

Composants Virtuels Comportementaux pour Applications de Compression d'Images

Guillaume SAVATON¹, Emmanuel CASSEAU¹, Eric MARTIN¹, Catherine LAMBERT-NEBOUT²

¹LESTER – Laboratoire « Electronique des Systèmes Temps Réel »

Université de Bretagne Sud, Centre de Recherche

Rue de Saint-Maudé – B.P. 92116

56321 LORIENT Cedex

{prénom.nom}@iuplo.univ-ubs.fr

²CNES – Centre National d'Etudes Spatiales

DTS/AE/SEA/ET

18 avenue Edouard Belin

31401 Toulouse Cedex 4

catherine.lambert@cnes.fr

Résumé – Afin de faire face aux nouveaux besoins des applications d'imagerie numérique et au volume croissant de données qu'elles manipulent, des techniques de compression de plus en plus élaborées sont requises, donnant naissance à de nouveaux standards tels que le tout récent *JPEG2000* pour le codage des images fixes. Dans cet article, nous nous intéressons à l'implantation sur architecture matérielle d'une chaîne de compression d'images à base de composants virtuels réutilisables. Face à la complexité des algorithmes à implanter et à la variété des profils d'applications supportés par *JPEG2000*, les méthodes traditionnelles de conception au niveau *RTL* trouvent leurs limites, c'est pourquoi nous proposons de rehausser le niveau d'abstraction de la spécification et de bénéficier des nouveaux outils de synthèse d'architecture du commerce afin d'introduire la notion de flexibilité architecturale d'un composant virtuel. Nous présentons ici l'application de notre méthodologie, développée dans le cadre du projet *RNRT MILPAT*, à la conception d'un composant virtuel de haut niveau pour la transformation en ondelettes discrète bidimensionnelle.

Abstract – Digital imaging applications have to deal with increasingly large images and require more and more efficient compression techniques. Recent standards for still image coding, such as *JPEG2000*, aim at fulfilling these new requirements. In this paper, we focus on the design of an image compression system based on reusing virtual components. We believe that traditional methods for hardware design at the *RTL* level suffer from heavy limitations that prevent them from efficiently addressing the complex algorithms to implement and the high flexibility required by the various application profiles supported by *JPEG2000*. This is the reason why we propose to raise the abstraction level of the specification and benefit from the emerging commercial behavioral synthesis tools in order to introduce the notion of architectural flexibility of a virtual component. Our method has been successfully applied to the design of a high-level virtual component for the 2D Discrete Wavelet Transform.

1. Introduction

Les applications d'imagerie numérique tendent à manipuler des images de dimensions croissantes. Le stockage et la transmission via des canaux de communication de tels volumes de données imposent des contraintes fortes sur le dimensionnement des support de mémorisation et de communication. La compression devient une étape indispensable, permettant de réduire le nombre de bits nécessaires à la représentation de l'information portée par une image et de tirer le meilleur parti des capacités de stockage et des bandes passantes disponibles.

Cependant, le fait de comprimer les images produites par un dispositif d'acquisition ou de traitement introduit de nouvelles contraintes liées aux performances du processus de compression lui-même. D'une part, les méthodes de compression non réversibles, intéressantes du point de vue des forts taux de compression qu'elles autorisent, se traduisent par des pertes d'information qui peuvent nuire à la qualité des images : selon les besoins des utilisateurs, un juste compromis doit être trouvé entre taux de compression et qualité. D'autre part, plus les images ont une taille importante, plus la compression nécessitera de temps de calcul et de quantité de mémoire. La réduction de la durée du processus de compression tend généralement à préférer aux solutions logicielles une implantation sous forme d'accélérateurs matériels à base de

circuits spécifiques du type *ASIC* ou *FPGA*. Lorsque les algorithmes de compression le permettent, la minimisation de la quantité de mémoire se fait en spécifiant un motif de traitement « au fil de l'eau », capable de produire les données de l'image compressée au fur et à mesure que les données image source lui sont présentées dans un ordre prédéterminé.

Cependant, les algorithmes de compression d'image – et ceux mis en place dans le standard *JPEG2000* en sont une bonne illustration – tendent à devenir de plus en plus complexes afin de répondre aux deux exigences contradictoires d'élever le taux de compression tout en conservant une bonne qualité de restitution. Spécifier une architecture matérielle compatible avec la norme *JPEG2000* exige un effort de conception conséquent tant pour la spécification de l'architecture que pour sa validation. De plus, la variété des paramètres correspondant aux différents profils d'application du standard *JPEG2000* apparaît comme un obstacle à la mise en place d'une stratégie de réutilisation de blocs fonctionnels – blocs « transformation en ondelettes » ou « codeur arithmétique », par exemple – déjà conçus par ailleurs sous forme de composants virtuels. Les méthodes de conception traditionnelles au niveau *RTL* ne permettent pas en effet d'atteindre le degré de flexibilité escompté et nécessiteraient de coûteuses et multiples conceptions de composants virtuels correspondant à chaque profil.

L'approche que nous proposons consiste à rehausser le niveau d'abstraction des composants virtuels en tirant profit des nouveaux outils de synthèse d'architecture émergeant sur le marché. L'objectif visé est d'une part de faciliter la spécification et la validation de ces composants au niveau fonctionnel, et d'autre part d'introduire une notion de flexibilité architecturale, autorisant l'instanciation, à partir d'une même description de haut niveau, d'architectures variées répondant à différents jeux de paramètres fonctionnels et respectant différentes contraintes de performances – en termes de nombre de portes, vitesse de traitement, quantité de mémoire, consommation, etc.

Dans cet article, nous présentons l'application de notre méthode, développée dans le cadre du projet *RNRT MILPAT*, à la spécification sous forme de composant virtuel d'un algorithme de transformation en ondelettes discrète bidimensionnelle intégrant dans une même description générique les paramètres supportés par la norme *JPEG2000*.

2. Normes de compression d'image

Parmi les techniques de compression des images fixes, la plus répandue est la norme *JPEG* (*Joint Photographic Experts Group, ITU-T T.81*) qui a su répondre aux besoins de nombreuses applications, notamment dans le domaine de la photographie numérique et de la transmission de fichiers images sur Internet (80% des images disponibles sur le web sont au format *JPEG*). Des limitations ont cependant été constatées, consistant notamment en l'apparition d'artefacts de blocs aux forts taux de compression. L'impact de ces altérations sur la qualité des images est ressenti plus fortement dans les domaines où une excellente qualité de restitution est exigée. C'est le cas par exemple de l'imagerie spatiale [7], de l'imagerie médicale, et de la photographie numérique.

Le comité *JPEG* a ainsi décidé de créer une nouvelle norme, *JPEG2000*, intégrant des techniques plus performantes afin de répondre aux nouveaux besoins [1]. Les exigences formulées par les utilisateurs potentiels, en termes de performances attendues et de fonctionnalités souhaitées, ont été regroupées en un ensemble de profils correspondant chacun à un domaine d'application : Internet, photographie numérique, imagerie médicale, télécommunications mobiles, imagerie SAR, télédétection optique, etc. La constitution de ces différents profils a servi de support à la sélection d'algorithmes répondant aux besoins des applications concernées.

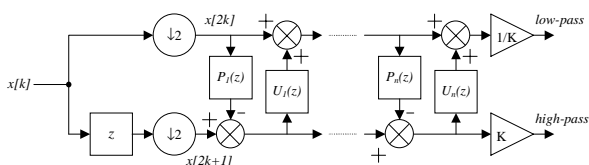


FIGURE 1. (a) Banc de filtre 1D pour une décomposition dyadique par convolution ; (b) structure *lifting* équivalente

La norme *JPEG2000* utilise un procédé de décorrélation basé sur la transformation en ondelettes discrète (*DWT*) – et non plus la transformation en cosinus discrète (*DCT*) – avec codage spécifique des sous-images [2][3]. Elle fournit égale-

ment une structure d'organisation des données compressées très flexible. Les stratégies adoptées pour améliorer la qualité aboutissent à des algorithmes de compression de forte complexité. Du fait de la variété des techniques mises en place pour répondre aux différents profils, *JPEG2000* s'apparente plus à une « boîte à outils » qu'à un schéma de codage unique.

3. La transformation en ondelettes discrète

3.1 Principe

La transformation en ondelettes discrète (*DWT*) bidimensionnelle repose sur la notion d'analyse multirésolution d'une image [4]. Celle-ci est décomposée en un ensemble de sous-bandes représentant l'information portée par l'image source à différents niveaux de résolution : l'image d'approximation (LL_N) est une version réduite et lissée de l'image initiale tandis que les images de détails – « horizontaux » (LH_n), « verticaux » (HL_n), « diagonaux » (HH_n) avec $1 \leq n \leq N$) contiennent uniquement des informations relatives à la texture locale et aux contours des régions de l'image, à une résolution n donnée et selon une direction donnée. L'algorithme de MALLAT [5] permet d'obtenir ces sous-images par application récursive, d'abord sur l'image source puis sur l'image d'approximation obtenue à chaque niveau, d'un banc de filtres passe-bas/passe-haut appliqué successivement selon les lignes et les colonnes de l'image à transformer.

La méthode connue sous le nom de « *Lifting Scheme* » [6], applicable dans le cas des transformations en ondelettes biorthogonales, consiste à remplacer la paire de filtres passe-bas/passe-haut par une structure en échelle mathématiquement équivalente (FIGURE 1). Les avantages de cette méthode comparée à l'algorithme de MALLAT sont d'une part une réduction de la complexité calculatoire, et d'autre part une exploitation plus efficace de la mémoire nécessaire au stockage des résultats partiels de calcul.

A la différence de la *DCT*, la *DWT* ne fournit pas une méthode de décomposition unique. La norme *JPEG2000* exploite cette flexibilité en donnant le choix entre deux ondelettes biorthogonales [2] : l'ondelette 9/7 pour la compression irréversible, et 5/3 pour la compression réversible. En outre, la profondeur d'exploration des niveaux de résolution est laissée au libre choix de l'utilisateur.

3.2 Transformation en ondelettes *lifting* au fil de l'eau

L'algorithme de transformation en ondelettes au fil de l'eau que nous avons développé est basé sur la méthode du *lifting scheme* avec l'ondelette 9/7. L'image à transformer étant parcourue par blocs de $M \times N$ pixels de la gauche vers la droite et du haut vers le bas, le processus de traitement réalise pour chaque bloc image source un calcul partiel de la *DWT* 2D multi-niveaux et donne naissance, d'une part à un bloc de $M \times N$ éléments de l'image transformée, et d'autre part à un ensemble de résultats partiels de calculs qui devront être mémorisés afin d'être réutilisés lors de la transformation de blocs ultérieurs.

Certains choix algorithmiques auront une influence significative sur les performances en vitesse et en surface de

l'architecture générée : le fait de spécifier une transformation sur des blocs de taille $M \times N$ autorise au premier niveau de décomposition N transformations horizontales à s'effectuer en parallèle, suivies de M transformations verticales également parallélisables. D'autre part, l'écriture d'un motif de transformation *lifting* 1D pour chaque transformation horizontale ou verticale autorise soit une implémentation de type **séquentiel**, dans laquelle les pas de *lifting* (P_i) et de dual *lifting* (U_i) (FIGURE 1) s'enchaînent les uns après les autres ; soit une implémentation de type **pipeline**, autorisant l'exécution en parallèle des calculs correspondant aux P_i et U_i [11].

4. Composants virtuels comportementaux

4.1 Définition

Un Composant Virtuel, ou *IP* pour *Intellectual Property*, se définit comme un ensemble de fichiers (modèle abstrait pour évaluation au niveau système, description synthétisable, bancs de test, scripts, etc.) permettant l'intégration d'un bloc fonctionnel préconçu – souvent acheté à un fournisseur extérieur – dans le flot de conception d'un système à implanter sur *ASIC* ou *FPGA*. Les formats de description et degrés d'abstraction d'un composant virtuel font l'objet de recommandations et standards, en particulier de la part du groupe *VSIA* (*Virtual Socket Interface Alliance*) [8], visant à garantir la « réutilisabilité » d'un bloc – adéquation entre les performances annoncées par le fournisseur et celles obtenues par l'utilisateur ; aptitude à interagir avec l'environnement dans lequel il sera implanté –, à standardiser les flots de conception et d'intégration d'un composant [9], et à fournir un cadre fiable pour le commerce électronique des composants virtuels. Le niveau d'abstraction le plus élevé admis par *VSIA* correspond aux composants virtuels dits *soft*, dont la description synthétisable est fournie au niveau transfert de registre (*RTL*), et destinés à des outils de synthèse *RTL* tels que *Design Compiler™* de *Synopsys*.

Une telle description, bien que paramétrable dans un domaine limité, ne possède pas de réelle flexibilité architecturale. C'est pourquoi nous proposons de rehausser le niveau d'abstraction des descriptions en introduisant la notion de *composant virtuel comportemental*, décrit sous une forme algorithmique dans un langage de haut niveau (*VHDL*, *SystemC*, etc.) et destiné à être synthétisé par des outils de synthèse d'architecture (*HLS* pour *High-Level Synthesis*) du commerce – capables de produire, automatiquement et en respectant un jeu de contraintes d'optimisation, une architecture *RTL* à partir d'une description de niveau comportemental. Le style de description adopté autorise un haut degré de flexibilité par le biais d'un jeu de paramètres génériques. Outre une description algorithmique de la fonction réalisée, un composant virtuel comportemental doit comporter une description de son protocole d'entrée/sortie, ainsi qu'un modèle flexible de la mémoire nécessaire au traitement.

4.2 Application à la conception d'une architecture flexible pour la *DWT* 2D

Comme nous avons pu l'observer, la variété des paramètres envisageables dans le cas d'une fonction telle que la *DWT*

est un défi à la spécification d'architectures matérielles pour *JPEG2000* : une architecture universelle aurait une forte complexité et serait surdimensionnée par rapport à une application donnée ; en revanche, la spécification d'architectures distinctes bien adaptée à chaque profil d'utilisation nécessiterait de nombreuses et coûteuses « *re-conceptions* ».

L'approche proposée dans le cadre du projet *RNRT MILPAT* consiste à mettre à profit les nouveaux outils de conception de haut niveau afin de permettre la génération automatique d'une variété d'architectures spécifiques à partir d'une unique description générique [11]. Cette description est rédigée dans le langage *VHDL* restreint aux instructions de niveau comportemental [10].

Le composant virtuel se décompose en quatre unités interconnectées : l'Unité de Traitement prend en charge la partie calculatoire de la *DWT* ; l'Unité de Contrôle gère l'état de la transformation (localisation spatiale du bloc à traiter dans l'image) ; l'Unité de Communication spécifie l'interface entre le *VC* et le système dans lequel il s'insère ; l'Unité Mémoire de Travail gère les accès à la mémoire de stockage des résultats de calculs partiels de transformations 1D verticales.

Les paramètres permettant d'adapter l'architecture à une application donnée peuvent être répartis en plusieurs catégories : 1 - Format des données manipulées (taille des images, taille des blocs, nombre de bits par pixel) ; 2 - Paramètres algorithmiques (choix de l'ondelette, niveaux de résolution, parallélisme des motifs *lifting* 1D) ; 3 - Adaptation à l'environnement d'intégration (taille des ports d'entrée/sortie, protocole de communication, cadence de traitement).

5. Résultats

Nous avons procédé à la synthèse de l'Unité de Traitement à l'aide de l'outil de synthèse de haut niveau *Monet™*, de *Mentor Graphics*, pour différents jeux de paramètres : a) nombre de niveaux de décomposition en ondelettes (1 ou 2 dans les résultats que nous présentons) ; b) degré de parallélisme des motifs de transformation 1D : séquentiel suivant les deux directions spatiales (S/S), parallèle horizontalement et séquentiel verticalement (P/S), parallèle horizontalement et verticalement (P/P) ; c) cadence de traitement.

Nous nous sommes restreints dans un premier temps à un modèle d'entrées/sorties dans lequel les données image source et transformée (blocs de 2×2 pixels pour une transformation sur 1 niveau ; de 4×4 pixels pour 2 niveaux) sont échangées avec l'Unité de Traitement sous forme de paquets de quatre pixels à raison d'un paquet par cycle.

Le TABLEAU 1 présentent les performances maximales que peut atteindre l'Unité de Traitement en supposant qu'un calcul élémentaire du type « *lifting step* » s'exécute en un cycle. La quantité de mémoire nécessaire pour la *DWT* verticale est donnée dans le cas d'une image de largeur égale à 256 pixels. Les FIGURES 2 et 3 illustrent les performances obtenues après synthèse comportementale, pour différents degrés de parallélisme et différentes contraintes de temps.

Les solutions obtenues permettent de délimiter l'espace d'exploration architecturale et offrent à l'utilisateur du *VC* la possibilité de sélectionner le bon compromis vitesse/surface/quantité de mémoire en fonction des contraintes

de l'application. La synthèse d'architecture nous a permis d'obtenir, pour chaque degré de parallélisme, des solutions dont la vitesse maximale est très proche de la limite estimée (TABLEAU 1). Quelle que soit la contrainte de temps, les architectures construites sur le motif « P/P » présentent la surface la plus importante et nécessitent la plus grande quantité de mémoire : ce motif est intéressant uniquement dans le cas d'applications exigeant une très grande vitesse de traitement.

TABLEAU 1. Performances estimées pour une DWT sur N niveaux

Motif (H/V)	Temps de traitement min estimé (cycles/pixel)		Mémoire de DWT horizontale (nombre de mots)		Mémoire de DWT verticale (nombre de mots)	
	N = 1	N = 2	N = 1	N = 2	N = 1	N = 2
	S/S	2,5	1,25	8	24	1024
P/S	1,5	0,75	22	66	1024	1600
P/P	0,5	0,25	22	66	2816	4224

A partir des courbes présentées en FIGURES 2 et 3 nous avons pu exhiber un jeu de solutions correspondant aux architectures les moins volumineuses et les moins coûteuses en mémoire pour une contrainte de temps donnée. Nous les avons réparties en trois classes dont nous résumons les caractéristiques dans le TABLEAU 2.

TABLEAU 2. Classes de solutions architecturales

Classe de solution	Motif (H/V)	DWT sur 1 niveau		DWT sur 2 niveaux	
		Temps de calcul (cycles/pixel)	Surface (kPortes)	Temps de calcul (cycles/pixel)	Surface (kPortes)
Rapides	P/P	< 1,5	50 à 75	< 0,75	210 à 230
Moyennes	P/S	1,5 à 2,5	45 à 48	0,75 à 1,25	160 à 180
Lentes	S/S	> 2,5	< 45	$\geq 1,25$	< 155

6. Conclusion

L'approche que nous avons mise en place dans le cadre du projet MILPAT nous a permis de spécifier à un haut niveau d'abstraction un composant virtuel réutilisable pour la transformation en ondelettes 2D, autorisant à partir d'une unique description paramétrable la génération d'une variété d'architectures adaptées à différents profils d'utilisation.

Cette méthode repose d'une part sur les outils de synthèse d'architecture du commerce, et d'autre part sur un modèle de spécification au niveau comportemental qui se veut général et indépendant des outils et des langages. Des travaux futurs nous permettront de valider notre méthodologie, notamment en envisageant la synthèse de l'algorithme de transformation en ondelettes par d'autres outils du commerce, comme *Behavioral CompilerTM* de Synopsys.

7. Remerciements

Les travaux présentés dans cet article ont bénéficié du soutien de la société Astrium (Vélizy, France), que nous tenons à remercier pour sa participation à l'étude sur les composants virtuels de niveau comportemental.

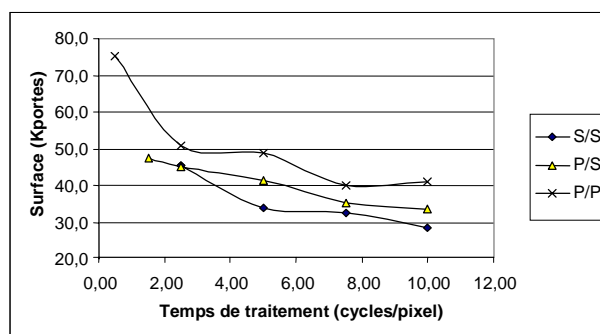


FIGURE 2. Performances des solutions architecturales pour une décomposition sur un niveau

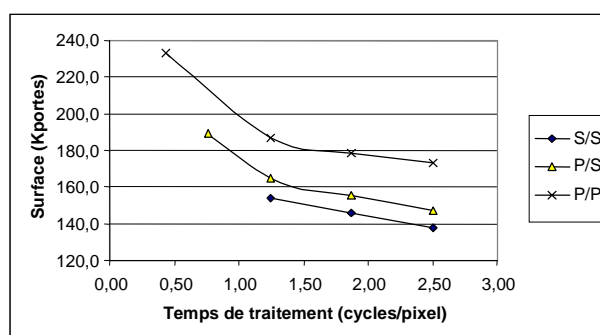


FIGURE 3. Performances des solutions architecturales pour une décomposition sur deux niveaux

8. Bibliographie

- [1] ISO/IEC JTC1/SC29 DWG1 – *JPEG2000 Requirements and Profiles Version 6.3* – 2000, <http://www.JPEG.org/JPEG2000.htm>
- [2] ISO/IEC JTC1/SC29 DWG1 – *JPEG2000 Part I Final Committee Draft 1.0* – 2000, <http://www.JPEG.org/CDs15444.htm>
- [3] David TAUBMAN – *High performance Scalable Image Compression with EBCOT* – IEEE Transactions on Image Processing, Vol. 9, n°7, pp 1158 à 1170, Juillet 2000.
- [4] M. ANTONINI, M. BARLAUD, P. MATHIEU, I. DAUBECHIES – *Image Coding Using Wavelet Transform* – IEEE Trans. on Image Processing, Vol 1, Nr 2, pp 205- 220, Avril 1992
- [5] S. G. MALLAT – *A Theory for Multi-Resolution Signal Decomposition: the Wavelet Representation* – IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol 11, Nr 7, July 1999
- [6] I. DAUBECHIES, W. SWELDENS – *Factoring Wavelet Transforms into Lifting Steps* – Journal of Fourier Analysis and Applications, Vol 4, Nr 3, pp 247-269, 1998
- [7] C. LAMBERT-NEBOUT, G. MOURY, J.-E. BLAMONT – *Status of Onboard Image Compression for CNES Space Missions* – Proceedings of SPIE'99, Vol 3808, pp 242-256, Octobre 1999
- [8] VSI Alliance – *Architecture Document* – Version 1.0, 1997, disponible à l'adresse <http://www.vsi.org>
- [9] M. KEATING, P. BRICAUD – *Reuse Methodology Manual for System-on-a-Chip Design* – Kluwer Academic Publishers, 1999
- [10] G. SAVATON, E. CASSEAU, E. MARTIN – *Behavioral VHDL Styles and High-Level Synthesis for IPs* – FDL 2000, Forum on Design Languages, pp. 107-115, Septembre 2000
- [11] G. SAVATON, E. CASSEAU, E. MARTIN – *Design and Synthesis of a Discrete Wavelet Transform Virtual Component for Image Compression* – International Workshop on IP Based Design, pp 103-108, Décembre 2000.