - C(Index(j)))^2, \quad (1)$$

where $w_{i,j}$ is a weighting function, large when $Index(i)$ is similar to $Index(j)$ and small when this two intensities are different. The commonly used weighting function by segmentation algorithms for measuring the similarity between two intensity $Index(i)$ and $Index(j)$ is the Gaussian function:

$$w_{i,j} = a \cdot e^{-\frac{(Index(i) - Index(j))^2}{2\sigma_i^2}},$$

with a , a positive real constant and σ_i^2 the local variance in a window around pixel i . Note that those weighting functions are unsymmetric ($w_{i,j}$ is not necessary equal to $w_{j,i}$). Also note that other weighting functions may be used as proposed in [5]. Those weighting functions give a value around 1 if $Index(i)$ and $Index(j)$ are close with respect to the local variance σ_i^2 and a value near 0 otherwise.

2.2. Specific colorization formulation

Note that the cost function E objective in [5] is to minimize the difference between the color of pixel i and the **weighted neighborhood colors**:

$$E(x) = \sum_{i=1}^{i=N} \left(x(i) - \sum_{j \in \mathcal{N}(i)} w_{i,j} x(j) \right)^2, \quad (2)$$

where x is either the U plane (of size N pixels), either the V plane (of size N pixels) of the color space YUV. The objective is to find the two unknown U and V. This energetic model owns the same form than the one used in segmentation algorithms based on normalized graph cuts [11]. The solution is obtained by computing the second smallest eigenvector of $D - W$ with W the $N \times N$ weighting matrix and D the diagonal matrix [5]. The minimization is simply proceeded trough eigenvector decomposition of the $N \times N$ high dimension matrix $D - W$.

With the specific case of palette-based watermarking schemes the cost function may be re-written:

$$E(C) = \sum_{i=1}^{i=N} \left(C(Index(i)) - \sum_{j \in \mathcal{N}(i)} w_{i,j} \cdot C(Index(j)) \right)^2, \quad (3)$$

where C is the unknown color palette. The equation form is not similar to the previous one (2) and we could not use the eigenvector decomposition approach. We then decide to re-write a close form for equation (3) in order to facilitate its analysis and to found an analytic solution for the optimization. We are moreover adding an initialization constraint for user colors guess. The user manually sets L couples (i_l, c_l) where i_l is a pixel position and c_l its associated color. In order to respect the user choices we impose that the color $C(Index(i_l))$ of the pixel i_l will be a color close to given color c_l i.e. minimize $(C(Index(i_l)) - c_l)^2$. The cost function is thus made of a constraint term on neighborhood differences (equation 1) and a constraint term on user colors guess:

$$E(C) = \sum_{i=1}^N \sum_{j \in \mathcal{N}(i)} w_{i,j} \cdot (C(Index(i)) - C(Index(j)))^2 + \lambda \sum_{l=1}^L (C(Index(i_l)) - c_l)^2. \quad (4)$$

The solution of equation (4) is obtained by canceling $\frac{\partial E}{\partial C(k)}$. The algorithm is iterative and for each k , the $C(k)$ colors (two chrominances U and V unknown) are updated until convergence such that (detail algorithm is given on Listing 1): $\forall k \in [1, K]$,

$$\begin{aligned} N(k) &= \sum_{i|Index(i)=k} \left(\sum_{j \in \mathcal{N}(i) \text{ and } Index(j) \neq k} w_{i,j} \cdot C(Index(j)) \right) \\ &+ \sum_{i|Index(i) \neq k} \left(\sum_{j \in \mathcal{N}(i) \text{ and } Index(j)=k} w_{i,j} \cdot C(Index(i)) \right) \\ D(k) &= \sum_{i|Index(i)=k} \left(\sum_{j \in \mathcal{N}(i) \text{ and } Index(j) \neq k} w_{i,j} \right) \\ &+ \sum_{i|Index(i) \neq k} \left(\sum_{j \in \mathcal{N}(i) \text{ and } Index(j)=k} w_{i,j} \right) \\ C(k) &= \begin{cases} \frac{\lambda \cdot c_l + N(k)}{\lambda + D(k)} & \text{if } \exists l, Index(i_l) = k, \\ \frac{N(k)}{D(k)} & \text{if } \nexists l, Index(i_l) = k \end{cases} \end{aligned} \quad (5)$$

3. RESULTS AND DISCUSSION

Two *colorization attacks* have been proceeded on the color-hiding scheme of Chaumont and Puech [10]. The pirate was familiar to signal processing and had only access to the watermark image (i.e the *index* image). The two concern images are *baboon* 256×256 and *kodak-13 mountain stream* 256×384 .

Figures 1.b and 2.b show colorization results obtained respectively with 5 and 6 user colors. Note that the user has been asked to choose few colors without any knowledge of the original color version. Figures 1.c and 2.c give the rebuilt

color images knowing the decoding key i.e knowing the hidden color palettes. One can note a difference between the colorized images and the rebuild images or between the palettes obtain by a colorization attack (Figures 1.e and 2.e) and the hidden palettes (Figures 1.f and 2.f).

Nevertheless, remember that the attack should produce a pleasant color version which could be print and sold. In those conditions, the visual quality of the obtained colorized images are very good. Also note that it was not asked to the user to spend time to tune its results. Those results are obtained in less than 5 tries by image which take less than 5 minutes. Improved results may have been obtained with more allocated time and with an adapted visual interface.

As it is clear that colorization attack may be potentially a security hole; counter-attack should be proposed in order to improved color-hiding watermarking schemes. An immediate solution for those watermarking scheme is to generate a watermark grey-level image which is far from the original luminance image. Other propositions have to be proposed in the future in order to improve color-hiding schemes.

4. CONCLUSION

To conclude, this paper introduces a new (indirect) attack specific to *color-hiding schemes*. This attack does not allow to retrieve the original color version of a watermark image, nevertheless a pleasant colorized image may easily be created. Future *color-hiding schemes* should then take into account this particular attack in order to be more robust.

5. REFERENCES

- [1] G. Burns, *Colorization (The Encyclopedia of Television)*, Museum of Broadcast Communications, <http://www.museum.tv/archives/etv/index.html>.
- [2] W. Markle and B. Hunt, "Coloring a Black and White Signal Using Motion Detection," in *Canadian patent no. 1291260*, Dec. 1987.
- [3] J. Silberg, "The Pleasantville Post Production Team that Focussed on the Absence of Color," Cinesite Press Article, http://www.cinesite.com/core/press/articles/1998/10_00_98-team.html, Oct. 1998.
- [4] NeuralTek, "BlackMagic Photo Colorization Software," 2003, <http://www.timebrush.com/blackmagic>.
- [5] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using Optimization," in *International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH'2004*, Los Angeles, California, 2004, pp. 689–694.

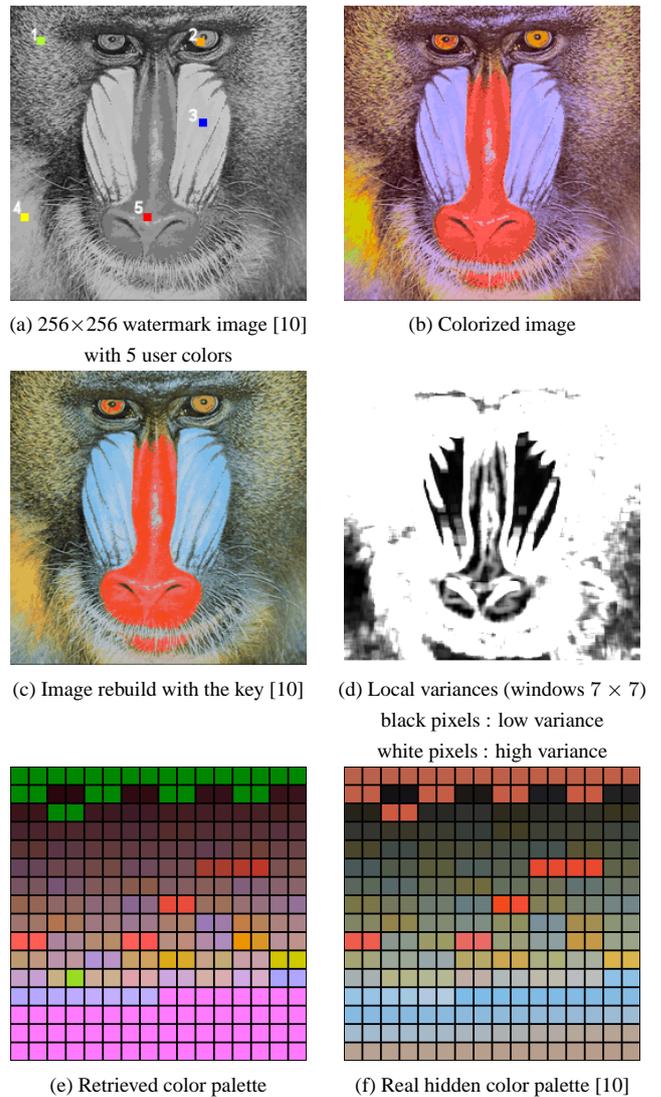


Fig. 1. Illustration of the colorization algorithm on baboon image

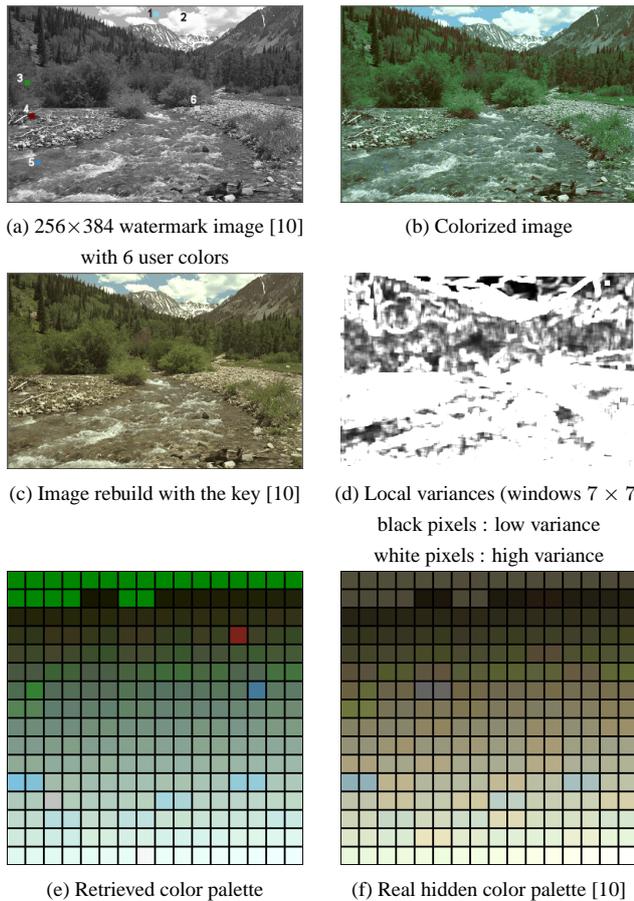


Fig. 2. Illustration of the colorization algorithm on kodak-13 mountain stream image

Listing 1. Colorization algorithm

```

const Integer NBITER; // maximum number of iteration

Procedure Colorization(): Palette
begin
    Palette Cnew, Cold; // current and previous color palette

    // TAKE INTO ACCOUNT THE L USER COLORS ;
    //  $\forall l \in [1, L], Cnew(Index(i_l)) \leftarrow c_l$ 
    init(Cnew);

    // ITERATIONS
    loop until NBITER reached or CONVERGENCE reached
    begin
        // COPY Cnew INTO Cold
        Cold  $\leftarrow$  Cnew;

        // UPDATE THE PALETTE Cnew WITH Cold KNOWLEDGE
        Apply equation 5;

        // SET IN CONFORMANCE EACH COLOR Cnew(k)
        // (U and V pixels  $\in [0, 255]$ )
        conformance(Cnew);
    end

    // RETURN PALETTE
    return Cnew;
end

```

- [6] P. Campisi, D. Kundur, D. Hatzinakos, and A. Neri, "Compressive Data Hiding: An Unconventional Approach for Improved Color Image Coding," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 2, pp. 152–163, 2002.
- [7] Y. Zhao, P. Campisi, and D. Kundur, "Dual Domain for Authentication and Compression of Cultural Heritage Images," *IEEE Transaction on Image Processing*, vol. 13, no. 3, pp. 430–448, 2004.
- [8] R. de Queiroz and K. Braun, "Color to Gray and Back: Color Embedding Into Textured Gray Images," *IEEE Transaction on Image Processing*, vol. 15, no. 6, pp. 1464–1470, 2006.
- [9] M. Chaumont and W. Puech, "A Fast and Efficient Method to Protect Color Images," in *IS&T/SPIE 19th Annual Symposium on Electronic Imaging, Visual Communications and Image Processing, VCIP2007, SPIE2007*, San Jose, California, USA, Jan. 2007, vol. 6508.
- [10] M. Chaumont and W. Puech, "A Grey-Level Image Embedding its Color Palette," in *IEEE International Conference on Image Processing, ICIP'2007*, San Antonio, Texas, USA, Sept. 2007.
- [11] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

3D-Face Model Tracking Based on a Multi-Resolution Active Search

Chaumont M. and Puech W.

Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE

ABSTRACT

This paper deals with face tracking in difficult conditions of non calibrated camera, strong head motions, thanks to a deformable 3D model. In those conditions, the proposed approach is able to detect and track a face. The novelty is mainly due to a multi-resolution Active Model search which allows to catch strong head motions. Results show an improvement between the single and the multi-resolution technique. Near real-time results are also provided.

Keywords: Multi-Resolution Active Model Search, 3D Face Tracking, Uncalibrated Camera, Near Real-Time.

1. INTRODUCTION

This paper deals with face tracking in video sequences with the used of a deformable 3D face-model. Current problems in face tracking are real-time constraint and the lack of robustness to luminosity variations, occlusions, fast motions and strong rotations. In this paper we focus especially on the fast motions, the strong rotations and the real-time problems. Our approach lies on a multi-resolution Active Model search (AM search). The novelties of our approach are the multi-resolution Active Model search and the near real-time performances.

In this paper we are limiting the test to an Active Model (AM) instead of an Active Appearance Model (AAM) as Dornaika and Ahlberg.¹ We do not then learn any texture modes variations. Our paper bring an improvement to Active Appearance Model approach² (the multi-resolution) but for the experiments and the approach justification we do not need to use the property of variation on textures and then we do not use a complete AAM approach. Additional improvements to the AAM approach, proposed in this paper, are the use of a 3D model and the illustration, through a complete implementation, that our face tracking solution is near real-time.

In comparison to La Cascia *et al*³ approach we use a deformable model which is richer than their rigid cylindrical model. Our model's deformations are proceeded directly during the tracking which gives additional informations about the face animation.

In comparison to 3D Model based tracking using an rigid model⁴ and a offline camera calibration, our tracking results are good despite no previous calibration. Our technique may thus be used for video sequences where camera characteristics are not known. Moreover, our approach allows a 3D model deformation. Some idea may nevertheless be catch from those matching points techniques. Indeed, as explain in,⁴ jitter and drift during tracking may be drastically reduced by using bundle adjustment and small number of matching points.

Lets also note that statistical approaches (particule filtering) give promising solutions. The probabilistic approaches provide better robustness to occlusions and could easily be added to deformable 3D model.⁵ The main problem of those solutions are their high CPU consuming time which make them unsuited for real-time applications.

Briefly, the first aim of this paper is to study the interest of *multi-resolution Active Model search* in the case of a tracking using a 3D-deformable face model in order to be more robust to strong motions. The second aim is to illustrate the real-time feasibility of approaches using a rough 3D deformable model. This paper is composed of two parts: the offline learning step (Section 2) where the pre-processing computations are explained and the tracking step (Section 3) where the *multi-resolution Active Model search* is analysed.

marc.chaumont@lirmm.fr; phone: +33 (0)4 67 41 85 14; william.puech@lirmm.fr; phone: +33 (0)4 67 41 86 85

2. OFFLINE LEARNING STEP

During the offline learning step the objective is to learn for a given person its specific 3D shape (Section 2.1), its specific texture (Section 2.2) and its specific update matrix (Section 2.3). Figure 1 shows the **shaped** 3D-face model and its associated texture obtained after the offline learning step. Note that this offline learning step may easily be extended to an AAM learning where a face data-base would have been used.¹ More details on shape learning and texture learning may be found in.⁶

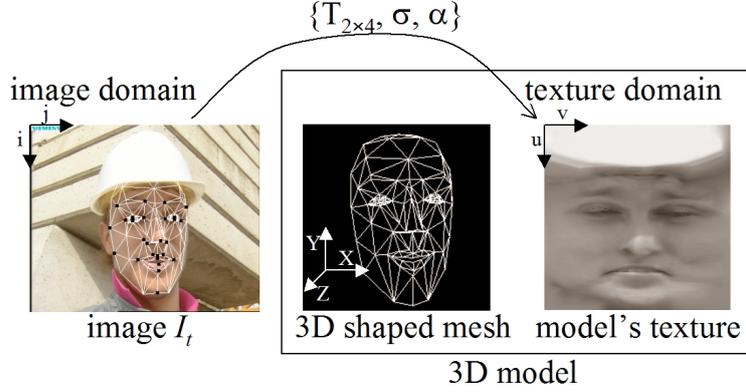


Figure 1. 3D shaped mesh and model's texture.

2.1. Shape learning

In order to learn the 3D shape of a specific person to track, we are using as input some 2D face feature points. The 2D feature points may be set manually or obtained thanks to an automatic approach.

A 3D-face model (CANDIDE-3⁷) is then deformed in order to best fit the 2D feature points. This deformation is proceed thanks to the minimization of the distance error E between the input set of 2D feature points $\{(u_i, v_i)^t\}$ and a set of 2D points $\{(u'_i, v'_i)^t\}$:

$$E = \sum_i (u_i - u'_i)^2 + (v_i - v'_i)^2. \quad (1)$$

The set of 2D points $\{(u'_i, v'_i)^t\}$ is obtained by applying on 3D vertex three linear operations: a shape deformation ($S_i \cdot \sigma$), an animation displacement ($A_i \cdot \alpha$) and a weak perspective projection* ($T_{2 \times 4}$) as expressed in the following equation:

$$\begin{pmatrix} u'_i \\ v'_i \end{pmatrix} = T_{2 \times 4} \cdot \underbrace{[A_i \cdot \alpha + S_i \sigma + M_i]}_{M'_i}, \quad (2)$$

where S_i and A_i are respectively the shape unit and the animation unit matrix, expressing the possible displacement of a vertex i . The displacement intensity is expressed by the weighting vectors σ and α . Equation 1 minimization gives parameters $T_{2 \times 4}$, σ , and α . Parameters $T_{2 \times 4}$ and σ are then used to deform the average 3D model and thus learn the specific face shape. More details on the minimization and the underlining hypothesis are given in.⁶

2.2. Texture learning

For an easier intelligibility we will name the image map: the *image domain* and the texture map: the *texture domain*. Once the 3D-model is shaped and the $T_{2 \times 4}$ pose is obtained (Section 2.1), the texture may be learned. This learning step is a simple warping procedure. The 3D mesh is projected onto the 2D image map in order to

*The weak perspective projection is also known as orthographic projection, graduate orthographic projection or affine projection.

give a 2D mesh in the image domain. The same process is done to obtain a 2D mesh in the texture domain. Then, each texture triangle from the 2D mesh of the image domain is warped to the corresponding triangle in the texture domain.

Lets note that the warping process needs two computational costly informations for each pixel: its associated triangle, and its three barycenter coefficients. Those informations are computed offline and do not change during the tracking process (indeed, the 2D mesh in the texture domain do not move). This pre-processing allows to pass of any 3D graphics card for image warping since it enables faster processing during the tracking step.⁸

2.3. Update matrix learning

Thus, once the texture, the 3D shape and the 3D pose are known, one compute the update matrix used for the tracking.

2.3.1. Single-resolution

During the tracking the objective is to project as well as possible the 3D model onto the image map in order to minimize the intensity difference computed between image I_t and projected model's texture. Without high lost of precision, one prefer minimizing the difference between the warped image[†] $W(I_t)$ and the model's texture I_m (Equ. 3.). This choice is done for real-time reason. Indeed, during the tracking step, the warping computation from image domain to texture domain is faster than the inverse warping due to offline pre-processed computation during texture learning (described in Section 2.2).

$$E(p) = \|r(p)\|^2 = \|W(I_t) - I_m\|^2. \quad (3)$$

Parameter p involved in minimization of Equ. (3) is composed of the animation vector (α) and the pose matrix ($T_{2 \times 4}$) (see Section 2.1). A first order Taylor expansion gives:

$$r(p + \Delta p) = r(p) + \frac{\delta r(p)}{\delta p} \Delta p.$$

During the model fitting, we wish to minimize the equation $\|r(p + \Delta p)\|^2$. So we are looking for the Δp which minimizes this equation. The solution of this least square problem is to choose Δp such that:

$$\Delta p = U.r(p),$$

$$\text{with } U = -(G^T G)^{-1} G^T,$$

$$\text{and } G = \frac{\delta r(p)}{\delta p}.$$

The update U matrix may be processed offline before the tracking. This update matrix is known as the negative pseudo-inverse of the gradient matrix G^\ddagger . G is computed by numeric differentiation such that the j^{th} column of G is estimated with:

$$G_j = \frac{r(p + hq_j) - r(p - hq_j)}{2h},$$

where h is a disturbance value and q_j is a vector with all elements zero except the j^{th} element that equals to one.

[†]The warping operation (image domain to texture domain) only transforms visible triangles. Hidden triangle regions in the texture domain will not be taken into account in the gradient descent computation ($r(p)$ equals zero in those regions).

[‡]G is a high dimension matrix. The number of lines is the number of parameters p , and the number of columns is the number of pixels.

2.3.2. Multi-resolution

With a single-resolution approach, the face target is lost when there is a strong head motion. To overcome that problem, we have chosen to use a multi-resolution tracking similarly to multi-resolution motion estimation.⁹ The multi-resolution approaches allow to keep valid the linear hypothesis near to the solution. Thus, multi-resolution pyramids are built (an image pyramid, a model’s texture pyramid and 2D mesh pyramids). A U_{r_i, r_t} matrix is then computed for each couple (r_i, r_t) where r_i is a given image resolution and r_t is a given texture resolution. Figure 2 shows few (r_i, r_t) possible couples.

During the tracking step, the low resolutions allow to catch strong motions (parameter p is roughly estimated) and high resolutions allow to catch motion details (parameter p is refine). Lets remark that experiments show that low resolutions are only of interest for the pose computation (and not for the facial animations) and that texture size should be similar to face-region size.

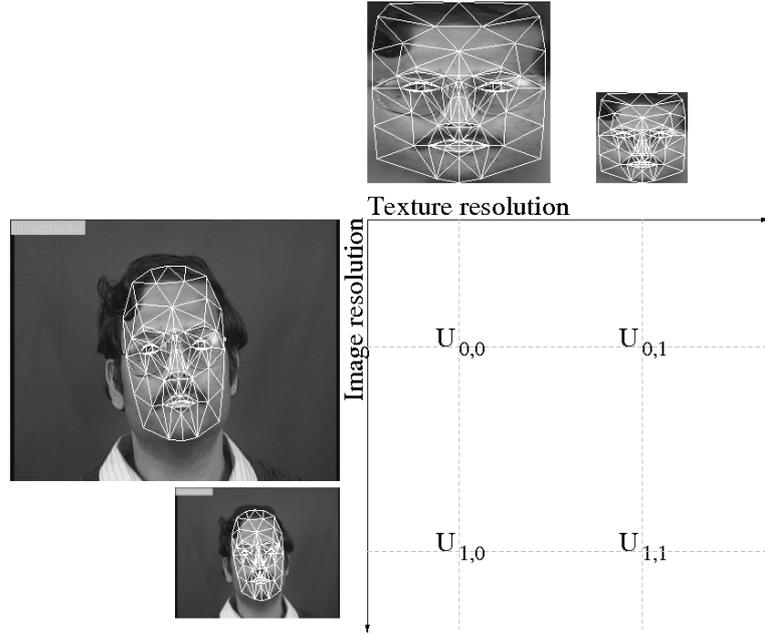


Figure 2. Multi-resolution matrix computations

3. TRACKING: ACTIVE MODEL SEARCH

3.1. Initialisation: face localization

In the case of face tracking, the face localization is in general a difficult problem.¹⁰ Even with a full implementation of an AAM approach one should initialize the 3D-face model pose relatively close to the solution. In the case of frontal view, many solutions have been proposed and the best results seem to be obtained by methods using a previous learning and a complete image scan (Neural Network, Hidden Markov Model, Support Vector Machine, Naive Bayes Classifier ...). We have then chosen to use one of this technique to localize the face.¹¹

3.2. Active Model Search

After the face localization, the Active Model search is proceeded by the multi-resolution gradient descent. For each *valid* couple (r_i, r_t) , the 3D model is iteratively updated until convergence or until a fixed number of iterations. The Active Model search algorithm is composed of four steps:

- Projection of the current image at resolution r_i into the texture domain at resolution r_t (knowing the current 3D model pose and its animation),

- Computation of the residue $r(p)$ of Equ. (3),
- Computation of the update parameter vector: $\Delta p = U_{r_i, r_t} \times r(p)$,
- Modification of the 3D model and its pose such that $p = p + \Delta p$.

The *valid* couples (r_i, r_t) used for the multi-resolution Active Model search are the *diagonal couples* (see the coordinate system on Figure 2) and the run goes from low resolutions to high ones. This scan choice allows to catch strong motions and is well fit for real-time constraint.

Lets remark that lots of computation are processed offline. Online computations are the image pyramid building, the 3D mesh projections, the images warpings (with the used of pre-processed accelerators), the image differences, the matrix products and the 3D model updates. Actually, the costly operations are the matrix products.

4. RESULTS

Tracking results are shown on two sequences. The sequence named *rotation* (30Hz, 360×288), comes from the M2VTS data-base and shows a person rotating the head and owning glasses. The second sequence named *erik* (CIF, 10Hz) shows a spoken person with lots of facial expressions and head motions. Below, the different steps of our multi-resolution face tracker are illustrated and commented.

For the first image, a face localization as to be proceeded. Remember that this may also be necessary with a complete AAM implementation. Figure 3 shows the result of a face localization, on the first image of the *rotation* sequence, by using a face detector.¹¹ The grey bounding box shows the face localization. Thanks to this bounding box, one deduce few 2D facial feature points and one minimize Equ. 1 in order to obtain an initial pose $T_{2 \times 4}$.⁶ The mesh (in white) represents the results of the 3D mesh projection obtained after this initial pose processing. Once this rough initialization as been computed, one run the multi-resolution active search (see Section 2.3.2), on this first image, in order to best fit in the 3D-face model.

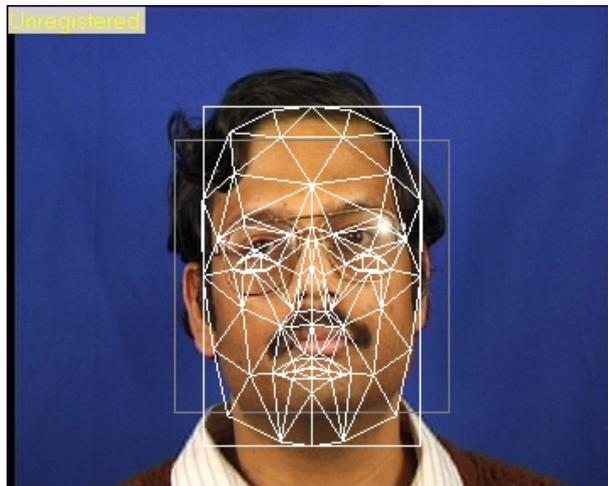
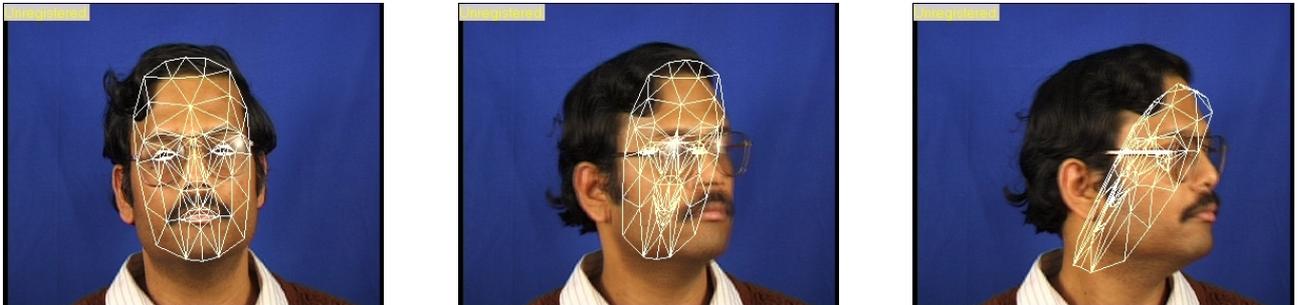


Figure 3. Localization of the face and rough model pose deduction

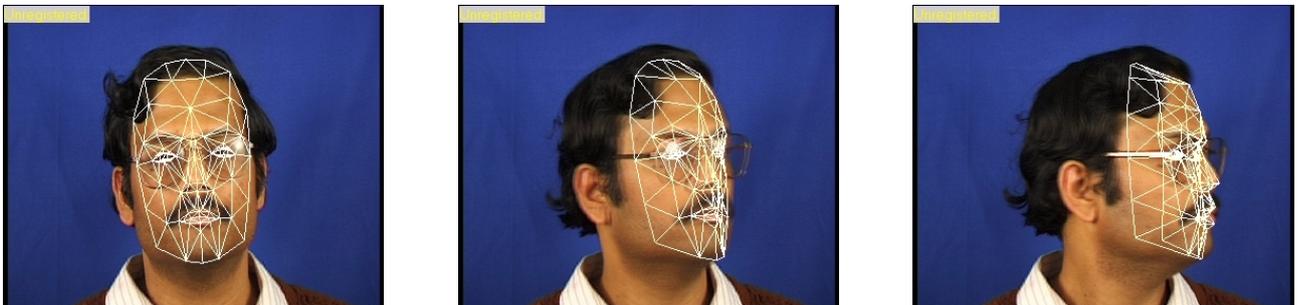
Figures 4 and 5 show the tracking results on few images. One see on the *rotation* sequence that the multi-resolution approach performs better results than the single-resolution approach. Indeed, there is a strong head motion in this sequence and the linearisation hypothesis is no more valid for the single-resolution approach. For the *erik* sequence, one observe that even if the face region is small, the facial expressions are well recovered. One should remark that those results are obtained without any previous camera calibration.



Images 3, 8 and 13 of the *rotation* sequence.



(a) Tracking **without multi-resolution** search; Images 3, 8 and 13 of the *rotation* sequence.



(b) Tracking **with multi-resolution** search; Images 3, 8 and 13 of the *rotation* sequence.

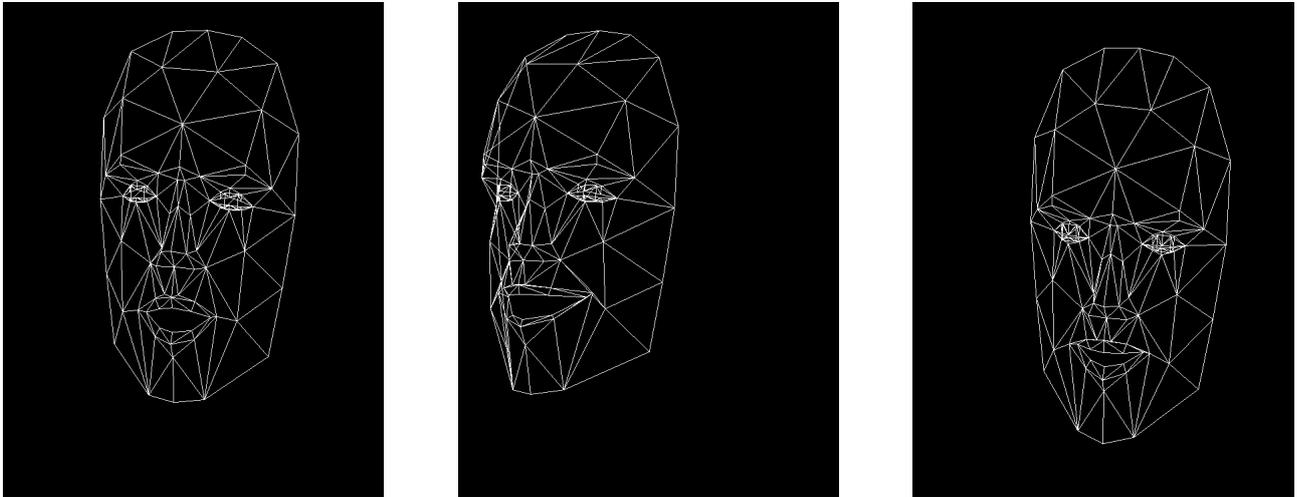
Figure 4. Comparison between a tracking with or without a multi-resolution search.



Images 3, 13 and 25 of the *erik* sequence.



Tracking with multi-resolution search; Images 3, 13 and 25 of the *erik* sequence.



3D Mesh for images 3, 13 and 25 of the *erik* sequence.

Figure 5. Illustration of multi-resolution tracking on *erik* sequence.

A software platform for web-cam acquisition, face tracking and screen displaying has been implemented using OpenCV library. Experimental conditions where a texture-image of size 100×100 , two levels of resolution, four iterations (resp. seven iterations) for resolution one (resp. resolution zero), non-natural desktop light conditions, and an Intel Pentium 2.4 GHz. With those difficult lighting conditions, the tracking have been released at 8Hz. One should note that this low frequency is due to the high texture-image size and to a non fully optimized code. A way to increase this frequency would be to decrease the texture-image size (40×40 is used in⁷) but results are less robust. Another solution would be to reduce the update matrix dimension (see Section 2.3.1) or to use local descriptors¹² instead of luminance information.

5. CONCLUSION

In this paper we proposed a face tracker based on a deformable 3D model catching facial expressions. The tracking is proceeded by a multi-resolution Active Model search (gradient descent). The novelties are the multi-resolution approach and the experimental proof of real-time possibilities. Results show an improvement of the tracking in the case of strong motions, the possibility to catch facial expressions even with small face-regions and an evaluation of real-time possibilities. Future works will deal with discriminant local descriptors, matrix dimension reduction and with statistical approach such as particle filtering.

REFERENCES

1. F. Dornaika and J. Ahlberg, "Fast and Reliable Active Appearance Model Search for 3D Face Tracking," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* **34**, pp. 1838–1853, Aug. 2004.
2. T.F.Cootes, G. Edwards, and C. Taylor, "Active Appearance Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2001* **23**, pp. 681–685, June 2001.
3. M. L. Cascia, S. Sclaroff, and V. Athitsos, "Fast, Reliable Head Tracking Under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2000* **22**, pp. 322–336, Apr. 2000.
4. L. Vacchetti, V. Lepetit, and P. Fua, "Stable Real-Time 3D Tracking Using Online and Offline Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2004* **26**, pp. 1385–1391, Oct. 2004.
5. F. Dornaika and F. Davoine, "Simultaneous Facial Action Tracking and Expression Recognition Using a Particle Filter," *IEEE International Conference on Computer Vision, ICCV'2005*, pp. 1838–1853, Aug. 2005.
6. M. Chaumont and B. Beaumesnil, "Robust and Real-Time 3D-Face Model Extraction," in *IEEE International Conference on Image Processing, ICIP'2005*, pp. 461–464, Sept. 2005.
7. J. Ahlberg, "CANDIDE-3 - Un Updated Parameterised Face," tech. rep., Department of Electrical Engineering, Linköping University, Jan. 2001.
8. J. Ahlberg, "Real-Time Facial Feature Tracking Using an Active Model With Fast Image Warping," in *International Workshop on Very Low Bitrate Video, VLBV'2001*, pp. 39–43, Oct. 2001.
9. S. Pateux, G. Marquant, and D. Chavira-Martinez, "Object Mosaicking via Meshes and Crack-Lines Technique. Application to Low Bit-Rate Video Coding," in *Picture Coding Symposium, PCS'2001*, (Seoul, Korea), Apr. 2001.
10. M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI'2002* **24**, pp. 34–58, Jan. 2002.
11. R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," in *IEEE International Conference on Image Processing, ICIP'2002*, pp. 900–903, Sept. 2002.
12. I.M.Scott, T.F.Cootes, and C. Taylor, "Improving Appearance Model Matching Using Local Image Structure," in *Information Processing in Medical Imaging, IPMI'2003*, pp. 258–269, July 2003.

Image watermarking schemes, watermarking schemes jointly to compression, data-hiding schemes.

Abstract

In this manuscript we address data-hiding in images and videos. Specifically we address robust watermarking for images, robust watermarking jointly with compression, and finally non robust data-hiding.

The first part of the manuscript deals with high-rate robust watermarking. After having briefly recalled the concept of informed watermarking, we study the two major watermarking families : trellis-based watermarking and quantized-based watermarking. We propose, firstly to reduce the computational complexity of the trellis-based watermarking, with a rotation based embedding, and secondly to introduce a trellis-based quantization in a watermarking system based on quantization.

The second part of the manuscript addresses the problem of watermarking jointly with a JPEG2000 compression step or an H.264 compression step. The quantization step and the watermarking step are achieved simultaneously, so that these two steps *do not fight* against each other. Watermarking in JPEG2000 is achieved by using the trellis quantization from the part 2 of the standard. Watermarking in H.264 is performed on the fly, after the quantization stage, choosing the best prediction through the process of rate-distortion optimization. We also propose to integrate a Tardos code to build an application for traitors tracing.

The last part of the manuscript describes the different mechanisms of color hiding in a grayscale image. We propose two approaches based on hiding a color palette in its index image. The first approach relies on the optimization of an energetic function to get a decomposition of the color image allowing an easy embedding. The second approach consists in quickly obtaining a color palette of larger size and then in embedding it in a reversible way.

Key words

Watermarking schemes, Dirty Paper Trellis Code, Rotation based embedding, Perceptual QIM, TCQ : Trellis Coded Quantization, H.264, JPEG2000, Color quantization, Color hiding, Color palette.

Schémas de tatouage d'images, schémas de tatouage conjoint à la compression, schémas de dissimulation de données.

Résumé

Dans ce manuscrit nous abordons l'insertion de données dans les images et les vidéos. Plus particulièrement nous traitons du tatouage robuste dans les images, du tatouage robuste conjointement à la compression et enfin de l'insertion de données (non robuste).

La première partie du manuscrit traite du tatouage robuste à haute capacité. Après avoir brièvement rappelé le concept de tatouage informé, nous étudions les deux principales familles de tatouage : le tatouage basé treillis et le tatouage basé quantification. Nous proposons d'une part de réduire la complexité calculatoire du tatouage basé treillis par une approche d'insertion par rotation, ainsi que d'autre part d'introduire une approche par quantification basée treillis au sein d'un système de tatouage basé quantification.

La deuxième partie du manuscrit aborde la problématique de tatouage conjointement à la phase de compression par JPEG2000 ou par H.264. L'idée consiste à faire en même temps l'étape de quantification et l'étape de tatouage, de sorte que ces deux étapes ne « luttent pas » l'une contre l'autre. Le tatouage au sein de JPEG2000 est effectué en détournant l'utilisation de la quantification basée treillis de la partie 2 du standard. Le tatouage au sein de H.264 est effectué à la volée, après la phase de quantification, en choisissant la meilleure prédiction via le processus d'optimisation débit-distorsion. Nous proposons également d'intégrer un code de Tardos pour construire une application de traçage de traîtres.

La dernière partie du manuscrit décrit les différents mécanismes de dissimulation d'une information couleur au sein d'une image en niveaux de gris. Nous proposons deux approches reposant sur la dissimulation d'une palette couleur dans son image d'index. La première approche consiste à modéliser le problème puis à l'optimiser afin d'avoir une bonne décomposition de l'image couleur ainsi qu'une insertion aisée. La seconde approche consiste à obtenir, de manière rapide et sûre, une palette de plus grande dimension puis à l'insérer de manière réversible.

Mots clés

Schémas de tatouage d'images, Dirty Paper Treillis Code, Insertion basée rotation, Perceptual QIM, TCQ : Quantification Codée par Treillis, H.264, JPEG2000, Quantification couleur, Dissimulation de la couleur, Palette de couleurs.