

Contribution à l'intégration matérielle du codeur vidéo pour l'imagerie médicale MM-Waaves

Yuhui Bai^{1,2}, Imen Mhedbi¹, Khalil Hachicha¹, Patrick Garda¹, Bertrand Granado¹, Syed Zahid Ahmed², Sébastien Topin³, Sylvain Hochberg³, Fayez Kaddouh⁴, Didier Heudes⁴

(1) LIP6 – Université Pierre et Marie Curie – CNRS – UMR 7606 – Paris

(2) ETIS – ENSEA – Université de Cergy – CNRS – UMR 8051 – Cergy

(3) Société CIRA – Paris

(4) APHP EGP – Paris

Résumé - Dans cet article, nous proposons un codeur vidéo pour système nomade haut débit pour le domaine médical basé sur un nouveau codeur vidéo Mask Motion Waaves à qualité médicale et une architecture de système sur puce pour la partie compression d'image du codeur. Sur le plan algorithmique, nous avons comparé trois techniques pour augmenter le taux de compression. Les résultats obtenus montrent une réduction significative du taux de compression (40% par rapport à l'encodeur WAAVES), tout en gardant la même qualité visuelle. Sur le plan architectural, nous décrivons la co-conception matériel / logiciel d'une architecture de système sur puce implémentée sur la plate-forme FPGA. Les résultats ont montré que la co-conception HW / SW à 100MHz pourrait atteindre jusqu'à un gain de 5x dans la partie critique du code source par rapport à solution purement logicielle exécutée sur un processeur à l'état de l'art à 3.2GHz.. La combinaison de ces résultats montre la faisabilité d'une mise en œuvre du codeur vidéo MMWaaves pour les applications e-santé.

Abstract - In this article, we present two studies that pave the way towards a mobile implementation of the WAAVES certified medical image compression encoder. On the algorithmic side, we compared three techniques to increase the compression rate. The obtained results show a significant bit-rate reduction, around 40% with respect to the WAAVES encoder, while keeping the same visual quality. On the architectural side, we describe the HW/SW co-design of an architecture implemented in a FPGA platform. By using code profiling, critical portions of the code were identified, then two methods for hardware acceleration were used to implement the critical part of the coder. The tests were done on FPGA and the results showed that HW/SW co-design running at 100MHz could achieve up to 5x performance gain in the critical portion compared to the high-end processor running at 3.2GHz. The combination of these results demonstrates the feasibility of materiel implementation of the WAAVES certified medical image coder suitable for e-health applications.

1 Introduction

Les images médicales, fixes ou en séquences vidéo, ne cessent de se développer en donnant une représentation de plus en plus précise des différentes parties du corps humain. Cependant plus l'image est précise plus la quantité des données engendrées est grande, leur compression est devenue une nécessité que ce soit à des fins de stockage ou de transport. Ceci est d'autant plus vrai dans le cadre de la eSanté où des appareils nomades sont de plus en plus utilisés. Nous présentons dans cet article nos travaux pour définir un codeur vidéo avec une qualité perceptuelle de niveau diagnostic médical. Ces travaux sont basés sur le codeur Waaves de la société CIRA, codeur certifié de qualité médicale, pour la compression des images fixes et sur des méthodes brevetées [1] au LIP6 sur la détection de mouvement pour la compression vidéo. Nos travaux concernent à la fois les aspects algorithmiques et les aspects d'architectures électroniques nécessaires pour embarquer le codeur MM-Waaves dans des appareils nomades.

2 MMWaaves

Pour générer un flux vidéo de qualité diagnostique médical nous avons combiné un algorithme de détection

de mouvement basé sur des champs de Markov [2] avec le codeur Waaves afin de créer le codeur vidéo MMWaaves (Mask Motion Waaves), dont l'architecture de principe est visible sur la figure 1. Dans ce codeur, une détection de mouvement est réalisée, nous avons étudié trois techniques pour la réaliser : la différence, la substitution et le calcul de ou exclusif. Ces opérations sont appliquées entre les pixels de l'image de référence et l'image courante.

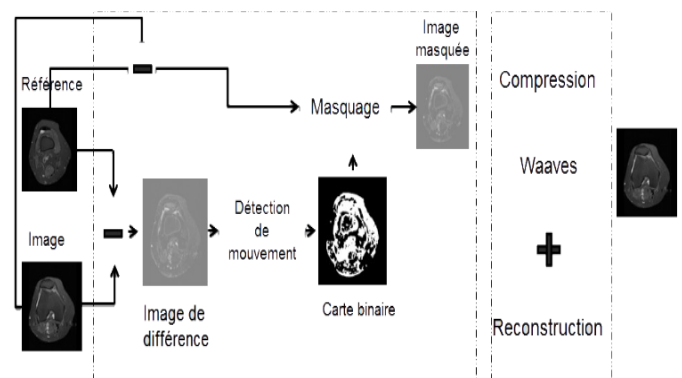


Figure 1: MM-Waaves

Les performances de l'algorithme de construction de la carte de mouvement sont liées à plusieurs paramètres [3] et notamment le seuil de binarisation. Nous avons donc évalué ces performances en fonction de l'évolution du seuil de binarisation. Nous avons considéré des images médicales 512*512.

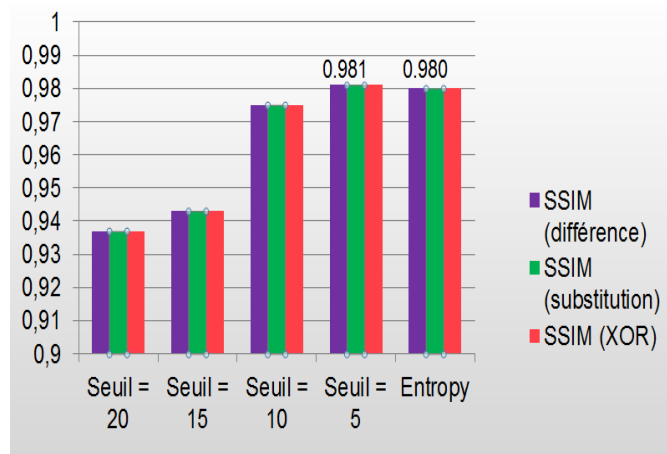


Figure 2: Influence du seuil sur l'évolution du SSIM

La première étape de l'algorithme de construction de la carte de mouvement consiste à faire une opération de soustraction entre l'image de référence et l'image courante, pour appliquer un algorithme de détection de mouvement basé sur les champs de Markov. La deuxième étape consiste à appliquer une des trois opérations : différence, substitution ou ou-exclusif entre les deux images et masquer le résultat avec la carte binaire de mouvement. Finalement, le bloc de reconstruction prenant en entrée l'image de référence et l'image masquée permet de retrouver l'image originale reconstruite. Afin de mesurer la performance de la qualité de cette image, nous appliquons ces deux mesures : une mesure objective basée sur le calcul du PSNR et une mesure psycho visuelle basée sur le calcul de l'indice SSIM [3]. Actuellement la plupart des évaluations de la qualité d'image sont réalisées à base d'une mesure quantitative du PSNR. Pour les images médicales qui nous intéressent, cette mesure fait l'objet de fortes critiques. Parfois cette même mesure est complétée par des évaluations subjectives souvent coûteuses. Le but est de proposer un outil capable d'améliorer la quantification de la qualité d'image médicale. Pour atteindre cet objectif nous exploitons les méthodes objectives basées sur la mesure de la qualité psycho visuelle d'une image. Ces méthodes consistent à mesurer l'erreur de visibilité entre une image dégradée et une image de référence en utilisant une variété de propriétés connues du système visuel humain. Ceci est traduit par la mesure d'une information structurale sous forme d'un index de similarité structurale.

Le diagramme de la figure 3 résume les résultats de l'influence du seuil de binarisation sur l'évolution de l'index SSIM. Nous avons comparé la compression des

images masquées (différence, substitution, Ou Exclusif) et la compression de l'image originale.

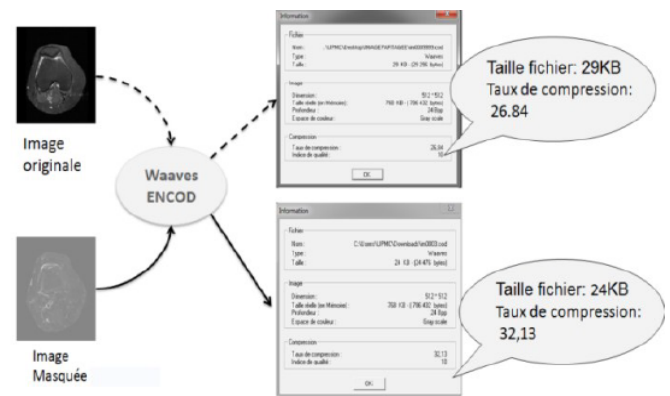


Figure 3: Compression Waaves

Le diagramme de la figure 4 illustre les résultats de l'influence du seuil de binarisation sur le taux de compression des différentes images. Nous avons un gain qui varie entre 21% pour la substitution et 46 % pour la différence pour un seuil fixe de valeur 5. Ces résultats ont montré la validité du codeur MM-Waaves qui permet d'augmenter le taux de compression tout en maintenant une qualité visuelle de niveau diagnostique médical. Fort de ce résultat nous sommes attelé à implémenter ce codeur, en débutant par le codeur d'image Waaves.

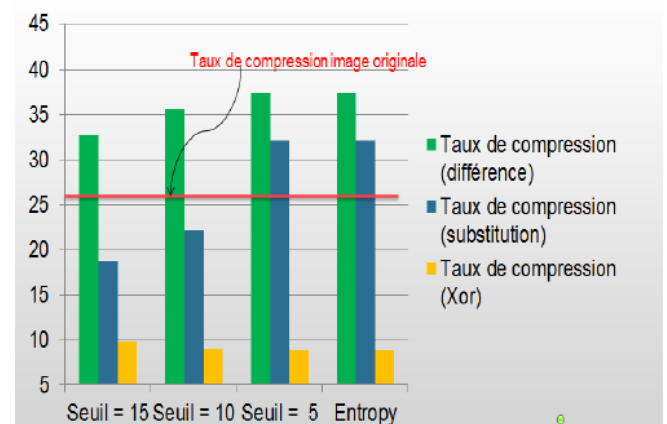


Figure 4: Influence du seuil sur l'évolution du taux de compression

3 Waaves sur le système embarqué

Des travaux préliminaires ont abouti à obtenir des performances de traitement d'une image RGB de taille 1Mo en 2s en compression ou décompression, sur un DSP C6748 cadencé à 1,5 GHz. Ces résultats défavorables, nous visons une image de 10Mo en une seconde, nous ont conduit à étudier la conception d'un système sur puce optimisé pour ce codeur. Nous avons pour cela mise en place une plate-forme expérimentale basée sur une FPGA Altera StratixIV Gx230 connecté à un capteur CMOS et un afficheur LCD. Un profilage sur un processeur Nios II a été réalisé pour déterminer les

lieux d'optimisation de l'algorithme, les résultats sont visibles sur la figure 5. A l'aide de ces résultats, nous avons réalisé une conception conjointe matérielle/logicielle, où les traitements les plus chronophages sont effectués au sein d'IP matérielles optimisées et les autres traitements au sein d'un processeur Nios II, dans un premier temps, ce processeur n'étant qu'un processeur d'expérimentation [4]. Ce système sur puce est illustré sur la figure 6, les tâches sont réparties entre le processeur du système, qui s'occupe de la conversion d'espace des pixels, de la transformée en ondelette et de la quantification, le codage adaptatif et le codage énumératif sont quant à eux implémentés sous forme d'accélérateur.

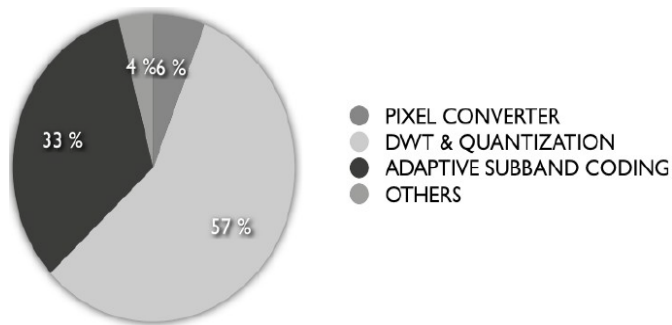


Figure 5: Repartition du temps d'exécution dans NiosII

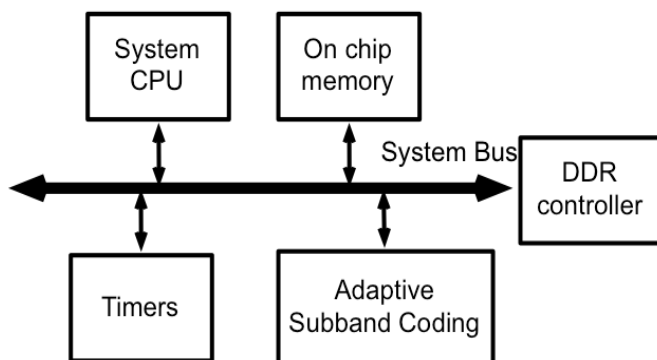


Figure 6: Système sur puce

3.1 Optimisation avec instructions personnalisées

Dans le code original, la transformée ondelette discrète [5] est calculée à l'aide de nombre en virgule flottante. Dans le processeur Nios II, il n'y a pas d'unité à virgule flottante (FPU). Pour pallier ce manque nous avons utilisé les instructions personnalisées disponibles pour le processeur NiosII qui permet de réaliser des accélérateurs matériels directement intégrés dans le chemin des instructions du processeur. Ceci permet d'augmenter les performances du système en déchargeant des portions de code logiciel à des accélérateurs matériels sans pour autant réaliser un interface spécifique [6].

3.2 Optimisation avec IP matériel personnalisé

Après avoir effectué la transformée ondelette et la quantification, nous obtenons une image décomposée en

plusieurs résolutions représentée par les coefficients des ondelettes quantifiés des différents sous-bandes. Le codage adaptatif qui suit permet d'encoder les coefficients de manière progressive entre les différentes sous bandes. Ce module est constitué d'une prédiction qui prédit la probabilité des coefficients d'être significatifs dans les sous-bandes de plus grande résolution à partir des sous-bandes de plus petite résolution. D'une mise à jour qui permet de réorganiser les coefficients quantifiés à partir de la prédiction pour obtenir un meilleur taux de compression. Du codage qui encode les coefficients réorganisés en utilisant le codage hiérarchique énumératif [7]. Dans la partie mise à jour est réalisé un tri intensif répétitif qui exige beaucoup d'accès mémoire externe aléatoire. Cette partie représente plus de 30% du temps total d'exécution de l'algorithme Waaves aussi bien sur le Nios II que sur un processeur flottant de type iCore7. Afin d'accélérer ce module, nous avons implémenté des IPs intégrés dans le système sur puce connecté au bus du système et géré comme des périphériques. Ces IP nécessitent un DMA qui a été implémenté entre le bus système et la mémoire externe. Dans la figure 7, l'architecture complète proposée est visible, les coefficients ondelette quantifiés sont enregistrés dans les mémoires externes et par les suites récupérées par l'IP de codage adaptatif via DMA. L'architecture est pipelinée entre les 3 parties du traitement pour pouvoir encoder les sous-bandes plus rapidement en utilisant du parallélisme de flux. L'IP a été conçue à l'aide du langage Verilog et encapsulé dans un composant via l'outil SOPC Builder d'Altera.

3.3 Performance

Les mesures de performances effectuées ont montré une accélération jusqu'à 20 fois par rapport aux traitements purement logiciels exécutés sur le processeur Nios II pour le module DWT à l'aide des instructions personnalisées. Dans la figure 8, une comparaison entre le temps d'exécution de l'IP matérielle de 3 images de taille 512x512 par rapport à l'exécution du code sur un DSP à l'état de l'art en virgule flottante de 1.5GHz et sur un processeur iCore7. Les résultats concernant l'architecture de l'IP de codage adaptatif à 100MHz ont montré un gain de 5.2 par rapport à l'exécution du code sur un processeur iCore7 à 3.2GHz [8].

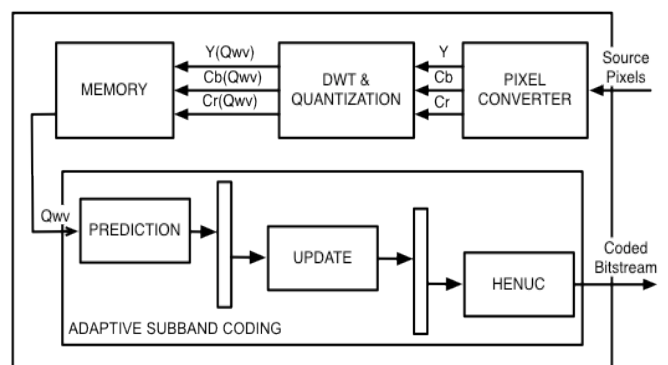


Figure 7: Flux du traitement dans le Soc

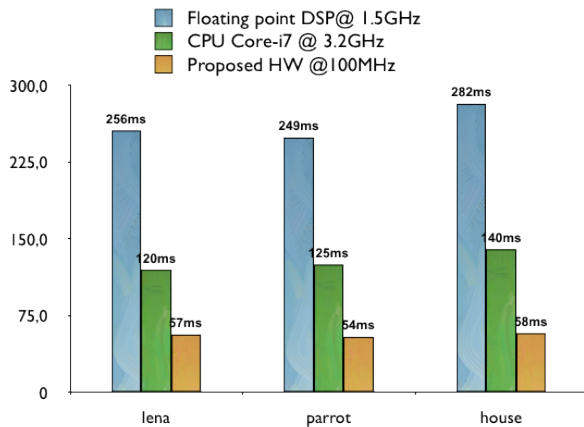


Figure 8: Performance/Evolutivité de l'architecture proposée

4 Conclusion

Dans cet article, nous avons présenté le codeur vidéo MM-Waaves combinaison du codeur Waaves et d'un algorithme de détection de mouvement basé sur des champs de Markov. Nous avons développé et validé MM-Waaves sur plusieurs modalités d'images médicales. À l'aide de deux mesures : une mesure objective (PSNR, SNR) et une mesure Psycho visuelle (index SSIM). Nous avons montré que le masquage des images de différences donne de meilleurs résultats que sans, et permet un gain de compression qui arrive à 45% et une qualité d'image élevée. Nous avons également, dans une optique d'implémentation de MM-Waaves proposé une architecture de conception jointe matérielle/ logicielle basée sur une plate-forme FPGA pour le codeur d'image Waaves. Nous avons proposé deux méthodes d'accélération matérielle avec une fréquence à 100MHz qui permettent une accélération

allant jusqu'à 5 fois par rapport à une accélération pour la performance du codeur par rapport à solution purement logicielle exécutée sur un processeur à l'état de l'art d'une fréquence de 3.2GHz.

[1] F.Lohier, P.Garda, L.Lacassagne, « Pocédé et dispositif de traitement de séquences d'images avec masquage », Brevet UPMC, Demande FR 62060 L du 3février 2000

[2] F.Lohier, P.Garda, L.Lacassagne, « Masked-Motion-JPEG2000 : A new reduced complexity video sequence compression scheme based on a MRF-motion detection algorithm toward inter frame masking », conf. On signal Processing Applications and technology, October 2000

[3] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[4] A. Ben Atitallah, P. Kadionik, F. Ghazzi, P.Nouel, N. Masmoudi and H. Levi, An FPGA implementation of HW/SW codesign architecture for H.263 video coding. International journal of Electronics and Communications, Dec 2006.

[5] M. W. Marcellin D. S. Taubman. JPEG2000: Image Compression Fundamentals, Standards, and Practice. Kluwer Academic Publishers, Nov 2001.

[6] Altera Nios custom instructions User Guide – Altera.

[7] K. Haapala, V. Lappalainen, T. D. Hämäläinen. Experimental parallel implementation of a wavelet-based still image encoder. Microprocessors and Microsystems, 29(4):155–167, 2005.

[8] Y.Bai, S.Z.Ahmed, B.Granado, FPGA Implemen-tation of Hierarchical Enumerative Coding for Locally IEEE International Conference on Field Programmable Logic and Applications (FPL), Sep, 2013.