HAL
open science

Génération, Compression et Rendu de LDI

Vincent Jantet, Luce Morin, Christine Guillemot

To cite this version:

Vincent Jantet, Luce Morin, Christine Guillemot. Génération, Compression et Rendu de LDI. Coresa, Oct 2010, Lyon, France. 6 p. hal-00530752

HAL Id: hal-00530752

https://hal.science/hal-00530752

Submitted on 29 Oct 2010

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.

Génération, Compression et Rendu de LDI

Vincent Jantet [1] Luce Morin [2] Christine Guillemot [1]

[1] INRIA Rennes, Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes – France

[2] IETR - INSA Rennes, 20 avenue des Buttes de Coësmes, 35043 Rennes – France

Résumé

Ce papier présente l'utilisation de Layered Depth Images (LDI) dans le cadre de la compression de vidéo MVD. Leur construction incrémentale (I-LDI) permet l'élimination d'une grande partie des corrélations inter-vues, sans nécessiter de caméras rectifiées. Leur compression par le codeur MVC permet en plus l'exploitation des corrélations temporelles. Une solution aux effets de fantômes (Ghosting) est proposée, basée sur une détection de contours dans les cartes de profondeur, suivie d'une classification locale premier plan/arrière plan. L'élimination des découvrements (Disocclusions) et craquelures de textures est permise par une projection ordonnée des pixels, suivie d'une projection inverse. Les tests sur des vidéos MVD montrent une réduction de 80% du nombre de pixels sur les couches supplémentaires des I-LDI. La comparaison des courbes débit-distorsion de MVC appliqué sur les vues, et de MVC appliqué sur les couches I-LDI, montre un apport de qualité pour des débits faibles.

Mots clefs

Codage vidéo, Réalité virtuelle, Vidéo Multi-Vues, Layered Depth Video, Interpolation de vues.

1 Introduction

La banalisation des dispositifs stéréoscopiques impose l'ajout d'une nouvelle dimension aux données multimédias classiques. L'acquisition d'une scène par plusieurs caméras synchronisées (MVV pour *Multi-View Video*) est un des procédés permettant l'estimation de la géométrie de la scène. La connaissance de cette géométrie est indispensable aux fonctionnalités 3D comme l'affichage en relief (3DTV) ou la sélection de points de vue (FVV pour *Free Viewpoint Video*).

Les informations géométriques peuvent être de différentes formes (maillage, fonction plénoptique, polygones, cartes de profondeur, ...). Chacune de ces formes constitue un format de données indépendant, et est associée à un algorithme de synthèse de vues approprié. Ces algorithmes de rendu peuvent être classés selon deux catégories, en fonction du type d'informations géométriques qu'ils utilisent.

- Les rendus basés géométrie (GBR) utilisent un modèle 3D évolué et très détaillé. Ils sont idéaux pour le rendu de scènes synthétiques, mais inadaptés à des données réelles, où le modèle géométrique est difficile à obtenir :
- Les rendus basés images (IBR) nécessitent une géométrie moins détaillée (souvent des cartes de profondeur), mais ont besoin de plus d'information de texture. Les données d'entrée sont moins compactes, mais les images synthétisées sont de meilleure qualité et plus réalistes.

Les LDI (pour *Layered Depth Images*) [1, 2] constituent l'une de ces approches IBR. La géométrie est modélisée par une image en plusieurs couches, où chaque couche est accompagnée de la carte de profondeur correspondante. Chaque point dans l'image en couche est donc associé à un ensemble de pixels, constitués chacun d'une couleur et d'une profondeur. Ces pixels représentent les objets (visibles ou masqués dans la scène) traversés par un rayon reliant le centre optique de la caméra au point image considéré. Cette représentation selon un seul point de vue, réduit efficacement les redondances inter-vues, tout en permettant une synthèse de vue réaliste pour des points de vue éloignés du point de vue de référence.

Deux méthodes de construction de LDI [3] sont détaillées dans la section 2, dont une approche incrémentale permettant d'éliminer les corrélations entre les couches. La compression de ces LDI [4, 5] par le codeur MVC [6] est analysée dans la section 3. Une méthode de synthèse de vues permettant de combler les découvrements et d'éliminer les effets de fantôme est décrite dans la section 4. Les résultats présentés dans la section 5 permettent la validation des trois étapes mises à la chaîne : Construction, Compression puis Rendu.

2 Construction des LDI

Les méthodes de construction de LDI s'appuient en entrée sur des vidéos multi-vues [3], où chaque vue est accompagnée de sa carte de profondeur (MVD pour *Multi-View + Depth*). Le logiciel de référence pour les construire (MVD2LDV du groupe MPEG [7]) présente un certain nombre de limitations :

- Les caméras d'entrée doivent être rectifiées, c'est-à-dire que les lignes épipolaires sont toutes alignées horizontalement. Cette hypothèse simplifie les équations de projection (présentées dans la section 2.1), mais impose aux caméras d'avoir des poses précises et des paramètres intrinsèques identiques. Cette propriété, difficile à avoir

dès l'acquisition, est généralement obtenue par un traitement sur les vues acquises.
– Le nombre de vues en entrée est limité à trois, et le nombre de couches générées est limité à deux.

Deux approches sont proposées pour remédier à ces limitations. La première, un peu naïve, est présentée dans la section 2.2. Simple à mettre en œuvre, elle présente le défaut de générer un grand nombre de couches, partiellement corrélées les unes aux autres. La seconde approche, détaillée dans la section 2.3, est une construction incrémentale. Les couches générées sont moins nombreuses, et moins remplies, ce qui facilite leur compression ultérieure.

2.1 Equations de projection

Dans une image, chaque pixel p est la projection perspective d'un point 3D M selon les équations (1). Ces équations nécessitent la connaissance des paramètres intrinsèques K, de la matrice de rotation R et de la position t de la caméra dans le repère global.

Les équations inverses (2) permettent de retrouver la position 3D $M = (X, Y, Z)$ d'un point, à partir du pixel p et de sa profondeur Z dans l'image.

$$\omega p = KR^{-1}(M - t) \tag{1}$$

$$M = \omega RK^{-1}p + t \tag{2}$$

avec ω le coefficient de normalisation.

Ces équations sont utilisées par les techniques de synthèse de vues basées sur des cartes de profondeur [2, 3]. Elles génèrent des effets de craquelures sur les textures et des zones de découvrements qui seront détaillés dans la section 4. Ces paramètres peuvent être estimés à partir des vidéos MVD, qui dans certains cas peuvent être rectifiés. Nous supposerons donc dans la suite que les paramètres caméras (K, R et t) sont connus.

2.2 Construction naïve des LDI

Une LDI est un concentré d'information selon un même point de vue. La méthode de construction naïve consiste donc à projeter toute l'information présente dans la MVD selon un unique point de vue de référence, puis à fusionner cette information et à l'organiser par couches.

Le point de vue de référence peut être choisi quelconque, et n'a pas à être identique à un des points de vue de la MVD. Les vues sont considérées comme des ensembles de pixels, chacun ayant une couleur et une profondeur. Tous les pixels de chaque vue sont projetés selon le point de vue de référence. Lorsque deux pixels p_1 et p_2, des vues respectives V_1 et V_2, représentent le même point 3D M, alors ils doivent être projetés sur le même pixel p' selon le point de vue de référence, et à la même profondeur Z. Pour éviter cette redondance, un test sur la profondeur est utilisé et les pixels trop proches (seuil Δ_Z) sont fusionnés. Cette construction simpliste est schématisée dans la figure 1.

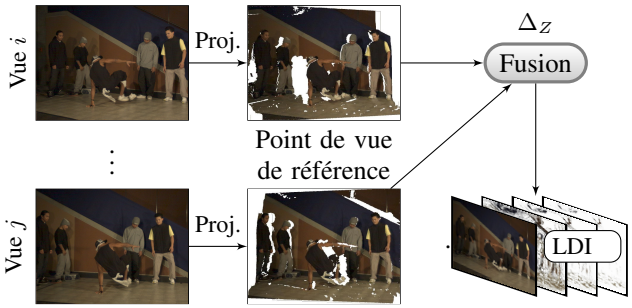


Figure 1 – *Schéma de construction naïf.*

Les trois premières couches d'une LDI sont présentées sur la figure 2. Chaque couche contient l'ensemble des pixels les plus proches de la caméra, qui ne soient pas déjà dans une des couches précédentes. Ainsi, la première couche contient tous les pixels visibles, alors que les autres couches sont partiellement vides car ne contiennent que les pixels masqués par un objet de la scène.

(a) 1ere couche (b) 2eme couche (c) 3eme couche

Figure 2 – *Premières couches d'une LDI naïve (Ref. = Vue 4) générée en utilisant les 8 caméras d'acquisition. ($\Delta_Z = 0.1$)*

Le seuil Δ_Z est critique pour l'élimination des redondances, mais difficile à déterminer à l'avance. Une autre méthode de construction est donc proposée, utilisant un critère totalement différent.

2.3 Construction incrémentale des LDI

Pour éliminer les corrélations entre les vues, une autre construction est proposée, désignée par I-LDI, pour Incremental-LDI [8]. Basée sur l'extraction d'information résiduelle [9], elle permet de réduire le nombre de couches et leur taux de remplissage.

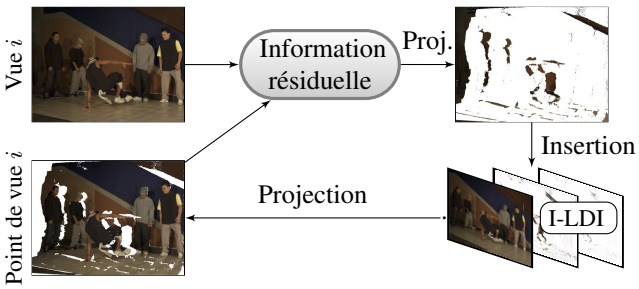


Figure 3 – *Extraction de l'information résiduelle de la vue i lors d'une itération de la construction I-LDI.*

L'algorithme peut se décomposer de la façon suivante :

Initialisation : Le point de vue de référence est fixé arbitrairement. Il n'a pas besoin d'être un des points de vue des caméras d'acquisition. La I-LDI est initialisée vide, elle ne contient aucun pixel.

Itérations : La I-LDI est projetée selon le point de vue d'une caméra d'acquisition, ce qui génère des zones de découvrement. Cette image virtuelle est comparée à l'image acquise de ce point de vue. L'ensemble des pixels permettant de combler les zones de découvrement est appelé information résiduelle. Seule cette information résiduelle est re-projetée selon le point de vue de référence pour être ajoutée dans la I-LDI. Cette étape est réitérée pour chacune des caméras d'acquisition, ajoutant de plus en plus de pixels dans la I-LDI.

Finalisation : L'ensemble des pixels est réparti par couches, chaque couche contenant les pixels les plus proches, non déjà contenus dans une couche précédente. Par construction, la première couche est l'image 2D+Z observée selon le point de vue de référence.

Les trois premières couches d'une I-LDI sont présentées dans la figure 4. La seconde couche ne contient que les textures des zones réellement occultées dans la scène.

(a) 1ere couche (b) 2eme couche (c) 3eme couche

Figure 4 – *Premières couches d'une I-LDI (Ref. = Vue 4), générée en utilisant les 8 caméras d'acquisition dans un ordre B-hiérarchique : (4 ; 2 ; 6 ; 1 ; 3 ; 5 ; 7 ; 0).*

La figure 5 présente le taux de remplissage des huit premières couches d'une LDI et d'une I-LDI. Dans les deux cas, la première couche est complète. Dans une LDI, le nombre de pixels cumulé de toutes les couches supplémentaires représente plus de 50% du nombre de pixels dans la première couche. Ce pourcentage est réduit à moins de 10% par la construction incrémentale. Dans ce cas, les couches à partir de la 3eme sont pratiquement vides.

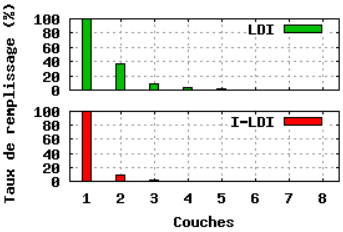


Figure 5 – *Taux de remplissage moyen des couches d'une LDI et d'une I-LDI*

3 Compression des LDI

Les vidéos MVD représentent un important volume de données, mais très fortement corrélées. Les corrélations inter-vues viennent s'ajouter aux corrélations temporelles, déjà présentes dans une vidéo classique. Pour compresser ces vidéos, une des propositions mise en œuvre par MPEG est le codeur MVC, présenté dans la partie 3.1. Ce codeur présente certaines limitations dans le cadre multi-vues, c'est pourquoi la partie 3.2 présente une adaptation de ce codeur aux couches d'une LDI.

3.1 Codeur MVC

Dans un codeur vidéo 2D comme H.264/AVC, quelques images clés servent de référence, et sont encodées indépendamment des autres. Les images restantes sont prédites à partir de ces images clés, par l'utilisation d'un champ de mouvement, codé et transmis avec les images clés. L'erreur de prédiction est également encodée, pour permettre d'être corrigée par le décodeur et d'améliorer la qualité. Ce principe à été étendu aux séquences multi-vues par le codeur MVC (pour *Multi-View Coding*, une extension de H.264/AVC). L'une des vues est choisie comme vue de référence, et est encodée comme une vidéo classique. Chaque image des vues supplémentaires est alors prédite à partir des images au même instant des vues déjà encodées et des images de la même vue mais à des instants différents.

Comparée à une compression Simulcast, où chaque vue est encodée indépendamment par H.264/AVC, la compression MVC présente les résultats suivants [6] :

- À débit équivalent, MVC apporte un gain en qualité entre 1 et 1.5 dB.
- À qualité équivalente, MVC apporte un gain de seulement 15% à 30% du débit.
- Plus de 90% des corrélations dans une vidéo multi-vues sont des corrélations temporelles.
- Le débit du flux MVC est linéairement lié au nombre de vues supplémentaires, rendant son utilisation difficile lorsque ce nombre de vues augmente.

3.2 Codage des LDI par MVC

Les déformations géométriques entre les vues sont prédites par MVC à l'aide d'un champ de mouvement par blocs, ce qui n'est pas adapté. L'avantage de la construction LDI est de corriger ces déformations géométriques, tout en éliminant une grande partie des corrélations inter-vues. Le codeur MVC est utilisé sur la séquence de LDI, en considérant chaque couche comme une vidéo différente. Pour cela, les couches partiellement vides sont complétées par le contenu de la couche précédente, de sorte que le codeur MVC n'ait pas à re-coder ces zones. MVC ne supportant que des vidéos à trois composantes, le codeur MVC est utilisé sur l'ensemble des textures, puis séparément sur l'ensemble des cartes de profondeur.

La figure 6 présente le graphique débit/distorsion obtenu par la compression MVC des LDI, et par la compression MVC des données MVD. En utilisant les quatre vues (4 ;

2 ; 6 ; 0) de la vidéo "Ballet" (MSR), les LDI sont générées d'une part, puis limitées à seulement deux couches. Les textures et les profondeurs sont encodées indépendamment par MVC, en utilisant les mêmes pas de quantification. Après décodage, les couches de la LDI sont utilisées pour générer la vue intermédiaire 5. Ces quatre mêmes vues sont encodées d'autre part directement par MVC (texture et profondeur au même pas de quantification). Les vues 4 et 6 décodées sont utilisées par VSRS, le logiciel de synthèse de vue de MPEG [7], pour synthétiser la vue 5.

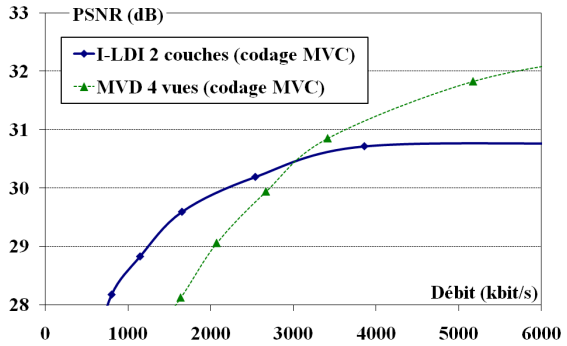


Figure 6 – *Comparaison débit-distorsion sur la vidéo "Ballet" (MSR), entre le format LDI codé par MVC et le format MVD codé par MVC.*

Le format LDI proposé présente un gain en qualité significatif comparé au standard MVC classique, lorsque les débits cibles sont faibles (de l'ordre de 3Mbits/s). Pour des débits plus élevés, le format LDI atteint rapidement une saturation, due aux erreurs de projection et d'alignement des contours.

4 Synthèse de vues

Les LDI contiennent suffisamment d'informations pour permettre la synthèse de vues virtuelles. Cette synthèse de vue se heurte à plusieurs difficultés. La section 4.1 présente les solutions classiques contre le problème de découvrement. La section 4.2 propose une solution plus complète, résolvant également les craquelures de textures. La section 4.3 explique l'apparition de fantômes, et propose une solution pour les éviter.

4.1 Inpainting

La projection de chaque pixel d'une image selon un nouveau point de vue génère des craquelures (erreurs d'échantillonnage) et des découvrements (texture inconnue), visibles sur la figure 7(a). Les craquelures peuvent être comblées par application d'un filtre médian sur la carte de profondeur, puis re-synthèse de la texture par projection inverse [10]. Les découvrements sont plus difficiles à compléter, puisqu'il manque de l'information. Plusieurs solutions d'inpainting ont été proposées, utilisant les textures avoisinantes.

- La bibliothèque OpenCV propose une méthode ne tenant pas compte de la profondeur. La figure 7(b) montre que la couleur d'avant plan semble s'étaler dans l'arrière plan.
- Une méthode dérivée utilise la profondeur pour remplacer temporairement l'avant plan par de l'arrière plan. L'inpainting est ensuite utilisé avec uniquement des textures d'arrière plan. Enfin, le remplacement inverse est effectué [10].

Ces deux méthodes ne permettent ni de détecter, ni de combler, les craquelures sur des textures d'avant plan. La solution proposée dans la section 4.2 s'appuie sur une projection ordonnée des pixels, dont le résultat est visible sur la figure 7(c). Un effet d'étirement est observable dans toute la zone remplie.

(a) Découvrements (b) Inpainting OpenCV (c) Proj. Ordonnée

Figure 7 – *Découvrements 7(a) et comparaison des résultats d'inpainting d'openCV 7(b) et de la projection ordonnée 7(c).*

4.2 Projection Ordonnée

McMillan [11] propose de projeter les pixels dans un ordre déterminé, pour appliquer l'algorithme du peintre et se passer de Z-Buffer. Cet ordre de parcours a pour second avantage de détecter les découvrements pendant la projection. La différence d'abscisse, après projection, entre un pixel et son voisin, permet de détecter les recouvrements et les découvrements. L'ordre est défini en fonction de l'épipôle, de sorte que les derniers pixels posés soient des pixels d'avant plan.

La figure 8(b) présente le principe lorsque les caméras sont rectifiées, c'est-à-dire que les pixels peuvent être traités par ligne. Soit un pixel $P_i = (x_i, y_i)$ de l'image d'origine se projetant en $P'_i = (x'_i, y'_i)$ dans l'image finale. L'égalité $y_i = y'_i$ implique que chaque ligne, que nous supposerons parcourue de gauche à droite, est indépendante des autres. Dans ce cas :

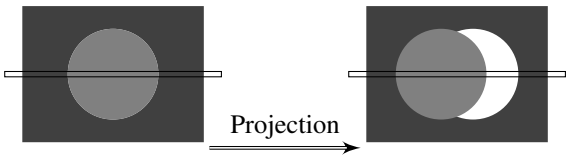
- Si $x'_i < x'_{i-1}$, alors il y a recouvrement, et le pixel courant est un pixel d'avant plan.
- Si $x'_i > x'_{i-1}$, alors il y a découvrement, et le pixel courant est un pixel d'arrière plan, pouvant servir à compléter.

Ce procédé est implémenté dans LDVRS, le logiciel de synthèse de vues rectifiées à partir des LDI de MPEG [7].

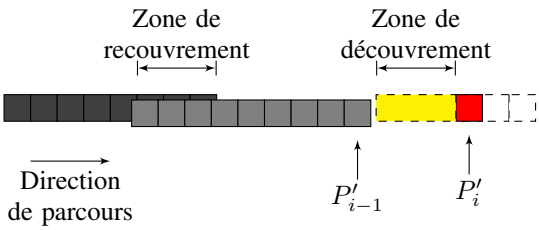
La figure 8(c) présente le principe dans le cadre de projections quelconques. L'égalité de y_i et de y'_i n'est plus assurée, et les lignes doivent être parcourues simultanément (en pratique, l'image est parcourue par colonne). Pour chaque ligne j de l'image finale, on définit la limite X^j_{max} des pixels déjà remplis par l'équation (3).

$$X^j_{max} = \max_{\{P'_i \,:\, y'=j\}} (x') \tag{3}$$

Dans ce cas :
- Si $x'_i < X^j_{max}$, alors il y a recouvrement.
- Si $x'_i > X^j_{max}$, alors il y a découvrement, et le pixel courant est un pixel d'arrière plan, pouvant servir à compléter la zone découverte.



(a)



(b) Détails d'une ligne de pixels pour une projection rectifiée.

(c) Détails de quelques lignes de pixels pour une projection non rectifiée.

Figure 8 – *Principe de la projection ordonnée.*

Cette projection n'est pas plus coûteuse en temps de calcul qu'une projection classique, mais du fait de son ordonnancement, son implémentation sur GPGPU est difficile.

4.3 Fantômes

Dans les images acquises pas les caméras, les contours des objets ne sont jamais nets. La couleur des pixels de contour reçoit la contribution de l'objet et celle de l'arrière plan. Par contre, la profondeur de ces pixels est tantôt celle de l'objet, tantôt celle de l'arrière plan. Après projection, ces pixels se retrouvent éloignés de la nouvelle frontière de l'objet, et génèrent un effet de fantôme.
La solution proposée est de ne pas tenir compte des pixels trop proches des frontières d'objets. Pour cela, un détecteur de Canny est appliqué sur la carte de profondeur, afin de localiser les contours. Une fenêtre de taille W est définie autour des contours, dans laquelle tous les pixels sont classés en fonction de leur profondeur. Les pixels dont la profondeur est supérieure à la profondeur moyenne dans la fenêtre, sont éliminés.

La figure 9 présente les résultats de synthèse de vues à partir d'une LDI à deux couches. Le fantôme 9(a) est détecté et effacé par cette méthode 9(b), puis remplacé par de l'inpainting 9(c).

(a) Fantôme. (b) Suppression. (c) Remplissage.

Figure 9 – *Suppression des effets de fantôme. ($W = 7$)*

5 Résultats

Des mesures de la qualité du rendu ont été effectuées, à partir des vidéos MVD "Ballet" et "Breakdancers" [12]. Les I-LDI à deux couches ont été générées selon le point de vue 4, à partir de la première image de chacune des 8 vues, utilisées dans un ordre B-hiérarchique (4 ; 0 ; 7 ; 2 ; 6 ; 1 ; 5 ; 3).
La figure 10 présente les PSNR, mesurés entre les vues générées et les vues acquises, pour la vidéo "Breakdancers". Plus la caméra virtuelle s'éloigne du point de vue de référence, plus les erreurs de projection sont accentuées, réduisant ainsi la qualité de l'image générée.

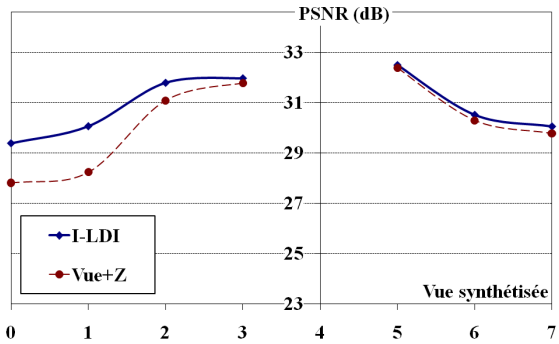


Figure 10 – *Qualité de rendu des vues virtuelles selon le point de vue. (Séquence : "Breakdancers" ; Ref. : Vue 4)*

Pour mesurer l'apport de la seconde couche de la I-LDI, les mêmes mesures ont été faites à partir uniquement de l'image 2D+Z de la vue de référence (c'est-à-dire la première couche de la I-LDI). Les PSNR obtenus sont toujours plus faibles qu'en utilisant les deux couches, mais dans des proportions très différentes selon la position de la caméra virtuelle. La géométrie de la scène fait que, pour une caméra virtuelle numérotée entre 0 et 3, les zones découvertes ont des textures uniformes, facilement estimées par les solutions d'inpainting. A l'inverse, pour une caméra virtuelle numérotée entre 5 et 7, des zones fortement texturées apparaissent, rendant pertinent le contenu de la seconde couche.
Le logiciel LDVRS de MPEG [7] à été utilisé pour comparer les résultats précédents avec ce qui se fait déjà en

synthèse de vue à partir de LDI. Ce logiciel utilise des LDV (semblables aux LDI à deux couches) pour générer des vues virtuelles rectifiées par rapport au point de vue de référence. VSRS [7] a été utilisé sur la vidéo MVD non compressée, pour générer un ensemble de vues rectifiées pouvant servir de références aux calculs de distorsions.
La figure 11 présente les résultats obtenus sur la vidéo "Ballet". Les mêmes I-LDI sont utilisées pour un rendu par projection ordonnée, et pour un rendu par LDVRen, un des algorithmes de LDVRS. Quel que soit le point de vue généré, les améliorations proposées dans cet article apportent un gain en PSNR comparé au rendu de LDVRen. En effet, LDVRen est conçu pour synthétiser des vues très proches de la vue de référence. Lorsque la caméra s'éloigne, il commet des erreurs qui peuvent le rendre moins performant que des algorithmes basés sur une seule vue 2D+Z.

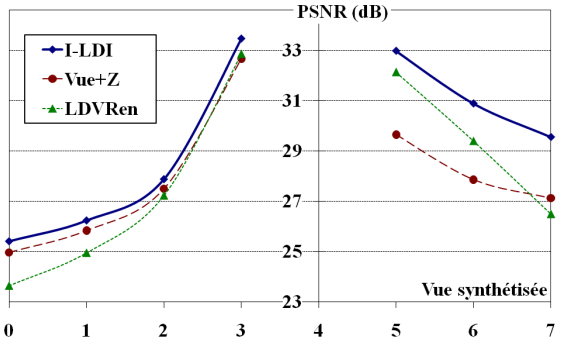


Figure 11 – *Qualité de rendu pour des vues virtuelles rectifiées. (Séquence : "Ballet" ; Ref. : Vue 4)*

6 Conclusions

Cet article propose une approche basée sur les LDI pour la compression et le rendu de vidéos MVD non rectifiées. La construction incrémentale permet d'éliminer les corrélations inter-vues tout en réduisant le nombre de couches produites. Ainsi, la seconde couche ne contient que 10% de pixels différents, comparée à la première couche. La compression MVC appliquée sur ces couches permet, en plus, l'exploitation des corrélations temporelles. Les I-LDI produites sont presque indépendantes du nombre de vues utilisées, ce qui rend le format très compétitif comparé à la compression MVC directement appliquée sur les vues. Quelques méthodes sont proposées pour améliorer la qualité des vues virtuelles générées. La totalité des zones découvertes sont comblées par une solution d'inpainting basée sur un ordonnancement des pixels lors de leur projection. Les effets de fantômes sont éliminés par une détection de contours dans la carte de profondeur. La comparaison avec LDVRen pour des vues rectifiées permet d'observer une amélioration de la qualité du rendu.
Les travaux futurs utiliseront un algorithme d'inpainting directionnel pour éliminer l'effet d'étirement produit par la projection ordonnée. Une étude sera également menée sur l'utilisation conjointe de plusieurs I-LDI.

Références

[1] J. Shade, S. Gortler, L. W. He, et R. Szeliski. Layered depth images. Dans *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, New York, NY, USA, 1998. ACM.

[2] S.-U. Yoon, E.-K. Lee, S.-Y. Kim, et Y.-S. Ho. A framework for representation and processing of multi-view video using the concept of layered depth image. *Journal of VLSI Signal Processing Systems for Signal Image and Video Technology*, 46 :87–102, 2007.

[3] X. Cheng, L. Sun, et S. Yang. Generation of layered depth images from multi-view video. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 5 :V –225–V –228, 16 2007-Oct. 19 2007.

[4] S.-U. Yoon, E.-K. Lee, S.-Y. Kim, Y.-S. Ho, K. Yun, S. Cho, et N. Hur. Coding of layered depth images representing multiple viewpoint video. 2006.

[5] J. Duan et J. Li. Compression of the layered depth image. *Image Processing, IEEE Transactions on*, 12(3) :365–372, Mars 2003.

[6] P. Merkle, A. Smolic, K. Müller, et T. Wiegand. Efficient prediction structures for multiview video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(11) :1461–1473, Novembre 2007.

[7] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, et Y. Mori. *Reference Softwares for Depth Estimation and View Synthesis*. ISO/IEC JTC1/SC29/WG11MPEG2008/M15377, April 2008.

[8] V. Jantet, L. Morin, et C. Guillemot. Incremental-ldi for multi-view coding. Dans *3DTV-Con2009*, Potsdam, Germany, Avril 2009.

[9] K. Müller, A. Smolic, K. Dix, P. Kauff, et T. Wiegand. Reliability-based generation and view synthesis in layered depth video. *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pages 34–39, Oct. 2008.

[10] K. J. Oh, S. Yea, et Y. S. Ho. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video. Dans *PCS'09 : Proceedings of the 27th conference on Picture Coding Symposium*, pages 233–236, Piscataway, NJ, USA, 2009. IEEE Press.

[11] L. McMillan. A list-priority rendering algorithm for redisplaying projected surfaces. Rapport technique, Chapel Hill, NC, USA, 1995.

[12] C.-L. Zitnick, S.-B. Kang, M. Uyttendaele, S. Winder, et R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3) :600–608, 2004.