

# Codage d’atlas de textures avec prédiction guidée par la topologie des maillages

Fatemeh NASIRI<sup>1</sup>, Navid MAHMOUDIAN BIDGOLI<sup>1</sup>, Frédéric PAYAN<sup>2</sup>, Thomas MAUGEY<sup>1</sup>

<sup>1</sup>Inria, Univ Rennes, CNRS, IRISA  
Campus de Beaulieu, 35042 Rennes

<sup>2</sup>Université Côte d’Azur, CNRS, I3S  
2000, Route des Lucioles, 06903 Sophia Antipolis

<sup>1</sup>fatemeh.nasiri@irisa.fr, navid.mahmoudian-bidgoli@inria.fr,  
thomas.maugey@inria.fr, <sup>2</sup>frederic.payan@univ-cotedazur.fr

**Résumé** – Nous présentons une technique de codage spécifique aux atlas de texture de modèles 3D. Inspirée des techniques utilisées en compression vidéo, notre approche exploite la topologie des maillages associés pour prédire de manière plus efficace les images de texture. Nous présentons aussi un nouvel ordonnancement des blocs, que nous avons combiné à une transformée basée graphe et à un codeur résiduel. Les résultats montrent que la prise en compte de la topologie du maillage améliore significativement les performances de compression des textures, notamment des cartes de diffusion.

**Abstract** – This paper proposes a novel prediction tool for improving the compression performance of texture atlases. Our algorithm takes advantage of the topology of the associated 3D meshes, in order to reduce the redundancies in the texture map. The concept of intra prediction used in video compression has been adapted to consider neighboring information on the 3D surface. We have then combined this prediction tool with an original block scanning strategy and a graph-based transform for residual coding. Results show that the knowledge of the mesh topology significantly improves the compression efficiency of texture atlases.

## 1 Introduction

Avec les développements récents des systèmes d’acquisitions 3D, les modèles 3D sont de plus en plus réalistes. Ils sont donc de plus en plus utilisés dans de nombreux domaines : architecture, héritage culturel, réalité virtuelle, augmentée, *etc.* Par conséquent, la transmission de ce type de données *via* les réseaux actuels est largement étudiée aussi bien dans le monde académique qu’industriel.

Pour qu’il soit réaliste, un modèle 3D est généralement composé d’un maillage (géométrie + connectivité) associé à une texture. Une texture peut être définie par une simple image RGB que l’on va plaquer sur le maillage. Pour des objets à l’aspect plus complexe, la texture peut être définie à l’aide d’un atlas qui contient une multitude de fragments d’images, comme le montre la Figure 1.

La compression de maillages 3D a été largement étudiée [6]. La compression de textures a été aussi étudiée [2, 4] dans un contexte de rendu efficace, c’est à dire pour permettre une décompression en temps réel au moment de l’affichage. Par contre, elle a été peu étudiée dans un contexte de communi-

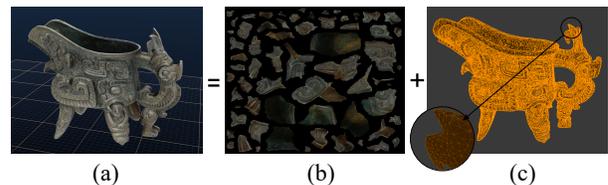


FIGURE 1 – Un modèle 3D réaliste (a) peut s’obtenir en ”plaquant” une texture 2D (b) sur un maillage surfacique triangulaire (c).

tion interactive, entre un serveur et un client qui reçoit l’information à afficher en fonction de la navigation de l’utilisateur.

Les textures sont souvent encodées sous des formats standards d’images, en JPEG par exemple. JPEG peut atteindre des taux de compression de 50 [4], mais reste néanmoins moins efficace que les techniques d’*encodage intra* utilisées en compression vidéo [7]. Pour exploiter d’avantage les redondances spatiales, une technique conventionnelle de codage intra partitionne une image 2D en *blocs*, et utilise ensuite les blocs adjacents déjà encodés pour prédire un bloc donné. L’erreur de prédiction qui en résulte, appelée aussi *signal résiduel*, est ensuite projetée dans un espace transformé pour éliminer la corrélation restante. Ces techniques performantes de codage *intra* ont été intégrés dans *Advanced Video Coding* (AVC) [8] et dans

Ce travail a été financé en partie par le GDR ISIS, dans le cadre du projet exploratoire ICON3D ainsi que par le laboratoire d’excellence Cominlabs, supervisé par l’ANR dans le cadre du programme ”Investing for the Future” avec la référence ANR-10-LABX-07-01.

High Efficiency Video Coding (HEVC) [10].

Un problème survient si l'on souhaite utiliser ce type de codage sur un atlas de texture. En effet, des triangles adjacents sur un maillage surfacique ne sont pas forcément associés à des régions adjacentes dans son atlas de texture, même si visuellement la texture sur la surface semble continue. Comme le montre la Figure 1, un atlas de texture est en fait un ensemble de fragments d'images déconnectés que l'on appellera *patches* par la suite. En conséquence, si on découpe un tel atlas en blocs pour effectuer un codage intra, de nombreux blocs seront à cheval sur les bords des patches, et n'auront donc pas un voisinage pertinent pour les prédire. La Figure 2 illustre ce problème : pour le bloc noté C, aucun des blocs voisins notés A et B ne sera efficace lors de la prédiction puisque ces blocs ne contiennent que des pixels noirs. Pour de tels blocs, un codage intra conventionnel n'est donc pas approprié.

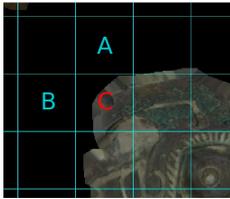


FIGURE 2 – Dans un contexte conventionnel de codage *intra*, la prédiction du bloc C à partir des blocs A et B est inefficace.

Durant ce travail nous supposons que le maillage est déjà encodé (géométrie et connectivité), et disponible au décodeur. Ainsi, notre prédiction pour les textures peut exploiter la topologie du maillage - notamment l'adjacence des triangles - afin d'être plus pertinente. Par conséquent, au lieu d'une prédiction conventionnelle par blocs adjacents dans l'atlas de texture qui peut s'avérer inefficace comme illustré précédemment, notre première proposition est de prédire un bloc de texture donné à partir de la *texture adjacente sur la surface du maillage* : voir Section 2. Ensuite, un atlas de texture étant composé de nombreux patches déconnectés, notre deuxième proposition est un nouvel ordonnancement des blocs également basé sur la connaissance du maillage, comme alternative au balayage classique proposé pour la vidéo : voir Section 3. Enfin, notre troisième contribution durant ce projet est d'utiliser une transformation basée graphe pour décorrélérer le signal résiduel des blocs en bordure de patches : voir Section 4.

## 2 Prédiction *intra*

D'une manière très générale, en compression vidéo, la prédiction *intra* est une technique qui permet de prédire des blocs de pixels à partir de pixels de blocs voisins déjà encodés. Par exemple, HEVC utilise une ligne et une colonne de pixels adjacents à un bloc (plus précisément la colonne de gauche et la ligne supérieure) comme référence pour prédire ce dernier le mieux possible parmi 35 modes prédéfinis (incluant la valeur moyenne, le mode planaire et 33 modes directionnels). Dans

notre contexte de compression de texture, une telle approche ne serait pas efficace dans de nombreux cas, car incapable de distinguer les parties pertinentes (les patches) et les parties non pertinentes (le "fond noir") de l'atlas. Pour les blocs en bordure de patches, cela produirait des signaux résiduels de grande énergie, puisqu'on essaierait de prédire ces pixels de texture à partir uniquement de pixels noirs (voir Figure 3).

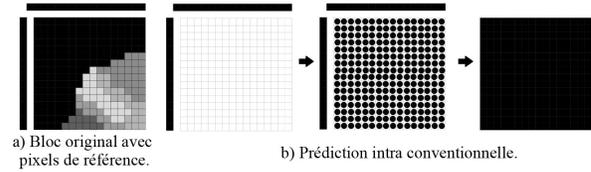


FIGURE 3 – Prédiction conventionnelle d'un bloc en bordure de patch (a). b) A gauche, pixels de référence ; au milieu, propagation par projection ; à droite, prédiction par interpolation.

Pour pallier ce problème, nous exploitons la topologie du maillage pour utiliser des pixels de références corrélés aux pixels du bloc que l'on souhaite prédire. Nous rappelons que les patches contenus dans un atlas de texture sont plaqués sur la surface de son maillage au moment du rendu, donnant l'impression qu'une seule texture a été plaquée de manière continue sur l'ensemble de l'objet. Pourtant, deux triangles adjacents dans le maillage peuvent récupérer leur texture dans deux patches éloignés dans l'atlas, et indépendants. C'est cette information que l'on va utiliser pour notre codage *intra*.

La Figure 4 présente une vision globale de notre approche : la Figure 4-(a) montre la partie d'un maillage définissant un objet 3D. Notre algorithme utilise les relations d'adjacence entre triangles pour déterminer les pixels de référence utilisés pour la prédiction d'un bloc à cheval sur la bordure d'un patch. Pour cela, on commence par projeter les triangles du maillage sur l'atlas, à l'aide des coordonnées de texture de chaque triangle (Figure 4-(b)). Ensuite, pour prédire la texture correspondant au triangle F7 par exemple (Figure 4-(c)), on va utiliser les pixels de son triangle voisin sur la surface, à savoir F1, bien que présent dans un autre bloc et éloigné dans l'atlas. Pour cela, on va "recopier" les pixels au bord de l'arête commune à F1 et F7, et on se servira de ces derniers comme pixels de référence lors de la prédiction de pixels de F7. Bien évidemment cela ne marchera que si les pixels de référence issus de F1 ont déjà été codés dans leur propre bloc (problème abordé dans la section suivante).

Avec une telle approche, les pixels de référence qui ont été recopiés aux bords des patches ne sont pas positionnés de manière rectiligne et orthogonale (selon une ligne en bas et une colonne à gauche) le long du bloc à prédire, comme c'est le cas avec les techniques de prédiction intra conventionnelles. Ceci rend plus complexe le traitement. Par exemple, l'algorithme de prédiction d'HEVC effectue une interpolation linéaire entre les pixels de référence afin d'obtenir les valeurs estimées d'un bloc [5]. Ce procédé, appelé "projection arrière", requiert d'avoir

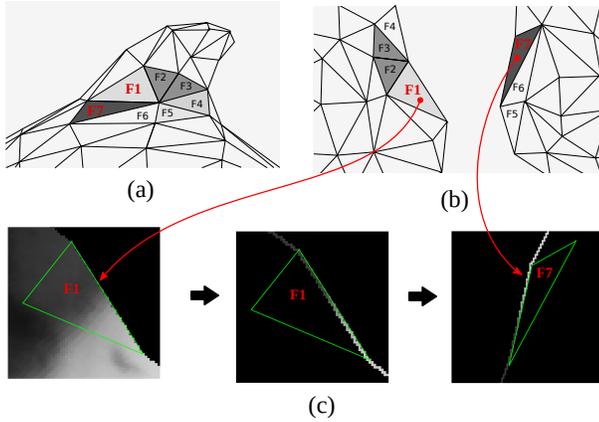


FIGURE 4 – Vision globale de notre prédiction. Le maillage d’entrée (a) est paramétré sur son atlas de texture (b). Cette paramétrisation est ensuite utilisée pour que les pixels contenus dans le triangle F7 soient prédits à partir des pixels du triangle adjacent F1 situés le long de l’arête commune, bien que distants dans l’atlas de texture.

des pixels de référence qui soient bien alignés en ligne et colonne, ce qui n’est pas le cas dans notre prédiction.

Afin de résoudre ce problème, notre algorithme effectue plutôt une projection “avant” des pixels de référence sur la grille, comme indiqué sur la Figure 5. L’algorithme procède en deux étapes : 1) projection “avant” des pixels de référence sur des positions non entières dans le bloc, 2) interpolation à partir des quatre voisins les plus proches pour obtenir les valeurs entières. A titre de comparaison, les Figures 3 et 5 montrent les résultats de prédiction en utilisant respectivement les méthodes conventionnelle et proposée.

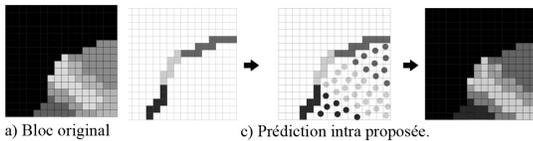


FIGURE 5 – Technique de prédiction proposée pour un bloc en bordure de patch (a). b) A gauche, pixels de référence ; au milieu, propagation ; à droite, prédiction par interpolation.

### 3 Ordonnement des blocs

L’utilisation de pixels de référence provenant potentiellement d’un bloc non adjacent au bloc que l’on souhaite prédire rend le balayage conventionnel impossible. En effet, avec notre approche, on risque d’avoir besoin de pixels de référence qui n’ont pas encore été codés, puisque l’on dépend de la topologie du maillage et non pas du parcours classique d’une image. Pour éviter l’apparition de ce problème, nous proposons un ordonnancement qui tire profit de la topologie du maillage, et notamment de l’adjacence des triangles.

Notre technique part d’un bloc qui contient le centroïde du

plus grand patch présent dans l’atlas. Une fois ce premier bloc codé, on choisit le prochain bloc à coder en suivant deux règles :

- **règle 1** : les candidats avec le plus grand nombre de blocs sont prioritaires.
- **règle 2** : en cas d’égalité, ceux dont les blocs de référence sont situés en haut à gauche sont prioritaires.

En partant du centroïde, tous les blocs du premier patch (*i.e.*, le plus grand) peuvent appliquer la technique conventionnelle de prédiction intra avec parfois la référence située en bas à gauche plutôt qu’en haut à droite. Pour le parcours des autres patches, nous détectons d’abord les triangles positionnés à la bordure des patches. Ensuite, nous trouvons leurs voisins dans les autres patches, grâce au maillage 3D. Dans notre méthode, le patch à remplir est choisi comme celui possédant le plus de pixels voisins dans sa bordure.

## 4 Codage résiduel

Les méthodes usuelles pour coder les résidus effectuent d’abord une Transformée en Cosinus Discrète (DCT), afin de compacter l’information pour la transmission. Malgré les nombreux avantages de la DCT, l’appliquer sur des blocs à la frontière d’un patch n’est pas efficace. Cela est dû au fait que la partie informative d’un tel bloc n’est pas carrée, ce qui impliquerait une grande énergie située dans les hautes fréquences.

Afin de résoudre ce problème, notre méthode utilise la connectivité du maillage pour détecter la frontière dans un bloc et appliquer une transformée basée graphe (GFT) [9] sur le résidu. La GFT est calculée de la manière suivante :

**Etape 1** Estimer un masque  $M$  à partir de l’information géométrique, qui indique quels pixels du bloc sont informatifs.

**Etape 2** Construire un graphe non-dirigé  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$  à partir de ce masque, dont les noeuds correspondent aux pixels dont la valeur du masque vaut 1. L’ensemble des arêtes relie les noeuds du graphe à ses 4 plus proches voisins (comme dans une image classique).

**Etape 3** Construire une matrice d’adjacence  $\mathbf{A}$  à partir des informations de connectivité.

**Etape 4** Calculer le degré de chaque noeud et construire une matrice de degré  $\mathbf{D}$  dont les éléments sur la diagonale correspondent au degré de chaque noeud.

La soustraction de la matrice  $\mathbf{A}$  avec  $\mathbf{D}$  forme le Laplacien du graphe,  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , qui est une matrice symétrique semi-positive et qui peut être diagonalisée :  $\mathbf{L} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U}$ , où  $\mathbf{U}$  est une matrice de vecteurs propres et  $\mathbf{\Lambda}$  une matrice diagonale contenant les valeurs propres. La matrice  $\mathbf{U}$  est utilisée pour calculer les coefficients de la GFT :  $\hat{R} = \mathbf{U} \mathbf{R}$ . La transformée inverse est calculée avec  $\mathbf{R} = \mathbf{U}^T \hat{R}$ . Un intérêt fort de la GFT est que le nombre de coefficients est exactement égal au nombre de noeuds dans le graphe, et donc au nombre de pixels informatifs dans le graphe. Ces coefficients sont ensuite quantifiés et codés grâce à un codeur arithmétique.

## 5 Resultats expérimentaux

Nous avons implémenté l’approche proposée dans une solution de codage complète. Les données testées sont des modèles 3D d’objets anciens [1]. Nous nous sommes focalisés uniquement sur la prédiction des cartes de diffusion qui représentent l’aspect naturel des objets. La méthode proposée, notée *GA* (pour *geometry aware*) est comparée à la méthode conventionnelle de prédiction intra, en terme de métrique de Bjontegaard (BD-R) [3]. Différents éléments de comparaison sont générés. Le premier (Anc-1) utilise un scanning conventionnel (dans le sens de la lecture) et une prédiction intra classique. Toutefois, cette méthode utilise la GFT pour coder les résidus. Le deuxième algorithme (Anc-2) diffère de Anc-1 par le fait qu’une DCT est utilisée en place de la GFT.

Par soucis de simplicité et sans perte de généralité, les performances sont évaluées selon deux conditions : un même codeur arithmétique est utilisé pour l’ensemble des schémas, et une taille de bloc fixe est utilisée pour toute l’image et chacune des méthodes. Toutefois, les résultats présentés sont des moyennes de chacun obtenus pour différentes tailles de blocs :  $16 \times 16$ ,  $32 \times 32$  et  $64 \times 64$ . Les PSNR ne sont calculés que sur les pixels informatifs reconstruits.

Le Tableau 1 compare les différentes configurations à l’algorithme *GA* proposé, en terme de BD-R. Chaque valeur négative indique le pourcentage de gain de compression obtenu avec *GA* par rapport aux deux autres méthodes testées. La Figure 6 montre les courbes débit-distortion pour chacune des configurations. Les résultats obtenus indiquent clairement un gain de compression apporté par chacun des outils proposés (prédiction, balayage et GFT).

Sample	Anc-1	Anc-2
Donnée-1	-16.1%	-29.6%
Donnée-2	-11.0%	-23.2%
Donnée-3	-17.3%	-19.4%

TABLE 1 – Performances de la méthode proposée (*GA*) et les deux méthodes de références en terme de BD-R (%).

## 6 Conclusion

Nous proposons une nouvelle approche pour compresser des textures de modèles 3D. L’idée est d’adapter le principe du codage intra proposé en vidéo aux spécificités de nos données d’entrée, à savoir des atlas de texture. Ce travail contient plusieurs contributions significatives, notamment pour les blocs de texture se trouvant à cheval sur les bords des patches : i) une prédiction guidée par l’information d’adjacence des triangles du maillage associé ; ii) un ordonnancement adapté à cette prédiction ; iii) l’utilisation d’une transformation basée graphe pour décorréler de manière plus efficace le signal résiduel. Notre prochaine étape est d’intégrer l’approche proposée dans des codecs existants, notamment *HEVC test Model (HM)*, afin de bénéficier de toutes les options proposées : CABAC, partitionnement par blocs flexibles *etc.*

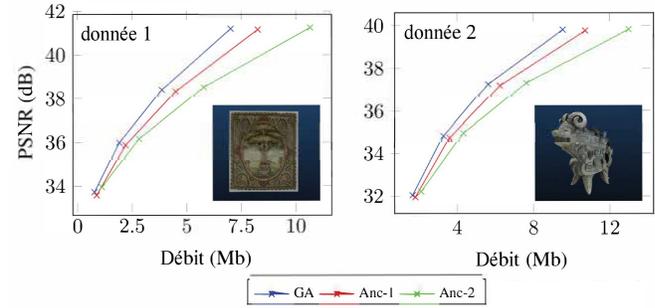


FIGURE 6 – Courbes débit-PSNR de l’algorithme proposé (*GA*) vs deux configurations de référence.

## Références

- [1] Smithsonian institution 3D program. <https://3d.si.edu/>. Accessed : 2018-10-29.
- [2] Andrew C. Beers, Maneesh Agrawala, and Navin Chaddha. Rendering from compressed textures. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96*, pages 373–378, New York, NY, USA, 1996. ACM.
- [3] Gisle Bjontegaard. Calculation of average PSNR differences between RD-curves. *VCEG-M33*, 2001.
- [4] Pavel Krajcevski, Srihari Pratapa, and Dinesh Manocha. GST : GPU-decodable supercompressed textures. *ACM Trans. Graph.*, 35(6) :230 :1–230 :10, November 2016.
- [5] J. Lainema, F. Bossen, W. Han, J. Min, and K. Ugur. Intra coding of the HEVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12) :1792–1801, Dec 2012.
- [6] Adrien Maglo, Guillaume Lavoué, Florent Dupont, and Céline Hudelot. 3D mesh compression : Survey, comparisons, and emerging trends. *ACM Comput. Surv.*, 47(3) :44 :1–44 :41, February 2015.
- [7] T. Nguyen and D. Marpe. Performance analysis of HEVC-based intra coding for still image compression. In *2012 Picture Coding Symposium*, pages 233–236, May 2012.
- [8] M. A. Saleh, H. Hashim, N. M. Tahir, and E. Hisham. Review for high efficiency video coding (HEVC). In *2014 IEEE Conference on Systems, Process and Control (ICSPC 2014)*, pages 141–146, Dec 2014.
- [9] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs : Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3) :83–98, May 2013.
- [10] G. J. Sullivan and T. Wiegand. Video compression - from concepts to the H.264/AVC standard. *Proceedings of the IEEE*, 93(1) :18–31, Jan 2005.