

Extraction Temps Réel d'Isosurfaces Adaptatives sur GPU

Hélène Perrier¹

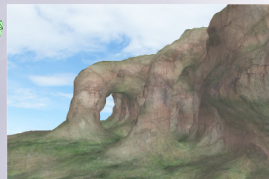
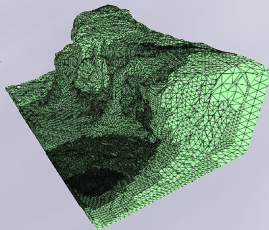
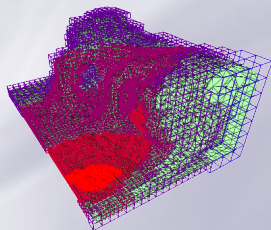
Jonathan Dupuy^{1,2}

Jean-Claude Iehl¹

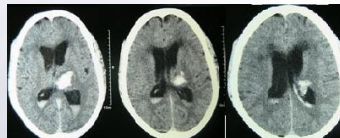
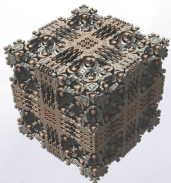
Jean-Philippe Farrugia¹

¹ Université Lyon 1, LIRIS, UMR5205

² Université de Montréal, LIGUM, Dept. I.R.O.,



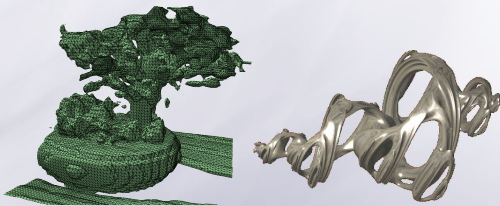
Rendu temps réel d'objets volumiques



- Très utilisés dans la modélisation, le médical ...
- Très peu de solution pour les afficher en temps réel

Rendu temps réel d'objets volumiques

Deux méthodes principales : Rasterization et Ray Tracing

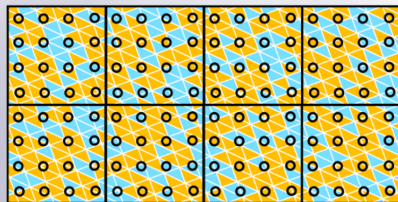
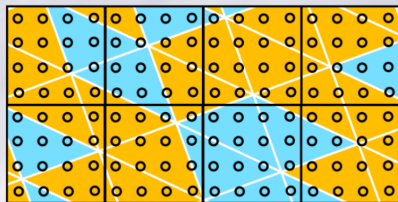


- Ray Tracing implique de trouver une intersection
- La Rasterization implique d'extraire une surface triangulée

Les GPU sont câblés pour faire de Rasterization ; elle se prête ainsi plus facilement au temps réel.

Rasterization

- Fonctionne par échantillonnage
- Loi de Shannon
- Contrainte sur la taille des triangles



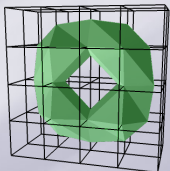
Méthode

- Extraction de surface
- En temps réel
- Adaptative
- Permettant de contrôler la taille des polygones
- Parallèle

Extraction de la surface

- **Marching Cubes [LC87]**

- Discrétise l'objet via une grille régulière
- Calcule la fonction volumiques aux sommets de cette grille
- Extrait la surface traversant chaque cellule
- Tâches isolées



- **Hautement parallèle**

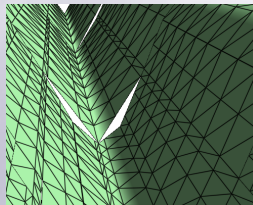
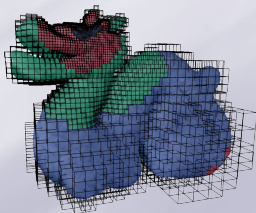
- Implantation efficace sur GPU

Méthode

- Extraction de surface
- Parallèle
- En temps réel
- Adaptative
- Permettant de contrôler la taille des polygones

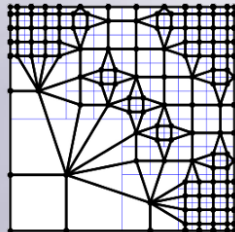
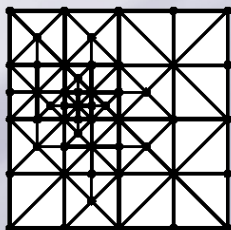
Marching Cubes adaptatif

- Appliquer Marching Cube sur une structure multi-résolution
- Apparition de trous



Éviter ces trous

- Certaines structures multi-résolution ne génèrent pas de trous [SBD13][SW04]
- Structures difficilement parallélisable
- La surface générée a une topologie propre
- Objectif visualisation



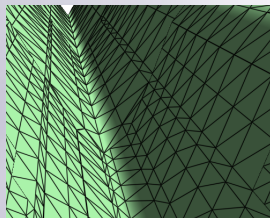
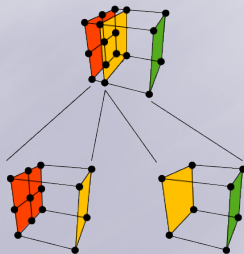
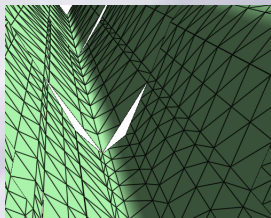
Boucher les trous

- Structures simples
 - Appliquer Marching Cubes sur un octree
- Boucher les trous
- Le maillage est potentiellement incorrect topologiquement
- La surface générée est visuellement sans trous

Lengyel [LAO10]

Bouche les trous avec des cellules de Transition

- Cellules ayant une face deux fois plus subdivisée que le reste
- Insérées au changements de résolution
- Triangulent les trous via un Marching Cube
- Travaille sur un octree restreint
- implantation CPU dans le C4Engine sous le nom "TransvoxelsTM"



Transvoxels [Len10]

- Deux passes de Marching Cubes
 - Une sur l'octree
 - Une pour boucher les trous
- Méthode parallèle ?
 - L'octree est une structure intrinsèquement récursive
 - L'identification des zones de changement de résolution implique de regarder les voisins

Méthode

- Extraction de surface
- Parallèle
- En temps réel
- Adaptative
- Permettant de contrôler la taille des polygones

Paralléliser [LAO10]

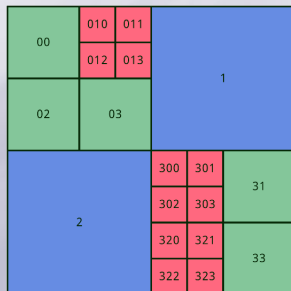
- Extraction de surface parallèle
- Gestion du Niveau de détail parallèle
 - Paralléliser l'octree
 - Paralléliser le critère de Niveau de Détail
 - Paralléliser la détection des zones de transition

Paralléliser l'octree

- Il existe une méthode pour manipuler efficacement des quadtrees restreints sur le GPU [DIP14]
- Représentation parallélisable du quadtree
- Avec un critère de Niveau de Détail parallèle
 - Basé sur la distance à la camera
- Contrôlant la taille des triangles à l'écran

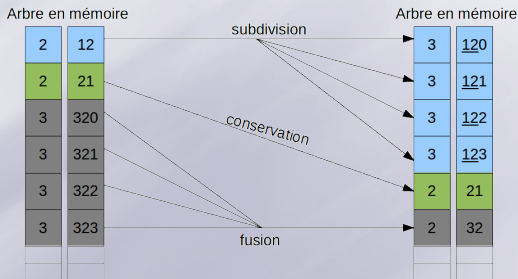
[DIP14]

- Identifie chaque cellule via un code de Morton
 - Position
 - Profondeur
 - Chemin à la racine



[DIP14]

- Arbre représenté par l'ensemble de ses feuilles
 - l'ensemble des cellules à trianguler
- Conserve cette liste dans un buffer
 - Fusion et Subdivision par ajout/suppression dans le buffer

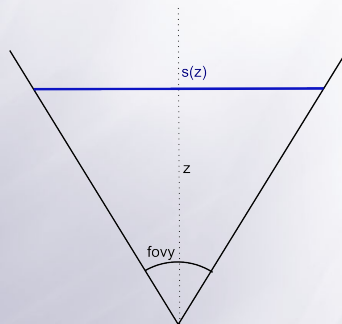
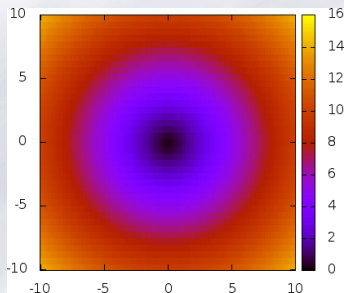


[DIP14]

- Parallélisable
- Condition
 - Le critère de Niveau de Détail doit être évaluable en parallèle
 - Pas d'accès aux cellules voisines
 - Cohérent sur tous les fils d'une cellule

Critère de Niveau de Détails

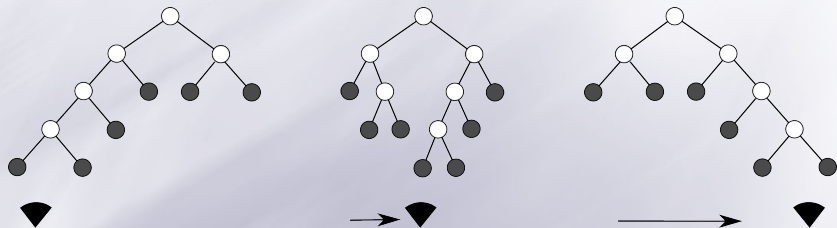
- Proposé par [DIP14]
- Critère basé sur la taille des triangles à l'écran.



$$s(z) = 2z \tan\left(\frac{fovy}{2}\right)$$

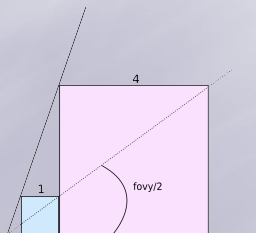
Critère de Niveau de Détails

Donne une coupe de l'arbre dépendante de la distance à la camera



Critère de Niveau de Détails

- Maintenir un octree restreint
- Critère de sélection basé sur le ratio $\frac{TailleCellule}{TailleProjete}$
 - Si $\frac{TailleCellule}{TailleProjete} > 2 \rightarrow$ Subdivision
 - Si $\frac{TailleCellule}{TailleProjete} < 0.5 \rightarrow$ Fusion
- Maintient un octree restreint tant que ratio < 4 et ratio > 0.25
- Tant que $fovy < 2 * atan(\frac{4}{5}) = 77^\circ$

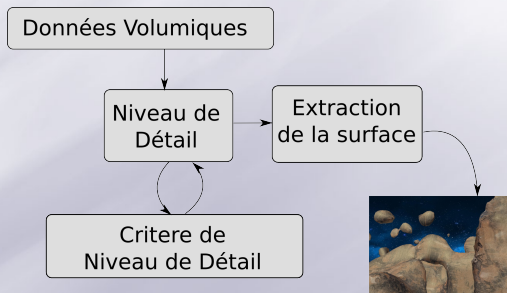


Méthode

- Extraction de surface
- Parallèle
- En temps réel
- Adaptative
- Permettant de contrôler la taille des polygones

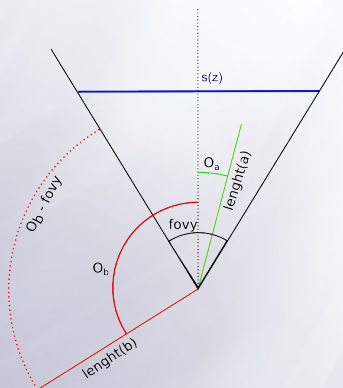
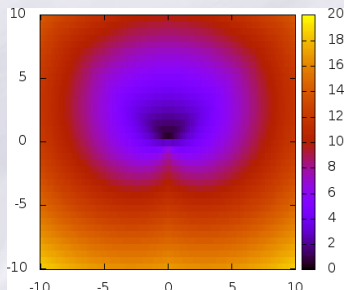
Méthode

- Combiner [LAO10] + [DIP14]
- Extraction d'isosurfaces adaptatives sur GPU



Critère de Niveau de Détails

- Prendre en compte la distance radiale



$$z' = z + \begin{cases} l(\theta - fovy), & \text{Si } \theta \geq fovy \\ 0, & \text{Sinon} \end{cases}$$

Critère de Niveau de Détails

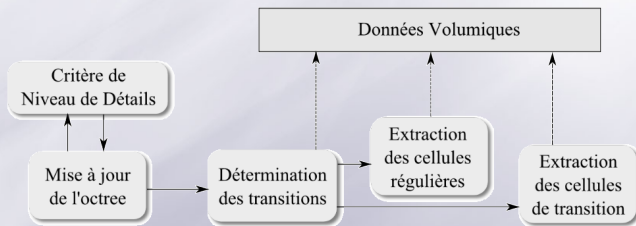
- Parallélisable
- Basé uniquement sur la distance au centre de la cellule
- Contenue dans le Code de Morton

- Identifier en parallèle les zones de transition
- Évaluer le critère sur le centre des cellules voisines
- Possible en parallèle grâce au Code de Morton

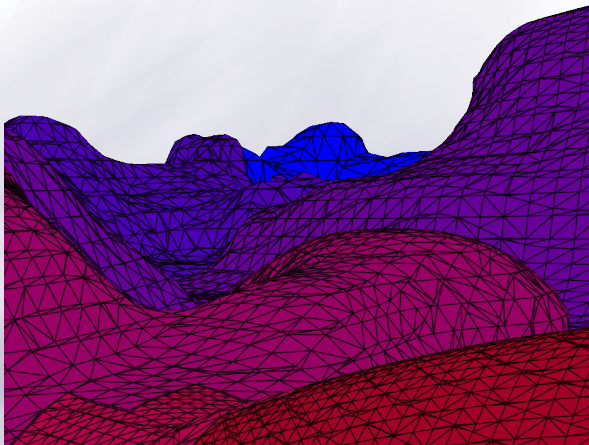
Méthode

- Extraction de surface
- Parallèle
- En temps réel
- Adaptative
- Permettant de contrôler la taille des polygones

Méthode



Taille des triangles contrôlée



Performances

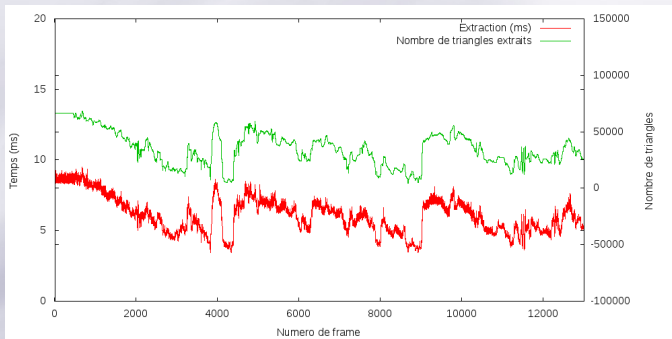
Mesurées sur une GeForce GTX 760 avec un objet 512³

Etape	Temps (ms)	%
Octree	0.30	5.3
Culling	2.05	36.3
Extraction	3.30	58.4
Total	5.65	100.0



Performances

Cellules Octree	Cellules Régulières	Cellules Transition
143K	4K	2K



Objet Volumique

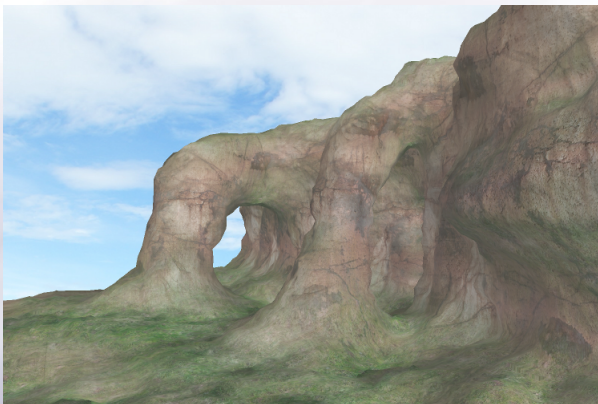
- Deux possibilités
- Fonction évaluée à la volée
 - Peu de consommation mémoire
 - Un peu plus lent à évaluer
- Texture 3D
 - Fonction complexes
 - Profite des mécanismes d'interpolation
 - Consommation mémoire importante

Exemples



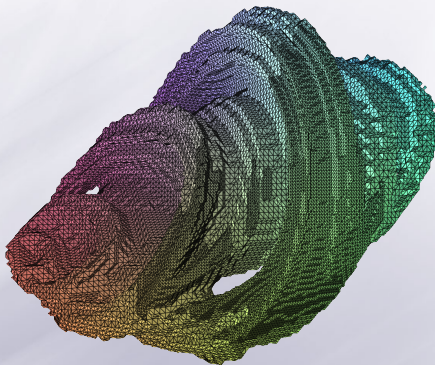
- $f(x, y, z) =$ Distance signée à la surface
- Résolution 256^3

Exemples



- $f(x, y, z) =$ Densité de matière
- Résolution 256^3
[PGGM09]

Exemples



- Objet animé

Conclusion

- Notre méthode
 - Rapide
 - Génère la géométrie à la volée, permet de manipuler des données dynamiques
 - Pas d'artefacts de rasterization
- Limitations
 - Limite à la résolution de l'objet représentable
 - Effet de popping

Références

[LC87] "Marching cubes : A high resolution 3d surface construction algorithm" W.E. Lorensen, and H.E. Cline, ACM Siggraph Computer Graphics, 1987.

[DIP14] "Quadrees on the GPU" J. Dupuy, J. C. Iehl, P. Poulin, GPU Pro 5, 2014

[LAO10] "Voxel-based terrain for real-time virtual simulations" E.S. Lengyel, J.D. Adviser-Owens, University of California at Davis, 2010

[PGGM09] "Arches : a framework for modeling complex terrains." A. Peytavie, E. Galin, J. Grosjean, S. Merillou, Computer Graphics Forum, 2009

[CHPR07] "Stochastic Simplification of Aggregate Detail" R.L. Cook, J. Halstead, M. Planck, D. Ryu, ACM Transactions on Graphics, 2007

[SBD13] "Level of Detail for Real-Time Volumetric Terrain Rendering". M. Scholz, J. Bender, C. Dachsbacher, VMV, 2013.

[SW04] "Dual marching cubes : Primal contouring of dual grids" S. Schaefer, J. Warren, Computer Graphics and Applications, 2004.

Questions

