ORIGINAL ARTICLE

# ETUDE DE CHIFFREMENT D'IMAGES UTILISANT LA STEGANOGRAPHIE PAR DWT ET LA COMPRESSIVE SENSING

## IMAGE ENCRYPTION STUDY USING DWT STEGANOGRAPHY AND COMPRESSIVE SENSING

**| Tianandrasana Romeo Rajaonarison *[1] | et | Paul Auguste Randriamitantsoa [2] |**

[1.] Université d'Antananarivo | Ecole Doctorale en Sciences et Technique de l'Ingénierie et de l'Innovation (ED-STII) | Laboratoire de Recherche en Télécommunication, Automatique, Signal et Images (TASI) | B.P 1500 Antananarivo101 | Madagascar |
[2.] Université d'Antananarivo | Ecole Doctorale en Sciences et Technique de l'Ingénierie et de l'Innovation (ED-STII) | Laboratoire de Recherche en Télécommunication, Automatique, Signal et Images (TASI) | B.P 1500 Antananarivo101 | Madagascar |

| Received 30 April, 2022 | | Accepted May 05, 2022 | | Published May 09, 2022 | | ID Article | Romeo-Ref07-ajira030522|

## RESUME

La compressive sensing (C.S) est un nouveau domaine qui suscite beaucoup d'intérêt au sein du traitement de signal et du traitement d'image. Des avancées majeures dans la C.S a conduit à un développement dans les systèmes d'acquisition des images en échantillonnant les signaux à des fréquences inférieures à la fréquence de Shannon-Nyquist. La C.S exploite la parcimonie des signaux ou des images dans le domaine de transformation afin de le pouvoir le reconstruire avec un nombre réduit de mesures. La C.S se distingue des systèmes d'acquisition courant par sa capacité à échantillonner et à compresser le signal d'intérêt simultanément. Combinée à la stéganographie, la compressive sensing (C.S) offre un schéma de sécurisation de données très efficace.
***Mots-clés**: Compresive sensing, DWT, Shannon-Niquist, Stéganographie.*

## ABSTRACT

Compressive sensing is a new emerging field that has attracted considerable attention in signals and image processing. Recent advances in compressive sensing have led to the development of imaging systems that senses signal at frequencies lower than the Nyquist rate. Compressive sensing exploits the property of a signal being sparse in some transform domain so that these sparse signals can be recovered from only small number of measurements. Compressive sensing distinguishes itself from the current digitalization system by its ability to sense and compress the signal at the same time. Robust signal recovery is possible from these measurements that are proportional to the sparsity level of the signal. This process of recovery is done through algorithms which give back the original sparse signal. Finally, merged with steganography, it offers new scheme to secure images.
***Keywords**: Compresive sensing, DWT, Shannon-Niquist, Steganography.*

## 1. INTRODUCTION

Chaque instant, des quantités phénoménales de données sont acquises et sont transmises à travers le réseau ou entreposées dans les supports de stockage. Même si des techniques de compression sont déjà mises en œuvre ; elles ne sont pas véritablement suffisantes. C'est la raison pour laquelle la compressive sensing (C.S) a été mise en place. Elle forme une toute nouvelle approche afin d'acquérir intelligemment et efficacement les signaux. En imagerie en particulier, un autre défi de taille est de sécuriser les images afin de les protéger. Si la compressive sensing offre d'ores et déjà une protection aux images, combinée à d'autres méthodes telles que la stéganographie elle garantit une protection supplémentaire plus sûre et plus efficace. Cet article consiste à faire connaitre l'intérêt de combiner la compressive sensing à la stéganographie pour le chiffrement de l'image dans les futurs systèmes de numérisation et de communication. Une simulation sur Matlab permet d'implementer une stéganographie utilisant la transformée en ondelette et la sécurisation d'image par une acquisition comprimée en utilisant la compression et le chiffrement en même temps.

## 2. MATERIELS ET METHODES

Avant que les informations ne transitent sur la voie publique, il faut les sécuriser au préalable. Pour les images en particulier, c'est là qu'entre en jeu la compressive sensing en tant qu'outils de cryptosystème. Afin de sécuriser davantage ces données, on combine la C.S à d'autres méthodes telles que la stéganographie.

### 2.1 Principe de la sécurisation d'images par C.S :

Le processus de cryptage d'image dans la C.S est appelé secure-compressive sensing. Il consiste en premier lieu à convertir l'image pour qu'elle devienne éparse, puis il faut générer la matrice de sensing qui va être la clé de décryptage. L'image dans son domaine de parcimonie constitue déjà une image compressée, cependant son produit avec la matrice sensing donne à la fois une image comprimée et chiffrée [1]. Le décryptage de l'image requiert la matrice sensing qui se comporte en quelque sorte comme la clé de l'image cryptée. En général, on peut résumer ce processus comme suit :

- Convertir l'image pour qu'elle soit éparse en utilisant des transformée telles que la Transformé en Cosinus Discrète (DCT), la Transformé de Fourier Discrète (DFT), ou la Transformé en ondelettes (DWT).

***Corresponding author Author & Copyright Author © 2022: | Tianandrasana Romeo Rajaonarison * |.** 

- Décider du rapport de compression CR ou Compressive Ratio pour l'image de taille $M \times N$ à compresser, on note :

$$m = CR \times M \quad (1)$$

- Générer la matrice de mesure $\Phi$ de dimension$m \times n$, cette matrice sera utilisée comme clé de cryptage.
- Effectuer la C.S de l'image en réalisant le produit des coefficients de l'image par celle de la matrice sensing

$$y_{m\times N} = \Phi_{m\times N} \times x_{N\times N} \quad (2)$$

- On obtient ensuite une image compressée et encodée qui est ensuite stockée ou partagée sur un réseau.

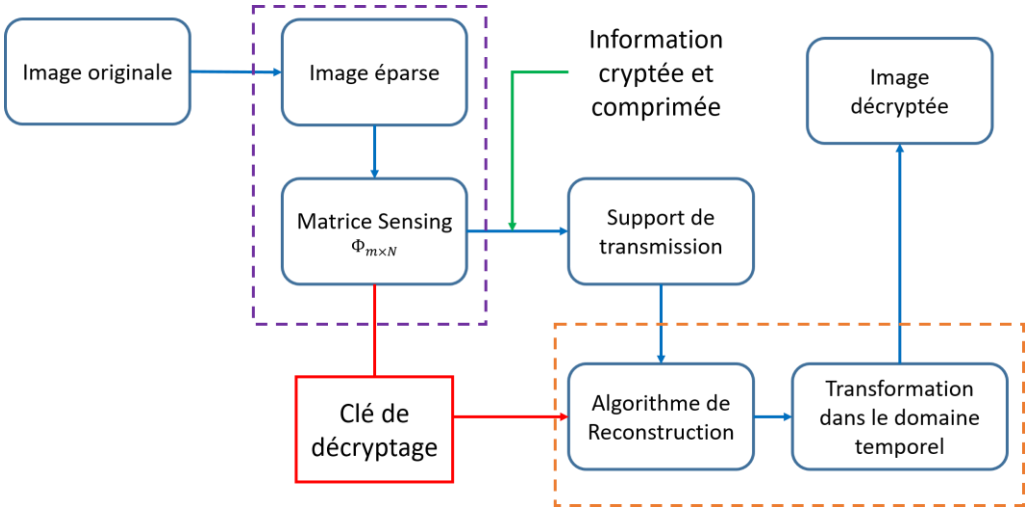



**Figure 1 :** la figure montre le principe de cryptage et de décryptage en utilisant la CS

Le processus de déchiffrement de l'image s'effectue grâce aux algorithmes de reconstruction ; pour ce faire il faut disposer de la matrice de sensing exacte et l'image compressée. Ces algorithmes reposent sur l'algèbre linéaire et des techniques d'optimisation. Si on désigne par $y$ l'image cryptée et $x'$ l'image obtenue après utilisation d'un algorithme de restitution qu'on note $AR$, les coefficients épars de l'image dans le domaine $\Psi$ représentent l'image décryptée par l'équation (3) [2,3] :

$$x' = AR(\Phi, y) \quad (3)$$

Où

- $x'$ est la matrice représentant les coefficients épars de l'image dans la base $\Psi$
- $AR$ dénote l'Algorithme de Reconstruction
- $\Phi$ est la matrice sensing
- $y$ est la matrice représentant l'image compressée et cryptée

## 2.2 Techniques de stéganographie d'images:

La stéganographie se définit comme l'art et la science d'écrire des messages cachés de telle sorte que seul le destinataire soit le seul à savoir l'existence de ce message. A la différence de la cryptographie où l'existence du message n'est pas déguisée mais obscurcie.

### 2.2.1 Stéganographie utilisant les LSB:

C'est une méthode relativement simple, mais pratique pour cacher des informations au sein d'une image. Elle utilise les bits de poids faible ou Least Significant Bits abrégés LSB de chaque pixel de l'image pour cacher les bits de poids d'une autre image. Si on choisit par exemple d'utiliser 4 bits pour cacher l'image secrète alors, il restera 4 bits pour l'image hôte [5,6].

**Tableau 1 :** Fonctionnement de la stéganographie Least Signifiant Bits.

| Image hôte | Image secrète | Image stéganographiée |
|---|---|---|
| **1010 0001** | **0011 1111** | 1010**0011** |
| **1111 0101** | **1001 1111** | 1111**1001** |

Pour restaurer l'image originale, il ne suffit que de connaître le nombre de bits utilisé pour entreposer l'image secrète dans l'hôte. Puis, on balaye l'image hôte pour prendre ses LSB correspondant aux nombres de bits utilisés. Ces bits extraits seront les bits de poids forts d'une image nouvellement construite.

### 2.2.2 Stéganographie utilisant la DCT:

Une autre méthode pour cacher une information au sein d'une autre image est l'utilisation de la DCT (Discrete Cosine Transform). Elle consiste à transformer l'image hôte dans le domaine DCT puis ajouter directement l'image secrète à cette image transformée. Puis, on effectue la transformée inverse de DCT pour obtenir l'image stéganographiée [5] [6].


**Figure 2 :** la figure montre la stéganographie utilisant la méthode Discrete Cosine Transform (DCT).

### 2.2.2 Stéganographie utilisant la DWT:

La stéganographie utilisant les ondelettes ou stéganographie DWT (Discrete Wavelet Transform) est la technique la plus performante car elle augmente la robustesse de l'information qui est dissimulée.
Cette technique fonctionne en prenant plusieurs ondelettes pour coder l'image entière. Par conséquent, l'image est hautement comprimée que les hautes fréquences formant les détails de l'image qui sont stockées séparément des basses fréquences. Dans cette technique, on altère les coefficients d'ondelettes de l'image hôte avec les coefficients d'ondelettes de l'autre image.

Pour éviter que les modifications de l'image ne soient visibles, seules les surfaces de basses fréquences de l'image sont altérées [5,6].


**Figure 3 :** la figure montre la stéganographie utilisant la méthode Discrete Wavelet Transform (DWT).

### 2.3 C.S combinée à la stéganographie:

Pour mieux appréhender le mode de fonctionnement de la compressive sensing combinée à la stéganographie nous allons faire une simulation sous MATLAB. Nous allons voir les différentes étapes mis en jeu pour avoir un système mieux sécurisé utilisant la compressive sensing et les résultats correspondants.

### 2.3.1 Génération des paramètres:

Afin de commencer, il faut générer les paramètres initiaux ; il faut choisir aléatoirement les positions des pixels $x_0$ et $y_0$ pour le zigzag. En effet, ces paramètres constituent la position initiale du zigzag. On peut générer leur valeur en utilisant la fonction MATLAB $randi$, cette fonction va générer une valeur entre la position du premier au dernier pixel. Ainsi le zigzag peut débuter à partir de cette valeur obtenue.

### 2.3.2 Algorithme de chiffrement:

Après avoir générés les paramètres initiaux $x_0$, $y_0$ on peut commencer la procédure de cryptage de l'image.
Premièrement, il faut choisir l'image de taille $M \times N$ que l'on veut transmettre. Par la suite, les coefficients de l'image sont dispersés en appliquant une confusion en zigzag sur ses coefficients à partir des coordonnées $x_0$ et $y_0$ générées précédemment. En faisant ainsi, on altère la position des éléments dans l'image pour les repositionner dans une nouvelle matrice de taille identique. Cette étape constitue d'ores et déjà un chiffrement de l'image.

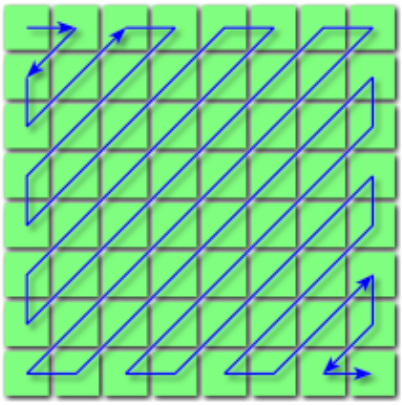


**Figure 4 :** la figure montre les igzag de l'image.

Ensuite, il faut appliquer la DWT sur l'image zigzaguée pour obtenir une nouvelle image de même taille que l'image originale mais constitue la représentation éparse de cette dernière.

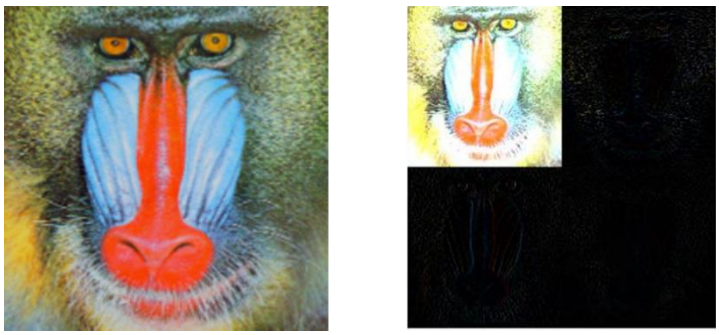


**Figure 5 :** la figure montre l'image transformée.

On se pose une limite dite valeur seuil à partir de laquelle on va sélectionner les coefficients. En effet les valeurs qui seront inférieures à ce seuil seront mises à zéro. Ce processus va augmenter la parcimonie sans pour autant affecter sa qualité. L'étape suivante consiste en génération de la matrice sensing $\Phi$ de dimension $\times N$ . Après le choix du facteur de compression $CR$ tel que $m = CR \times N$, une des solutions les plus simples mais efficace consiste à générer une matrice de distribution gaussienne unitaire c'est-à-dire la norme de chaque colonne est 1. On peut utiliser également une matrice partielle de Fourier ou une matrice de Bernoulli. Après, il ne reste plus qu'à effectuer la C.S sur l'image en effectuant son produit avec la matrice sensing pour obtenir l'image compressée.

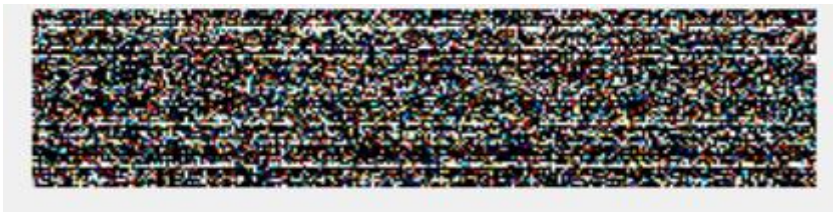


**Figure 6 :** la figure montre l image compressée.

L'image compressée est ensuite quantifiée afin que ces éléments aient une valeur entre 0 et 255. On obtient les coefficients cette matrice quantifiée en procédant de la sorte :

$$y_1(i,j) = floor\left(255 \times \frac{y(i,j) - min}{max - min}\right) \quad (4)$$

Où :

- $y_1(i,j)$ coefficient de l'image quantifié aux positions $i$ et $j$
- $y(i,j)$ sont les coefficients de l'image à quantifier aux coordonnées $i$ et $j$
- $floor(x)$ se charge de trouver l'entier le plus grand entier inférieur à $x$

- $min$ et $max$ sont les valeurs minimales et maximales de $y$

**Figure 7 :** la figure montre l'image après quantification.

La dernière étape consiste à choisir une image porteuse à laquelle on va insérer cette image récemment quantifiée. Pour ce faire, il faut appliquer la stéganographie par DWT sur l'hôte ou l'image porteuse et ajouter à ces composantes d'ondelette de basse fréquence de l'image cryptée et compressée.

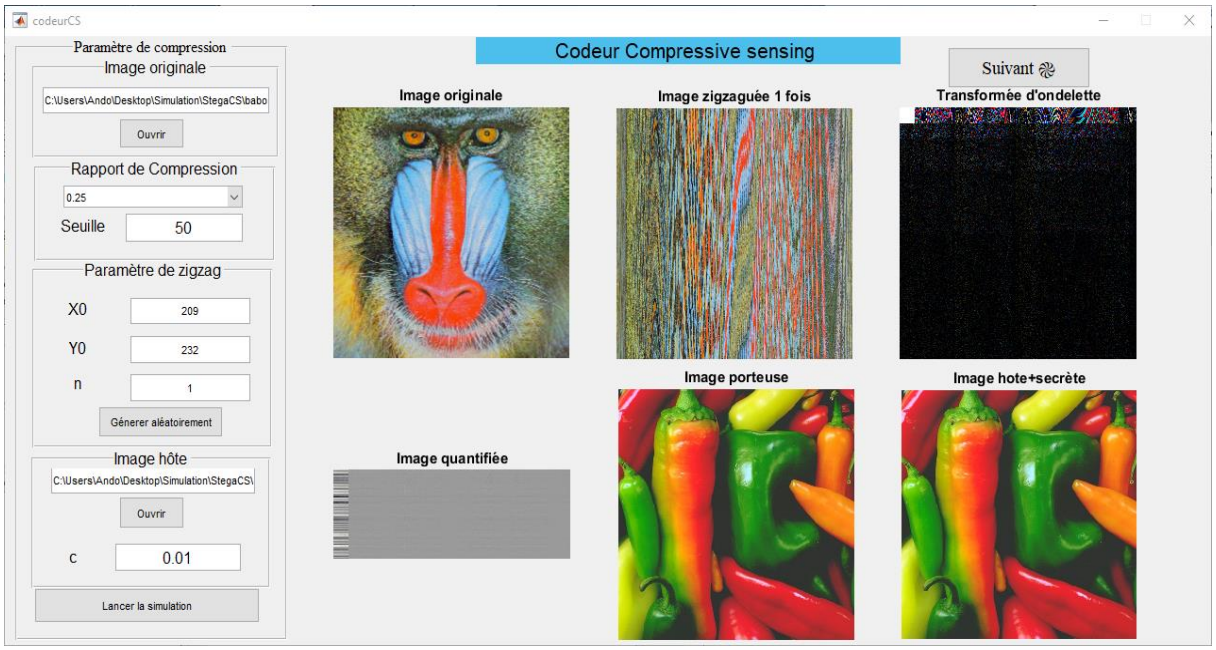



**Figure 8 :** la figure montre le processus de chiffrement.

### 2.3.3 Algorithme de décryptage

A la réception on obtient tout d'abord l'image quantifiée. Il faut effectuer l'inverse de la quantification sur cette image, ce qui nous donne l'image compressée. L'inverse de la quantification peut être obtenu par l'expression (5) :

$$y(i,j) = \frac{y_1(i,j) \times (max - min)}{255} + min \qquad \textbf{(5)}$$

Où :

- $y_1(i,j)$ est la valeur du pixel de l'image quantifiée aux positions $i$ et $j$.
- $y(i,j)$ est la valeur du pixel de l'image après quantification inverse aux positions $i$ et $j$.
- $max$ valeur maximale contenue dans l'image avant quantification.
- $min$ valeur minimale contenue dans l'image avant quantification.

Avec la matrice sensing envoyée secrètement au récepteur, la restitution de l'image décompressée peut être entamée en utilisant un algorithme de reconstruction adéquat pour obtenir l'image éparse dans le domaine des ondelettes.
La transformée d'ondelette inverse est ensuite effectuée pour obtenir l'image zigzaguée. Finalement, on effectue le zigzag inverse à partir des positions $x_0$ et $y_0$ pour décrypter l'image et pour obtenir l'image finale.

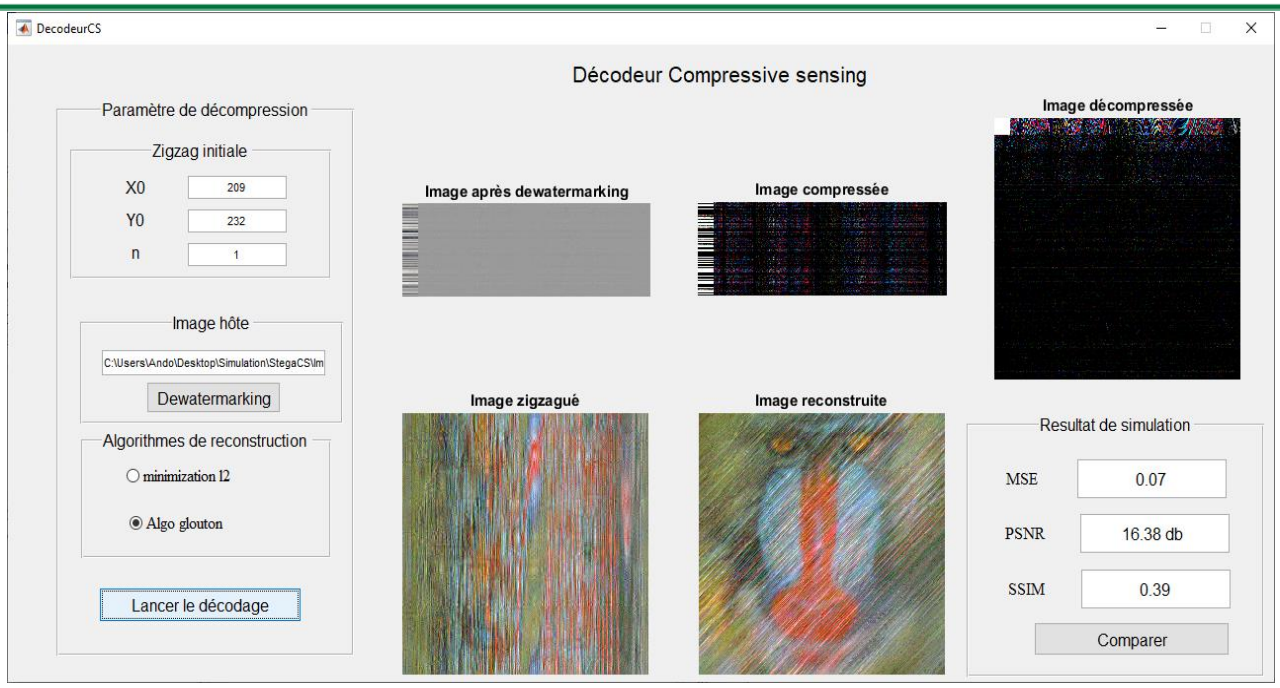



**Figure 9 :** Processus de déchiffement.

## 2.3 Mesure de la qualité d'image:

Après une telle transformation de l'image, il est plus que nécessaire de mesurer la déformation subite par l'image comparée à l'image originale. Ces mesures donnent une idée sur la qualité et la performance de la méthode de cryptage et de décryptage [3,4].

### 2.3.1 Mean Square Error

Pour obtenir la MSE ou Mean Square Error d'une image, il faut tout d'abord trouver l'erreur présente entre l'image originale et celle décryptée.

Cette erreur se définit comme la différence entre chaque valeur de pixel da
ns l'image de référence avec chaque élément de l'image obtenue. Pour une image de taille $M \times N$, on a :

$$erreur[M,N] = I_{ref}[M,N] - I_{CS}[M,N] \quad (6)$$

Où :

- $\text{erreur}$ est une matrice de taille $M \times N$ contenant l'erreur,
- $I_{ref}$ est l'image de référence,
- $I_{CS}$ est l'image obtenu après transformation,

La MSE va donner l'erreur absolue présente entre les 2 images et se calcule de la manière suivante :

$$MSE = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} (erreur)^2 \quad (7)$$

### 2.3.2 Peak Signal to Noise Ratio

En général, la MSE ne considère pas la variation dynamique de l'image. On fait recours au PSNR ou Peak Signal to Noise Ration pour trouver le rapport entre la puissance maximale de l'image de référence et la puissance du bruit de l'image traitée.

En général, il est représenté dans une échelle logarithmique en décibels parce que les images peuvent avoir un large intervalle de dynamique. Le PSNR se calcule de la manière suivante :

$$PSNR = 10\, log\left(\frac{(2^n - 1)^2}{MSE}\right) \quad (8)$$

Où $n$ représente la dynamique de l'image, c'est-à-dire le nombre de bits nécessaire pour représenter un pixel de l'image. Pour une image codée sur 8 bits :

$$PSNR = 10\,log\left(\frac{255^2}{MSE}\right) \quad \textbf{(9)}$$

Pour une image en couleur constituée de 3 plans de niveau de gris, le PSNR s'obtient en calculant une par une chaque plan de niveau de gris, puis le PSNR de l'image s'obtient en moyennant chacun des PSNR obtenus sur les 3 plans.

$$PSNR = \frac{\left(PSNR_{Rouge} + PSNR_{Vert} + PSNR_{Bleu}\right)}{3} \quad \textbf{(10)}$$

### 2.3.3 Structure Similarity

L'indice SSIM ou Structure Similarity est une méthode de mesure de la similarité entre 2 images. Il constitue une mesure complète se basant sur l'image initiale comme référence.
Il est développé pour mesurer la qualité visuelle d'une image qui a subi une transformation par rapport à l'originale. L'idée fondamentale est de mesurer la similarité de structure plutôt qu'une différence pixel à pixel comme pour le PSNR.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_x\sigma_y + c_2)(cov_{xy} + c_3)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)(\sigma_x\sigma_y + c_3)} \quad \textbf{(11)}$$

Où :

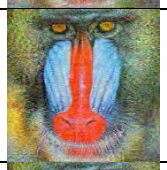
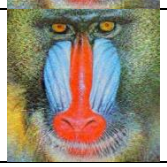
- $\mu_x$ est la moyenne de $x$
- $\mu_y$ est la moyenne de $y$
- $\sigma_x^2$ est la variance de $x$
- $\sigma_y^2$ est la variance de $y$
- $cov_{xy}$ est la covariance de $x$ et $y$
- $c_1$, $c_2$ et $c_3$ sont des constantes de stabilisation

## 3. RESULTATS

Dans le cadre de cette simulation, nous allons utiliser une image en couleur dont la dimension est de $256 \times 256$. Nous allons comparer une à une la performance des algorithmes gloutons face à la minimisation $l_2$. Nous allons utiliser un tour de zigzag afin de réduire la corrélation existant entre les pixels voisins dans l'image. Nous fixons la valeur du seuillage à 50 pour chaque simulation faite.
Nous avons récapitulé dans le tableau 2 les résultats des simulations utilisant différents algorithmes de reconstruction. Pour l'utilisation de l'OMP (Orthogonal Matching Pursuit) nous obtenons :

**Tableau 2 :** Tableau présente les résultats de l'utilisation de l' Orthogonal Matching Pursuit.

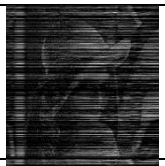
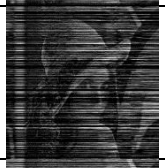
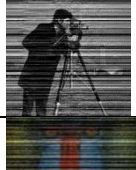
| Image originale | Taux de compression | Image reconstruite | MSE | PSNR | SSIM |
|---|---|---|---|---|---|
|  | 0.25 |  | 1.45 | 22.46 | 0.38 |
| | 0.5 |  | 0.61 | 26.20 | 0.53 |
| | 0.75 |  | 0.14 | 32.57 | 0.75 |
|  | 0.25 |  | 2.25 | 20.55 | 0.25 |

| | | | | | |
|---|---|---|---|---|---|
| | 0.5 |  | 1.06 | 23.82 | 0.36 |
| | 0.75 |  | 0.28 | 29.58 | 0.54 |
|  | 0.25 |  | 9.60 | 19.07 | 0.59 |
| | 0.5 |  | 6.68 | 20.66 | 0.68 |
| | 0.75 |  | 2.93 | 24.24 | 0.82 |

**MSE : Mean Square Error ; PSNR : Peak Signal to Noise Ratio ; SSIM : Structural SIMilarity.**

Dans le tableau 3, nous utilisons la méthode du pseudo-inverse, ou minimisation $l_2$ et nous avons obtenu les résultats suivants:

**Tableau 3 :** Le tableau résultats de l'utilisation de la minimisation $l_2$.

| Image originale | Taux de compression | Image reconstruite | MSE | PSNR | SSIM |
|---|---|---|---|---|---|
|  | 0,25 |  | 31,66 | 9,07 | 0,04 |
| | 0,5 |  | 26,55 | 9,84 | 0,06 |
| | 0,75 |  | 14,57 | 12,44 | 0,15 |
|  | 0,25 |  | 45,26 | 7,52 | 0,02 |
| | 0,5 |  | 36,25 | 8,48 | 0,05 |
| | 0,75 |  | 19,84 | 19,84 | 0,11 |
|  | 0,25 |  | 13,72 | 7,52 | 0,11 |

| | | | | | |
|---|---|---|---|---|---|
| | 0,5 |  | 10,09 | 8,84 | 0,20 |
| | 0,75 |  | 6,94 | 11,35 | 0,34 |

## 4. DISCUSSION

Au premier abord, on constate qu'il est possible grâce à la compressive sensing de réaliser un taux de compression plus élevé allant jusqu'au quart de l'image originale. Ensuite, pour chaque rapport de compression choisi, on remarque que l'OMP procure un taux d'erreur plus bas, un PSNR plus grand et SSIM plus proche de 1 que lors de l'utilisation de la technique de minimisation de la norme $l_2$. En effet la minimisation $l_2$, si bien qu'elle soit rapide n'est pas optimale pour la recherche de la solution la plus éparse. L'OMP à l'opposée exploite au mieux la parcimonie pour reconstruire l'image d'où son efficacité comparée à la méthode de pseudo-inverse. Pour chaque méthode utilisée, on constate également une augmentation du PSNR, et du SSIM au fur et à mesure que le nombre de mesures augmente. Toutefois, on peut déjà obtenir un résultat proche de l'image d'origine en utilisant l'OMP à partir d'une compression au quart.

Les résultats du chiffrement et du déchiffrement sont donnés par les figures 8 et 9. On a bien remarqué que l'image chiffrée est incompréhensible et le système permet bien de reconstituer l'image originale.

En définitive, nous avons vu la possibilité de sécuriser efficacement les images grâce à l'utilisation de la C.S et de la stéganographie. Grâce à la simulation, on a pu voir l'efficacité de cette méthode pour la sécurisation des données. En outre, la C.S offre un taux de compression élevé tout en promettant une restitution quasi totale de l'image compressée.

## 5. CONCLUSION

En conclusion, la compressive sensing est une technique qui exploite la parcimonie des signaux dans le but de faciliter sa compression et sa transmission. Elle ne se limite pas uniquement à la compression des données mais elle permet aussi de chiffrer les images. En dépit de sa simplicité, elle redéfinit les principes de base de la numérisation et offre un tout nouveau visage aux méthodes d'acquisition. Combinée à la stéganographie, elle offre une sécurisation supplémentaire aux images.

Une perspective serait de trouver des algorithmes de reconstruction plus performants et plus efficaces pour la décompression.

## 6. REFERENCES

1. Rohit M. THankI. Multibiometric Watermarking with Compressive Sensing Theory *Springer*, 2015.
2. Chi Wah Kok et Win Sham Tam. Digital Image interpolation, CRC Press, 2014.
3. Vishal M.Patel et Rama Chellapa. Sparse Representation and Compressive sensing for imaging and vision, *Springer*, 2013.
4. Yonina C.Eldar, Gitta Kutinyok. Compressed Sensing Theory and Applications, Cambridge University,2012.
5. Angshul Majumdar, Compressed sensing for engineers, CRC Press, 2018.
6. Sendil AB Image Steganography, https://www.mygreatlearning.com/blog/image-steganography-explained/
7. https://www.geeksforgeeks.org/image-steganography-in-cryptography/ Image Steganography in cryptographie

**Cite this article: Tianandrasana Romeo Rajaonarison and Paul Auguste Randriamitantsoa.** ETUDE DE CHIFFREMENT D'IMAGES UTILISANT LA STEGANOGRAPHIE PAR DWT ET LA COMPRESSIVE SENSING**.** *Am. J. innov. res. appl. sci. 2022; 14(5): 179-*187.