Vers un tatouage d'images mou

Toward a soft image watermarking

par François CAYRE*, Franck DAVOINE**

* Laboratoire TELE, Université catholique de Louvain, Place du Levant 2, 1348 Louvain-La-Neuve, Belgique (cayre@tele.ucl.ac.be).

** Laboratoire HeuDiaSyC, Université de Technologie de Compiègne, BP 20529, 60205 Compiègne, France (fdavoine@utc.fr).

résumé et mots clés

Cet article introduit une nouvelle méthode de tatouage pour la protection d'images fixes. La méthode permet de cacher une signature dans une image, sous la forme de w paquets de r bits. Le schéma de tatouage est additif, et la marque elle-même est calculée par addition de produits de couples de fonctions orthogonales. Nous montrerons comment le choix des fonctions orthogonales peut être fait, de façon à rendre le tatouage robuste face à différents types d'attaques. Puis, nous donnerons quelques résultats permettant d'apprécier la robustesse du tatouage, et discuterons des améliorations possibles pouvant être apportées à une telle méthode.

Tatouage, protection du copyright, ondelettes, attaques géométriques, fonctions orthogonales.

abstract and key words

This paper introduces a new approach of watermarking for copyright protection. The goal of the method is to hide signatures composed of w segments of r bits in digital images. The framework itself is founded upon a wavelet transformed domain, and an additive embedding rule using products of orthogonal basis functions. We will show how the choice of different kinds of orthogonal functions allows to improve the robustness of the watermarking scheme to signal processing or geometric attacks.

Watermarking, copyright protection, wavelets, geometric attacks, orthogonal basis functions.

1. introduction

Le tatouage d'images est une technique encore jeune, qui a pour but de dissimuler au sein même de l'information visuelle d'une image numérique, une information cachée identifiant son propriétaire, ou son contenu. L'information ainsi ajoutée ne doit pas être perçue par l'œil humain et peut être codée de deux manières, suivant que l'on veuille répondre à la question « *cette information est-elle présente dans l'image ?* », ou bien « *quelle information est cachée dans cette image ?* ». La première question appelle une réponse binaire de type oui ou non, tandis que la seconde implique de pouvoir relire l'information entière dans l'image. Pour se référer à la première problématique, nous parlerons de tatouage de *signature*, alors que pour la seconde nous parlerons de tatouage de *message*. D'une manière générale, l'information ajoutée doit être indélébile, indétectable et/ou illisible par une personne non autorisée et par quelque moyen que ce

Les auteurs remercient le RNRT pour avoir soutenu ce travail réalisé dans le cadre du projet Aquamars, au sein du laboratoire HeuDiaSyC.

soit. Ces conditions ne peuvent bien sûr pas être rigoureusement toutes vérifiées en même temps, et les schémas de tatouage proposés doivent tenir compte de l'application visée, et de la nature des données à traiter. L'information sera cachée dans des régions de l'image aptes à la porter, sémantiquement significatives, en tenant compte de leur capacité et de leur aptitude à la rendre imperceptible [6]. Le principe de l'algorithme de tatouage peut en outre être connu ou secret, même si une méthode de tatouage sûre réclamerait que son fonctionnement puisse être rendu public (principe de Kerckhoff). Le tatouage est souvent symétrique dans le sens où les paramètres utilisés pour incruster (on parle de clé privée) puis extraire l'information sont identiques. Des travaux récents portent sur le tatouage asymétrique [9,11], qui utilise des paramètres différents pour l'insertion et l'extraction de l'information, et qui permet donc de la relire à l'aide d'une seule clé publique. Les méthodes de tatouage proposées dans la littérature [3,4] reposent en grande partie sur la théorie de l'information et des communications numériques (capacité, codes orthogonaux, codes correcteurs, multiporteuses, partage de canaux), le traitement et l'analyse des images (représentations multirésolutions orthogonales ou redondantes, filtrage et estimation de paramètres, segmentation, détection), les statistiques (décision, tests d'hypothèses, mesures de confiance, fusion, reconnaissance), et la cryptographie (gestion de clés publiques et privées). Leur conception implique de prendre en compte les mécanismes psychologiques et physiologiques qui permettent de percevoir les dégradations des images, des couleurs, mais également les différents types d'attaques pouvant altérer l'information cachée.

Notre objectif ici est de proposer une nouvelle méthode de tatouage aveugle d'images fixes, basée sur une transformation en ondelettes discrète, et utilisant un ensemble de produits de fonctions orthogonales. La méthode permet de dissimuler w paquets de r bits dans une image. La paramétrisation des fonctions permet d'adapter la robustesse du tatoueur à deux principaux types d'attaques : les attaques par traitement du signal (plus particulièrement le filtrage, et la compression JPEG), et les déformations géométriques locales du contenu de l'image. Nous décrirons le principe de la méthode et testerons sa robustesse face aux attaques visées, puis discuterons des améliorations possibles. Cette méthode est une extension d'une autre déjà existante, permettant de cacher une ou plusieurs signatures, vers une méthode permettant de cacher un message à l'aide d'une clé secrète.

2. cacher une signature

2.1. principe d'une méthode de référence

Nous exposons ici le principe et les résultats d'une méthode de référence, permettant de cacher une signature à l'intérieur de l'image. Cette méthode est exposée plus avant par Barni *et al.* [1] (1999). On se propose ici de cacher une signature dans l'image, puis d'être ensuite en mesure de dire si cette signature est présente ou non dans l'image. À la relecture, on devra donc disposer de l'image tatouée, ainsi que d'une signature dont on souhaite vérifier la présence ou non dans l'image tatouée. Comme cette méthode appelle une réponse binaire de la part du système de tatouage, on souhaite pouvoir disposer d'une mesure de la confiance dans la réponse. Cette confiance est exprimée sous forme d'une probabilité d'erreur P_f fixée à l'avance. Afin de bénéficier d'une telle mesure de confiance, l'information à cacher doit vérifier des propriétés statistiques que l'on pourra estimer lors de la relecture de la signature. Pour cela, il est naturel de matérialiser l'information à cacher par une séquence pseudo-aléatoire dont la moyenne et la variance sont connues. C'est le germe K du générateur de nombres pseudo-aléatoires qui servira d'information à cacher. En effet, on ne pourra que répondre à la question « *la séquence de germe K est-elle présente dans l'image ?* ».

Nous rappelons comment cacher cette séquence : l'espace de tatouage associé à cette méthode est composé des trois sous-bandes de plus fins détails de la transformée en ondelettes de l'image. Le tatouage proprement dit d'une image de taille $(2m \times 2n)$ s'effectue à l'aide d'une séquence x pseudo-aléatoire de germe K, de moyenne nulle, de variance unité, et de longueur $3 \times m \times n$. Les coordonnées sont données relativement à une sous-bande et I_s représente une des trois sous-bandes (indexées par s).

Afin de tenir compte du contenu de l'image, on adapte localement l'intensité du tatouage à l'aide d'un masque psycho-visuel ω tiré de travaux sur la compression et la quantification [7]. Pour en contrôler globalement la force, on définit un paramètre α. On

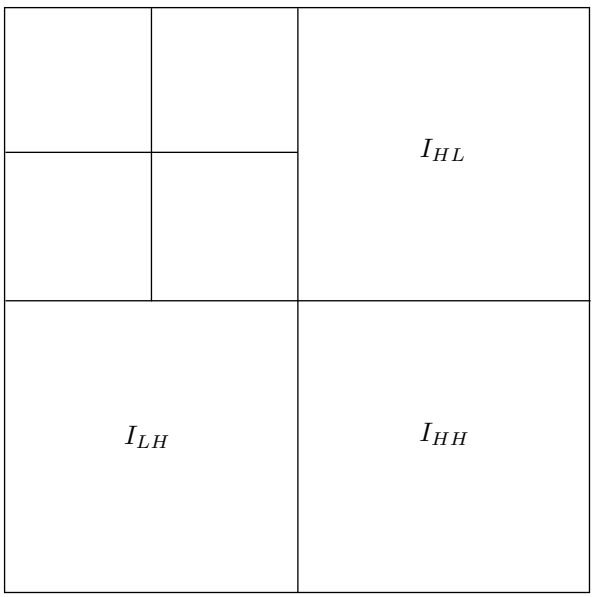

Figure 1. – Le domaine de tatouage : les trois sous-bandes de plus fins détails.

choisit de voir la marque comme un vecteur 2D formé par le produit du masque psycho-visuel ω et la séquence x initialisée avec le germe K. L'insertion de la marque est donc de type additif. Nous avons donc l'expression de la marque :

$$W_s(i,j) = \omega_s(i,j) \times x_s(i,j) \tag{1}$$

Qui donne le schéma d'insertion suivant :

$$I_s^W(i,j) = I_s(i,j) + \alpha \times |I_s(i,j)| \times W_s(i,j) \tag{2}$$

Lors de la détection, on calcule la corrélation entre la séquence obtenue à partir de la signature à tester K' et l'image tatouée. La probabilité d'erreur permet de fixer un seuil pour la valeur de cette corrélation, au-delà de laquelle on déclarera la signature présente dans l'image. Le seuil T est calculé en fonction d'un estimateur σ et de la probabilité d'erreur souhaitée :

$$P_f \leqslant \frac{1}{2} erfc\Big(\frac{T}{\sqrt{2\sigma^2}}\Big), \tag{3}$$

qui conduit à une expression du seuil fonction uniquement de l'image tatouée (c'est le rôle de l'estimateur, que nous ne détaillons pas, *cf.* [1]) :

$$T \leqslant 3.97\sqrt{2\sigma^2}, \tag{4}$$

pour une probabilité d'erreur fixée à 10^{-8}. On calcule donc la corrélation ρ entre l'image tatouée et la séquence x' associée au germe K' :

$$\rho = \sum_s \sum_{i=1}^{n} \sum_{j=1}^{m} I_s^W(i,j) x'_s(i,j). \tag{5}$$

L'étape de décision est constituée uniquement d'une comparaison entre ρ et T. Si ρ est supérieure à T, on déclare alors que la signature $K' = K$ est effectivement présente dans l'image.

2.2. discussion

La méthode décrite brièvement ici permet de cacher une signature (le système de tatouage fournit une réponse binaire), nous nous attacherons dans la suite de cet article à pouvoir cacher un message. En outre, cette méthode repose sur la corrélation, comme de nombreuses autres, même si d'autres pistes permettent de se passer de l'opérateur de corrélation et de le remplacer par une projection dans un réseau de neurones par exemple [9]. La simple idée de pouvoir obtenir une réponse lors de la corrélation implique d'avoir conservé la synchronisation pixel à pixel entre l'image tatouée et la marque à tester. Pourtant, cette synchronisation peut être altérée imperceptiblement par une attaque consistant à modifier localement et aléatoirement l'image tatouée [12], l'image attaquée est alors visuellement identique à la version tatouée brute, mais si la marque est encore présente, il est par contre devenu impossible d'obtenir une quelconque réponse à la corrélation du fait de la désynchronisation locale des pixels. Cette attaque est essentiellement de type *géométrique* car la mise en défaut du système de relecture consiste à établir une différence pixel à pixel entre l'image tatouée attaquée et la marque associée au germe K'. Nous choisissons de regrouper dans cette catégorie d'attaques le fenêtrage, l'enlèvement ou la permutation de lignes/colonnes, etc...

Outre les attaques susmentionnées, ils sont des traitements que l'on fait subir de manière usuelle à une image, comme la compression, le filtrage, *etc*. Ces manipulations ne sont pas des attaques au sens où un pirate l'entendrait, mais elles altèrent néanmoins la marque. On souhaite pouvoir résister à ces opérations usuelles. Nous regroupons ces manipulations sous le terme générique d'attaques de type *filtrage* ; même si toutes ne procèdent pas d'une volonté délibérée de nuire, elles sont par contre toutes susceptibles d'attaquer la marque. Plus généralement, nous classons dans cette catégorie les attaques entraînant une perte de qualité lors de la corrélation non due à une désynchronisation.

Il convient maintenant d'isoler les parties du système de tatouage entrant en jeu dans la résistance à tel ou tel type d'attaque. Le fait de résister à des attaques de type *filtrage* relève évidemment de la manière dont la marque a été insérée dans l'image. Le choix de l'espace de tatouage est donc primordial, ainsi que la manière dont les coefficients de l'image sont modifiés dans cet espace. Afin d'être plus clair par la suite, nous regroupons sous le terme de *modus operandi* le choix d'un espace de tatouage et d'une modification des coefficients. À l'évidence, le *modus operandi* de la méthode précédemment décrite est satisfaisant, nous avons pu retrouver la marque après une compression JPEG avec un facteur de qualité de 20. Le *modus operandi* n'est pas la partie du système nécessitant une remise en cause. Nous le conserverons donc dans notre extension, notre schéma sera toujours de type additif et agira toujours sur les trois sous-bandes utilisées jusqu'à présent, ce qui équivaut à utiliser les $3/4$ de la surface de l'image.

À l'inverse, la mise en échec du système de relecture lors d'attaques *géométriques* repose uniquement sur la structure même de la marque. En effet, la marque que nous avons générée nécessite de retrouver au pixel près sa position relative par rapport à l'image tatouée. Il existe des méthodes de resynchronisation entre la marque et l'image tatouée pour de petites déformations locales [10], mais ces méthodes nécessitent de longs calculs, et surtout une version de l'image non attaquée *géométriquement* (une version tatouée et compressée peut faire l'affaire). Nous chercherons donc à dégager une autre structure pour la marque, permettant de s'affranchir de quelques-unes des attaques de type

géométrique. D'ores et déjà, il nous faut renoncer aux commodités induites par les séquences pseudo-aléatoires puisque rien ne nous garantit plus qu'elles constituent l'unique moyen de coder l'information à cacher.

3. une méthode générique de tatouage

La méthode de tatouage d'images que nous décrivons ici utilise le même *modus operandi* que précédemment. Mais les objectifs que nous nous assignons diffèrent : nous voulons à présent cacher un message dans l'image et non plus une simple signature. Cela veut dire que la sortie du module de relecture sera constituée de bits et non plus d'une assertion binaire. En outre, nous voulons rendre la lecture du message possible uniquement si l'on fournit la bonne clé, secrète dans notre cas. Des algorithmes de tatouage asymétriques ont été proposés et même étudiés ([9],[11]) mais leurs propriétés de relecture sont moins bonnes que dans le cas qui nous concerne. Pour la suite, nous donnons un aperçu de la méthode que nous proposons, et notamment la conversion des chaînes de bits (clé et message) en fonctions orthogonales. Ensuite, nous formaliserons ces idées de manière à dégager un schéma générique de tatouage dont nous donnerons deux instantiations pour illustrer notre propos (correspondant chacune à des ensembles de fonctions différents).

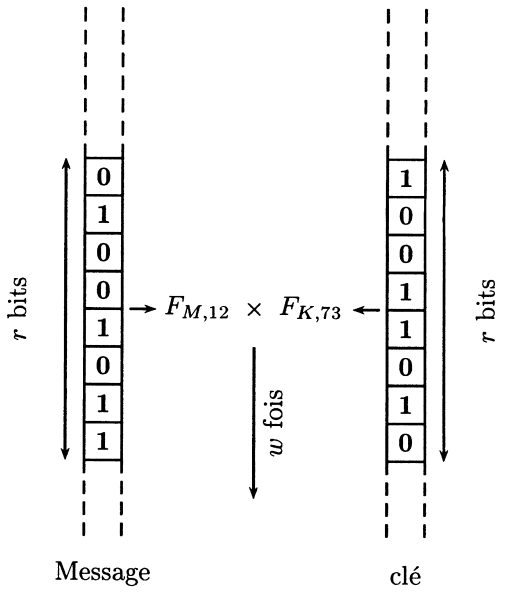

Figure 2. – Principe d'encodage pour r bits : on trouve la fonction associée à la configuration binaire de r bits pour la clé (ici la 73ème fonction de la famille F_K : $F_{K,73}$) et le message $(F_{M,12})$, et on effectue le produit de ces fonctions. Pour généraliser, on itère sur w paquets en sommant ces produits $(F_{M,p} \times F_{K,q})$

3.1. principe

Nous commençons par donner le schéma générique que nous allons utiliser (et qui permet éventuellement de retrouver le schéma initial), nous définissons pour cela des contraintes faibles sur la clé et le message. Introduisons maintenant les méthodes de tatouage d'images, puis de relecture des bits du message : le principe consiste non pas en une division spatiale de l'image (et cacher un bit par bloc [13]) mais plutôt en une projection du couple formé du message et de la clé sur des espaces orthogonaux de fonctions.

On considère pour cela à la fois la clé et le message comme une concaténation de motifs binaires de taille r. Sans perte de généralité et à des fins de présentation, on suppose dans la figure 2 que la clé et le message ont la même taille, et sont chacun formés de w paquets de r bits concaténés.

Chaque segment (de clé ou de message) peut donc être associé à une des 2^r configurations binaires possibles. On fait également correspondre à chacune de ces configurations binaires sur r bits une fonction (les conditions sur ces fonctions seront définies plus bas, notamment, deux ensembles de fonctions différents sont nécessaires pour la clé puis le message). Ainsi, nous avons une bijection entre une suite de r bits et une famille de 2^r fonctions [1], et ceci est vrai séparément pour la clé et le message. Afin d'assurer le secret, on multiplie les fonctions relatives à un segment de la clé et à un segment du message.

Pour la suite, nous considérerons le cas où les tailles de segments diffèrent pour la clé et le message. Toutefois, notre implémentation pose des tailles identiques pour les segments de clé et de message. Nous détaillons par la suite les schémas d'insertion et de relecture du message en formalisant les idées exposées brièvement en figure 2.

3.2. insertion du message

Soit une clé K dite *privée*, utilisée pour interdire la relecture d'un message M à toute personne ne connaissant pas cette clé. Les conditions suivantes sur K et M doivent être vérifiées :

$$M = \bigoplus_{i=1}^{w} M_i^r \ , \quad M_i^r \in \{0;1\}^r \ , \ \forall i \in [1;w] \tag{6}$$

$$K = \bigoplus_{i=1}^{w} K_i^l \ , \quad K_i^l \in \{0;1\}^l \ , \ \forall i \in [1;w] \tag{7}$$

$$\text{et } \forall (i,j) \in [1;w]^2 \quad K_i^l \neq K_j^l \tag{8}$$

où $\bigoplus$ est l'opérateur de concaténation. La clé et le message sont donc respectivement découpés en w paquets (segments binaires) de l et r bits. Notons ici que la construction des segments K_i^l

1. Cette mise en correspondance est classiquement utilisée dans les schémas de modulation M-aire, afin d'augmenter les débits de transmission.

(resp. M_i^r) pourrait se faire en choisissant aléatoirement les bits dans K (resp. M), pour augmenter la sécurité globale du schéma de tatouage, comme nous le verrons par la suite. Les segments de clé doivent également être tous différents (équation 8), de façon à éviter toute interférence. Cette contrainte impose une limitation sur l'espace des clés possibles ; chaque segment de clé devant être différent des autres, le nombre total de clés disponibles est donc :

$$N_K(w,l) = \prod_{i=0}^{w-1} 2^l - i \tag{9}$$

On voit par là qu'il est possible d'avoir des longueurs de clé et de message différentes, et ainsi augmenter la taille de l'espace des clés de manière artificielle en posant $l \geqslant r$, nous verrons malgré tout que l ne peut être arbitrairement grand. Soit une image originale I de taille $(2m \times 2n)$, et deux ensembles de fonctions de base :

$$F_{M,k} \ : \ [0;m] \times [0;n] \rightarrow \mathbb{R} \ , \quad k \in \mathbb{N} \tag{10}$$

$$F_{K,k} \ : \ [0;m] \times [0;n] \rightarrow \mathbb{R} \ , \quad k \in \mathbb{N} \tag{11}$$

avec les contraintes d'orthogonalité inter- et intra-ensembles suivantes :

$$\forall (\alpha,\beta) \in \mathbb{N}^2 \quad \langle F_{M,\alpha} \ , \ F_{M,\beta} \rangle = \delta_{\alpha,\beta} \tag{12}$$

$$\forall (\alpha,\beta) \in \mathbb{N}^2 \quad \langle F_{K,\alpha} \ , \ F_{K,\beta} \rangle = \delta_{\alpha,\beta} \tag{13}$$

$$\forall (\alpha,\beta) \in \mathbb{N}^2 \quad \langle F_{M,\alpha} \ , \ F_{K,\beta} \rangle = 0 \tag{14}$$

Chaque segment binaire est vu comme un index, de la manière suivante : le segment M_i^r du message indexe une des 2^r fonctions $F_{M,k}$, le segment K_i^l de la clé indexe une des 2^l fonctions $F_{K,k}$. Soit $b_\nu : \{0,1\}^\nu \rightarrow \mathbb{N}$ une bijection, permettant d'associer une valeur d'index à un segment binaire. L'équation suivante définit une fonction de marquage W permettant de « porter » le message M composé de $w \times r$ bits :

$$W_s(i,j) = \frac{1}{w} \sum_{p=1}^{w} F_{K,b_l(K_p^l)}(i,j) \times F_{M,b_r(M_p^r)}(i,j) \tag{15}$$

Soit l'image I, sur laquelle nous calculons une transformée en ondelettes discrète. Notons I_s ($s \in HL,\ LH,\ HH$) les trois images de plus fine résolution, chacune de taille $(m \times n)$, et composées des coefficients d'ondelettes $I_s(i,j)$. L'insertion du message dans l'image se fait selon un schéma additif, pour accentuer le marquage des coefficients d'ondelettes les plus significatifs. Soit I_s^W la version tatouée de I_s, et ω_s un masque de pondération psychovisuelle [7] calculé à partir de la représentation multirésolution de l'image I. Pour chacune des trois images I_s, le masque ω_s fournit une valeur de pondération réelle par coefficient d'ondelette. Soit enfin le coefficient α permettant de contrôler la force du tatouage. L'insertion du message se fait de la façon suivante :

$$I_s^W(i,j) = I_s(i,j) + \alpha \times \omega_s(i,j) \times W_s(i,j) \tag{16}$$

L'image tatouée I^W est ensuite obtenue par transformation en ondelettes inverse, à partir des images de coefficients I_s^W.

3.3. lecture du message

La lecture des w paquets de r bits à partir de l'image I^W peut se faire assez facilement, si les conditions d'orthogonalité (équations 12, 13 et 14) sont vérifiées. Nous proposons ici la méthode simple suivante : soit une clé K et une image tatouée. La clé est découpée selon (7), pour extraire chacun des segments K_p^l. Pour chaque segment K_p^l, il reste à calculer les corrélations entre I_s^W et les produits $\omega'_s \times F_{K,b_l(K_p^l)} \times F_{M,b_r(M_q^r)}$ pour chaque $q \in \{1, 2^r\}$:

$$T_s^p(i,j) = I_s^W(i,j) \times \omega'_s(i,j) \times F_{K,b_l(K_p^l)}(i,j) \tag{17}$$

$$C^q(K_p^l) = \frac{1}{3mn} \sum_s \sum_{i=1}^{n} \sum_{j=1}^{m} T_s^p(i,j) \times F_{M,b_r(M_q^r)}(i,j) \tag{18}$$

Le masque ω'_s est recalculé sur l'image tatouée et éventuellement attaquée. Pour un segment K_p^l donné, on ne conserve que les fonctions $F_{M,b_r(M_q^r)}$ qui retournent la corrélation maximale $C_1 = \max_q \{C^q(K_p^l)\}$, et la deuxième corrélation C_2, maximum des $2^r - 1$ corrélations restantes. La fonction $F_{M,b_r(M_q^r)}$ associée au segment K_p^l est jugée correcte si la différence $(C_1 - C_2)$ est supérieure à un seuil fixé, calculé de la façon suivante :

$$C_1 - C_2 > \mathcal{T} \times C_1, \quad \text{avec } 0 < \mathcal{T} < 0.3 \tag{19}$$

Si le test (19) réussit, la séquence binaire M_q^r constituant une partie du message M est retrouvée à partir de la réciproque b_r^{-1}. On peut noter dès à présent que des compromis devront être trouvés sur les longueurs l (resp. r) des segments de clé (resp. message). La longueur r ne doit pas être trop élevée car sinon, un nombre trop important de bits peut être perdu, en cas de fausse détection. La lecture de la marque par corrélations devient également trop coûteuse en temps de calculs. Et enfin il deviendrait difficile d'introduire un nombre trop important de fonctions orthogonales dans un support d'image de taille limitée

(nous n'utilisons que les trois quart de la surface de l'image originale pour cacher le message). Cette dernière remarque impose de limiter également la longueur l.

Nous venons d'exposer la manière d'obtenir les valeurs des corrélations à partir de l'image tatouée. Néanmoins, nous pouvons encore améliorer le schéma en affinant la politique de détection. En effet, suivant que l'on a intérêt ou non à pouvoir relire coûte que coûte le message (tatouage robuste ou fragile), on implémentera différemment le module de décision. Il apparaît en effet en pratique que lorsqu'un segment de message est mal décodé, la corrélation correspondante au segment effectivement inséré fait souvent partie des P plus grandes valeurs (nous avons pour l'instant $P = 2$). Afin de diminuer les erreurs de relecture, on pourrait choisir de prendre en compte davantage de corrélations pour mieux détecter le message. En outre, ce procédé pourrait être complété par l'utilisation de codes correcteurs définis sur des mots (Reed-Solomon) : on pourrait ainsi discriminer plus facilement entre deux segments litigieux (ou plus). La difficulté ici vient du fait que l'on ne peut disposer dans le cas générique d'aucun outil statistique pour estimer la confiance dans le message relu. En conséquence, une autre méthode est à envisager, peut-être fondée l'utilisation de codes correcteurs.

4. implémentation

Nous présentons ici deux exemples d'implémentation de la méthode générique décrite dans la section 3, en utilisant deux types de fonctions de base, puis des résultats expérimentaux seront donnés dans la section 5. La transformation de l'image originale est calculée à partir d'ondelettes biorthogonales, selon un schéma « lifting ». Nous nous fixons pour objectif d'introduire un message de 64 bits dans une image monochrome de taille 256×256, codée sur 8 bits par pixel. Nous vérifierons les qualités de l'algorithme en terme de robustesse face à des attaques volontaires ou pas, de type JPEG ou géométriques.

Barni *et al.* [1] ont montré la robustesse de leur algorithme de tatouage par ondelettes et étalement de spectre, en utilisant des fonctions pseudo-aléatoires réelles centrées, de variance unité, et supposées être suffisamment orthogonales. De façon à tester la robustesse de notre algorithme face aux mêmes attaques que dans [1], nous avons utilisé le même type de fonctions pour F_M et F_K lors des deux expérimentations que nous avons menées : la première avec des séquences pseudo-aléatoires (pour vérifier la validité du schéma générique), et la seconde avec des fonctions plus régulières (pour introduire le tatouage mou). Notons ici que ces deux ensembles de fonctions nous permettent de cacher w paquets de r bits dans une image, et non plus un seul bit, comme dans [1]. La lecture du message se fait par calcul de corrélations entre une fonction pseudo-aléatoire extraite de l'image tatouée et un ensemble d'autres fonctions, proposées par le propriétaire de l'image originale. Ceci impose donc que la géométrie globale de l'image ne soit pas trop modifiée après tatouage. Dans le cas contraire, les correspondances entre composantes des fonctions sont perdues (les fonctions sont dites « désynchronisées ») et le message ne peut pas être relu. Ce type de méthode de tatouage simple ne résiste donc pas à des déformations géométriques appliquées sur la totalité du support de l'image tatouée. Une solution proposée par différents auteurs est de cacher dans l'image des points ou motifs « d'ancrage », faciles à retrouver avant lecture du message, et permettant de compenser les déformations géométriques. Mais ces points peuvent éventuellement être effacés de l'image tatouée [14], et ce problème de resynchronisation *globale* n'a d'ailleurs rien de commun avec les objectifs premiers que se fixe le tatouage mou : pallier les déformations géométriques *locales* de manière aveugle ; même s'il reste toujours possible de considérer le problème en se servant du schéma générique et en inventant de nouvelles fonctions.

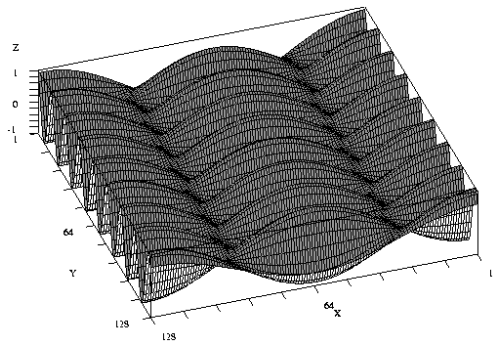

Figure 3. – Exemple de paquet : un produit de deux sinusoïdes orthogonales $F_M \times F_K$.

Nous proposons ici d'utiliser des fonctions orthogonales pour F_M et F_K, plus régulières que des séquences pseudo-aléatoires, de façon à être moins sensibles aux problèmes de désynchronisation locale. Les algorithmes d'attaque tels que *Stirmark* appliquent de faibles déformations locales sur l'ensemble du support de l'image, tout en préservant une qualité d'image suffisante. Nous proposons dans notre cas d'utiliser des sinusoïdes orthogonales de basses fréquences, horizontales pour F_M (figure 3), et verticales pour F_K, de périodes suffisamment grandes par rapport à la taille des déformations locales introduites par exemple par *Stirmark* :

$$F_{M,k}(i,j) = \cos\left(\frac{2 \times \pi \times k \times i}{\beta_M}\right) \tag{20}$$

$$F_{K,k}(i,j) = \cos\left(\frac{2 \times \pi \times k \times j}{\beta_K}\right) \tag{21}$$

Nous choisissons arbitrairement les longueurs $l = r = 8$ et $\beta_K = 2 \times \beta_M = 256$. Le message sera introduit avec une force

$\alpha = 0,65$ de façon à préserver la qualité visuelle des images marquées, et la valeur de $\mathcal{T}$ qui permet de relire chacun des segments du message est expérimentalement fixée à 0.10.

5. résultats

Afin de montrer l'influence de différentes attaques sur la lecture du message, nous choisissons d'illustrer nos résultats sur la seule image Lena, sachant que des résultats similaires ont été obtenus sur d'autres images « photographiques » de même taille. Comme nous l'avons souligné précédemment, nous devons prendre garde à la taille de la famille de fonctions associées au message puisque toutes (au pire des cas) devront être testées lors de la corrélation pour trouver le segment de message caché. Pour notre implémentation, nous avons utilisé un PII-350 et le temps de relecture du message est d'environ 1 minute 30 secondes ($l = r = 8$). Cette méthode ne concerne donc pas les applications orientées temps-réel (vidéo), à moins bien sûr de développer un circuit intégré dédié. Nous donnons figure 4 les images produites par l'algorithme : l'image tatouée, et la différence entre l'image originale et l'image tatouée (amplifiée).

5.1. tatouage par étalement de spectre

Nous vérifions que des fonctions pseudo-aléatoires pour F_M et F_K permettent de retrouver les 8 paquets de 8 bits, dans l'image Lena, après des attaques simples de type « traitement du signal » telles qu'un filtrage passe-bas 3×3, une compression JPEG (facteur de qualité 50 %), et la conservation d'un quart de la surface de l'image, choisi aléatoirement. La région extraite dans ce dernier cas doit bien sûr pouvoir être replacée à sa place sur le support de l'image originale pour resynchroniser les portions du message (on parle dans ce cas de *pseudo-cropping*). Le message est également retrouvé si une partie seulement de l'image tatouée est déformée. Elle est par contre effacée (par désynchronisation) si la totalité de la géométrie de l'image est déformée *localement*, même faiblement avec des algorithmes tels que *Stirmark*. Néanmoins, malgré ses moins bonnes performances que celles de la méthode de référence du fait du compromis capacité/robustesse, nous avons pu vérifier la validité du schéma générique. Nous devons dès à présent souligner que les performances en terme de robustesse seront meilleures dans le cas du tatouage mou, y compris en cas d'attaque de type filtrage.

5.2. tatouage mou

Afin de tester cette méthode, nous considérerons les deux attaques suivantes : une forte compression JPEG, et une déformation géométrique importante de la totalité de l'image tatouée. Nous parlons ici de tatouage mou car nous comptons que la régularité des fonctions de base fera en sorte que la marque se déformera avec l'attaque, de la même manière qu'une structure mécanique doit être flexible pour résister aux sollicitations. Malgré tout, nous espérons que la déformation sera suffisamment faible par rapport aux fonctions non attaquées, à la fois pour que la réponse soit encore significative à la corrélation, mais aussi pour ne pas coder un autre segment de message et générer ainsi des erreurs de relecture.

Figure 4. – Lena tatouée ($\alpha = 0,65$) et différence avec l'originale ($\times 32$). PSNR : 34,66dB. Instantiation du schéma générique : cosinus basses fréquences.

On soulignera ici la nécessité qu'il y a à ce que la marque soit insérée pixel à pixel dans l'image, plus exactement que la marque soit *spatialement* solidaire de l'image. Ainsi le tatouage mou n'est-il possible qu'en travaillant sur un espace de tatouage rendant compte de la géométrie de l'image. Jusqu'à présent, deux représentations sont possibles : le domaine spatial brut (ce qui impliquerait de modifier notre *modus operandi* mais permettrait de travailler plus finement sur l'image), ou bien une représentation par ondelettes.

À l'évidence, les représentations multirésolution demeurent dans notre cas un espace de choix car l'information est organisée de façon spatio-fréquentielle, permettant une optimisation du *modus operandi* à laquelle nous n'avons pas souhaité nous consacrer. Nous avons cherché à montrer la validité de notre modèle à l'aide de fonctions de bases simples : des cosinus de basses fréquences. Ce choix est très limitatif et offre des failles de sécurité évidentes, mais il permet néanmoins de vérifier que des fonctions douces (régulières) peuvent coder l'information à cacher et offrir une conception intuitive des fonctions de base en profitant directement de leur spécificités spatiales. Il est patent qu'un travail plus profond se révèle nécessaire sur ces fonctions offrant une paramétrisation fine, presque graphique, du tatouage.

En pratique, on constate que des erreurs peuvent se produire après des distortions géométriques fortes, quand le module de relecture confond deux fonctions (codant un segment de message) proches l'une de l'autre. Afin de diminuer le BER, on s'arrange pour que deux fonctions proches (en terme d'index) codent des segments binaires proches (en terme de distance de Hamming). En effet, si un segment est confondu avec un autre, autant que la différence (en bits) entre eux soit minime. La bijection b_r sera donc une disposition en code Gray des 2^r motifs binaires possibles. Ainsi, le nombre de bits potentiellement perdus est-il minimal.

Nous ne considérons dans cette partie que les attaques sur la géométrie de la marque n'ayant entraîné aucune perte de bit. En effet, un paramètre rendant compte d'une déformation géométrique progressive est dépendant du type de déformation (le BER dépend de chaque attaque et de son affinité avec les fonctions de base) ; et les résultats pratiques montrent que le module de relecture n'est pas optimisé puisque pour la grande majorité des tests (causant une perte de bits), seuls 6 bits en moyenne sont faux à l'approche des conditions limites d'attaque (l'image devient fortement dégradée du fait de l'attaque). Cela est causé par la détection de fonctions proches (en index) des fonctions effectivement insérées. Deux pistes d'amélioration sont envisageables : optimiser la géométrie des fonctions de base (maximiser les différences de corrélation entre deux fonctions d'index proches – et probablement diminuer la valeur de r), et prendre en compte davantage de valeurs de corrélations afin de pouvoir discriminer sur un ensemble de fonctions détectées plus grand à l'aide de codes correcteurs.

5.3. attaques JPEG

Nous vérifions expérimentalement que les 64 bits sont retrouvés, à partir de l'image Lena tatouée (avec $\alpha = 1.8$), puis compressée avec un facteur de qualité de 17 %. L'image a bien sûr dans ce cas perdu toute valeur commerciale, mais le résultat laisse penser qu'une dégradation moindre permettra de retrouver le message plus facilement, même si la force de marquage α doit être réduite afin d'assurer une totale transparence de la marque. Nous constatons le même type de résultats sur d'autres images photographiques de même taille. La figure 5 illustre l'étape de lecture des huit paquets de huit bits du message. Pour un paquet donné, nous affichons à chaque instant les valeurs des corrélations maximales intermédiaires. Les deux derniers pics représentent donc les valeurs de corrélations maximales C_2 et C_1, que l'on compare selon (19).

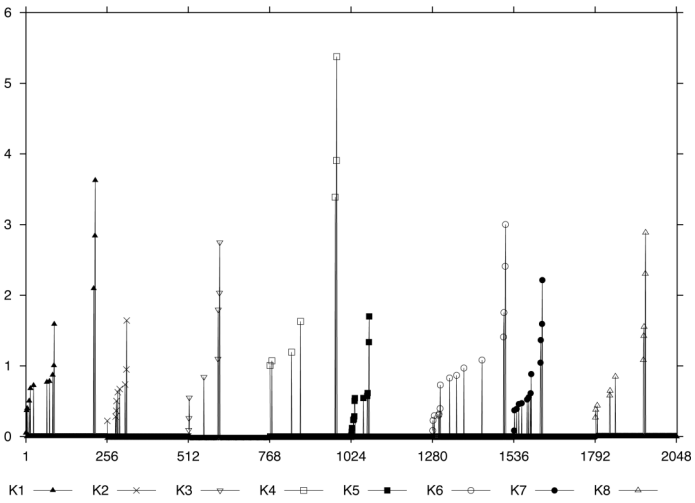

Figure 5. – Lecture du message après attaque JPEG. Pour chaque paquet du message, nous affichons à chaque instant les valeurs des corrélations maximales intermédiaires. Les deux derniers pics représentent les valeurs de corrélations maximales C_2 et C_1, qui sont comparées.

À l'inverse des attaques géométriques, la compression JPEG permet de donner une estimation du BER. La figure 6 donne une idée de ce BER même si davantage de tests auraient permis d'avoir une meilleure représentation. Nous rappelons qu'aucun code correcteur d'erreur n'a été employé, et que le module de relecture peut être largement optimisé. Il s'agit ici d'une évaluation brute du tatouage mou face à la compression JPEG classique.

On peut considérer que tel quel, le *modus operandi* permet de résister convenablement (sans plus) à la compression JPEG. Agissant comme un passe-bas, elle est relativement clémente avec nos cosinus, que l'on retrouve sans ambiguïté jusqu'à un facteur de qualité de 30 %. Au-delà, la plus grande corrélation par segment de clé peut ne pas être celle à laquelle correspond la fonction codant effectivement un segment de message. Il arrive dans peu de cas que l'erreur (en terme d'index) soit grande, mais généralement elle est faible : on détecte une fonction souvent distante d'un ou deux index au plus. La plus grande corrélation pour le segment de message relu correspond donc à une fonction parmi $F_{M,k+b_r(M_p^r)}$ avec $-D \leqslant k \leqslant D$ lorsque des erreurs dues à une attaque se produisent. En pratique, nous avons souvent constaté que si nous autorisions simplement de pouvoir choisir intelligemment parmi les corrélations situées à une distance $D = 2$ de la plus grande corrélation détectée, le

modus operandi actuel permettrait malgré sa simplicité de pouvoir tolérer des taux de compression plus élevés encore (correspondant à un facteur de qualité de l'ordre de 20 %, ce qui correspond à un facteur de qualité bien en-deçà de celui d'une image utilisable).

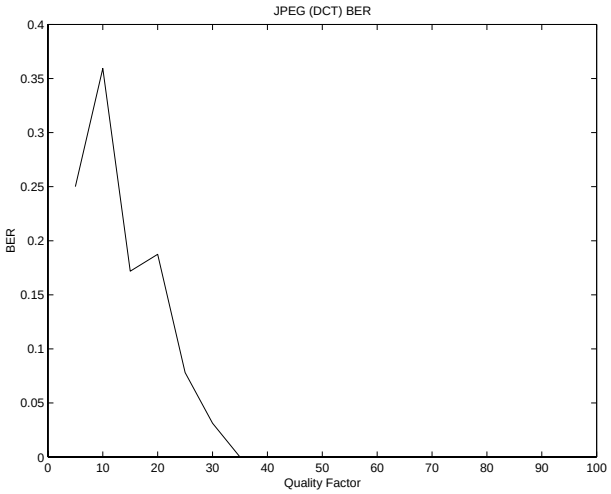

Figure 6. – Allure du BER fonction du facteur de qualité JPEG. $\alpha = 0,65$ **(PSNR : 34,66dB). Tatouage mou (cosinus BF).**

5.4. attaques géométriques

Une transformation globale sinusoïdale d'une amplitude de 4 pixels et de période de 96 pixels est appliquée sur l'image Lena tatouée (figure 7). Cette déformation a été préférée à celles causées par *Stirmark*, moins visibles. Telle quelle, cette méthode ne peut résister convenablement à toutes les attaques disponibles dans *Stirmark*, notamment celles entraînant une désynchronisation globale de la marque. En effet, la structure même des cosinus fait qu'aucune sorte d'origine ou de référence ne peut être retrouvée en cas de fenêtrage. L'information est toujours présente mais illisible. Il s'agit, outre l'apparition de pics dans la transformée de Fourier, de la principale critique contre l'utilisation de cosinus. Nous rappelons que nous voulions avant tout vérifier la vraisemblance de notre schéma de tatouage mou.

Les limites de résistance aux attaques géométriques reposent entièrement sur les fonctions utilisées. Nos simples cosinus nous ont permis de retrouver *en aveugle* la marque après application de petites déformations locales causées par *Stirmark*, dont l'amplitude aux bords ne devait pas excéder 2 pixels. Les mêmes attaques avait causé à la fois l'effacement de la signature de notre méthode de référence et celui du message s'il était codé à l'aide de séquences orthogonales. Cela est évidemment dû au fait que la marque s'est déformée en restant cohérente avec elle-même (avant attaque). Il reste à présent à dessiner des fonctions répondant aux attaques géométriques que l'on veut pallier.

Naturellement, le problème du fenêtrage est ici central car il implique en quelque sorte que la marque puisse être décodée depuis n'importe quel *Minimum Watermark Segment*. Cela impose des contraintes fortes sur les fonctions de base, que nous n'avons pas encore eu le loisir de développer plus avant dans nos travaux.

L'utilisation de fonctions régulières pour F_M et F_K permet de retrouver les huit paquets de huit bits (figure 8), dans l'image Lena de taille 256×256 attaquée et représentée figure 7. Bien évidemment, le tatouage est ici appliqué trop fortement, mais il s'agit de montrer la validité du principe. En tatouant moins fort, on parvient à retrouver le message après de plus petites déformations (amplitude : 2 pixels), mais sans chercher à optimiser la géométrie des fonctions de base. Nous vérifions qu'il est possible de cacher puis relire un message de 128 bits dans l'image Lena 512×512, attaquée par déformation cosinusoïdale. L'orthogonalité fréquentielle et spatiale des fonctions des ensembles F_M et F_K est dans ce cas mieux assurée. La plus grande surface disponible dans le domaine de tatouage donne

Figure 7. – Déformation sévère de l'image Lena. Le tatouage mou permet de retrouver la marque en aveugle sur de petites déformations locales comme celles-ci ($\alpha = 1,8$**).**

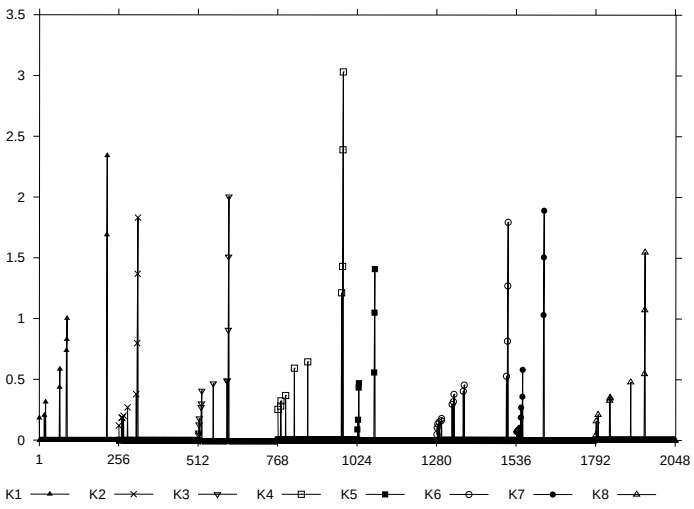

Figure 8. – Lecture du message après attaque géométrique.

des corrélations numériquement bien mieux marquées que sur la version 256×256 pixels (que nous avons considérée pour tous nos tests), la robustesse globale est meilleure sur la version 512×512 pixels.

La figure 9 montre d'autres exemples d'attaques qui n'effacent pas les 8×8 bits cachés dans Lena 256×256. Nous avons pour cela passé en revue quelques traitements d'un logiciel courant de manipulation d'images.

Figure 9. – Relecture de 64 bits dans les images attaquées localement (*warp localisé, pseudo-cropping, flash*), et globalement (bruit gaussien, *rehaussement de contraste, effet canvas*).

6. conclusions et perspectives

Nous avons proposé dans cette article une nouvelle méthode générique de tatouage d'images, permettant de cacher un message de w paquets de r bits (typiquement 8×8 bits) dans une image photographique monochrome de taille 256×256. La méthode insère dans l'image une somme de produits de fonctions orthogonales, et ces dernières peuvent être choisies en fonction du type d'attaques auxquelles le tatoueur devra faire face. La méthode s'est révélée être robuste face à des attaques simples de type traitement du signal, mais aussi à des attaques géométriques locales, réparties sur tout le support de l'image. Elle n'introduit pas de points ou de motifs connus dans l'image, et ne permet donc pas de compenser une déformation globale de l'image, telle qu'une rotation de 45 degrés par exemple. Cette assertion est vraie dans le cas général mais demande à être nuancée dans le cas de fonctions de base cosinusoïdales : la transformée de Fourier de l'image tatouée met en évidence des pics correspondant aux cosinus utilisés. Il devient évident que le choix des fonctions de base demande à être optimisé en fonction de critères dépendant de l'application visée. D'une manière générale, on peut affirmer que notre message est effaçable mais reste illisible pour qui n'est pas autorisé à le lire. La méthode générique peut bien sûr être améliorée. Nous proposons ici plusieurs solutions. L'utilisation de fonctions pseudo-aléatoires permet de calculer des seuils statistiques dépendant d'une probabilité fixée à l'avance de fausse détection des segments du message, comme l'ont déjà proposé différents auteurs, pour cacher un bit dans une image [2,1]. Les fonctions régulières proposées dans l'article (des cosinus) sont orthogonales, mais elles perdent cette propriété s'il est nécessaire d'en cacher un grand nombre dans les trois quarts de la surface de l'image à tatouer (les images des coefficients d'ondelette de plus fine résolution). Il reste donc à trouver des fonctions orthogonales régulières, plus locales, pouvant être disposées à différents endroits de l'image. Ces fonctions peuvent en outre être secrètes (ex: [5]), afin de rendre plus difficile la recherche du message. Il est également possible de considérer d'autres transformations en ondelettes, à base par exemple d'ondelettes complexes [8], qui présentent l'avantage d'être réversibles, et d'offrir un plus grand nombre de coefficients à tatouer (et donc « plus de place »). Les w paquets de r bits ont été introduits dans l'image, indépendamment les uns des autres. Des codes correcteurs d'erreurs par paquets (ex : Reed Solomon) pourraient donc être considérés. La procédure de détection utilisée dans l'article peut également être améliorée, par exemple en ne considérant pas que les deux seules corrélations les plus élevées, par segment de message. Enfin, une gestion plus efficace des clés et des messages en s'inspirant de résultats de cryptographie renforcera la sécurité du schéma global de tatouage.

BIBLIOGRAPHIE

[1] M. Barni, F. Bartolini, V. Cappellini, A. Lippi and A. Piva, « A DWT-based technique for spatio-frequency masking of digital signatures », *SPIE, vol. 3657, Conference on Security and Watermarking of Multimedia Content, Electronic Imaging*, San Jose, January 1999.

[2] M. Barni, F. Bartolini, V. Cappellini and A. Piva, « A DCT-domain system for robust image watermarking », *Signal processing*, Vol. 66, pp. 357-372, 1998.

[3] W. Bender, D. Gruhl and N. Morimoto, « Techniques for data hiding », in *Proc. of the SPIE*, San Jose, USA, Feb. 1995, pp. 2420-2440.

[4] I. Cox, J. Killian, T. Leighton and T. Shamoon, « Secure spread spectrum watermarking for multimedia », *IEEE Transactions on Image Processing*, 6(12) :1673-1687, December 1997.

[5] J. Fridrich, Lt.A.C. Baldoza, R.J. Simard, « Robust digital watermarking based on key-dependent basis functions », LNCS 1525, *Intl. Information Hiding Workshop*, Portland, USA, April 1998.

[6] J.F. Delaigle, C. DE Vleeschlouwer and B. Macq, « Watermarking based on a human visual model », *Signal Processing: Image Communication*, 66 (3) : 319-335, May 1998

[7] A. Lewis and G. Knowles, « Image compression using the 2-D wavelet transform », *IEEE Transactions on Image Processing*, 1 (2), pp. 244-250, April 1992.

[8] P. Loo and N. Kingsbury, « Digital watermarking using complex wavelets », *IEEE Intl. Conference on Image Processing*, Vancouver, Canada, 10-13 Sept. 2000.

[9] J. Picard and A. Robert, « Public key watermarking using neural networks », *SPIE, Security and Watermarking of Multimedia Contents III*, San Jose, USA, 22-25 Jan. 2001.

[10] P. Loo and N. Kingsbury, « Motion estimation based registration of geometrically distorded images for watermarking recovery », *SPIE, Security and Watermarking of Multimedia Contents III*, San Jose, USA, 22-25 Jan. 2001.

[11] T. Furon, I. Venturini and P. Duhamel, « Unified approach of asymmetric watermarking schemes » *SPIE, Security and Watermarking of Mutimedia Contents III*, San Jose, USA, 22-25 Jan. 2001.

[12] F. Petitcolas, R. Anderson and M. Kuhn, « Attacks on copyright marking systems », *Information hiding, 2nd International Workshop*, Portland, USA, 15-17 Avr. 1998.

[13] E. Koch and J. Zhao, « Towards robust and hidden image copyright labeling », *Proceedings of IEEE Workshop on non-linear processing*, pp. 452-455, June 1995.

[14] S. Voloshynovskiy, A. Herrigel and Y. Rytsar, « Watermark template attack », *SPIE, Security and Watermarking of Multimedia Contents III*, San Jose, USA, 22-25 Jan. 2001.

Manuscrit reçu le 18 avril 2001

LES AUTEURS

François CAYRE

François CAYRE est né en 1977. Il a obtenu son diplôme d'ingénieur en Génie informatique et le DEA Contrôle des Systèmes à l'Université de Technologie de Compiègne, en 2000. Il prépare actuellement une thèse dans le laboratoire de Télécommunications et Télédétection de l'Université catholique de Louvain, en Belgique, sur le thème du tatouage d'objets tridimensionnels.

Franck DAVOINE

Franck DAVOINE est né en 1967. Il a obtenu son doctorat en Signal Image Parole à l'INPG en décembre 1995. Après un séjour postdoctoral de 18 mois à Linköping (Suède), il est nommé Maître de Conférences à l'Université de Technologie de Compiègne. Il est actuellement Chargé de recherche au CNRS, au sein de l'UMR Heudiasyc (UTC). Les domaines d'applications privilégiés de ses recherches sont le tatouage d'images, et l'analyse de visages.