Tatouage robuste d'images par turbo TCQ

Robust image watermarking using Turbo TCQ

Gaëtan Le Guelvouit [1]

[1]Orange Labs – France Telecom R&D, 4, rue du Clos Courtel 35512 Cesson-Sévigné cedex
gaetan.leguelvouit@orange-ftgroup.com

Manuscrit reçu le 15 avril 2008

Résumé et mots clés

Cet article se concentre sur la mise en pratique du schéma idéal de Costa pour la problématique du tatouage de contenus multimédia. Après un rappel de la théorie, nous faisons une utilisation détournée de techniques de quantification pour construire un code correcteur adapté aux canaux avec information adjacente. La suite de l'article est consacrée à l'application de ce code pour le tatouage robuste d'images en niveaux de gris. Les résultats des expérimentations montrent des performances encourageantes, au niveau des papiers de référence du domaine, avec une mise en œuvre simple et efficace.

Tatouage, codes correcteurs, traitement d'image, quantification, principe turbo, TCQ

Abstract and key words

Robust watermarking is the art of embedding secret data within an host document. This watermark must be as transparent as possible, in order to preserve fidelity between host and marked document. It must also be robust, i.e. even if marked document is attacked – this concerns both usual multimedia transforms and malicious modifications – it should be possible to read the embedded message. A watermarking scheme is a compromise between transparency, robustness and capacity, i.e. embedded message length. It is widely admitted that watermarking is a communication problem. Thus, recent literacy dealt with digital communications tools to improve watermarking's compromise. This leaded to the re-discovery of channels with side information available at the encoder (see Fig. 2), and of the corresponding seminal paper by Costa[1]. The author demonstrated how to dramatically improve channel capacity. Unfortunately, his demonstration is impossible to implement in practice.

This paper deals with a new and simple code following Costa's paradigm, and with its application to image water marking. After the recall of some theory (Sec. 1), we show how to use scalar quantization to design surjective codebooks, as shown by Fig. 1 This first approach is improved using Trellis-Coded Quantization (Sec. 2.2). Defined by a transition function (see Eq. 4), a trellis is a set of states linked by valued arcs, as illustrated by Fig. 3. In the case of TCQ, those arcs are valued by quantizers. Each possible binary message corresponds to a path, which defines a sequence of scalar quantizers according to Eq. 5. In order to encode or decode a message, we use a customized Viterbi algorithm to find the best path and thus the best sequence of quantizers. For encoding, trellis transitions are enforced by strong a priori in order to output a path equal to the message to be encoded, while decoding process deals with complete trellis. Thanks to iterative principles, Turbo TCQ (TTCQ) improves this coding scheme using two parallel TCQ and interleaving (see Fig. 4).

While TTCQ ensures informed coding (i.e. the ability of fitting codewords to side information), we also use informed embedding according to Eq. 6. Experimental results shows a good level of performance (see Fig. 5) : for an embedding rate of 1/1 and to reach an error bit rate lower than 10^{-5}, our scheme is 5.5 dB better than SCS [6] – a well-known implementation of Costa's theory.

In Sec. 3, TTCQ-based code is applied to image watermarking. We use an experimental setup similar a reference paper [8]. In order to adapt watermark embedding rate, a spread transform of host DCT samples is done to get side informa-

tion (see Eq. 7). Then, message to be embedded is encoded thanks to TTCQ and inverse spread transform is computed. Final watermark is added to DCT samples shaping perceptual weights. For that purpose, we use a waterfilling algorithm, given by Alg. 1.
Results detailed in Sec. 4 confirms TTCQ-based code to be a good compromise between performance and complexity. Even with the addition of Gaussian noise, a set of 1200 watermarks has been extracted from 1200 images without any error (see Fig. 6). JPEG-based attack shows our scheme performs better than reference paper's, as seen in Fig. 7. Nevertheless, our approach is not completely robust to scaling attacks. This is confirmed by Figs. 8 and 9.
Finally, we conclude this article with a discussion concerning the use of vectorial quantizers. We show that quantization-based channel coding should not use the same lattices as source coding. An example is given by Fig. 10.

Watermarking, error correcting codes, image processing, quantization, turbo principle, TCQ

Introduction

Le tatouage robuste de contenus multimédia consiste à insérer au sein même d'un document hôte un message sous la forme d'une marque. Cette marque doit être la plus transparente possible afin que le contenu marqué reste normalement exploitable. La marque doit aussi être robuste afin de résister à des traitements : entre le moment où le contenu est marqué et celui où l'on tente de retrouver le message qui y est caché, le contenu a probablement subi des traitements (compression avec pertes, filtrages, dégradations volontaires, etc.). Enfin, la taille du message codé par la marque indique le débit du canal de tatouage. Une technique de tatouage peut donc être caractérisée par un compromis entre transparence, robustesse et capacité.
De nombreuses études ont tenté de modéliser le tatouage pour trouver les techniques et les paramètres permettant d'élargir les frontières du compromis que nous venons de voir. Ces études ont abouti sur une vision largement acceptée aujourd'hui qui considère le tatouage comme un problème de communication. En effet, cacher un message dans un contenu pour le retrouver plus tard, et ce malgré des altérations, est assimilable à la transmission d'informations sur un canal bruité. Suivant ce paradigme, les études académiques se sont alors focalisées sur l'utilisation de codes correcteurs, de techniques de modulation, etc. de façon plus ou moins formelle, pour améliorer le compromis transparence-robustesse-capacité.
Ces nouvelles approches ont permis la redécouverte d'un type de canaux de communication, particulièrement adapté au problème du tatouage [3] : les canaux avec information adjacente disponible à l'encodage. La suite de cet article décrit une approche simple et performante pour construire un schéma de tatouage exploitant les caractéristiques de ces canaux de communication. Dans la première partie, nous revenons sur la notion d'information d'adjacente dans les canaux de communication, puis sur l'étude de Costa [1]. Dans la section suivante, les techniques de quantification pour le tatouage sont abordées et nous décrivons plus particulièrement notre proposition, utilisant un code correcteur construit à base de turbo TCQ (*Trellis-Coded Quantization*). Les sections 3 et 4 montrent un exemple de mise en pratique (le tatouage robuste d'images) et présentent les résultats que nous obtenons face à des attaques communes. Enfin, nous terminons par une discussion sur l'amélioration de la technique.

1. Rappels sur les canaux avec information adjacente

Comme vu lors de l'introduction, nous considérons le tatouage comme un problème de communication : nous cherchons à transmettre un message $\mathbf{m}$ sous la forme d'une marque $\mathbf{w}$, via un canal bruité par le contenu hôte $\mathbf{x}$ et par des attaques[1]. Cette première partie rappelle le principe du schéma de Costa et son implication dans le problème du tatouage.
Considérons un canal gaussien avec une contrainte d'émission. Un signal $\mathbf{w}$ est émis avec une énergie limitée définie par

$$\frac{1}{n}\sum_{i=0}^{n-1}\mathbf{w}[i]^2 \leq P.$$

Le canal de communication est perturbé par un bruit $\mathbf{z}$ modélisé par une variable aléatoire suivant une loi Normale $Z \sim \mathcal{N}(0,N)$. La capacité du canal – c'est-à-dire le nombre de bits utiles par symbole $\mathbf{w}[i]$ émis – est donnée par [2]

1. Les attaques étant toutes les modifications subies par le contenu entre son marquage et le moment où l'on souhaite retrouver le message.

$$C = \frac{1}{2}\log\left[1 + \frac{P}{N}\right]. \quad (1)$$

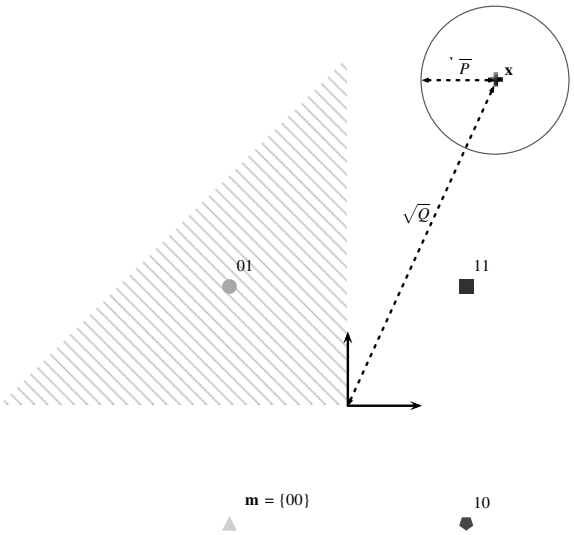
Maintenant, modélisons le tatouage par un canal de ce type[2]. Cela est illustré par la figure 2. La marque **w** – limitée en puissance pour des contraintes de transparence – est ajoutée au signal hôte **x** que nous modélisons par une variable aléatoire suivant une loi Normale $X \sim \mathcal{N}(0,N)$. Le tout est attaqué par un bruit **z**. Comme les deux sources X et Z sont indépendantes, la capacité du canal de tatouage est donc :

$$C = \frac{1}{2}\log\left[1 + \frac{P}{Q+N}\right]. \quad (2)$$

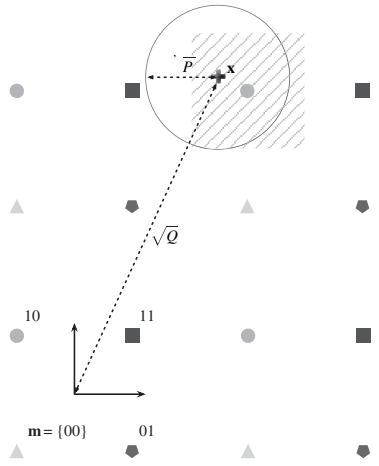
La capacité de la marque serait donc limitée par N et surtout par Q. En effet, $Q >> N$ car malgré l'attaque, le document doit rester exploitable.

La perturbation **x** présente la particularité d'être parfaitement connue au moment de l'émission de **w**. Cet état initial du canal est une information adjacente. En 1983, Costa [1] a démontré que cette information n'avait aucune influence sur la capacité du canal si l'on utilisait les techniques de codage adéquates. Cela ouvrit des perspectives importantes pour la problématique du tatouage, car on entrevit la possibilité de construire un schéma dont les performances dépendraient uniquement de l'attaque et pas du tout du signal hôte. Les gains espérés étaient énormes.

Pour atteindre la capacité, Costa propose dans son article un schéma théorique basé sur la construction d'un dictionnaire surjectif, puis sur une technique d'émission consistant à *diriger* l'information adjacente vers le mot de code, plutôt que de simplement l'ajouter. Il y a donc un codage informé et une émission informée, spécifiques aux canaux avec information adjacente. Dans la suite, nous proposons un codage adapté s'appuyant sur des techniques de quantification, et une méthode d'émission prenant en compte des critères perceptuels pour minimiser l'impact du tatouage sur la qualité des documents marqués.



(a) Code correcteur classique (un mot de code par message) : si Q est trop grand, il peut être impossible d'atteindre la zone du mot de code visé en respectant la contrainte P.



(b) Code surjectif basé sur la quantification scalaire : les mots de codes sont répétés régulièrement, si bien qu'un mot de code correspondant au message à transmettre est accessible où que soit **x**.

Figure 1. Représentation schématique de la répartition des mots de codes dans un espace de dimension n = 2.

2. Codes correcteurs basés sur la quantification

Costa démontre dans son article la formule de capacité en construisant un dictionnaire avec des mots de code tirés aléatoirement. C'est impossible à réaliser en pratique, car sans structure entre les mots de code, le décodage (qui correspond à la recherche du mot de code le plus proche du signal reçu) nécessite une recherche exhaustive dans l'ensemble du dictionnaire. La complexité est en $\mathcal{O}(2^n)$, n étant le nombre de bits utiles. Depuis la redécouverte de l'article de Costa, de nombreux travaux ont proposé des mises en pratique de son schéma idéal [4]. La plus connue est le schéma de Costa scalaire (SCS) [6], qui utilise la quantification scalaire de **x** pour encoder un message. La première partie de cette section rappelle les principes du SCS, la seconde introduit l'application de la turbo TCQ pour le codage d'information, et enfin la troisième compare ces deux techniques dans le cadre d'une émission informée.

2. C'est un modèle très simplifié, les signaux rencontrés étant rarement simplement gaussiens.

2.1. Schéma de Costa scalaire

Par soucis de simplicité, nous considérons une transmission binaire avec un message $\mathbf{m} \in \{0,1\}^n$. Pour un pas de quantification Δ donné, le SCS définit le dictionnaire $\mathcal{U}$, produit de dictionnaires scalaires $\mathcal{U} = \mathcal{U}[0] \times \mathcal{U}[1] \times \ldots \times \mathcal{U}[n-1]$, avec

$$\mathcal{U}[i] = \left\{ k\frac{\Delta}{2} + \mathbf{d}[i],\ k \in \mathbb{Z} \right\},$$

et où $\mathbf{d} \in [-\Delta/2, +\Delta/2]^n$ représente un bruit de dithering formant une clef secrète. Chaque message possible $\mathbf{m}$ est associé à un sous-dictionnaire $\mathcal{U}_\mathbf{m} \subset \mathcal{U}$ défini par

$$\mathcal{U}_\mathbf{m}[i] = \left\{ k\Delta + \mathbf{d}[i] + \frac{\Delta \mathbf{m}[i]}{2},\ k \in \mathbb{Z} \right\}.$$

L'encodage de $\mathbf{m}$ consiste à rechercher le mot de code de $\mathcal{U}_\mathbf{m}$ le plus proche de $\mathbf{x}$, c'est-à-dire plus formellement

$$\mathbf{u}^\star = \arg\min_{\mathbf{u} \in \mathcal{U}_\mathbf{m}} \|\mathbf{u} - \mathbf{x}\|^2. \tag{3}$$

L'encodage correspond donc à décoder le signal hôte en se limitant au sous-dictionnaire $\mathcal{U}_\mathbf{m}$.

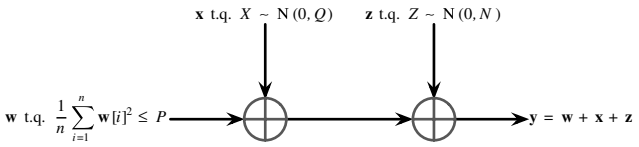


Figure 2. Canal de communication simplifié du tatouage.

2.2. Codes basés sur la turbo TCQ

La TCQ (*Trellis-Coded Quantization*) [5] est une technique de quantification utilisant un ensemble de quantificateurs plus simples (ici des quantificateurs scalaires, comme pour le SCS), organisés au sein d'un treillis. Ce treillis est défini par une fonction de transition :

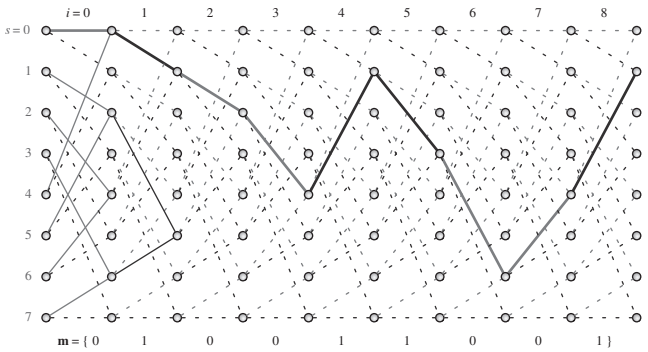


Figure 3. Exemple de treillis à 8 états utilisé pour l'encodage du message $\mathbf{m} = \{010011001\}$.

$$\begin{array}{rcl} \mathcal{S} \times \{0,1\} & \longrightarrow & \mathcal{S} \\ t : (s_i, \mathbf{m}[i]) & \longmapsto & s_{i+1}, \end{array} \tag{4}$$

avec $\mathcal{S} = \{0,1,\ldots,2^r - 1\}$ ensemble des états possibles. Les sous-dictionnaires sont définis en fonction des états du treillis par

$$\mathcal{U}_\mathbf{m}[i] = \left\{ k\Delta + \mathbf{d}[i] + \frac{\Delta s_i}{2^r} + \frac{\Delta \mathbf{m}[i]}{2},\ k \in \mathbb{Z} \right\} \tag{5}$$

et le mot de code le plus proche $\mathbf{u}^\star \in \mathcal{U}_\mathbf{m}$ de $\mathbf{x}$ est calculé par un algorithme de Viterbi avec un *a priori* fort, afin de s'assurer que le mot de code trouvé appartient bien à $\mathcal{U}_\mathbf{m}$ (équation 3). L'algorithme de Viterbi est une application de la programmation dynamique qui permet d'élaguer le treillis à chaque étape et de retrouver le chemin le plus probable *a posteriori* (c'est-à-dire la suite des états s_i) en une complexité $\mathcal{O}(n)$.

La figure 3 illustre cette technique pour l'encodage d'un message $\mathbf{m} = \{010011001\}$. Nous introduisons au début de l'algorithme un *a priori* pour supprimer les branches du treillis qui ne correspondent pas aux bits du message à encoder (sur la figure 3, seules les branches correspondant au bit 0 sont conservées). L'algorithme de Viterbi va ensuite élaguer l'arbre pour chaque étape et chaque état en fonction de la distance entre l'échantillon hôte et les pas de quantification définis par l'équation 5 (pour reprendre l'exemple de la figure 3, à $i = 2$, la transition allant de l'état 5 vers l'état 2 est supprimée, à cause du chemin en gras plus avantageux). Le décodage utilise aussi l'algorithme de Viterbi, mais sans *a priori* forcé sur les bits du message. Le chemin renvoyé le plus probable correspond à la suite de bits du message décodé.

La turbo TCQ [7] est une amélioration de la TCQ reprenant les approches itératives que l'on trouve habituellement appliquées dans les codes correcteurs (principe turbo). Le quantificateur est composé de deux treillis parallèles, comme montré sur la figure 4. Un premier treillis prend en entrée le signal à quantifier, tandis que le second travaille sur une version entrelacée. Les métriques *a posteriori* issues du premier quantificateur servent de métriques *a priori* au quantificateur parallèle. Le processus est itéré jusqu'à ce que les sorties convergent. Par rapport à une simple TCQ, l'utilisation de deux quantificateurs associés à un entrelaceur répartit les mots de code plus uniformément dans l'espace multi-dimensionnel des signaux. La distance entre les mots de code est plus uniforme, ce qui réduit la distance minimale du code, et augmente sa performance.

Nous utilisons cette technique pour construire un code adapté aux canaux avec information adjacente. L'encodage consiste à quantifier $\mathbf{x}$ en forçant les transitions des treillis pour qu'elles correspondent au message à encoder. De cette façon, nous nous assurons que le mot de code obtenu $\mathbf{u}^\star$ appartient bien à $\mathcal{U}_\mathbf{m}$. Le décodage reprend le même principe, mais en laissant toutes les transitions libres. Le chemin choisi par le décodeur forme le message décodé. Dans les mises en pratique de la suite de cet article, nous prenons un treillis à 2^9 états.

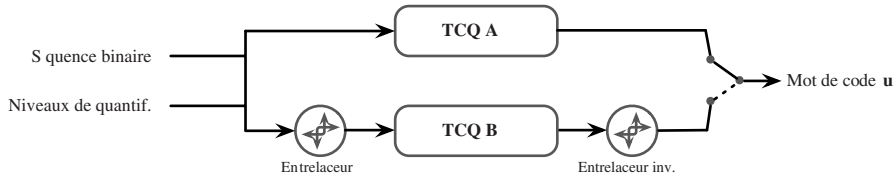


Figure 4. Principes de la turbo TCQ : deux TCQ fonctionnent en parallèle sur des versions entrelacées d'un même signal.

2.3. Émission informée et simulations

Comme déjà vu dans la section 1, le schéma de Costa comprend en plus d'un dictionnaire adapté une technique d'émission informée. L'information adjacente est dirigée vers le mot de code, par l'introduction d'un paramètre α :

$$\mathbf{w} = \alpha \left(\mathbf{u}^{\star} - \mathbf{x}\right). \tag{6}$$

Nous reprenons $\alpha = P/(P+N)$, comme défini par Costa. Les expériences montrent que cette formule donne les meilleurs résultats. La figure 5 présente les performances du code turbo TCQ et le gain obtenu par rapport à des techniques plus classiques : le SCS, la technique QIM (qui correspond au SCS sans émission informée, c'est-à-dire avec $\alpha = 1$) et un code TCQ. Un signal hôte est marqué en utilisant les techniques précitées, puis un bruit gaussien est ajouté. L'abscisse présente le rapport marque-à-bruit (WNR[3]) et l'ordonnée le taux d'erreur binaire. Avec un rendement de 1/1 (un bit transmis par échantillon de $\mathbf{x}$) et pour nous assurer un taux d'erreur inférieur à 10^{-5}, nous avons un gain de 5,5 dB par rapport au SCS. L'utilisation du principe turbo fait gagner 3,5 dB. Nous remarquons également le décrochage très rapide du code turbo TCQ, qui le rend totalement inadapté à des WNR inférieurs à 8 dB.

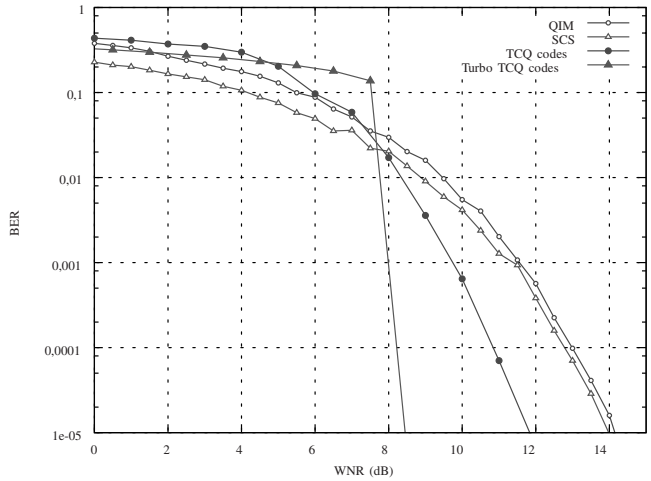


Figure 5. Taux d'erreur en fonction du rapport WNR sur canal gaussien avec information adjacente, avec un code de rendement 1/1 (un bit utile par échantillon de signal hôte), pour des treillis à 2^9 états et 500 étapes.

3. *Watermark-to-Noise Ratio*.

Initialement, la turbo TCQ est été imaginée pour le codage de source. Or, comme discuté par les auteurs de cette technique [7], le processus itératif ne converge pas toujours dans ce cas : le signal d'entrée peut se trouver à équidistance de plusieurs mots de code, et le processus itératif oscille alors entre ceux-ci. Dans le cas du codage canal, comme dans l'application que nous faisons de la TTCQ, la phase de marquage dirige le signal hôte vers un mot de code (équation 6), si bien que le signal d'entrée sur lequel travaille la TTCQ lors de l'extraction de la marque est statistiquement centré sur un mot de code. Ceci est d'autant plus vrai que la dimension des signaux est importante. En pratique, nous n'avons pas constaté de problème de convergence lors des expériences illustrée par la figure 5 ni lors de la mise en application décrite dans la section suivante.

3. Application au tatouage robuste d'images

Dans cette section, nous appliquons le code correcteur basé sur la turbo TCQ pour construire un schéma de tatouage d'images robuste. Pour ses résultats en termes de capacité et de robustesse, l'article de de Miller *et al.* [8] fait référence dans le domaine du tatouage d'images. Afin de comparer cette approche avec la notre, nous reprenons une bonne partie de leurs choix : transformée fréquentielle, modèle perceptuel, attaques, etc.

Nous nous limitons aux images en niveaux de gris. L'image à marquer est passée dans le domaine DCT en blocs 8×8. Pour chacun des blocs, douze coefficients sont sélectionnés : le coefficient AC est évité, et nous gardons les douze coefficients suivants dans l'ordre zigzag. Ces échantillons sont regroupés dans un vecteur $\mathbf{s}$ et sont projetés sur un vecteur pseudo-aléatoire[4] $\mathbf{g} \in \{-1;+1\}^{12n}$ afin de former une valeur unique :

$$\mathbf{x}[i] = \sum_{j=0}^{11} \mathbf{s}[12i+j] \times \mathbf{g}[12i+j]. \tag{7}$$

4. Ce vecteur – commun à la phase d'insertion et d'extraction – fait partie de la clef secrète du schéma.

L'ensemble des valeurs obtenues par projection (une valeur par bloc DCT) forme le signal **x**. Cette technique de projection nous permet de moduler le rendement du code. Comme vu dans la section 2.3, le rendement d'origine du code turbo TCQ est de 1/1, et nous le passons ainsi à 1/12. Nous calculons ensuite $\mathbf{u}^\star$ en utilisant la turbo TCQ dont les transitions seront forcées par les bits du message à insérer, puis calculons la marque **w** grâce à l'équation 6.

Pour la projection inverse, il est possible de prendre en compte une mesure perceptuelle. La marque **w** – initialement calculée dans le domaine projeté – est distribuée sur les douze coefficients à marquer en fonction de leur importance psychovisuelle. Nous utilisons pour cela un *waterfilling*, décrit par l'algorithme 1 : à partir du signal **x**, nous nous approchons du mot de code – sans le dépasser – en modifiant en priorité les éléments les moins significatifs perceptuellement. Le paramètre ε utilisé à la ligne 14 joue sur la précision du remplissage. Nous l'avons fixé à 10^{-3} pour nos expériences. La pondération perceptuelle choisie est le modèle de Watson, également utilisé dans l'article que nous prenons comme référence [8]. Pour optimiser son utilisation dans l'algorithme 1, nous utilisons l'exposant 4/3 (lignes 6 et 25), issu d'une minimisation lagrangienne de la distance de Watson [9].

L'extraction de la marque reprend les mêmes étapes : transformation de l'image considérée dans le domaine DCT, application de la projection sur le vecteur **g** puis décodage de la suite de valeurs obtenues.

D'un point de vue pratique, nous insistons sur la faible complexité de notre schéma. L'insertion de la marque dans une image de 368 × 240 pixels – processus qui comprend la lecture du fichier, la DCT, la calcul du modèle perceptuel, la projection, la TTCQ, la DCT inverse et l'écriture du fichier résultat – prend en moyenne deux dixièmes de seconde sur une machine récente. L'extraction est deux fois plus rapide. La vitesse de traitement est un point faible de la méthode prise en référence [8], l'insertion informée utilisant un algorithme de Monte Carlo très coûteux[5].

Algorithme 1. Algorithme de waterfiling *pour prendre en compte une pondération perceptuelle lors de la projection inverse.*

Entrées
– **s** : m coefficients DCT à marquer,
– **p** : pondérations perceptuelles correspondantes,
– **g** : vecteur de projection de dimension m,
– **x** : coefficients projetés selon l'équation 7,
– **u** : mot de code **u** de dimension $n = m/12$.
Sorties
– **t** : coefficients DCT tatoués.

1 **début**
2
3 *// Initialisations*
4 **pour** $j = 0\,; j < n$ **faire**
5 $\quad \text{complet}[j] = \mathbf{w}[j] = 0\,;$
6 $\quad \mathbf{b}[j] = \sum_{i=0}^{11} \mathbf{p}[12j+i]^{4/3}\,;$
7 $t = \alpha^2 \sum_{i=0}^{n-1} (\mathbf{u}[i] - \mathbf{x}[i])^2\,;$
8
9 *// Algorithme de remplissage pour calculer la marque*
10 **répéter**
11 $\quad d = 0\,;$
12 $\quad$ **pour** $j = 0\,; j < n$ **faire**
13 $\quad\quad$ **si** $\text{complet}[j] = 0$ **alors**
14 $\quad\quad\quad \mathbf{w}[j] += \text{sign}(\mathbf{u}[j] - \mathbf{x}[j]) \times \epsilon \mathbf{b}[j]\,;$
15 $\quad\quad$ **si** $\mathbf{w}[j]^2 > (\mathbf{x}[j] - \mathbf{u}[j])^2\,;$
16 $\quad\quad\quad \text{complet}[j] = 1\,;$
17 $\quad\quad\quad \mathbf{w}[j] = \mathbf{u}[j] - \mathbf{x}[j]\,;$
18 $\quad\quad d += \mathbf{w}[j]^2\,;$
19 **jusqu'à** $d \geq t$ **;**
20
21 *// Projection inverse pour modifier les coefficients DCT*
22 **pour** $j = 0\,; j < n$ **faire**
23 $\quad c = \sum_{i=0}^{11} \mathbf{p}[12j+i]^{4/3}\,;$
24 $\quad$ **pour** $i = 0\,; i < 12$ **faire**
25 $\quad\quad \mathbf{t}[12j+i] = \mathbf{s}[12j+i] + \mathbf{g}[12j+i]\mathbf{w}[j]\dfrac{\mathbf{p}[12j+i]^{4/3}}{c}\,;$
26 **fin**

4. Résultats

Notre schéma de tatouage est testé sur un ensemble de 1200 images en niveaux de gris de 368 × 240 pixels. Nous insérons une marque de 1380 bits utiles par image (soit un bit par bloc DCT). La distorsion d'insertion (paramétrée par le pas de quantification Δ utilisé dans la turbo TCQ) est fixée telle que la distance de Watson moyenne entre image d'origine et image marquée soit proche de celle de l'article de référence. Sans prise en compte du modèle de Watson à l'insertion, nous obtenons une distance moyenne de 96,84 (contre 101,52 pour la référence [8]), et 31,04 avec prise en compte (contre 31,6). Les quatre figures suivantes présentent les taux d'erreur par message après attaque des images marquées. L'abscisse mesure la force de l'attaque et l'ordonnée le taux d'erreur par message. Nous avons placé un repère à 0,2, taux d'erreur sous lequel le schéma de tatouage est considéré comme *robuste* selon l'article que nous prenons comme référence.

La figure 6 montre la robustesse de notre schéma face à l'ajout de bruit gaussien. Un bruit suivant une loi Normale est ajouté aux pixels de l'image marquée. L'écart type du bruit varie entre

5. La technique d'insertion a été remplacée ensuite [10].

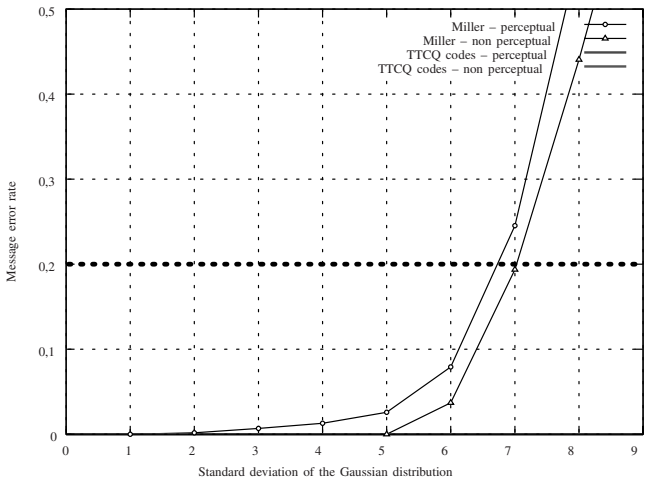


Figure 6. Robustesse contre l'ajout de bruit gaussien : aucune erreur pour la TTCQ.

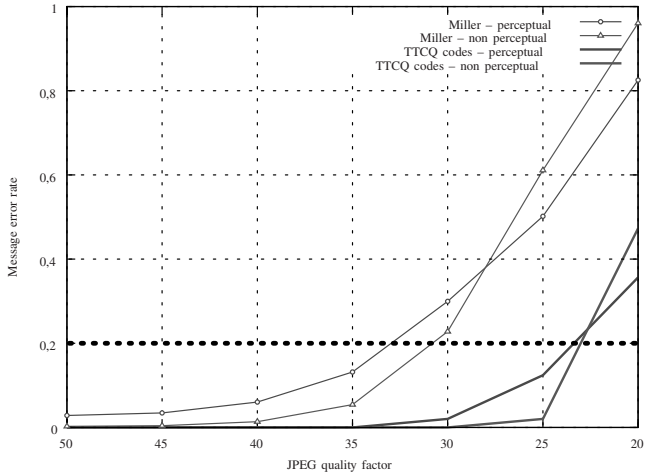


Figure 7. Robustesse contre la compression JPEG : le schéma est considéré robuste jusqu'à un facteur de qualité de 23.

0 et 9. Sur les 1200 images, nous n'avons noté aucune erreur pour un bruit d'écart type jusqu'à 9. Le schéma de référence n'est plus considéré comme robuste (plus d'un message extrait sur cinq présente au moins une erreur) à partir d'un écart type de 7.

La figure 7 présente les résultats face à la compression JPEG. Les images attaquées sont compressées (ce qui correspond en réalité à une quantification des coefficients DCT), avec des facteurs de qualité allant de 50 (acceptable) à 20 (image très dégradée quasi-inutilisable). Notre schéma reste robuste jusqu'à un facteur de qualité de 23 (celui de référence résiste jusqu'à 30 dans le cas le plus favorable).

De part la construction même du dictionnaire, la TTCQ – comme toutes les techniques basées sur la quantification de l'information adjacente – n'est pas intrinsèquement robuste aux changements d'échelle, car le pas de quantification Δ ne correspond plus à la marque présente dans le signal attaqué. Néanmoins, les expériences montrent une résistance non négligeable à ce genre d'attaques. Ainsi, notre schéma de tatouage d'image reste robuste à des changements d'échelle (chaque pixel est multiplié par une constante) compris entre 0,8 et 1,2 (figure 8). Notons que malgré l'invariabilité intrinsèque du schéma de référence aux changements d'échelle, la saturation des pixels limite sa résistance au facteur 1,3.

Face à un filtre de flou gaussien (filtre passe-bas), le schéma TTCQ est robuste jusqu'à un filtre de largeur 0,5 (figure 9). Ce filtre agit comme un facteur d'échelle sur les coefficients DCT utilisés pour le marquage, et nous retrouvons donc les mêmes faiblesses que pour l'attaque précédente. Les deux derniers résultats sont certes moins performants que ceux du schéma que nous avons pris comme référence, mais ils laissent entrevoir d'importantes améliorations par l'utilisation de techniques de synchronisation.

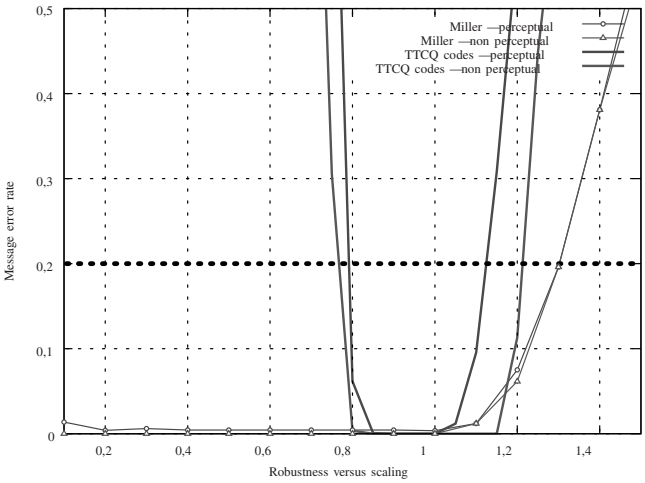


Figure 8. Robustesse contre les facteurs d'échelle : la version non perceptuelle résiste à un facteur compris entre 0,8 et 1,2.

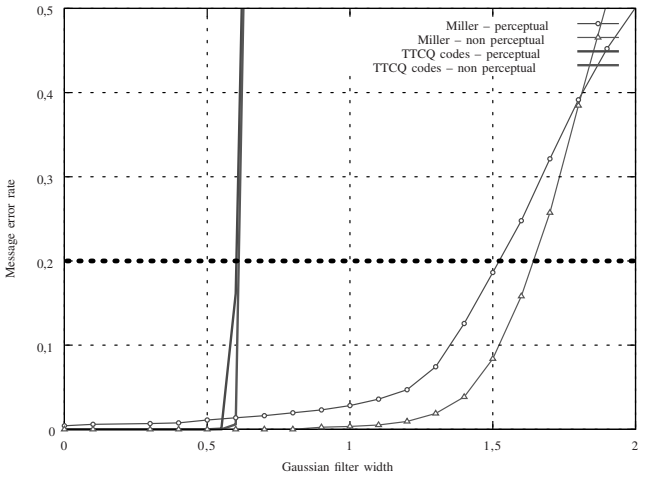


Figure 9. Robustesse contre le flou par filtre gaussien : résistance jusqu'à une largeur de filtre de 0,5.

5. Discussion sur l'utilisation de quantificateurs vectoriels

Les techniques présentées précédemment s'appuient sur des quantificateurs scalaires, si bien qu'un bit est inséré par échantillon. Pour réduire ce rendement et augmenter la robustesse du schéma, nous utilisons une projection. D'un point de vue codage, la projection est sous-optimale et nous éloigne de la limite théorique de capacité. Il serait plus judicieux de construire directement un code de rendement voulu.

Nous avons modifié la turbo TCQ pour qu'elle utilise des quantificateurs à deux dimensions et l'avons comparée à la TTCQ scalaire précédée par une projection de rendement 1/2. La version vectorielle fait mieux de 0,31 dB (le taux d'erreur binaire de 10^{-5} est atteint dès un rapport WNR de 5,19 dB contre 5,5 dB). Néanmoins, utiliser des quantificateurs vectoriels impose quelques précautions.

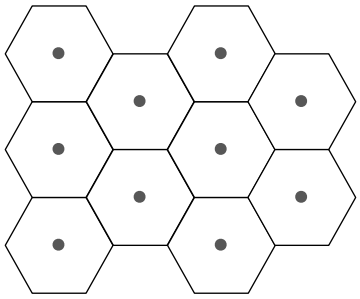
La figure 10(a) montre le pavage optimal d'un espace en deux dimensions, obtenu par la lattice A_2 [11]. Dans ce cas, la distance entre un mot de code et ses six voisins les plus proches est constante. Naïvement, on pourrait espérer de bons résultats en utilisant ce type de quantificateur pour le code TTCQ étendu à deux dimensions : chaque quantificateur implémenterait alors une lattice A_2 et serait décalé par rapport aux autres. Or, il n'en est rien. Comme on le constate sur la figure 10(b), l'union de ces deux lattices donne une lattice Z_2, ce qui correspond au cas scalaire, qui est bien sûr sous-optimal en deux dimensions.

La bonne utilisation est donné par la figure 10(c). C'est l'union des quantificateurs utilisés dans le treillis qui doit implémenter la lattice visée, et non chaque quantificateur. On remarque dans ce cas que les quantificateurs implémentent de simples lattices Z_2 et que le décalage entre les deux lattices est judicieusement choisi pour former au final A_2. La distance entre deux mots de codes issus de quantificateurs différents est alors maximale (et constante).

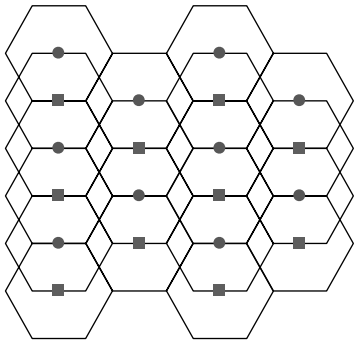
Conclusion

Cet article a présenté une mise en pratique simple des principes décrits dans l'article de Costa sur les canaux avec information adjacente. Nous avons montré que l'utilisation d'un code correcteur adapté au canal du tatouage et d'une technique d'insertion informée donne un schéma de tatouage avec d'excellentes performances, à la fois en termes de capacité (plus de 1300 bits dans des imagettes de 368×240 pixels) et de robustesse.

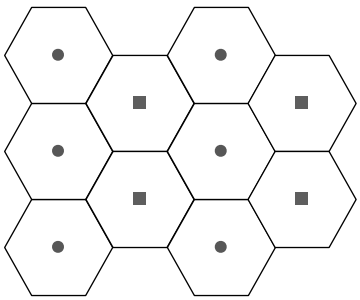
De plus, l'utilisation d'une fonction de projection avant le code correcteur permet à la fois de moduler le rendement global du code pour adapter le compromis robustesse/capacité, et de



(a) Pavage obtenu par le meilleur quantificateur.



(b) R partition sous-optimale des mots code partir de deux quantificateurs optimaux.



(c) R partition id ale des mots de code.

Figure 10. Pavages en deux dimensions obtenus par l'assemblage de quantificateurs.

prendre en compte des pondérations externes pour adapter la marque au document hôte (poids perceptuels comme ici, mais aussi paramètres issus de la théorie des jeux [12], marque de type PSC [13], etc.).

Néanmoins, nous avons vu dans la section 4 que l'utilisation de quantificateurs pour le tatouage montre ses limites face aux changements d'échelle. Pour construire un schéma résistant à une plus large palette d'attaques (filtrages, changements de contraste ou de luminosité, etc.), il nous faut inclure en plus une technique pour synchroniser le pas de quantification Δ, avant et après attaque. La littérature propose de nombreuses techniques pour synchroniser les quantificateurs lors de la phase d'extraction. Les approches par normalisation en fonction des caractéristiques du signal nous semblent prometteuses [14] : le pas de quantification Δ est non plus fixe, mais calculé en fonction du signal à marquer, si bien que les altérations sur le signal attaqué sont répercutées sur la valeur de Δ utilisée pour l'extraction de la marque. De plus, le schéma proposé ici offre une certaine résistance face aux changements d'échelle, ce qui offrirait une souplesse intéressante quant à l'utilisation de ces techniques de synchronisation (le système peut tolérer une marge d'erreur dans la synchronisation).

Gaëtan **Le Guelvouit**

Diplômé du département Informatique de l'INSA de Rennes en 2000, Gaëtan Le Guelvouit est aujourd'hui ingénieur de recherche à Orange Labs (France Telecom R&D). Ses travaux de recherches ont débuté par un doctorat à l'IRISA, puis se sont poursuivis au Laboratoire des Signaux et Systèmes de Supélec Paris. Depuis 2005, ses thématiques de prédilections – tatouage numérique, traitement d'images et communications numériques – se sont élargies à la protection des contenus et à la sécurité logicielle.

Références

[1] M. H. M. COSTA, « Writing on dirty paper », *IEEE. Trans. on Information Theory*, Vol. 29, No. 3, 1983, p. 439-441.

[2] T. M. COVER, J. A. THOMAS, « Elements of information theory », Wiley-Interscience, 1991.

[3] I. J. COX, M. L. MILLER, A. L. MCKELLIPS, « Watermarking as communications with side information », *IEEE J. Selected Areas Communi.*, Vol. 16, No. 4, 1998, p. 587-593.

[4] P. MOULIN, R. KOETTER, « Data-hiding codes », *Proceedings IEEE*, Vol. 93, No. 12, 2005, p. 2083-2127.

[5] M. W. MARCELLIN, T. R. FISHER, « Trellis-coded quantization of memoryless and Gauss-Markov sources », *IEEE Trans. on Communications*, Vol. 38, 1990, p. 82-93.

[6] J. J. EGGERS, R. BAÜML, R. TZCHOPPE, B. GIROD, « Scalar Costa scheme for information embedding », *IEEE Trans. on Signal Processing*, 2003.

[7] V. CHAPPELIER, C. GUILLEMOT, S. MARINKOVIC, « Turbo trellis-coded quantization », *Proc. of Int. Symp. on Turbo Codes*, 2003.

[8] M. L. MILLER, G. J. DOËRR, I. J. COX. « Applying informed coding and informed embedding to design a robust, high capacity watermark », *IEEE Trans. on Image Processing*, Vol. 3, No. 6, 2004, p. 792-807.

[9] I. J. COX, M. L. MILLER, J. A. BLOOM, « Digital watermarking », Morgan Kaufmann Publishers, 2002.

[10] L. LIN, G. J. DOËRR, I. J. COX, M. L. MILLER, « An efficient algorithm for informed embedding of dirty paper trellis codes for watermarking », *Proc. of Int. Conf. on Image Processing*, Vol. 1, 2005, p. 697-700.

[11] J. MARTINET, « Les réseaux parfaits des espaces euclidiens », Masson, 1996.

[12] S. PATEUX, G. LE GUELVOUIT, « Practical watermarking scheme based on wide spread spectrum and game theory », *Signal Processing : Image Communication*, No. 18, 2003, p. 283-296.

[13] J. K. SU, B. GIROD, « Power-spectrum condition for energy-efficient watermarking », *Proc. of Int. Conf. on Image Processing*, 1999, p. 301-305.

[14] Q. LI, I. J. COX, « Improved spread transform dither modulation using a perceptual model : robustness to amplitude scaling and JPEG compression », *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, 2007.