Une méthode de compression et de cryptage simultanés des images multiples : principe et implantation PFGA

M. JRIDI et A. ALFALOU

ISEN-Brest/L@bISEN/VISION, ISEN CS 42807

20 rue Cuirassé Bretagne, 29228 Brest Cedex 2
{maher.jridi, ayman.al-falou}@isen.fr

Résumé

Nous présentons ici une nouvelle méthode de fusion spectral d'images multiples basée sur la transformée en cosinus discret (DCT) et une méthode de filtrage spécifique. Pour diminuer la taille des données multiplexées, nous proposons une méthode de compression basée sur une quantification adaptative. Chaque fréquence est codée avec un nombre optimisé de bits par rapport à sa position dans le domaine DC. Cette compression constitue un premier niveau de cryptage. Un second niveau est réalisé en utilisant des informations biométriques. Pour évaluer les performances de la méthode proposée nous calculons le MSE et le PSNR. Les résultats obtenus sont meilleurs par rapport au standard JPEG. Une deuxième partie de ce travail consiste à la proposition d'architecture matérielle afin de pouvoir réaliser des traitements embarqués. Une réalisation directe (par opposition à la séparation ligne/colonne) de la DCT en 2D est proposée. Les résultats de synthèse montrent que le triplet rapidité, énergie et consommation de ressources matérielles est optimisé.

Mots clefs

Compression et cryptage simultanés, DCT, implantation FPGA.

1 Introduction

De nos jours, les besoins en compression et en cryptage sont de plus en plus grandissants. La réalisation de ces deux opérations en cascade pose un certain nombre de problèmes. En effet, compresser des données et les crypter par la suite ou vice versa, peut réduire le taux de compression ainsi que la robustesse du cryptage. Pour résoudre ce problème, une nouvelle méthode de compression et de cryptage simultanés des images a été proposée dans [1]. Cette méthode est basée sur le multiplexage fréquentiel des images cibles [1-4]. Une première version d'un schéma multiplexant quatre images est proposée et validée dans notre laboratoire [1] en utilisant les propriétés de la DCT (transformation en cosinus discrète).

Cependant, le spectre fusionné des quartes images, ne permet pas une compression réelle des images à transmettre et/ou à stocker car la dynamique du spectre est très grande et nécessite un nombre très élevé de bits pour les coder. Pour résoudre cette limitation, nous proposons une méthode avec trois optimisations (i) permettre d'augmenter le nombre d'images cibles en entrée du système, (ii) réaliser une compression réelle des informations fusionnées dans le domaine spectral et (iii) développer une implantation matérielle tout numérique sur des cibles programmables de types FPGA. Pour y arriver, nous avons utilisé et adapté la technique développée par Naughton et al dans [4] pour permettre le codage du fichier multiplexé à transmettre et/ou à stocker sur un nombre de bits bien défini. Finalement, une optimisation du cryptage est proposée en se basant sur le cryptage DRP [2] (Double Random Phase encryption system) pour rendre le système plus résistant et plus robuste contre les tentatives de piratage du système.

2 Multiplexage spectral des images

La méthode de multiplexage développée dans cet article consiste à fusionner les informations utiles issues de différentes images cibles dans le domaine fréquentiel. Pour passer dans le domaine fréquentiel, nous appliquons séparément la transformation en Cosinus Discrète (DCT) à chacune des images conformément à la Figure 1. Le choix de la DCT est motivé par :

(1) L'utilisation de la DCT dans des standards tels que JPEG et MPEG et sa capacité de regrouper les informations utiles, pour la reconstruction d'une image, dans le coin gauche haut de son plan fréquentiel.
(2) La proposition, dans un travail antérieur, d'un algorithme optimisé implantant la DCT sur FPGA [5-7].

En utilisant la propriété de regroupement des informations utiles dans les fréquences hautes gauches, nous multiplions chacun des plans fréquentiels avec un filtre passe-bas (Figure 1). A l'issue de ce filtrage, les informations fréquentielles seront réduites selon la taille du filtre considéré. La taille du filtre, exprimée en pixels,

dépend de la taille des images cibles en pixels et du nombre d'images considérées, elle est donnée par

$$(Fs_x, Fs_y) = \left(\frac{Is_x}{N}, \frac{Is_y}{N} \right) \quad (1)$$

Le couple (Fs_x, Fs_y) définit la taille du filtre suivant x et suivant y, (Is_x, Is_y) la taille en pixels de l'image cible et N le nombre des images cibles considérées en entrée du système. Ainsi en utilisant l'équation (1) la taille du plan fréquentiel est réduite par un rapport N . Ensuite chacune des parties filtrées subit une rotation et un décalage bien définies, de sorte à ce que chaque 4 DCTs forment un groupement comme cela est montré en sortie de la Figure 1. Le fait de regrouper les DCTs sous cette forme constitue un premier niveau de cryptage. En effet, le résultat de ce regroupement ressemble à un spectre obtenu avec une TF (transformation de Fourier). De plus les images se chevaucheront en sortie après transformation.
Cependant, pour coder le plan de sortie multiplexant les différentes DCTs, nous avons besoin d'utiliser un nombre de bits très élevé i.e. cela peut constituer une augmentation de la taille de fichier à stocker et/ou à transmettre. En effet, les amplitudes obtenues avec la DCT ont une dynamique très élevée nécessitant un nombre de bits élevé. Pour résoudre ce problème, nous proposons de coder les différentes fréquences avec un nombre de bits réduit et optimal.

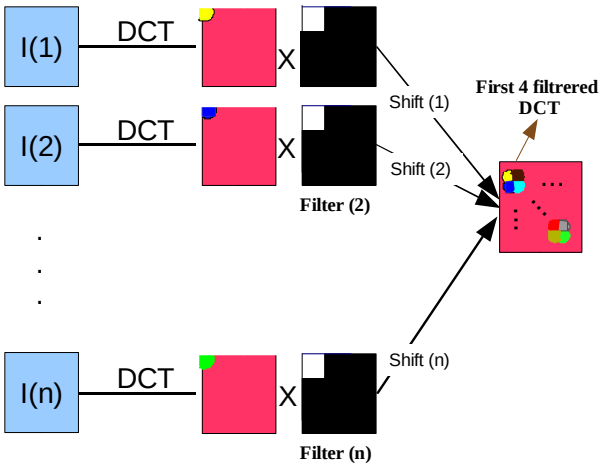

Figure 1 - *Diagramme synoptique représentant la méthode de multiplexage des images multiples, n étant le nombre des images cibles. La rotation, le décalage et le regroupement 4 par 4 constitue un premier niveau de cryptage.*

3 Technique de compression

La méthode de compression développée permet de réduire au maximum la taille en pixel de chacune des DCTs filtrées comme indiqué dans la Figure 2. Par conséquent, nous divisons la partie filtrée de chacune des DCTs en plusieurs zones, en commencent par le point haut gauche (Figure 2-d). Ici, nous avons choisi des zones de largeur égale à « 2 » pixels comme cela est montré sur la Figure 2-d i.e. chacune des couleurs représente une zone différente.

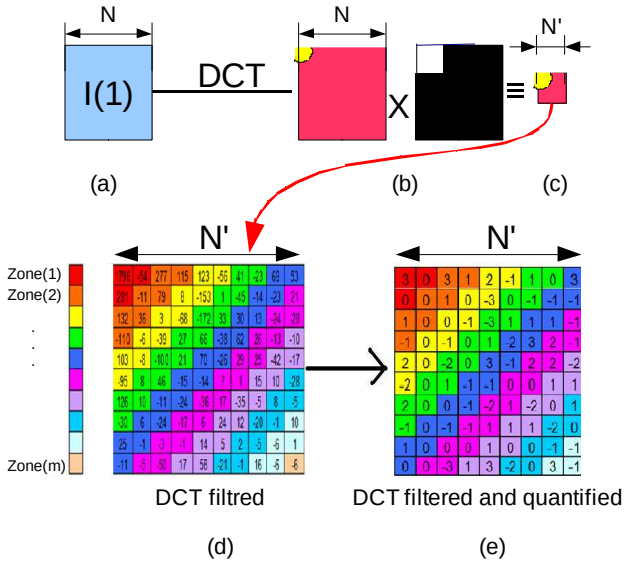

Figure 2 - *Principe de la compression : (a) image cible, (b) DCT et le filtre passe-bas, (c-d) exemple d'une DCT filtrée repartie en zones, (d) la DCT filtrée et quantifiée.*

Ensuite, nous quantifions chacune des valeurs des différentes zones considérées (Figure 2-e) en utilisant la formule suivante :

$$V'_{zone_l}(i,j) = round\left(\frac{\left(2^{m-1}-1\right) \times V_{zone_l}(i,j)}{MAX\left(V_{zone_l}\right)} \right) \quad (2)$$

$V_{zone_l}(i,j)$ et $V'_{zone_l}(i,j)$ définies les valeurs DCT avant et après quantification de la zone (l), (i,j) sont les coordonnées considérées, $MAX\left(V_{zone_l}\right)$ désigne la valeur maximale pour la zone (l), *m* est le nombre de bits considéré pour coder les valeurs de la zone (l) et *round* définie la partie entière. La Figure 2-e présente un exemple quantifié d'une DCT filtrée (N', N') pixels d'une image (N, N) pixels.

Le taux de compression utilisé dans le reste de cet article est défini par

$$T_c = \left(1 - \frac{nb_{out}}{nb_{in}}\right) \times 100 \quad (3)$$

T_c est le taux de compression, nb_{in} est la taille en bits des images multiples considérées en entrée et nb_{out} désigne la taille de l'image compressée. Pour optimiser le taux de compression nous pouvons utiliser un nombre de bits adapté à chacune des zones (Figure 2-e). Ce type de quantification répartie en zones constitue un deuxième niveau de cryptage.

4 Validation de la technique de compression et du cryptage simultanés

Pour crypter ainsi les images comprimées par la méthode de multiplexage –DCT proposée dans Figures 1 et 2, nous proposons une nouvelle méthode de cryptage. Cette dernière est basée sur l'utilisation de deux clés de cryptage aléatoires et d'une image biométrique (une empreinte digitale). La technique du multiplexage (Figure 1) ainsi que la quantification spectacle développé dans Figure 2 sont considérées comme deux clés supplémentaires de cryptage. Le principe de la méthode de cryptage réel proposée est présenté sur le digramme synoptique Figure. 3.

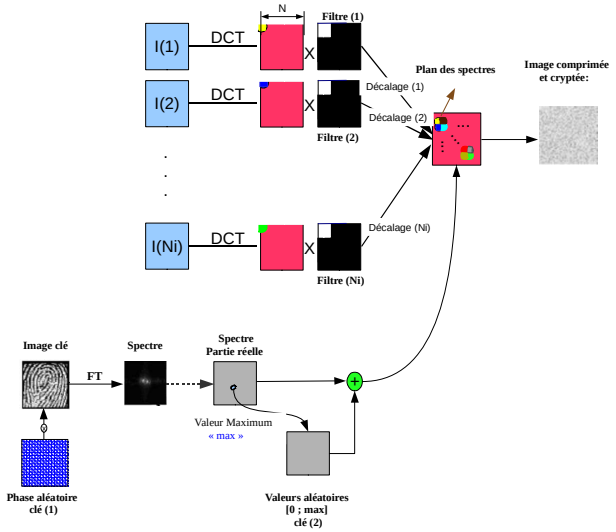

Figure 3 - *Schéma synoptique de cryptage optimisé réel.*

La première étape consiste à réaliser la transformation de Fourier d'une image biométrique (empreinte digitale) multipliée avec un premier masque de phase aléatoire. Ensuite, nous sélectionnons seulement la partie réelle de ce spectre. Une fois la valeur *max* est déterminée, nous réalisons une image aléatoire des valeurs réelles comprises entre [0 et max]. Ensuite, nous additionnons la partie réelle du spectre de l'empreinte avec cette deuxième clé d'amplitude aléatoire. Finalement, le plan fréquentiel est crypté avec le résultat, pour obtenir en sortie un plan à la fois comprimé et crypté. Comme cela est présenté en sotie de la Figure 4. Pour mesurer le taux de cryptage de notre méthode, nous avons calculé le PSNR entre l'image cible et l'image décryptée sans connaitre les clés i.e. IDCT du résultat de compression et cryptage. Pour cela nous avons considéré une séquence de 4 images couleurs présentés Figure 5-a. Après avoir décomposé les images couleurs en utilisant la base RGB, nous obtenons 12 images. Ces 12 images sont introduites dans notre système de compression et de cryptage. En sortie nous obtenons l'image comprimée et cryptée présenté Figure 5-b. Ainsi, la valeur moyenne du PSNR obtenue avec cette dernière image (Figure5-b) est égale à 4,2 dB. Cette valeur PSNR très faible montre bien le bon comportement de notre méthode de cryptage.

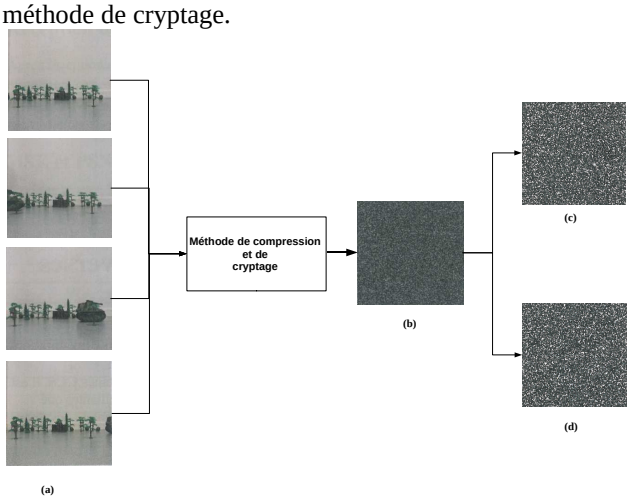

Figure 4 - *Résultats de compression et de cryptage de 4 images couleurs à l'entrée du système de compression et de cryptage simultané : (a) les 4 images couleurs utilisées, (b) l'image comprimée et cryptée, (c) une image piratée en réalisant une DCT de l'image et en utilisant plusieurs clés aléatoires de décryptage et (d) une image piratée avec une FFT.*

Pour valider la robustesse de notre méthode de cryptage, nous avons appliqué deux attaques brutes. La première consiste à réaliser une DCT de l'image, le résultat de cette attaque est présenté dans la Figure 4-c. La deuxième attaque consiste à relaisser une transformation de Fourier de l'image, le résultat est présenté (Figure 4-d). Les images piratés, Figures 4-c et Figures 4-d, montrent bien que le pirate ne trouve pas l'image en claire, de plus les valeurs des PSNR restent très élevés (à hauteur de 48 dB) même en essayant de multiplier l'image de la Figures 4-b par une clé aléatoire de sorte à supprimer la valeur aléatoire greffée à limage en sortie. En effet, le PSNR entre deux images doit être au dessus de 15 dB pour que l'image reconstituée soit visible et déchiffrable. Les tests réalisés dans notre laboratoire montrent bien que quelques soit le nombre d'itération, nous avons un PSNR < 5 dB. Ainsi, la méthode de cryptage s'avère être très performante.

5 Implantation matérielle

L'algorithme de calcul utilisé dans la méthode proposée est la DCT. Afin d'employer cette méthode dans un système portable, une implantation matérielle optimisée sur FPGA est nécessaire. La partie de cryptage peut être générée en différé. Dans ce travail, nous nous occupons que de la partie de compression et nous optons pour la réalisation directe de la DCT, par opposition à la

réalisation à base de la séparation ligne/colonne. Ainsi, nous pouvons nous affranchir de l'utilisation d'une mémoire de transposition.

Pour cette réalisation directe, nous nous sommes basés sur les travaux de Hallapuro dans [8] pour l'étendre à une DCT de taille 8x8 voire des tailles bien supérieures. Ici, il faut signaler que les méthodes de calculs de la DCT direct en 2D sont rares malgré que cela offre une alternative intéressante pour optimiser le triplet rapidité, consommation d'énergie et consommation de ressources matérielles. En effet, pour un bloc X de pixel de taille 8x8, la représentation matricielle des coefficients Y de la DCT peut s'écrire sous cette forme :

$$Y = COS \times X \times COS^t \tag{3}$$

La matrice COS est une matrice 8x8 définie par

$$COS = \begin{pmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & -c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & -a & -c & e & -g \end{pmatrix} \tag{4}$$

Les différentes constantes de la matrice COS peuvent être trouvées dans [9].

Comme dans [8], nous pouvons calculer Y par :

$$Y = B \times C \times X \times C^t \times B^t = Y_{std} \otimes E \tag{5}$$

Ainsi, nous pouvons calculer les matrices de tailles 8x8 C, B et E. Ces dernières sont définies par :

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & c/a & e/a & g/a & -g/a & -e/a & -c/a & -1 \\ 1 & f/b & -f/b & -1 & -1 & -f/b & f/b & 1 \\ 1 & -g/c & -a/c & -e/c & e/c & a/c & g/c & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -a/e & g/e & c/e & -c/e & -g/e & a/e & -1 \\ 1 & -b/f & b/f & -1 & -1 & b/f & -b/f & 1 \\ 1 & -e/g & c/g & -a/g & a/g & -c/g & e/g & -1 \end{pmatrix} \tag{6}$$

$$B = \begin{pmatrix} d & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & f & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & g \end{pmatrix} \quad E = \begin{pmatrix} d^2 & da & db & dc & d^2 & de & df & dg \\ ad & a^2 & ab & ac & ad & ae & af & ag \\ bd & ba & b^2 & bc & bd & be & bf & bg \\ cd & ca & cb & c^2 & cd & ce & cf & cg \\ d^2 & da & db & dc & d^2 & de & df & dg \\ ed & ea & eb & ec & ed & e^2 & ef & eg \\ fd & fa & fb & fc & fd & fe & f^2 & fg \\ gd & ga & gb & gc & gd & ge & gf & g^2 \end{pmatrix} \tag{7}$$

Pour obtenir une architecture optimisée d'un point de vue temps de calcul et ressources occupées, nous calculons Y_{std} au lieu de Y et nous incluons la matrice E dans la matrice de quantification/déquantification. De plus, il faut signaler que les matrices B et C sont orthogonales et le calcul de la DCT inverse se fera avec les mêmes matrices, sous la forme simple ou transposée.

Ainsi, la représentation matricielle proposée permet le calcul de la DCT avec un nombre réduit de multiplieurs. Les résultats de synthèse de l'architecture proposée pour un bloc d'entrée de taille 4*4 sont résumés dans le tableau Tab. 1 et montrent les bonnes performances du circuit proposé.

La description structurelle de l'architecture matérielle implantée est donnée dans la Figure 5.

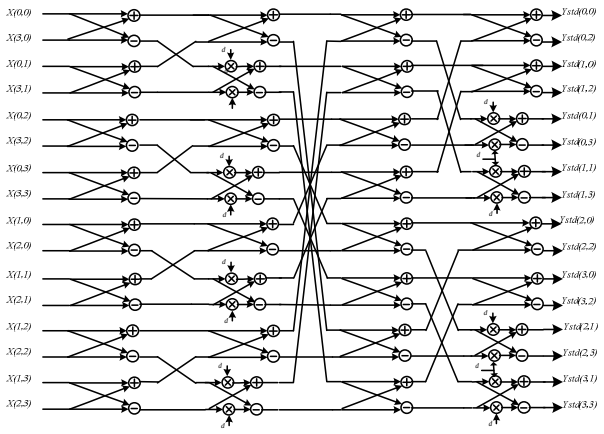

Figure 5 - *Description structurelle de l'architecture proposée pour une taille de 4x4.*

Cette description montre que l'architecture proposée présente un flot de données symétrique. De plus, une description équivalente de la DCT proposée pour une taille de bloc de 8x8 a été élaborée (elle n'est pas présentée ici pour ne pas encombrer le document). Des DCT de tailles supérieures peuvent être obtenues par la même méthode de décomposition matricielle. En effet, il est intéressant pour la méthode de compression et de cryptage simultanés de proposer des DCT de tailles élevées. La raison est que le spectre de toutes les images multiplexées est stocké dans une matrice dont les tailles correspondent à la taille de la DCT. Ainsi, pour pouvoir augmenter le nombre d'images à l'entrée du système (donc augmenter le rapport de compression et la robustesse du cryptage), il faut augmenter la taille de la DCT. Par exemple, une DCT de taille 16x16 est suffisante pour multiplexer le spectre de 16 images compressées tout en offrant un bon rapport de compression (89%) et une image reconstruite de bonne qualité.

De plus, plusieurs schémas de compression ont été simulés pour comparer le PSNR entre l'image d'origine et l'image compressée de plusieurs formes de DCT (théorique, par multiplication matricielle, proposée) et en variant à la fois l'arithmétique utilisée (point fixe et point flottant) et la taille des blocks des pixels employés (4*4,

8*8) et le pas de quantification. Ces différentes simulations sont reportées sur la Figure 6 et confirment que la DCT proposée n'induit pas une perte en PSNR (ou induit une faible perte), mais comme indiqué dans le Tab 1, elle présente un gain en surface silicium et en temps d'exécution.

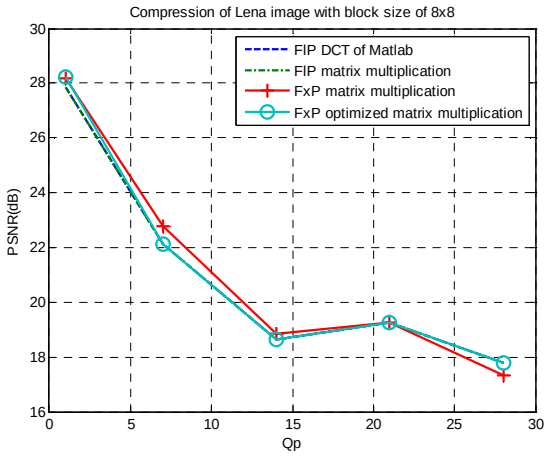

Figure 6 - *Compression de l'image Lena avec des tailles de blocs de 8x8. Utilisation de plusieurs DCT : Exact DCT de Matlab, DCT à base de multiplication matricielle en flottant et en point fixe et DCT proposée en point fixe.*

6 Conclusion

Pour résumer, nous avons investigué le problème qui consiste à trouver une manière optimale pour multiplexer le spectre de plusieurs images ainsi que l'implantation d'une méthode de compression et de cryptage simultanés. Par rapport à la méthode expliquée dans ce manuscrit, les résultats trouvés dépassent celles qui peuvent être obtenus avec le standard JPEG. De plus, la méthode proposée offre deux niveaux de cryptage : le premier est le résultat de la fusion des spectres et de la quantification adaptative, le second vient du fait que nous utilisons des clés biométriques pour crypter le spectre fusionné. Un autre aspect de ce travail a consisté à proposer une réalisation directe de la DCT implantée sur FPGA. Une des perspectives de ce travail est autour de la proposition d'architecture matérielle pour réaliser la compression et le cryptage en temps-réel sur FPGA.

Tab 1 : Résultats de synthèse sur FPGA Xilinx Spartan3E XC3S500

	Séparation ligne/colonne		Réalisation directe	
	Algorithme de Chen	Algorithme de Loeffler	Théorique	Proposée
Multiplications	40	24	128	16
Additions	72	72	96	64
Latence relative	$5\times(2T_A+T_M)$	$5\times(2T_A+T_M)$	$4T_A+2T_M$	$4T_A+2T_M$
Latence absolue	15	15	6	6
Slice LUT	502	341	1260	431
Fréquence(MHz)	126	119	138	203
Temps d'exécutions (µs)	50,2	34,1	75,6	10,6
Puissance consommée	-	37,21	-	17,29

Références

[1] A. Alfalou and C. Brosseau, "Optical image compression and encryption methods," Adv. Opt. Photon. 1, 589 (2009).

[2] B. Javidi, ed. *Optical and Digital Techniques for Information Security*, (Springer Verlag, New York, 2005).

[3] Philippe Refregier and Bahram Javidi, "Optical image encryption based on input plane and Fourier plane random encoding," Opt. Lett. 20, 767-769 (1995)

[4] Y. Frauel, A. Castro, T. J. Naughton, and B. Javidi, "Resistance of the double random phase encryption against various attacks," Opt. Express 15, 10253 (2007).

[5] Maher Jridi, Pramod Kumar Meher and Ayman AlFalou, "Zero Quantized DCT Coefficients Prediction Technique for Intra-Frame Video Encoding," *IET image processing*, Vol. 7, Iss. 2, pp. 165–173 (2013).

[6] Maher Jridi, Ayman AlFalou, and Pramod Kumar Meher, "Optimized Architecture using a Novel Subexpression Elimination on Loeffler Algorithm for DCT-Based Image Compression," *VLSI Design Journal*, Vol. 2012, pp. 1-9, (2012). (http://www.hindawi.com/journals/vlsi/aip)

[7] Maher Jridi and Ayman AlFalou, " Joint Optimization of Low-power DCT Architecture and

Effcient Quantization Technique for Embedded Image Compression", Book Chapter in *VLSI-SoC: Forward-Looking Trends in IC and System Design*, J. Ayala A. Alonso and R. Reis (Eds.) IFIP WG 10.5, Springer Verlag, ISBN 978-3-642-28565-3, Feb. 2012, pp. 155-181. DOI : 10.1007/978-3-642-28566-0, (http://dx.doi.org/10.1007/978-3-642-28566-0)

[8] A. Hallapuro, M. Karczewicz H. Malvar, "Low Complexity Transform and Quantization - Part I: Basic Implementation", *Document of JVT*, pp. 227-230, (Feb. 2002).

[9] Maher Jridi, Yousri Ouerhani and Ayman AlFalou, " Low complexity DCT engine for image and video compression", *Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 8656, Real-Time Image and Video*