



HAL
open science

Z-QuadTree: une nouvelle structure de quadtree pour cartes de profondeur

Thomas Colleu, Dorra Riahi, Luce Morin

► **To cite this version:**

Thomas Colleu, Dorra Riahi, Luce Morin. Z-QuadTree: une nouvelle structure de quadtree pour cartes de profondeur. ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, INRIA Grenoble Rhône-Alpes, Jun 2011, Praz-sur-Arly, France. inria-00596246

HAL Id: inria-00596246

<https://inria.hal.science/inria-00596246>

Submitted on 26 May 2011

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.

Z-QuadTree : une nouvelle structure de quadtree pour cartes de profondeur

T. Colleu¹

D.Riahi²

L.Morin²

¹ Orange Labs Rennes

² IETR/INSA Rennes

IETR/INSA Rennes, 20 avenue des buttes de Coësmes, 35043 Rennes
thomas.colleu@insa-rennes.fr

Résumé

Cet article présente une structure de données "Z-quadtree" permettant de réduire le nombre de quads dans un quadtree représentant une carte de profondeur. La méthode se focalise en particulier sur les zones de discontinuités de profondeur. Dans un quadtree classique, la décomposition est très fine autour des discontinuités, même si les surfaces en avant plan et arrière plan sont planes. La méthode proposée consiste à fusionner les quads d'arrière plan ensemble, tout en conservant les quads d'avant plan formant le contour d'un objet. Les premiers résultats ont montré que le nombre de quads a pu être réduit de 20 à 25% sans dégradation de la qualité visuelle.

Mots Clef

Quadtree, cartes de profondeur, modelisation 3D.

Abstract

This paper presents a data structure "Z-quadtree" that helps reducing the number of quads in a quadtree extracted from a depth map. The method focuses on areas around depth discontinuities. In a classical quadtree, the decomposition is very fine around discontinuities, even if background and foreground surfaces are planar. The proposed method consists in fusing background quads together, while keeping foreground quads that describe the outline of an object. First results have shown that the number of quads could be reduced by 20 to 25%, without decrease of the modeling quality.

Keywords

Quadtree, depth maps, 3D modeling.

1 Introduction

De nombreuses applications de vidéo 3D utilisent en entrée des données de type "vidéo + profondeur", c'est à dire des vidéos simples ou multivues pour lesquelles une carte de profondeur est associée à chaque image [1]. L'information de profondeur permet de réaliser des synthèses de vues virtuelles et sa qualité est alors cruciale, en particulier la précision des discontinuités de profondeur. Les

méthodes standards de compression basées blocs ou ondelettes sont peu efficaces pour coder les cartes de profondeur et induisent des artefacts visuels sur les contours. Plusieurs méthodes alternatives de représentations et codage adaptées aux cartes de profondeur sont basées sur une décomposition en quadtree, qui permet de modéliser finement les contours de discontinuité [2]. Par contre elles génèrent un grand nombre de petits blocs ou *quads* (figure 1(a)), dont le coût de codage est élevé.

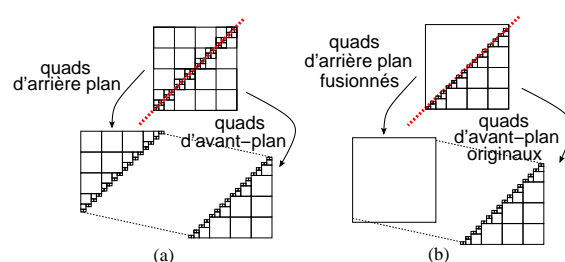


FIGURE 1 – Divisions autour d'une discontinuité. (a) Grand nombre de petits quads. (b) Le quad fusionné en arrière plan remplace les petits quads.

Dans cet article, nous proposons une structure appelée "Z-quadtree" adaptée à la représentation compacte de cartes de profondeur et dans laquelle un grand nombre de petits quads sont remplacés par des quads plus grands et donc moins coûteux (figure 1(b)). Cette structure a été intégrée dans une méthode de modélisation de séquences multivues par soupe de polygones [3]. Son principe peut être appliqué à d'autres types de données.

2 Fusion des quads d'arrière plan

Dans un quadtree de carte de profondeur, la décomposition est très fine autour des discontinuités. On obtient ainsi des petits quads en avant plan (Fg) et en arrière plan (Bg) de la discontinuité. Pour réduire le nombre de quads, nous proposons de fusionner les quads Bg qui correspondent souvent à des surfaces planes, tout en conservant les quads Fg.

Structure du quadtree. En général, la fusion consiste à regrouper 4 quads fils dans leur quad père, c'est l'in-

verse de la division. On remonte ainsi dans le quadtree, des feuilles vers le noeud racine. Dans notre méthode la fusion ne concerne que les quads d'arrière plan (Bg) par rapport à une discontinuité. Ainsi, 1, 2 ou 3 quads Bg sont fusionnés dans leur quad père, tandis que les quads Fg sont conservés en tant que quad fils. Pour signaler dans la structure du quadtree qu'un quad a été fusionné, un flag supplémentaire de 1 bit est utilisé. S'il vaut '1', alors le quad contient un quad et aussi un ou des quads fils Fg. La figure 2 illustre cette fusion et la modification de la structure du quadtree.

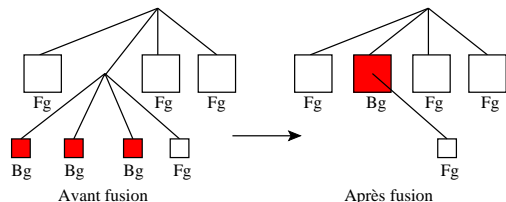


FIGURE 2 – Avant la fusion, de nombreux quads sont Bg mais il y a aussi des quads Fg qui ne doivent pas être fusionnés. Après la fusion, le noeud père contient le quad Bg fusionné et aussi le quad fils Fg non fusionné.

Création d'un quad fusionné. Pour fusionner les quads Bg, le principal problème est de définir la position des coins et la texture dans la zone occultée par les quads fils Fg. La figure 3 illustre ce problème. Les coins occultés sont calculés par extrapolation des autres coins non-occultés. Pour valider ce coin, un critère de préservation de la géométrie est testé entre la carte de profondeur et le quad. Ce critère permet de ne pas fusionner les quads si la profondeur dans cette région n'est pas plane. Ensuite, une texture transparente est attribuée à la zone du quad qui est occultée. Cette zone de transparence est délimitée par les quads Fg, elle peut donc être retrouvée à partir de la structure du quadtree ce qui évite de devoir transmettre cette information. Cette méthode de fusion est résumée dans l'algorithme 1 ci-dessous.

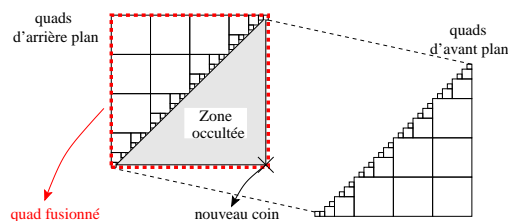


FIGURE 3 – Création d'un quad fusionné (en pointillés rouges). Les zones occultées doivent avoir une texture transparente. La profondeur du nouveau coin doit être calculée.

3 Résultats

La méthode a été testée sur deux séquences appelées *Breakdancers* et *Ballet*. Une réduction de 26% de quads est obtenue pour la première (de 23558 quads à 17334) et

Objectif : Parcourir un quadtree (bottom-up) et fusionner les quads d'arrière plan quand c'est possible.

```

foreach quad do
  if DiscontinuityInside (quad) then
    identify Bg and Fg children quads;
    Compute new corner(s) and fuse Bg children quads ;
    if PreservesGeometry (fusedQuad) then
      Insert fused quad in quadtree;
      Add transparency to occluded area of fused quad;
      Remove Bg children quads from quadtree;
    end
  end
end

```

Algorithme 1: Fusion des quads d'arrière plan

20% pour la seconde (de 27615 quads à 22186). De plus, la qualité visuelle des images synthétisées est conservée. On peut donc espérer une baisse significative du débit une fois que l'étape de compression sera implémentée.

4 Conclusion

Cet article présente une structure de données "Z-quadtree" permettant de réduire le nombre de quads dans un quadtree représentant une carte de profondeur.

Les premiers résultats ont montré que le nombre de quads a pu être réduit de 20 à 25% sans dégradation de la qualité visuelle. Une évaluation du coût de codage de cette structure est à présent nécessaire pour la valider complètement. Outre le gain en codage, cette structure peut être utilisée pour extrapoler judicieusement l'information d'arrière-plan au niveau des discontinuités de profondeur, à la fois en termes de géométrie et de texture, les quads fusionnés peuvent alors être vus comme des petits imposteurs.

Références

- [1] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3) :600–608, 2004.
- [2] J. Evers-Senne, J. Woetzel, and R. Koch. Modeling and rendering of complex scenes with a multi-camera rig. In *Conference on Visual Media Production (CVMP)*, 2004.
- [3] Thomas Collet, Stéphane Pateux, Luce Morin, and Claude Labit. A polygon soup representation for multiview coding. *Journal of Visual Communication and Image Representation*, In Press, Accepted Manuscript, jan 2010.

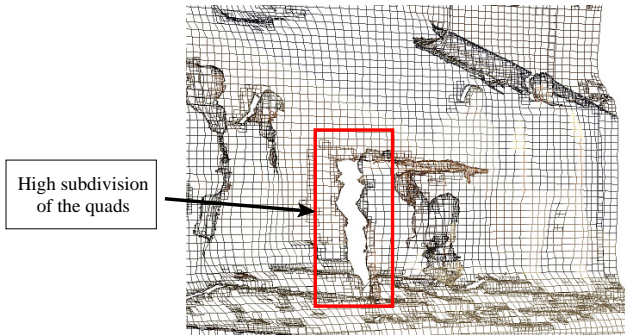


FIGURE 4 – Soupe de polygones obtenue par decomposition en quadtree des cartes de profondeur [3]. La division des quads est très fine autour des discontinuités de profondeur.

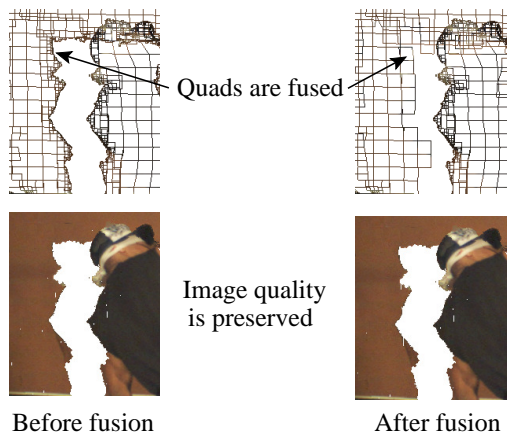


FIGURE 5 – Comparaison de la représentation avant et après la fusion des quads en arrière plan.