

# Tatouage d'images avec utilisation de la transformée en cosinus discrète

Lysandre Debut et Alexis Vialaret

**Abstract**—Depuis la nuit des temps l'homme cherche à tatouer des images numériques. Pour cela il a d'abord inventé les ordinateurs, puis s'est mis en quête de méthodes et formules dans le but d'un jour pouvoir insérer des images dans d'autres images. Ce travail absolument novateur et inédit atteint enfin ce but, et permet à l'humanité entière de s'élever à un niveau jamais atteint auparavant.

## I. INTRODUCTION

Le tatouage d'image est une technique permettant d'ajouter des informations dans une image numérique. Ces informations sont codées sous la forme de bits et peuvent donc contenir un message, une image, du son ou encore une vidéo. Ces "tatouages" aussi appelés marques peuvent être visibles (exemple de copyright) ou bien invisibles, ce qui permet de cacher le message. Pendant ce projet nous avons cherché à cacher le message, et plusieurs techniques existent pour y parvenir. Ces techniques se séparent en deux catégories : les méthodes qui travaillent dans le domaine spatial et les méthodes qui travaillent dans le domaine spectral.

Nous avons précédemment étudié une méthode qui consistait à encoder un message sur le bit de poids faible de chaque pixel composant l'image (méthode du LSB). Chaque pixel étant fait de trois composantes : rouge, verte et bleue, l'encodage se fait généralement sur une seule des composantes. Ceci permet d'encoder un message de manière relativement discrète, le bit de poids faible ne représentant que 1/255ème de la couleur lorsqu'elle est encodée sur 8 bits. Cette technique travaille dans le domaine spatial. Cette manière d'encoder un message est considérée comme fragile car si l'image se trouve un tout petit peu modifiée, la marque ne sera pas identifiable (modifiée en couleur, en taille). De plus, cette méthode n'offre aucune sécurité intrinsèque, ce qui peut être résolu en encodant le message avec une clé.

Les méthodes dans le domaine spectral permettent de pallier le problème, en étant plus robustes face aux modifications, et c'est là-dessus que nous nous sommes penchés pour ce projet. La méthode utilisée est appelée "DCT", la transformée en cosinus discrète. Cette transformation se rapproche de la transformée de Fourier discrète (DFT), cependant la DCT crée des coefficients réels car étant basée sur l'utilisation d'une fonction cosinus, contrairement à la DFT qui crée des coefficients complexes car basée sur l'utilisation d'une exponentielle complexe. Cette transformée est très utilisée en traitement du signal, de l'image mais aussi dans les opérations de compression car c'est une transformation sans pertes (le signal original peut être retrouvé à l'identique en effectuant l'opération inverse). L'encodage d'information dans une image à l'aide de cette transformée se fait grâce à

une clé de cryptage, qui définit les fréquences dans lesquelles l'information sera encodée. L'utilisation de cette clé de cryptage fournit une sécurité intrinsèque. Le tatouage en utilisant cette méthode se fait par l'utilisation de blocs, sur lesquels est encodée une partie de l'information. Ceci assure une robustesse optimale en encodant le message sur la totalité de l'image, mais offre une densité d'information beaucoup plus faible que la méthode du bit de poids faible.

Afin de réaliser ce projet, nous avons utilisé le logiciel Matlab, qui permet de faire de la gestion de matrices et donc de la gestion d'images, de texte et de vidéo.

## II. MÉTHODES

### A. Tatouage d'image dans une autre image

Nous avons voulu rendre notre programme le plus flexible possible. Il sera capable de tatouer des images ou des vidéos de différentes résolutions avec des marques en noir et blanc elles aussi de résolutions différentes sans problèmes, et ce dans plusieurs couches de l'images. Il y a cependant tout de même des limites strictes imposées par la méthode de tatouage, et des limites plus souples pour la discrétion et la robustesse de la marque. Il faut aussi prendre en compte une quatrième contrainte qui est le temps de traitement nécessaire pour effectuer le traitement. Afin de trouver le compromis le plus satisfaisant entre ces quatre composantes (densité d'information, discrétion, robustesse, temps de traitement), nous allons appliquer des marques de tailles différentes, avec des puissances de marquage différentes et comparer les résultats obtenus. L'image source sera une image en RGB de 512x512 pixels. Les images tatouées dessus seront des logos de l'ESEO en noir et blanc dont les tailles vont de 8x8 pixels à 128x128 pixels.

Afin de générer une clé de cryptage pertinente, il est important de comprendre le fonctionnement de la DCT. Cette transformation regroupe les informations des images par fréquences : Les plus hautes fréquences sont situées en haut à gauche, et décroissent en s'éloignant. Afin de se placer à mi-chemin entre la discrétion et la robustesse, nous choisissons donc d'encoder nos informations sur les moyennes fréquences. Encoder sur les hautes fréquences revient à placer l'information sur les bits de poids fort de l'image et la rend donc peu discrète. En revanche, la mettre sur les basses fréquences est équivalent à modifier les bits de poids faible, qui sont plus susceptibles d'être altérés par des compressions, déformations, etc...

La génération de la clé se passe donc de la manière suivante : on définit d'abord un masque qui préserve les hautes et basses fréquences qu'on étire afin qu'il soit de la taille des blocs d'encodage ( $M \times N$ ). Le nombre total de



Fig. 1. Génération de la clé de cryptage

blocs est égal au nombre de bits dans la marque. On génère ensuite une image (MxN) contenant une répartition aléatoire et uniforme de -1, 1, et 0 sur laquelle on applique le masque. C'est notre clé de cryptage finale.

Nous divisons ensuite notre image source en une série de blocs sur lesquels nous appliquons la DCT. C'est ces blocs que nous marquons avec plus ou moins de force ( $\alpha$ ) en y ajoutant ou retranchant la clé pondérée :

$$0_{\text{binaire}} = \text{bloc} - \alpha * \text{cle} \quad (1)$$

$$1_{\text{binaire}} = \text{bloc} + \alpha * \text{cle} \quad (2)$$

Ces blocs ne seront marqués que sur une seule de leurs couches, soit la rouge, la verte ou la bleue. Nous avons commencé par marquer la couche bleue car les différences de couleurs dans ce domaine spectral sont moins visibles pour l'œil humain.

Une fois que le tatouage dans les filtres rouge, vert et bleu seront opérationnels, nous effectuerons une transformation de l'image du domaine "RVB" vers le domaine "HSL" (Teinte, saturation, luminosité), afin de pouvoir mettre le tatouage dans une de ces trois couches.

Une fois le tatouage effectué, il ne reste plus qu'à appliquer la DCT inverse sur chacun des blocs, puis à les recoller ensemble pour retrouver l'image originale.

Nous avons accès à plusieurs paramètres qui nous permettent de jouer sur les propriétés de marquage afin de nous placer où nous voulons dans l'espace des contraintes : la taille des images (source et marque), la force de marquage, les couches tatouées...

### B. Tatouage de texte dans une image

Pour pouvoir ajouter une dimension intéressante à notre programme, nous avons choisi d'offrir la possibilité d'encoder du texte dans l'image. Pour ce faire, nous avons écrit une fonction qui convertit un texte en image, ensuite utilisée en tant que marque puis tatouée sur la source. Vu que les marques sont en noir et blanc, chaque pixel correspond à un bit d'encodage. La conversion de texte en image se fait de la manière suivante : le texte, encodé en UTF-8, est converti en caractères ASCII. Ces caractères ASCII sont encodés sur 8 bits, ce qui signifie que 8 pixels seront utilisés pour chaque caractère que nous souhaitons encoder. Nous rajoutons les caractères de fin de chaîne '\0' afin que le programme comprenne où s'arrêter de lire le texte, ce qui prend 16 bits d'encodage supplémentaires. Le programme

détermine la taille minimale de l'image en fonction du nombre de caractères dans le fichier '.txt' fourni. Le calcul de la taille de l'image se fait de la manière suivante : la taille minimale de la hauteur et de la largeur de l'image est définie par la première puissance de deux supérieure ou égale à :  $\sqrt{n + 16}$  avec n la taille du vecteur binaire correspondant aux caractères ASCII. Une fois la taille de la marque définie, une matrice de cette taille est créée dans laquelle les valeurs ASCII précédemment trouvées sont écrites, suivies des caractères d'échappement, qui sont ensuite suivis d'un bourrage de bits nuls. Le procédé de marquage est ensuite identique à celui précédemment énoncé. Le programme est rendu semi-automatique en détectant si le fichier "marque" fourni est de type ".png" ou ".txt". Dans ce dernier cas, le programme procédera à la transformation du texte en image qu'il utilisera en tant que marque. Nous avons choisi d'encoder la phrase suivante : "Lorem ipsum dolor sit amet, consectetur adipiscing elit" pour effectuer nos tests.

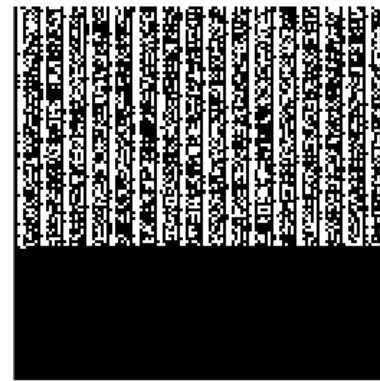


Fig. 2. Conversion de texte en image (texte de

Afin d'obtenir le texte en décodant l'image, il faut tout d'abord extraire la marque de l'image comme précédemment expliqué. La marque obtenue est donc normalement exactement la même marque que celle codée dans l'étape précédente. À partir de cela, la matrice est redimensionnée en matrice de huit colonnes, de manière à avoir chaque ligne représentant un caractère. Ces octets sont convertis en caractères, ce qui crée un vecteur de caractères. Ce vecteur, une fois transposé, représente une chaîne de caractères qui sera ensuite découpée avant le caractère d'échappement '\0' pour obtenir le message encodé.

Etant donné que chaque caractère est encodé sur 8 bits et que la marque est en noir et blanc, chaque caractère doit être encodé sur 8 pixels ce qui représente une densité d'information extrêmement faible.

### C. Tatouage dans une vidéo

On peut imaginer plusieurs types de tatouages sur une vidéo : un bit par trame, une seule image tatouée, toute la vidéo tatouée... Nous avons choisi de ne tatouer qu'une seule trame de notre vidéo, en plein milieu. En utilisant les objets Matlab Videoreader et Videowriter on peut extraire une image de la vidéo, la tatouer, et la replacer dans la vidéo. Là encore, le programme de tatouage repère l'extension .mp4

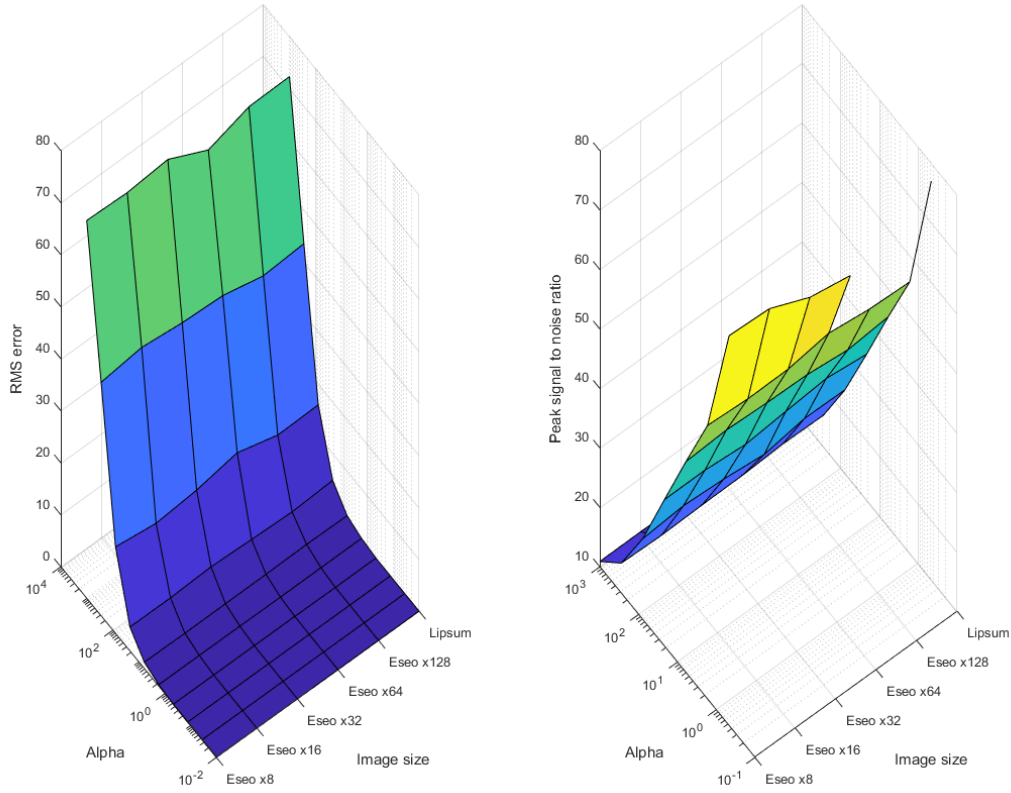


Fig. 3. Robustesse (RMSE) et discrétion (PSNR) en fonction de la taille de l'image

du fichier vidéo et effectue les actions de transformation requises automatiquement.

#### D. Méthodes d'évaluation des résultats

Nous allons tatouer plusieurs marques (8x8, 16x16, 32x32, 64x64, 128x128) sur la même image source. Nous appliquerons ces marques plusieurs fois chacune avec des forces de marquage différentes, allant de 0.01 à 1000 (échelle logarithmique). Pour toutes les images obtenues, nous mesurons la discrétion de la marque en calculant le PSNR et la distance euclidienne avec l'image source, et la robustesse en observant le taux d'erreur entre la marque décryptée et la marque originale.

### III. RÉSULTATS

Comme vu précédemment nous cherchons un équilibre entre les quatre propriétés suivantes dans le tatouage de l'image : la densité d'information, la discrétion de la marque, la robustesse du tatouage, et le temps de traitement. La densité d'information est fixée par les dimensions relatives des deux images (marque et source). La source est divisée en autant de blocs qu'il y a de bits dans la marque, c'est pourquoi plus les deux images sont proches en terme de taille, plus la densité d'information est importante. Avoir une forte densité d'information est cependant peu intéressant puisque très détrimentaire à la fois à la robustesse et à la discrétion. Le temps de traitement est proportionnel à la taille

en bits de la marque, on voudra donc avoir une marque de taille modeste afin de gagner en temps de traitement. Par exemple, une image en noir et blanc de taille 32x32 (1024 bits) prend 96 fois moins de temps à traiter qu'une image en RGB en 64x64 (98304 bits). Dès lors que la taille de l'image source est largement supérieure à celle la marque, alors nous avons le loisir de régler la force de marquage pour trouver le bon compromis entre discrétion et robustesse. Tout simplement, plus la force de marquage est élevé, plus le tatouage sera robuste, au détriment de la discrétion, et vice versa.

En modifiant la force de marquage ainsi que le filtre dans lequel nous encodons le message, nous pouvons remarquer des différences dans la discrétion du marquage. Dans le domaine RGB, le filtre bleu semblait altérer le moins l'image, du moins pour notre oeil qui est moins sensible à la différence d'intensité bleue. Dans le domaine HSL, la différence la moins visible se voyait dans la saturation. Les figures représentant un tatouage pour une force 30 et 100 pour chaque filtre sont disponibles en annexe.

Nous voyons sur le premier graphique que l'erreur RMS entre l'image source et l'image tatouée croît très peu pour alpha inférieur à 100, mais augmente drastiquement après. De plus on voit que la dimension relative de la source et de la marque a une influence négligeable sur la discrétion du tatouage. Le deuxième graphique représente l'évolution du PSNR de l'image source et de l'image tatouée. On y voit sans

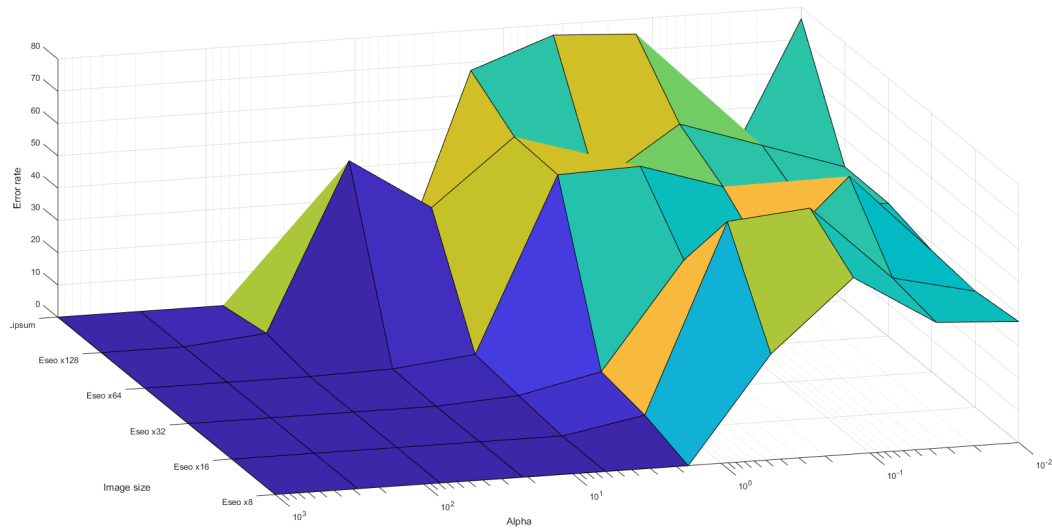


Fig. 4. Taux d'erreur entre la marque originale et celle décodée en fonction d'alpha et de la taille de marque



Fig. 5. Tatouage d'une image dans la couche bleue avec alpha = 30



Fig. 7. Tatouage d'une image dans la couche saturation avec alpha = 30

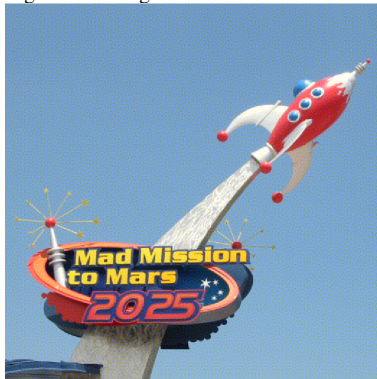


Fig. 6. Tatouage d'une image dans la couche bleue avec alpha = 100



Fig. 8. Tatouage d'une image dans la couche saturation avec alpha = 100

surprise que la force du signal (équivalent à l'image source) diminue quand alpha croît, cela signifie que plus alpha est grand, moins l'image tatouée ressemble à l'image source. On remarque que certaines valeurs du PSNR sont absentes quand la force de marquage devient très petite. En effet, lorsque la force tend vers 0, l'image source avant tatouage et après tatouage sont identiques, le PSNR tend alors vers

l'infini, ce qui ne permet pas une représentation graphique. La figure 4 nous renseigne sur le taux d'erreur. On voit qu'ici contrairement aux précédentes mesures, la taille de l'image a un fort impact sur la qualité de la marque au décryptage, et requiert une force de marquage plus élevée pour compenser.

On peut donc se servir de ces graphiques et en particulier du premier et du dernier pour régler les paramètres de

tatouage dans le but d'obtenir un marquage répondant aux spécifications requises.

Cependant, ces graphiques ne nous disent pas tout : les yeux étant plus sensibles au vert qu'au bleu, un tatouage sur la couche verte est plus visible qu'un tatouage sur la couche bleue, et ce pour les mêmes paramètres, ce qui n'est pas reflété dans la figure.

#### IV. DISCUSSION

Le but de cette étude était d'identifier si la transformée en cosinus discrète était une méthode intéressante pour effectuer un tatouage d'image de qualité (équilibre entre les quatre propriétés atteint). Nous avons pour référence la méthode de l'encodage sur le bit de poids faible, que nous avons effectuée l'an dernier. En utilisant la méthode de la DCT nous avons réussi à encoder une image et du texte dans une image ainsi que dans une vidéo. Nous l'avons fait de manière à ce que la faible modification de l'image ou la vidéo source, que ça soit en taille ou par ajout de bruit n'altère pas le message placé.

Cette dernière méthode s'est montrée plus difficile à appliquer, plus coûteuse en temps et moins dense en information, cependant la robustesse est beaucoup plus élevée. Ceci peut être expliqué par le domaine dans lequel ces méthodes travaillent : le domaine spectral pour la DCT et le domaine spatial pour la méthode du bit de poids faible, ce qui justifie les résultats. L'utilisation de cette méthode nous donne un avantage supplémentaire, qui est la possibilité de modifier la force du marquage de l'image, et ainsi de gérer sa capacité à résister à la modification/au bruit. L'aspect sécurité est également très intéressant avec cette méthode puisque contrairement à la méthode du LSB, il est nécessaire de connaître la clé de cryptage en plus de la méthode pour lire le message.

Nous aurions pu envisager d'autres transformées afin d'identifier quelle méthode nous aurait permis d'obtenir les meilleurs résultats et la meilleure robustesse. Parmi ces transformées nous pouvons notamment noter la transformée en sinus discrète (travail dans les imaginaires purs) et la transformée de Fourier discrète (travail dans les réels et imaginaires). La transformée en cosinus discrète que nous avons utilisé durant ce projet s'est cependant montrée très intéressante pour une robustesse optimale.

Nous pourrions aussi répartir l'information sur les trois couches de l'image. On divise ainsi la densité d'information par couche par trois, ce qui nous permet de meilleures performances en robustesse et discrétion. On pourrait aussi étendre cela à la vidéo, en intégrant une vidéo dans une vidéo avec par exemple les images sur un canal, et le son sur un autre.

Pour tester la robustesse dans un contexte d'utilisation réel de cette méthode de tatouage, il serait intéressant d'imprimer et de scanner l'image marquée et/ou de créer un outil permettant de réaliser des compressions successives en JPEG afin de visualiser la détérioration du message.

Nous avons fait le choix de nous baser sur une clé de cryptage qui privilégiait les moyennes fréquences pour l'encodage du message. L'étude de l'encodage dans les hautes ou basses fréquences se montrerait aussi intéressante que l'étude des différentes transformées, cependant nous avons été pris par le temps.