



HAL
open science

Vers un système de projection alternatif pour le cinéma numérique

Nicolas Bertrand, Jean-Denis Durou, Vincent Charvillat

► **To cite this version:**

Nicolas Bertrand, Jean-Denis Durou, Vincent Charvillat. Vers un système de projection alternatif pour le cinéma numérique. COMpression et REprésentation des Signaux Audiovisuels (CORESA 2012), May 2012, Lille, France. (support électronique). hal-04584908

HAL Id: hal-04584908

<https://hal.science/hal-04584908>

Submitted on 24 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vers un système de projection alternatif pour le cinéma numérique

Nicolas BERTRAND¹

Jean-Denis DUROU¹

Vincent CHARVILLAT¹

¹ IRIT, UMR CNRS 5505, Toulouse

nicolas.bertrand@isf.cc durou@irit.fr vincent.charvillat@enseeiht.fr

Résumé

La conversion des salles de cinéma vers le numérique est en cours. Le format JPEG2000 a été choisi par le consortium Digital Cinema Initiative (DCI) pour la compression des images. Ce format permet de compresser efficacement des images de grande taille. Néanmoins, la mise en œuvre de l'encodage et du décodage de ce format est complexe : elle entraîne une grande consommation de temps de calcul. Les solutions les plus performantes actuellement sont matérielles (utilisation d'architectures de type VLSI et, en particulier, de FPGA). Il existe peu de solutions logicielles performantes pour la décompression d'images en temps réel. Notre objectif pour le cinéma numérique dans les salles Art & Essai est de pouvoir décompresser un flux d'images allant jusqu'à 30 images par seconde au format 2K (images de taille 2048×1080) en développant des logiciels libres. Des solutions à base de GPU ont été proposées pour l'encodage JPEG2000. Dans cet article, nous commençons par présenter JPEG2000 et les spécificités de ce format liées au cinéma numérique, puis des solutions d'architecture logicielle pour la parallélisation de sa décompression.

Mots clefs

Cinéma numérique, projection, JPEG2000, décompression.

1 Introduction

L'exploitation cinématographique est entrée dans un nouveau paradigme : le numérique. En France, au début de l'année 2012, plus de 60 % des salles de cinéma sont équipées de projecteurs numériques, mais seulement 44 % des exploitants (sources cinego.net). Les multiplexes, comprenant beaucoup de salles, sont déjà équipés, mais de nombreux petits cinémas indépendants ou ayant le label Art & Essai sont toujours équipés de projecteurs de bobines 35 mm. Pendant ce temps, les principaux laboratoires photochimiques annoncent la fin du tirage sur bobines d'ici un à deux ans.

Le consortium *Digital Cinema Initiative* (DCI) a été créé en 2002 par sept *majors* hollywoodiennes, afin d'accélérer la transition de l'industrie du cinéma vers le numérique. La principale motivation de cette transition a été économique, afin de diminuer le coût du tirage des bobines de films. Cela

est expliqué dans [1], ainsi que les risques que cette transition présente pour les salles indépendantes. Le DCI a publié une spécification [2] expliquant le modèle du cinéma numérique, de la distribution à la projection.

Le *master* numérique est appelé *Digital Cinema Distribution Master* (DCDM). Il comprend les pistes image, son, sous-titres, etc. Seules les images sont compressées. Le format de compression choisi est le JPEG2000. Il n'y a pas de codage temporel de la vidéo : on code image par image. La copie numérique issue du DCDM est appelée *Digital Cinema Package* (DCP).

Dans le paragraphe 2, nous proposons un système de projection alternatif adapté aux salles indépendantes. Dans le paragraphe 3, nous précisons les notions de JPEG2000 nécessaires aux paragraphes suivants. Le paragraphe 4 explique les raisons des contraintes imposées par le DCI au JPEG2000. Le paragraphe 5 présente des pistes de parallélisation logicielle pour le décodage en temps réel des images compressées.

2 Problème des salles indépendantes

Le modèle du cinéma numérique tel qu'il est proposé actuellement, de la distribution à la projection en salle, convient au mode de fonctionnement des multiplexes mais n'est pas adapté aux cinémas indépendants en raison du coût (de 70 k€ à 100 k€), de la complexité et de la sécurité (lutte contre le piratage) des équipements.

Le but de notre travail, initié par les cinémas Utopia, est de proposer une solution alternative libre et à moindre coût, adaptée aux distributeurs et exploitants Art & Essai. Les domaines couverts sont la création, la projection et la distribution de copies numériques avec une bonne qualité d'image optimisée pour les petits écrans (c'est-à-dire de largeur inférieure à 10 m).

Dans cet article, nous nous intéressons spécifiquement à la projection. La problématique générale qui s'y rapporte est présentée dans [3]. Les images d'un DCP sont compressées au format JPEG2000. Nous cherchons une solution permettant de décoder les fichiers JPEG2000 à une cadence de 25/30 images par seconde, au format 2K (images de taille 2048×1080 , comportant 3 composantes de couleur et une profondeur de 12 bits) sur des ordinateurs courants, afin de réduire le coût du matériel de projection. Les constructeurs de l'industrie cinématographique proposent du matériel dédié mais, pour un écran de largeur inférieure à 10 m,

Projecteur	DCI	Alternatif
Définition	2048 × 1080	1920 × 1080
Profondeur	12 bits	10 bits
Débit supporté	3 Gbits/s	3 Gbits/s
Sous-échantillonnage	Aucun	4 : 2 : 2
Coût	60 k€	[20,25] k€
Largeur max. d'écran	30 m	10 m

Tableau 1 – Comparaison des propriétés des projecteurs pour le cinéma numérique.

il est possible d'utiliser des projecteurs moins chers, fondés sur la même technologie que les projecteurs 2K. Des tests ont montré que cela permettait d'obtenir une bonne qualité d'image à la projection [4]. Ces deux types de projecteurs ont des propriétés très similaires (cf. tableau 1).

Le décodage se fait par un *player*, un logiciel décodant les images à une cadence d'au moins 24 image par secondes (cf. figure 1). Il prend en entrée des DCP standards, non cryptés dans un premier temps, et pourra être connecté soit à un projecteur 2K, soit à un projecteur alternatif. Dans le cadre de notre étude, nous nous intéressons d'abord au projecteur alternatif. Pour le projecteur 2K, il faudra de plus assurer une liaison cryptée [2, p. 88].

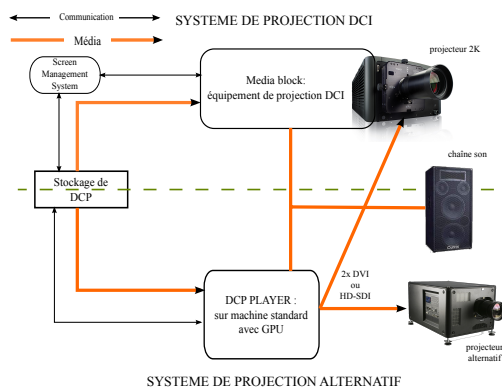


Figure 1 – Système de projection DCI et proposition d'un système alternatif. Le SMS (Screen Management System) est une application permettant de gérer les projections dans une salle. Dans le système alternatif, elle pourrait être intégrée au *player*.

Les besoins du *player* pour le projecteur alternatif sont : (i) décompresser en temps réel (jusqu'à une cadence de 30 images par seconde) ; (ii) diminuer la profondeur de 12 à 10 bits ; (iii) réduire la taille de l'image pour être compatible avec le format 1920 × 1080 ; (iv) faire un sous-échantillonnage chromatique. Le rapport d'aspect d'un film au moment du tournage vaut souvent 2.39 ou 1.85, ce qui donne, respectivement, une résolution de 2048 × 857 ou 1998 × 1080 (les pixels du DCI sont carrés). Il ne faut donc que très légèrement modifier l'image pour être compatible

avec le projecteur alternatif. Cette mise à l'échelle peut se faire pendant ou après la décompression. Le passage de 12 bits à 10 bits de profondeur faciliterait la décompression, car il y aurait moins de plans de bits à traiter dans EBCOT (cf. paragraphe suivant). Les outils de production cinématographique (caméras ou logiciels de montage) utilisent souvent le sous-échantillonnage chromatique. Le DCI interdisant tout sous-échantillonnage, on sera amené à sur-échantillonner pour pouvoir produire un master numérique, bien que cela n'apporte pas d'information supplémentaire. Il faudra donc sous-échantillonner, et décider si cela doit être fait pendant ou après la décompression.

Le coût de cette solution alternative se situe entre 12 et 15 k€ : un ordinateur coûte 2 k€ et le projecteur alternatif est vendu à 12 k€ environ. Il n'y pas de coût lié au logiciel car seules des solutions libres sont prévues.

Il n'existe à ce jour aucun logiciel libre permettant la décompression en temps réel d'images JPEG2000 au format du cinéma numérique. Le logiciel *Kakadu* de compression et de décompression de l'équipe de Taubman, dans sa version 6.0, est censé décoder les images au format 2K avec un rapport d'aspect de 2.39 (2048 × 857) et un débit de 244 Mbits/s à une fréquence de 24 images par seconde sur des ordinateurs à 8 cœurs. Notons que le rapport d'aspect choisi correspond aux images les plus petites du format 2K. Nous souhaitons développer un logiciel de décompression libre, ce qui n'est pas le cas de *Kakadu*. Le laboratoire Fraunhofer IIS propose une solution pour le cinéma numérique, *easyDCP*, qui elle aussi est propriétaire.

3 JPEG2000

Nous présentons ici le JPEG2000 du point de vue de la décompression, c'est-à-dire le passage du fichier binaire à l'image.

3.1 Organisation du fichier binaire

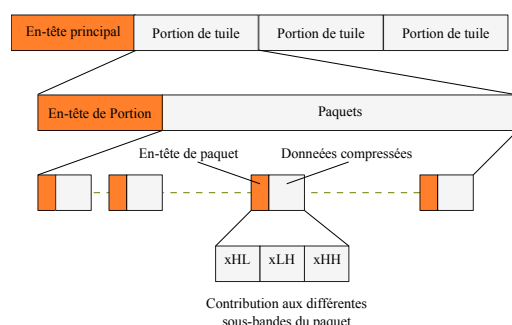


Figure 2 – Organisation du fichier binaire : une fois les contributions d'un paquet décompressées, leur ensemble forme un *precinct*.

Le fichier binaire est composé d'un en-tête principal et de données (cf. figure 2). Ces dernières sont structurées en une succession de portions de tuiles (*tile-parts*), une tuile étant une zone spatiale de l'image (il peut y avoir une ou

plusieurs tuiles par image). Chaque portion de tuile commence par un en-tête et comporte une succession de paquets. Les en-têtes du fichier binaire et d'une portion de tuile sont construits de la même manière : ils contiennent des marqueurs indiquant les paramètres (taille, nombre de composantes, ...) de l'image. Ces marqueurs peuvent être obligatoires ou optionnels. Un même marqueur peut se retrouver à la fois dans les en-têtes du fichier binaire et d'une portion de tuile. Chaque paquet inclus dans une portion de tuile dispose d'un en-tête spécifique qui précède les données compressées issues d'une petite région de l'image.

3.2 Organisation des coefficients d'ondelettes

Nous décrivons ici les données compressées. Comme dit précédemment, une image peut être découpée en une ou plusieurs tuiles. On suppose disposer ici d'images en couleur à trois composantes. Sur une tuile, on applique d'abord un changement d'espace colorimétrique. Ensuite, une transformation en ondelettes multi-niveaux est appliquée à chaque composante.

À chaque niveau de décomposition, on obtient 3 sous-bandes qu'on peut noter x_{HL} , x_{LH} et x_{HH} , au niveau de décomposition x . Au niveau le plus grossier, on a une seule sous-bande notée LL . Les coefficients issus de la transformation en ondelettes sont regroupés en code-blocs puis en precincts (cf. figure 3).

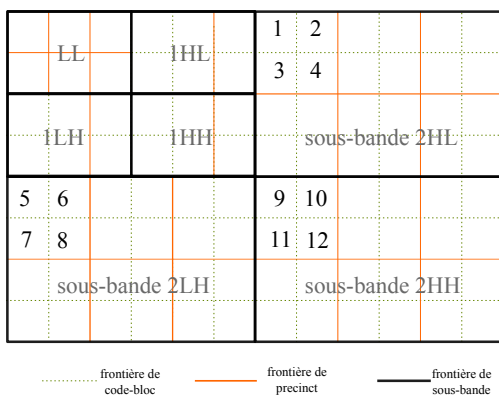


Figure 3 – Code-blocs et precincts pour la transformation en ondelettes d'une composante. Sur cet exemple, il y a 3 niveaux de décomposition. La taille d'un code-bloc est toujours la même, celle du precinct peut varier en fonction du niveau de décomposition. Les 6 precincts de la sous-bande LL sont plus petits que ceux des niveaux supérieurs. Ici, leur taille coïncide avec celle d'un code-bloc.

Les code-blocs sont toujours carrés avec un côté impliquant un nombre de coefficients en 2^n (où, généralement, $n \in [3,6]$). Des code-blocs issus de différentes sous-bandes sont regroupés en precincts, ces derniers correspondant à une même zone spatiale. Ainsi, dans la figure 3, les code-blocs numérotés 1, 2, 3, 4 de la sous-bande $2HL$ sont

regroupés avec les code-blocs 5 à 12 pour former un precinct. On voit aussi sur la figure 3 qu'un precinct de seulement 6 code-blocs est bâti depuis les sous-bandes $1HL$, $1LH$ et $1HH$ au second niveau de décomposition, comme conséquence de la taille rectangulaire de l'image initiale. Dans l'exemple de la figure 2, la charge utile d'un paquet correspond ainsi aux coefficients d'un precinct compressés selon un mécanisme que nous rappellerons dans le prochain paragraphe. Il faut noter que la compression JPEG2000 introduit aussi des couches de qualité au niveau des precincts et donc des paquets, que nous n'utiliserons pas pour le cinéma numérique.

3.3 Codage par plans de bits

La compression des coefficients d'ondelettes issus d'un même code-bloc s'opère selon le codeur contextuel de JPEG2000 nommé EBCOT, qui a été proposé par Taubman dans [5] et détaillé dans [6]. Dans le sens de la décompression, EBCOT est un décodeur à deux étages (Tier) : le Tier-1 est composé d'un décodeur par plans de bits et d'un décodeur arithmétique (MQ-Coder) ; le Tier-2 extrait les données depuis le train de bits.

Le principe de fonctionnement du Tier-1 est identique pour le codage et le décodage. EBCOT décode indépendamment chaque code-bloc d'un paquet. Le Tier-2 extrait depuis les paquets les données compressées de chaque code-bloc qu'il fournit au Tier-1. Le décodeur par plans de bits du Tier-1 applique à chaque plan trois passes : une passe de propagation de la signification (*significance propagation pass*), une passe d'affinage de l'amplitude (*magnitude refinement pass*) et une passe de nettoyage (*clean-up pass*). Quatre primitives de codage sont utilisées par ces passes (*run-length coding*, *zero coding*, *magnitude refinement coding* et *sign coding*). Le parcours des coefficients d'un code-bloc sur un plan de bits est effectué par bandes de quatre lignes ; chaque bande est parcourue par colonne de la gauche vers la droite, comme indiqué sur la figure 4.

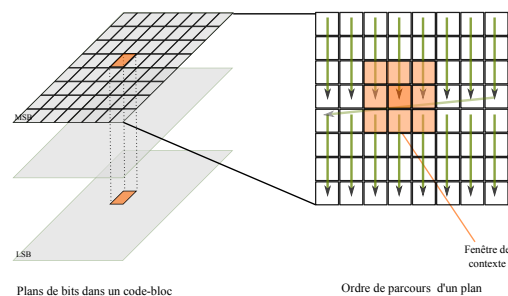


Figure 4 – Parcours en plans de bits dans un code-bloc de taille 8×8 .

La compression réalisée par EBCOT est contextuelle. Le contexte est une valeur générée par le décodeur de plans de bits en fonction des bits voisins, formant une fenêtrage de contexte (cf. figure 4) et de variables d'état (issues des

passes et des primitives de codage). Le MQ-Decoder utilise le contexte et le train de bits pour estimer les probabilités que le bit à décoder soit égal à 0 ou 1, sachant le contexte. Le MQ-Decoder adapte les distributions conditionnelles tout au long du parcours des plans de bits.

Il existe dans EBCOT des options (appelées RESET/TERMINATION) permettant de réinitialiser ces distributions de probabilités à chaque passe [6, p. 503]. L'option CAUSAL supprime quant à elle la dépendance entre bandes au sein d'un même plan binaire. Lors de la génération d'un contexte, on utilise en effet une fenêtre de contexte, des bits de cette fenêtre pouvant se retrouver dans la bande précédente ou dans la bande suivante : dans le cas de l'utilisation de l'option CAUSAL, ces bits sont supposés égaux à 0. Dans la suite (paragraphe 4), nous discutons de l'utilisation spécifique de JPEG2000 dans le cadre du cinéma avant d'envisager l'utilisation de telles options pour la parallélisation du décodage traitée dans le paragraphe 5.

3.4 Mesure de performances

Le tableau 3.4 compare les performances de décompression JPEG2000 des deux solutions libres les plus utilisées. L'image *crowd run* est réelle, tandis que *sintel* est une image de synthèse. Les deux images ont été compressées en JPEG2000 avec les options liées au cinéma numérique au format 2K. *OpenJpeg* est plus rapide pour décompresser, surtout au niveau de la transformation en ondelettes inverse (IDWT). *Jasper* est plus rapide sur le décodage Tier-1, car il gère moins d'options de codage qu'*OpenJpeg*.

	Image <i>crowd run</i>			Image <i>sintel</i>		
	Tier-1	IDWT	Tuile	Tier-1	IDWT	Tuile
<i>Jasper</i>	324	206	586	333	148	522
<i>OpenJpeg</i>	344	52	427	342	43	408

Tableau 2 – Comparaison des performances de décodage entre *OpenJpeg* et *Jasper* (en ms).

4 JPEG2000 et contraintes du cinéma numérique

Des options pour l'encodage JPEG2000 sont imposées par le DCI. En 2004, le DCI a choisi JPEG2000 comme format de compression. Tout d'abord, le DCI impose que chaque image soit compressée séparément. Le DCI a souhaité avoir un format ouvert [7], qui supporte l'encodage des images avec une profondeur de bits importante (12 bits) et qui supporte aussi le format d'espace colorimétrique CIE 1931 XYZ. En outre, un même fichier image doit pouvoir comporter les niveaux de définition du 2K (2048 × 1080) et du 4K (4096 × 2160). Les auteurs de [7] donnent les contraintes imposées par le DCI sur le JPEG2000, mais n'explicitent pas les raisons de ce choix : c'est ce que nous nous proposons de faire dans ce paragraphe.

Une image ne peut pas être décomposée en plusieurs tuiles. Une seule tuile est autorisée, le DCI craignant qu'à la fron-

tière entre deux tuiles, on ait une perte de qualité d'image. Une décomposition en tuiles est surtout utile pour de très grandes images, comme par exemple les images satellitaires.

Le DCI a choisi d'utiliser la compression avec perte. Avec la compression sans perte, l'espace disque utilisé par l'ensemble des images d'un film serait trop volumineux pour la transmission dématérialisée des copies numériques.

4.1 Contraintes sur la transformation en ondelettes

Lors de la compression avec perte, on applique une transformation en ondelettes irréversible. Le DCI régleme cette perte en imposant que, lors de la décompression, la transformation inverse utilise au moins 16 bits de précision, car le calcul est fait en virgule fixe.

Le DCI n'oblige pas l'utilisation de la transformation en ondelettes, elle en impose toutefois au moins une pour le format 4K, pour pouvoir en extraire une au format 2K, afin de pouvoir projeter sur un projecteur 2K une copie numérique au format 4K. Elle fixe aussi le nombre maximal de transformations à cinq pour le 2K, et à six pour le 4K. Pour la projection, il n'est pas nécessaire d'avoir plusieurs niveaux de décomposition. De plus, l'utilisation de nombreux niveaux de décomposition augmente le temps de calcul lors de la décompression. Néanmoins, l'utilisation de la transformation en ondelettes réduit la redondance des données.

4.2 Contraintes sur le codage Tier-2

Un code-bloc doit avoir comme taille 32×32 et la taille d'un precinct est 256×256 , sauf pour la bande de fréquences la plus faible (*LL*), qui a comme taille 128×128 . Pour une image 2K avec une fréquence de 24 images par seconde, le taux de compression maximal est égal à 1302083 octets par image (en cumulant les trois composantes et les en-têtes). De plus, le taux de compression maximal pour une couleur ne peut excéder 1041666 octets par composante (en-têtes de composantes compris). Une contrainte séparée sur le taux de compression des composantes permet qu'une composante ne soit pas trop prédominante par rapport à une autre et autorise ainsi une meilleure perception de l'image.

À titre de comparaison, la valeur 1302083 correspond à un débit maximal égal à 250 Mbits/s, ce qui constitue une valeur élevée par rapport aux débits du MPEG2 HD (50 Mbits/s) et du MPEG2-DVD (9 Mbits/s). Avec un tel débit, le DCI a choisi d'avoir la meilleure qualité de compression possible, au détriment de l'espace disque. Une copie numérique d'un film peut avoir une taille de 200 à 300 Go pour le format 2K, et jusqu'à plus de 500 Go pour le format 4K.

4.3 Contraintes sur la formation du train de bits

Il ne peut pas y avoir de région d'intérêt. Cette fonctionnalité n'est pas utile pour des images dédiées à la projection.

Les marqueurs *CODing style Default* (COD), *CODing style Component* (COC), *Quantization Default* (QCD) et *Quantization Component* (QCC) doivent apparaître uniquement dans l'en-tête principal du fichier, afin d'empêcher des codages spécifiques pour une composante de couleur et d'assurer une meilleure inter-opérabilité du train de bits du JPEG2000 [8, p. 41].

Dans les marqueurs COD et COC, le DCI bloque l'utilisation des options de codage CAUSAL, RESET et TERMINATION qui pourraient faciliter la parallélisation de EBCOT. Le fait de bloquer ces options de codage a comme but d'assurer, d'une part, une meilleure inter-opérabilité du train de bits, en imposant des restrictions sur les paramètres du JPEG2000. D'autre part, cela préserve l'efficacité de la compression. Les pertes dues à ces options seraient minimes selon [6, p. 502], mais permettent une parallélisation du traitement des plans de bits plus aisée.

L'ordre de progression de la décomposition d'une image doit être du type *Component-Position-Resolution-Layer*. Cela signifie que, lors du codage, on organise les precincts d'abord par composante de couleur, puis par position du precinct dans toutes les sous-bandes. Les sous-bandes sont parcourues dans l'ordre *LL, 1HL, 1LH, 1HH, 2HL, ... nHH* (cf. figure 3). La notion de couche de qualité n'a pas de sens car une seule couche de qualité est imposée par le DCI. Le DCI impose donc qu'une image soit décompressée selon un parcours spatial et qu'elle soit affichée totalement en une seule fois : ceci afin de ne pas avoir d'effet de transition entre deux images du flux vidéo.

4.4 Contrainte sur le décodeur

Le décodeur doit décomposer chaque composante de couleur avec 12 bits de profondeur pour chaque échantillon, et il ne doit pas faire de sous-échantillonnage chromatique. Sur ce point, nous nous écartons du standard du DCI, dans le cas du projecteur alternatif. De plus, le DCI spécifie que le décodeur ne doit pas ignorer des couches de qualité ou des niveaux de résolution pour gérer des pics de charge lors du décodage. Là aussi, nous pourrions nous permettre un écart dans le cas du projecteur alternatif, si nous montrons que la dégradation de l'image n'est pas perceptible à la projection.

5 Parallélisation logicielle du format JPEG2000

Les solutions les plus utilisées pour la compression et la décompression JPEG2000 en parallèle sont des solutions matérielles à base de FPGA [9, 6]. C'est ce type de solution que l'on retrouve chez tous les constructeurs de matériel de projection pour le cinéma numérique.

Le JPEG2000 est un codeur complexe demandant beaucoup de ressources de calcul. Les deux principaux verrous sont les transformations en ondelettes et surtout EBCOT (cf. [10], table 1). Dans ce paragraphe, nous nous intéressons au traitement parallèle logiciel de ce dernier, sur des ordinateurs à 4 ou 8 cœurs équipés d'une carte graphique

avec GPU. De tels ordinateurs sont classiques aujourd'hui. Dans [6, p. 508], les auteurs présentent deux modes, macroscopique et microscopique, du traitement parallèle d'EBCOT. Dans le mode macroscopique, on parallélise l'intégralité de l'encodage et le décodage des code-blocs, sachant que ces derniers sont traités indépendamment par EBCOT. En mode microscopique, on parallélise de manière plus fine, à l'intérieur d'un code-bloc, en cherchant à traiter séparément les plans de bits ou les bandes. Dans ce cas, il faut réussir à rendre indépendants les traitements du contexte et des variables d'état de EBCOT.

5.1 Mode macroscopique

Vu l'organisation des paquets des fichiers JPEG2000 du cinéma numérique, l'approche la plus simple consiste à traiter séparément chaque composante de couleur, car les données du fichier sont séparées en trois : une par composante. À l'intérieur d'une composante, on parallélise le traitement des code-blocs. De plus le fait de n'avoir qu'une seule couche de qualité facilite la décompression d'un code-bloc. La parallélisation du traitement des composantes pourrait être faite par différents cœurs et les opérations plus fines, comme les transformations en ondelettes et le traitement des code-blocs, par les GPU.

Dans le projet CUJ2K [11], les changements d'espace colorimétrique et les transformations en ondelettes sont traités en parallèle sur GPU, et EBCOT est réalisé par les différents CPU. De bons gains de performance sont atteints pour la transformation en ondelettes, mais 90 % du traitement est pris par EBCOT.

Dans [10], les auteurs expliquent que le mode macroscopique n'est pas adapté au traitement sur GPU. Le traitement d'EBCOT sur un code-bloc n'est pas adapté à l'architecture de type *Single-Instruction Multiple-Thread* (SIMT) des GPU, en raison de la complexité de l'algorithme. Lors du traitement d'EBCOT les nombreuses branches conditionnelles (*if, switch*) n'optimisent pas le temps d'utilisation des processeurs.

5.2 Mode microscopique

Les publications traitent plus souvent le cas de l'encodage que celui du décodage. Pour EBCOT, il y a une symétrie car la création du contexte nécessite le parcours du plan de bits dans les deux cas. Il est souligné dans [6, p.663] que l'implémentation parallèle logicielle du MQ-Decoder est plus facile à mettre en œuvre que son implémentation matérielle, mais les algorithmes parallèles du MQ-Decoder ne sont pas donnés car ils font partie du logiciel propriétaire Kakadu.

Dans [6, p. 508], les auteurs proposent une approche parallèle microscopique de l'encodage et du décodage avec EBCOT, en utilisant les options CAUSAL, RESET et TERMINATION. Elles permettent de traiter à l'intérieur d'un code-bloc les différentes passes en parallèle. Toutes les variables nécessaires au codage de chaque passe ne sont pas indépendantes, mais l'utilisation de ces options en réduit le nombre. On peut aussi légèrement décaler dans le

temps le traitement d'une passe pour une bande associée. Par exemple, dès que les deux premières colonnes d'une bande pour une passe sont traitées, on peut lancer le traitement de la passe suivante sur la même bande.

5.3 Repenser EBCOT ?

Les auteurs de [10] expliquent qu'il est nécessaire de repenser EBCOT pour l'adapter aux architectures des GPU. Ils proposent un codeur par plans de bits et un codeur arithmétique parallèle. Le codeur par plans de bits décrit est celui du JPEG2000. Par contre, le codeur arithmétique utilisé n'est pas le MQ-Coder. La principale contribution apportée par [10] est la suppression des dépendances entre bits d'un même plan. La suppression des dépendances est réalisée en faisant de la prédiction sur les variables d'état du codeur par plans de bits. L'algorithme qui suit cette méthode de prédiction n'est pas précisé dans l'article. Les résultats d'encodage avec cette méthode montrent que, plus la taille des code-blocs choisie est petite, plus les GPU sont utilisés, et donc plus l'encodage est réalisé rapidement. Pour augmenter la rapidité d'encodage sur code-blocs plus grands (64×64), il est préconisé d'utiliser les options CAUSAL, RESET et TERMINATION car elles suppriment les dépendances inter-bandes et inter-plans de bits.

Une approche microscopique pour implémenter le codeur par plans de bits sur GPU est également proposée dans [12], qui va dans le même sens que [10] : la parallélisation par code-bloc n'est pas assez fine, et il est donc nécessaire de paralléliser à l'intérieur d'un code-bloc le codage de chaque bit d'un plan. Les auteurs proposent aussi d'utiliser la prédiction de contexte. L'utilisation de l'option CAUSAL ne permettant pas un parallélisme assez fin, ils proposent une méthode ne l'utilisant pas. Leur méthode supprime les trois passes d'EBCOT. L'équivalence avec la méthode à trois passes est prouvée de manière formelle.

6 Conclusion et perspectives

Cet article a pour but de présenter un système de projection alternatif adapté aux salles de cinéma indépendantes, et de détailler les contraintes imposées par le cinéma numérique, au travers de la spécification du DCI, sur le format JPEG2000. Pour pouvoir projeter des copies numériques, il faut paralléliser le traitement de JPEG2000 et l'adapter à l'architecture des cartes graphiques avec GPU.

Pour la décompression du JPEG2000, notre prochaine étape consistera à implémenter des prototypes sur GPU basés sur les approches de [10] et [12], qui sont similaires.

En ce qui concerne le décodage et la projection, nous nous permettrons quelques écarts au standard du DCI pour les systèmes de projection alternatifs. Nous nous donnerons notamment la possibilité de ne pas décoder entièrement une image, ou bien de faire du sous-échantillonnage chromatique si l'information perdue induit une baisse de qualité imperceptible à la projection.

Les prototypes seront basés sur le logiciel libre OpenJpeg, qui n'implémente pas encore d'algorithme parallèle.

Le projet de recherche, au-delà du contexte technique, vise à garantir à tous les films, notamment aux films à petit budget, l'accès aux salles obscures, et ainsi à préserver la diversité culturelle cinématographique.

Remerciements : ce sujet de recherche est soutenu par les cinémas Utopia au travers de l'association ISF (Indépendants, Solidaires et Fédérés).

Références

- [1] J.-M. Mas Ribés, S. Baudry, P. Correa, D. Doyen, B. Macq, Y. Maetz, O. De Marneffe, P. Nguyen, J.-F. Nivart, et I. Wolf. *European Digital Cinema Security Whitebook*. Presses Universitaires de Louvain, 2007.
- [2] DCI. *Digital Cinema System Specification*. Digital Cinema Initiatives, 1.2 édition, Mars 2006.
- [3] N. Bertrand. *Cinéma numérique : état de l'art d'un point de vue Art et Essai*. Mémoire de stage de master 2 recherche, Université Paul Sabatier, Toulouse, Juillet 2011.
- [4] C. Angelopoulos. *On opportunities and challenges for european cinema in the digital era*. Rapport technique, European Commission, Septembre 2010.
- [5] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7) :1158–1170, Juillet 2000.
- [6] D.S. Taubman et M.W. Marcellin. *JPEG2000 : image compression fundamentals, standards, and practice*, volume 1 de *The Kluwer international series in engineering and computer science*. Kluwer Academic Publishers, 2002.
- [7] A. Bilgin et M.W. Marcellin. JPEG2000 for digital cinema. Dans *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems – ISCAS 2006*, pages 3878–3881. IEEE, Mai 2006.
- [8] ITU-T. *Information technology – JPEG 2000 image coding system – Part 1 : Core coding system*, Août 2002.
- [9] G. Rouvroy, F.-X. Standaert, F. Lefèbre, J.-J. Quisquater, Benoît Macq, et J.-D. Legat. Reconfigurable hardware solutions for the digital rights management of digital cinema. Dans *Proceedings of the 4th ACM Workshop on Digital Rights Management, DRM'04*, pages 40–53, 2004.
- [10] R. Le, I.R. Bahar, et J.L. Mundy. A novel parallel Tier-1 coder for JPEG2000 using GPUs. Dans *Proceedings of the 9th IEEE Symposium on Application Specific Processors – SASP 2011*, pages 129–136, Juin 2011.
- [11] D. Weiss et M. Heide. CUJ2K : a JPEG2000 encoder in CUDA. Septembre 2009.
- [12] J. Matela, V. Rusnák, et P. Holub. Efficient JPEG2000 EBCOT Context Modeling for Massively Parallel Architectures. Dans *Proceedings of the 2011 Data Compression Conference*, pages 423–432, Mars 2011.