Modélisation de Scènes Naturelles à Partir de Séquences Vidéos Multi-vue plus Profondeur (MVD)

Youssef Alj[1,2] Guillaume Boisson[1] Philippe Bordes[1] Muriel Pressigout[2] Luce Morin[2]

[1] Technicolor
[2] INSA Rennes

{youssef.alj,guillaume.boisson,philippe.bordes}@technicolor.com
{muriel.pressigout,luce.morin}@insa-rennes.fr

Résumé

Dans cet article, un schéma de modélisation de séquences Multi-vues Vidéo plus profondeur (MVD) est présenté. Le but est de réduire la redondance de profondeur et de texture présentes dans les séquences MVD. Pour ce faire, la fusion de cartes de profondeurs utilisant une représentation volumétrique est proposée. Les voxels sont "carvés" itérativement pour chaque vue en utilisant la technique de traçage de rayons (ray tracing). La surface fusionnée est extraite à partir de cette représentation en utilisant l'algorithme de Marching Cubes. Finalement, le problème de plaquage des textures sur cette surface résultante est abordé. L'algorithme proposé sélectionne parmi toutes les textures le meilleur candidat pour texturer un triangle de la surface résultante. Ce choix est fait en utilisant une métrique dite de photocohérence. Les tests et les résultats sont fournis pour des images fixes en utilisant les séquences MVD usuelles.

Mots clefs

Multi-view Video plus Depth (MVD), 3DTV, FTV, depth map fusion, Space Carving, multi-view texture mapping.

1 Introduction

L'Imagerie Multi-Vue (IMV) a suscité un vif intérêt durant les dernières décennies. Grâce au développement des écrans stéréoscopiques, l'IMV fournit une sensation réaliste de profondeur à l'utilisateur et une navigation virtuelle autour de la scène observée, ouvrant par conséquent un large éventail de sujets de recherche et d'applications telles que la télévision 3D (TV3D) ou la FTV (*Free-viewpoint TV*). Cependant, de nombreux défis techniques ont entravé la rapide apparition de ces applications dans les marchés de masse. Ces défis peuvent être liés à l'acquisition de la scène et à sa représentation d'une part ou à la transmission des données représentées d'autre part. Dans le contexte de la représentation de scènes naturelles, de nombreux efforts ont été fournis afin de surmonter ces difficultés. Les méthodes proposées dans la littérature peuvent être classées en trois catégories : les représentations basées image, les représentations basées géométrie ou les représentations intermédiaires.

Les représentations basées image regroupent l'utilisation de MVV (Multi-view Video), 2D+Z [1] ou MVD (*Multi-view Video plus Depth*) [2], où chaque vue est constituée de l'image acquise et d'une carte de profondeur estimée. Les représentations basées géométrie s'appuient sur l'utilisation d'un modèle géométrique à base de surface maillée [3], de modèle volumétrique [4] ou de nuage de points [5]. Les représentations intermédiaires incluent l'utilisation des LDI (*Layered Depth Images*) [6], des représentations dites *billboards* [7] ou de soupe de polygones [8]. L'approche adoptée dans cet article consiste en une méthode hybride s'appuyant sur l'utilisation des séquences MVD, afin de conserver le photo-réalisme de la scène observée, combinée avec un modèle géométrique, à base de maillage triangulaire, renforçant ainsi la compacité de la représentation.

L'utilisation des représentations volumétriques pour la modélisation de scènes a connu un grand succès ces dernières années grâce à la robustesse de ces méthodes aux différents bruits. Typiquement, ces méthodes supposent l'existence d'un volume englobant dans lequel la scène d'intérêt se situe. Ce volume est ensuite subdivisé en éléments cubiques appelés voxels. Durant l'étape de sculpture de l'espace (*Space Carving*), chaque voxel est étiqueté comme étant opaque ou transparent selon sa cohérence avec la scène. Ceux qui sont cohérents sont déclarés comme opaques et les autres voxels sont étiquetés comme transparents. La surface finale est obtenue à partir de cette classification binaire en utilisant l'algorithme de *Marching Cubes*. La mesure de la cohérence d'un voxel avec la scène peut être basée sur l'information de silhouette extraite des images d'entrées, cette méthode est connue sous le nom de *Shape from silhouette* (modélisation à partir des silhouettes) [9] ou directement à partir de l'information contenue dans les images d'entrées ce qui est connue sous le nom de *Shape from photoconsistency* (modélisation à partir de la photocohérence) [10]. Les approches alternatives comme VRIP [11] ou KinectFusion [12] utilisent les cartes de profondeur comme données d'entrée et les fusionnent en discrétisant une fonction de distance signée sur le volume englobant,

la valeur de cette distance en chaque voxel correspond à la distance signée du voxel à la plus proche surface vis à-vis du point de vue. Dans VRIP, une surface explicite est extraite à partir de cette représentation volumétrique grâce l'algorithme de Marching Cubes, tandis que, contraints par le rendu en temps réel, cette surface est simplement prédite par la technique de traçage de rayons dans le cas de KinectFusion. La méthode proposée dans cet article est similaire à ces dernières approches utilisant les cartes de profondeur. Néanmoins, l'objectif recherché est de construire un modèle 3D explicite destiné à transmettre l'information contenue dans les cartes de profondeur originales en réduisant les redondances. La cohérence du modèle 3D avec les données d'entrée est donc le critère de qualité qui nous intéresse dans cette étude. En particulier, ce modèle doit être photocohérent avec les images d'entrée, la photocohérence est donc renforcée grâce à l'algorithme de plaquage de texture proposé. Les contributions de cet article sont doubles : d'abord, un schéma volumétrique dédié à la fusion des cartes de profondeur en une surface maillée est proposé, c'est l'objet de la section 2. Ensuite, un nouvel algorithme de plaquage texture multi-vues est présenté dans la section 3. Enfin, les résultats sont présentés dans la section 4.

2 Modélisation géométrique

Dans cette section, notre schéma volumétrique dédié à la fusion des cartes de profondeur est présenté. Considérant le cas général, les séquences d'entrées sont calibrées mais ne sont pas nécessairement alignées ni rectifiées. Le problème d'estimation de la disparité inter-vue n'est pas abordé dans cet article et les cartes de profondeurs fournies sont supposées être fiables. Le but est donc de fusionner ces cartes de profondeur en une seule surface cohérente avec les vues originales. Une vue d'ensemble de notre schéma de fusion est présentée dans la figure 1. Premièrement, un maillage triangulaire en haute résolution est extrait à partir de chaque carte de profondeur. Ensuite, le volume englobant tous ces maillages est défini puis discrétisé en voxels. Un voxel peut avoir deux statuts : opaque ou transparent. Pour chaque caméra, l'ensemble des voxels se situant dans son cône de vue sont déterminés et étiquetés comme opaque. A l'étape suivante, l'Enveloppe Volumétrique du Maillage (EVM) est déterminée pour chacune des vues. Étant donné un maillage correspondant à un point de vue, l'EVM définit l'ensemble des voxels situés sur la surface maillée. Notre formulation de l'algorithme de *Space Carving* repose sur l'EVM calculée. En effet, pour chaque vue, des rayons sont tracés à partir du centre de la caméra à chaque voxel constituant l'EVM correspondant à cette vue. Les voxels se situant sur chaque rayon sont déclarés transparents.

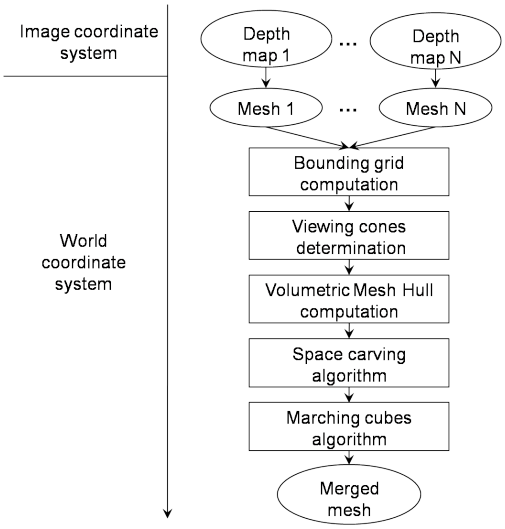

Figure 1 – *Structure du schéma de fusion.*

2.1 Construction des maillages à partir des cartes de profondeur

La première étape de l'algorithme est la génération d'un maillage pour chaque carte de profondeur donnée. Pour chaque vue, un maillage haute résolution est construit dans le repère image. Chaque sommet du maillage représente un pixel dans la carte de profondeur. Les maillages ainsi construits sont alignés en exprimant chaque sommet du maillage dans le repère monde en utilisant l'équation suivante :

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} & R & & T \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} f_x & S_{uv} & c_u & 0 \\ 0 & f_y & c_v & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} su \\ sv \\ s \\ 1 \end{pmatrix}$$

Où X, Y, et Z représentent les coordonnées d'un point 3D dans le repère monde. u et v sont les coordonnées de la projection du point 3D dans le repère image, et $s = z_{cam}$ est la profondeur du pixel (u, v) dans le repère caméra. R et T définissent respectivement la matrice de rotation et de translation du repère caméra par rapport au repère monde. f_x, f_y, S_{uv}, c_u et c_v sont les paramètres intrinsèques de la caméra.

2.2 Calcul de la grille englobante

Les maillages ainsi construits sont les entrées de notre représentation volumétrique. La structure de données utilisée dans cette représentation est une grille volumétrique. Cette grille est construite par subdivision régulière de la boîte englobante de l'ensemble des maillages en éléments parallélépipédiques appelés voxels. Un étiquetage binaire est utilisé pour attribuer à chaque voxel l'étiquette opaque ou transparent. L'étiquette de chaque voxel est initialisée à transparent. Les étiquettes des voxels sont modifiées tout

au long de l'algorithme selon leur pertinence pour représenter la scène.

2.3 Détermination des cônes de vue

Dans cette étape le but est de trouver l'ensemble des voxels se situant dans le cône de vue de chaque caméra. Les voxels en dehors de l'union des cônes de vue sont étiquetés comme transparent. Le cône de vue de chaque caméra est modélisé par un frustum défini par quatre plans délimitant l'ensemble des sommets visibles selon chaque axe (voir Figure 2). Le but est donc de déterminer pour chaque caméra l'ensemble des voxels se situant dans le frustum correspondant. Les équations des plans du frustum sont donc calculés pour chaque caméra. Ces équations nous permettent de situer chaque voxel par rapport aux plans calculés et donc de définir les voxels dans chaque frustum.

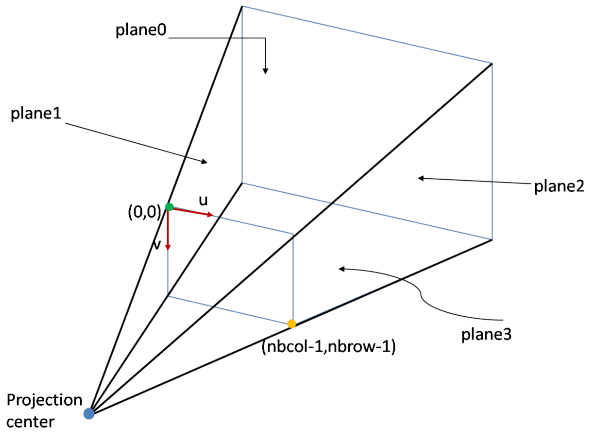

Figure 2 – *Calcul des plans du Frustum de projection.*

2.4 Construction de l'Enveloppe Volumétrique du Maillage EVM

Dans cette section, le calcul de l'Enveloppe Volumétrique du Maillage (EVM) est présenté. Cette EVM sera utilisée durant le processus de *Space Carving*. Pour chaque maillage construit, l'ensemble des voxels intersectant la surface du maillage définit l'EVM associée à ce maillage. A la fin de la construction de chaque EVM, chaque voxel aura une information additionnelle définissant à quelle EVM ce voxel appartient. Pour chaque maillage, les triangles sont parcourus et analysés selon la grille volumétrique en utilisant un algorithme de *Scan Line 3D*, cette analyse permet de définir les voxels se situant sur chaque triangle. L'ensemble des voxels analysés seront marqués comme appartenant à l'EVM courante en mettant à jour l'information additionnelle associée au voxel. L'algorithme d'analyse des triangles d'un maillage est présenté dans l'algorithme 1.

```
// Parcours des triangles
pour chaque triangle T faire
    pour chaque axe X, Y et Z faire
        Calculer les limites entières b_min et b_max de T
        selon chaque axe.
        pour m ← b_min to b_max faire
            Calculer l'intersection du triangle T avec le
            plan à la coordonnée m.
            Trouver les voxels au long de cette
            intersection en utilisant la ligne de Bresenham.
            Ajouter ces voxels à l'EVM.
        fin
    fin
fin
```

Algorithm 1: Détermination de l'Enveloppe Volumétrique du Maillage (EVM).

2.5 Algorithme de *Space Carving*

Jusqu'à cette étape de l'algorithme, un voxel opaque est un voxel qui se situe dans au moins un cône de vue et un voxel de l'EVM est un voxel qui se situe sur un des maillages. Dans cette étape, le *Space Carving* est effectué en mettant à jour les étiquettes des voxels d'opaque à transparent pour tous les voxels situés devant l'EVM. Plus précisément, les différents EVMs sont considérés successivement et un critère de cohérence géométrique est utilisé afin de mettre à jour les étiquettes des voxels. Ce critère est basé sur la position relative du voxel par rapport au maillage considéré et le point de vue correspondant à ce maillage. Un voxel est dit géométriquement cohérent avec une surface du maillage s'il se situe derrière cette surface par rapport au point de vue correspondant. Les voxels détectés comme géométriquement incohérents sont étiquetés comme transparents. Le processus de *Space Carving* met à jour le volume sculpté pour chaque caméra. Ce volume est initialisé à l'union des voxels situés dans les cônes de vues calculés dans la section 2.3 (voir Figure 3). Le même volume est ensuite raffiné itérativement avec l'information géométrique provenant de chacune des vues, voir Figures 3 et 4), en sculptant les zones occultantes relatives à chaque vue. Pour chaque vue, les rayons sont tracés du centre de la caméra aux voxels de l'EVM correspondant à cette caméra. Pour chaque rayon, les voxels se situant sur le rayon courant sont scannés en utilisant l'algorithme de Bresenham 3D de traçage de lignes. Comme ils sont géométriquement incohérents, ces voxels seront creusés (i.e. étiquetés comme transparents). A la fin de cette étape de *Space Carving*, l'ensemble des voxels opaques définit le modèle volumétrique de nos données MVD.

2.6 Marching Cubes

Finalement, la surface fusionnée est extraite à partir du volume binairement étiqueté en utilisant l'algorithme des

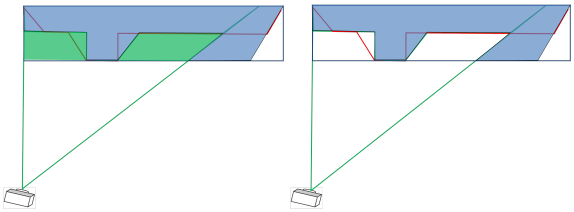

Figure 3 – *Space carving vis-à-vis de la caméra gauche. Les rayons sont tracés du centre de la caméra à l'EVM correspondante. Les zones vertes sont carvées (schéma à gauche). Le résultat du Space Carving vis-à-vis de la caméra gauche est montré dans le schéma à droite.*

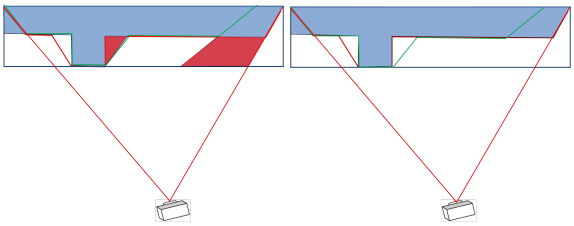

Figure 4 – *Space carving pour la caméra droite. Les zones rouges sont carvées. Le résultat est montré dans le schéma de droite.*

Marching Cubes. L'algorithme de *Marching Cubes* a été introduit par Lorenson et Cline [13]. Cet algorithme prend comme entrée une structure de données de type volume régulièrement subdivisé en voxels. A chaque voxel de ce volume est attribué une valeur. Durant le traitement, chaque sommet de voxel ayant une valeur supérieure ou inférieure à une valeur prédéfinie est marqué. Tous les autres sommets restent non-marqués. Par conséquent, le résultat de l'algorithme de *Marching Cubes* est le maillage triangulaire connectant tous les sommets qui ont été marqués.

3 Plaquage des textures

La contrainte de photocohérence est renforcée en texturant la surface fusionnée extraite de l'algorithme de *Marching Cubes*. Les images de textures des séquences MVD sont

```
// Parcours des vues
pour chaque caméra faire
    Extraire la position de la caméra.
    Extraire l'EVM correspondante.
    pour chaque voxel dans l'EVM faire
        Trouver la ligne de Bresenham 3D entre la
        position de la caméra et le centre du voxel.
        Mettre à jour les labels des voxels se situant sur
        cette ligne à transparent.
    fin
fin
```

Algorithm 2: L'algorithme de *Space Carving*.

utilisées afin de texturer cette surface et un nouvel algorithme de plaquage de textures multi-vues basé sur une métrique de photocohérence est proposé. Les principales étapes de cet algorithme sont présentées dans la Figure 5. D'abord, la visibilité est déterminée pour chaque triangle. Ensuite, chaque texture est projetée sur les triangles visibles de cette surface. Les erreurs de projections photométriques sont calculées pour chaque triangle et pour chaque vue et sont sauvegardées dans des images d'erreurs. La meilleure texture pour chaque triangle est celle qui minimise une métrique de photocohérence. Les triangles qui ne sont visibles par aucune caméra sont texturés avec la texture de la caméra intermédiaire. Les vues synthétisées sont extraites en faisant le rendu du maillage texturé, chaque triangle étant texturé avec la meilleure texture selon la métrique de photocohérence.

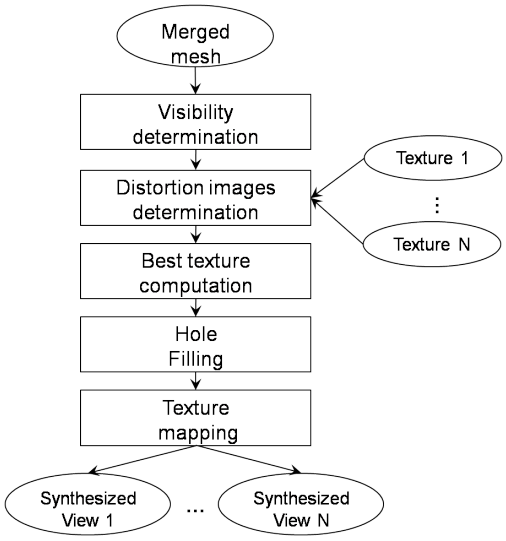

Figure 5 – *Le schéma global de notre algorithme de plaquage de texture basé sur la métrique de photocohérence.*

A partir de l'ensemble des images de texture $\mathcal{I} = \{I_1, \ldots, I_n\}$ et le maillage $\mathcal{M}$, le but est de trouver pour chaque triangle $T \in \mathcal{M}$ la meilleure image de texure $\hat{I}_T \in \mathcal{I}$. Texturer un triangle T est formulé comme un problème de minimisation d'énergie. La meilleure texture pour T est calculée en minimisant une fonction de coût décrivant la photocohérence du triangle T avec l'ensemble des images de texture. La photocohérence est estimée en projetant le maillage texturé sur les vues $\mathcal{V} = \{V_1, \ldots, V_n\}$ correspondant au images d'entrées $\mathcal{I}$. Le triangle t (Voir Figure 6) est d'abord texturé avec la texture I_1 de la caméra 1 (flèche rouge pour l'opération de plaquage de texture) puis projeté sur les caméras 3, 4 et 5, i.e. les caméras dans lesquelles le triangle t est visible (flèche verte pour l'opération de projection). Les erreurs quadratiques entre le triangle texturé et chacune de ses projections sur les images I_3, I_4 et I_5 sont calculées. L'erreur de texturer le triangle t avec la texture I_1 i.e. la métrique de photocohérence relative à la

texture I_1 est la somme de toutes ces erreurs quadratiques. Ce procédé est répété pour les textures I_2, I_3, I_4 et I_5 et la métrique de photocohérence est donc calculée pour chacune de ces textures. La texture minimisant cette métrique est considérée comme meilleure texture pour le triangle t. Voir [14] pour plus de détails sur les algorithmes utilisés.

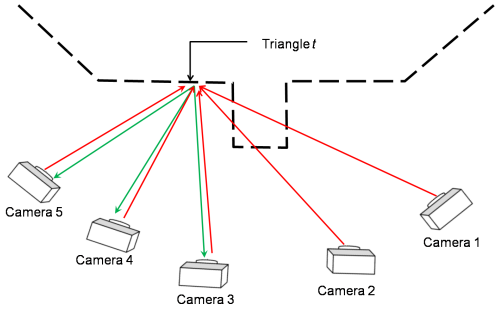

Figure 6 – *Plaquage de texture en utilisant la métrique de photocohérence. Par simplicité, le maillage à texturer est dessiné en pointillés représentant les triangles du maillage.*

4 Résultats

Les résultats sont présentés pour deux séquences. La séquence *Breakdancers*, fournie par Microsoft, a été utilisée par l'ancien groupe FTV de MPEG. Cette séquence est capturée par huit caméras disposées en arc. Les vues ne sont pas rectifiées. Les cartes de profondeur sont de bonne qualité. La deuxième séquence *Balloons* est fournie par l'Université de Nagoya et partagée dans le groupe MPEG 3DV. Les vidéos ont été capturées avec des caméras parallèles, et les vues sont rectifiées. Les cartes de profondeur sont de moins bonne qualité que celle de *Breakdancers*, néanmoins elles représentent un matériel plus réaliste pour une application pratique. Les deux séquences sont à la résolution XGA (1024x768).

4.1 Modélisation géométrique

La figure suivante illustre la surface obtenue après notre modélisation géométrique avec une résolution volumétrique moyenne.
On distingue des artefacts dans les zones de discontinuités de profondeur. A de telles résolutions, la quantification de profondeur présente aussi des artefacts notamment sur les surfaces planes. Ces deux types d'artefacts sont atténués grâce à notre algorithme de plaquage de texture.

4.2 Plaquage des textures

Calcul de distorsion : La distorsion a été calculée entre les images rendues et les images originales en utilisant le PSNR. Il est important de souligner que la distorsion est calculée uniquement sur les triangles visibles par au moins une caméra.
Même si la distorsion en termes d'erreur quadratique est grande, les résultats en termes de qualité visuelle sont plutôt satisfaisants. Grâce à notre algorithme de plaquage de texture, notre schéma de modélisation est robuste à la qualité des cartes de profondeur. Néanmoins, il est évident qu'une estimation plus exacte des cartes de profondeur donnerait de meilleurs résultats notamment dans la phase de modélisation géométrique.

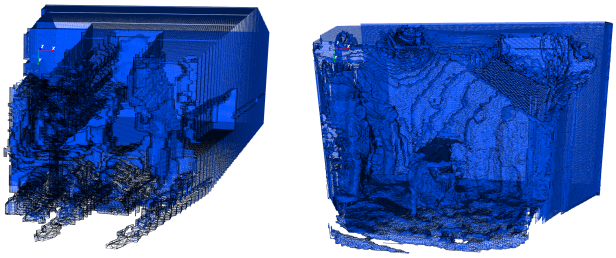

Figure 7 – *Résultats de notre schéma de fusion avec une résolution volumétrique de 250x250x250. A gauche : le modèle issu de Balloons. A droite : le modèle issu de Breakdancers.*

Figure 8 – *Résultats du plaquage de texture basé photocohérence pour Balloons. De haut en bas et de gauche à droite l'image de référence correspondant à la caméra 1, l'image de référence correspondant à la caméra 5, l'image synthétisée correspondant à la caméra 1 et l'image synthétisée correspondant à la caméra 5.*

5 Conclusion

Dans cet article, le problème de modélisation 3D de séquences vidéos à partir de données Mulit-vue plus Profondeur (MVD) a été abordé, sans aucune hypothèse sur la configuration des caméras. Un nouveau schéma de modélisation géométrique permettant la fusion des cartes de profondeur en un seul maillage triangulaire a été présenté. A cet égard, une représentation volumétrique reposant sur un algorithme de *Space Carving* itératif a été utilisé. Les cartes de profondeur ont été représentées par des voxels

Figure 9 – *Résultats du plaquage de texture basé photo-cohérence pour Breakdancers. De haut en bas et de gauche à droite : l'image de référence correspondant à la caméra 0, l'image de référence correspondant à la caméra 7, l'image synthétisée correspondant à la caméra 0, l'image synthétisée correspondant à la caméra 7.*

Sequence	PSNR (dB)		
Breakdancers	Camera 0	Camera 4	Camera 7
	29.52	33.51	30.27
Balloons	Camera 1	Camera 3	Camera 5
	28.94	31.83	30.92

Tableau 1 – *Distorsion des images synthétisées pour les séquence breakdancers et balloons.*

dont le statut (opaque ou transparent) est mis à jour selon leur cohérence géométrique avec ces cartes de profondeur. La frontière du modèle volumétrique est finalement convertie en maillage triangulaire grâce à l'algorithme de *Marching Cubes*. Un nouvel algorithme de plaquage de textures multi-vues a été ensuite proposé. Cet algorithme attribue à chaque triangle de la surface fusionnée la texture optimale en minimisant les distorsions entre les vues synthétisées et les vues originales. Le modèle 3D proposé et composé de texture et de géométrie nous permet de synthétiser toute vue se situant entre les caméras initiales avec peu d'artéfacts visibles. Comme perspective, le problème de la transmission de notre modèle 3D sera abordé en réduisant le coût de codage du modèle géométrique et en formalisant le signal de texture à transmettre.

Références

[1] C. Fehn, P. Kauff, M.O. De Beeck, F. Ernst, W. Ijsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, et I. Sexton. An evolutionary and optimised approach on 3d-tv. Dans *Proc. of IBC*, volume 2, pages 357–365, 2002.

[2] P. Merkle, A. Smolic, K. Muller, et T. Wiegand. Multi-view video plus depth representation and coding. Dans *ICIP 2007*, volume 1, pages I–201. IEEE, 2007.

[3] R. Balter, P. Gioia, et L. Morin. Scalable and efficient video coding using 3-d modeling. *Multimedia, IEEE Transactions on*, 8(6) :1147–1155, 2006.

[4] K. Mueller, A. Smolic, P. Merkle, B. Kaspar, P. Eisert, et T. Wiegand. 3d reconstruction of natural scenes with view-adaptive multi-texturing. Dans *3DPVT 2004*, pages 116–123. IEEE, 2004.

[5] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, et M. Gross. Scalable 3d video of dynamic scenes. *The Visual Computer*, 21(8) :629–638, 2005.

[6] L. He, J. Shade, S. Gortler, et R. Szeliski. Layered depth images. Dans *Proceedings of the 25th annual conference on computer graphics and interactive techniques (SIGGRAPH 1998), July*, pages 19–24, 1998.

[7] M. Waschbüsch, S. Würmlin, et M. Gross. 3d video billboard clouds. Dans *Computer Graphics Forum*, volume 26, pages 561–569. Wiley Online Library, 2007.

[8] T. Colleu, S. Pateux, L. Morin, et C. Labit. A polygon soup representation for multiview coding. *Journal of Visual Communication and Image Representation*, 21(5-6) :561–576, 2010.

[9] W.N. Martin et JK Aggarwal. Volumetric descriptions of objects from multiple views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2) :150–158, 1983.

[10] K.N. Kutulakos et S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3) :199–218, 2000.

[11] B. Curless et M. Levoy. A volumetric method for building complex models from range images. Dans *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[12] S. Izadi et al. Kinectfusion : real-time 3d reconstruction and interaction using a moving depth camera. Dans *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[13] W.E. Lorensen et H.E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4) :163–169, 1987.

[14] Y. Alj, G. Boisson, P. Bordes, M. Pressigout, et L. Morin. Space carving mvd sequences for modeling natural 3d scenes. volume 8290, page 829005. SPIE, January 2012.