Revue Électronique Francophone d'Informatique Graphique, Volume 2, Numéro 1, pp. 61–71, 2008

Visualisation Focus+Contexte pour l'Exploration Interactive de Maillages Tétraédriques

Sébastien Barbier [1] et Georges-Pierre Bonneau[1]

[1]Grenoble Universités - LJK - INRIA - CNRS

Résumé

L'exploration et l'analyse visuelle de grands maillages tétraédriques restent des tâches coûteuses en temps lorsque les ensembles de données sont affichés dans leur globalité. Pourtant, dans la plupart des cas, l'utilisateur n'explorera que de petites zones compactes où se concentrent les informations qu'il juge remarquables. Se basant sur ce constat, nous proposons une approche focus+contexte reposant sur une double résolution des données. Dans l'espace objet, une Région Locale d'Intérêt (RLI) - le focus - est extraite du maillage précis originel et est entourée par une représentation grossière globale - le contexte. Pour unir les deux résolutions, une connexion topologiquement valide est créée interactivement. Les techniques de rendu classiques y sont intégrées.
De plus, quand le focus est déplacé, l'extraction de la RLI ainsi que son affichage sont accélérés en utilisant la cohérence temporelle. Les dernières cartes graphiques sont utilisées afin d'accélérer le Rendu Volumique Direct.
Notre approche focus+contexte réduit de manière significative le nombre de primitives affichées ce qui permet une exploration interactive de grands maillages tétraédriques.

Mots clé : Visualisation Scientifique, Focus+Contexte, MultiResolution, Maillages Tétraèdriques

1. Introduction

Les simulations numériques de phénomènes naturels ou pour l'ingénierie sont réalisées sur des maillages composés au moins de millions de cellules. Le champ scalaire résultant est analysé visuellement mais l'affichage global de tels ensembles de données reste un processus coûteux en temps. Cependant, un rendu global précis est rarement nécessaire car l'utilisateur concentre souvent son attention sur des zones remarquables dans lesquelles une grande précision est essentielle alors qu'une simplification peut être réalisée à l'extérieur pour conserver un contexte informatif. De plus, il est capital de pouvoir explorer interactivement les alentours de la région d'intérêt afin de la situer pleinement dans son contexte.

La majorité des approches précédentes se basent sur une structure multirésolution pour n'afficher que les sous-parties visibles à la meilleure précision. Au cours d'une étape de précalcul, des structures de données multirésolution sont extraites via un algorithme de simplification. Au cours de l'exécution, la résolution est adaptée ce qui permet de réduire le nombre de tétraèdres, donc d'accélérer la phase de rendu.

Plus récemment, des algorithmes de focus+contexte monorésolution ont été proposés, restreignant le focus selon les besoins de l'utilisateur. La plupart d'entre eux ont été implémentés sur des grilles régulières ou s'appuient sur une extraction haut-niveau de l'information.

Dans cet article, nous proposons un algorithme pour des maillages tétraédriques qui mélange l'approche focus+contexte avec une structure bi-résolution. Le focus ou Région Locale d'Intérêt (RLI) peut être déplacé interactivement n'importe où dans l'ensemble de données, en particulier dans les zones remarquables. Une résolution grossière entoure la RLI comme contexte afin de guider l'exploration. Les techniques de rendu classiques telles qu'isosurfaces, plans de coupe texturés ou encore Rendu Volumique Direct (DVR), sont pleinement intégrées.

Les principales contributions sont les suivantes :
– Une approche focus+contexte efficace et centrée sur

l'utilisateur reposant sur une structure à deux résolutions pour explorer interactivement de grands maillages tétraédriques. Une telle approche permet de réduire les temps de précalculs tout en simplifiant les structures de données par rapport aux structures multirésolution.
– L'utilisation de la cohérence temporelle pour accélerer le déplacement et le rendu du focus au sein des zones remarquables.
– Des améliorations de méthodes existantes pour le DVR reposant sur l'exploitation des dernières fonctionnalités des cartes graphiques. Le tri des tétraèdres est en partie réalisé sur GPU via un algorithme de depth-peeling et un geometry shader permet d'accélerer la méthode de projection des primitives.

La suite de cet article est organisée de la manière suivante : la section 2 est un bref état de l'art sur les précédentes approches. Notre structure bi-résolution est détaillée dans la section 3 et son accélération via la cohérence temporelle dans la section 4. Les améliorations sur le pipeline graphique concernant le DVR sont décrites dans la section 5. Des résultats sont fournis dans la section 6 et nous concluons dans la dernière section.

2. État de l'Art

2.1. Simplification et Multirésolution

Afin de visualiser de grands ensembles de données, des algorithmes de simplification et multirésolution ont été proposés pour diminuer le nombre de tétraèdres affichés. Une majorité d'entre eux reposent sur des opérations de contractions d'arêtes. La simplification se base sur un ordonnancement des arêtes au sein d'une pile par rapport à l'évaluation d'une somme pondérée d'une métrique d'erreur scalaire [THJ99, GVW99] qui permet de préserver les variations du champ scalaire ; et d'une métrique d'erreur de domaine [CCM*00, NE04] qui interdit des contractions qui causeraient une grande déformation de l'aspect global du maillage. Avant chaque contraction, celle-ci est testée afin d'éviter toute modification topologique ou géométrique qui rendrait le maillage simplifié incorrect.

Les structures multiresolution sont construites en aval de ces simplifications afin d'afficher dynamiquement des maillages tétraèdriques dépendant du point de vue. Elles permettent ainsi d'afficher un maillage tétraèdrique à haute résolution dans la zone visible tout en assurant une dégradation douce vers une résolution grossière dans les parties invisibles à l'utilisateur [Lue98, XV96]. Staadt et Gross [SG98], Cignoni et al. [CFM*04] et Sondershaus et Strasser [SS05] utilisent des forêts d'arbres binaires pour stocker les dépendances entre les contractions de manière compacte. Callahan et al [CCSS05] généralisent leur système nommé HAVS pour afficher des niveaux de détails globaux. Sondershaus et Strasser [SS06] développent une segmentation des ensembles de données. Un octree guide la construction d'une hiérarchie multirésolution stockée dans un graphe direct sans cycle (DAG).

Cignoni et al. [CMS94] introduisent un outil local, la *MagicSphere*, à l'intérieur de laquelle ils appliquent un filtre multirésolution pour l'extraction d'isosurfaces. Deux maillages tétraèdriques sont utilisés : un fin et un grossier. A chaque étape, une région sphérique est définie dans l'espace objet. L'ensemble des deux maillages est entièrement parcouru afin de déterminer si leurs tétraèdres se trouvent respectivement dans la région sphérique ou en-dehors. Une isosurface est ainsi recalculée à chaque modification de la région d'intérêt, réalisant une extraction fine dans la sphère d'intérêt et une extraction grossière à l'extérieur. Aucune connexion topologique et géométrique entre les deux résolutions n'est construite, mais une transition douce est simulée via l'utilisation du alpha buffer pour mélanger les triangles de bord appartenant aux l'une des deux résolutions. Une possible accélération de la méthode est proposée se basant sur un découpage de l'espace afin de ne parcourir qu'un sous-ensemble des tétraèdres dans les deux résolutions en fonction de leurs intersections avec la sphère d'intérêt.

2.2. Approches Focus+Contexte

Récemment, des approches focus+contexte monorésolution ont été proposées pour visualiser les volumes de données selon les besoins de l'utilisateur.

Les méthodes focus+contexte les plus populaires sont élaborées pour des données médicales régulières segmentées. Hadwiger et al. [HBH03] proposent d'appliquer différentes techniques de rendu pour mettre en évidence les informations remarquables. Wang et al. [WZMK05] définissent une *Magic Volume Lens* qui agrandit dans l'espace image les données qu'elle contient. Le contexte devant tenir dans un espace image plus petit est compressé aux abords de la loupe. Viola et al. [VFSG06] proposent des solutions pour déterminer automatiquement les zones remarquables.

Pour les maillages tétraédriques, les méthodes focus+contexte existantes [DGH03, PKH04] mélangent des techniques de rendu global avec une représentation 2D des données scalaires afin de localiser les zones remarquables dans l'espace et dans le temps.

2.3. Rendu Volumique Direct

Le Rendu Volumique Direct [Max95] est une technique de rendu pour visualiser des maillages volumiques. Nous rappelons certains algorithmes permettant de la réaliser.

Les méthodes de Ray-Casting ont été implémentées sur GPU [WKME03]. Mais elles restent contraintes par la taille mémoire des cartes graphiques ce qui ne permet pas de visualiser de grands ensembles de données.

Les techniques de projection de cellules réalisent l'inté-

gration des rayons en déterminant la projection des tétraèdres dans l'espace image [ST90, WMFC02, MMFE06]. Ces méthodes nécessitent en amont un tri des cellules selon leur visibilité [Wil92, CMSW04].

Le système *HAVS* [CICS05] est une approche hybride qui exécute deux tris consécutifs sur les faces des tétraèdres : le premier calcule un ordre partiel sur le CPU et le second un ordre total sur le GPU. Mais les limitations actuelles du GPU peuvent produire des tris locaux incorrects.

3. Focus+Contexte via une structure birésolution

Nous proposons une approche focus+contexte pour des maillages tétraédriques reposant sur des résolutions fine et grossière des données. Une telle stucture birésolution a tout d'abord été introduite par Cignoni et al. [CMS94] mais uniquement pour l'extraction d'isosurfaces. Nous proposons une nouvelle approche permettant de construire une connexion valide entre les deux résolutions de manière efficace en temps pour obtenir un unique maillage et utilisant la cohérence temporelle. Contrairement aux approches multirésolution, aucune structure de données complexe n'est nécessaire. De plus, toutes les techniques de rendu classiques sont pleinement intégrées.

Nous détaillons ci-après les étapes de la construction de notre approche. L'algorithme est illustré par des schémas sur des surfaces dans un souci de clarté.

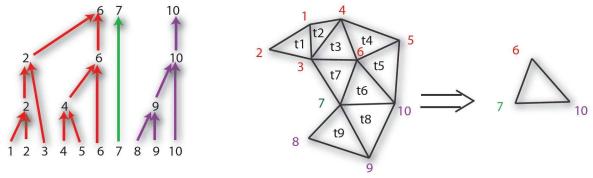

Figure 1: *Processus de Simplification. Les sommets sont regroupés en 3 clusters C_0^v={1,2,3,4,5,6}, C_1^v={7} et C_2^v={8,9,10} ; et les triangles dans C_0^t={$t_1,t_2,t_3,t_4,t_5,t_6,t_7$}, C_1^t={t_6,t_7,t_8,t_9} et C_2^t={t_5,t_6,t_8,t_9}.*

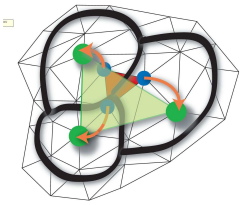

Figure 2: *Création du maillage grossier à partir de la clusterisation. Le triangle rouge appartient à trois clusters (délimités par les traits épais). Les sommets fins bleus sont projetés sur les grossiers verts transformant le triangle fin rouge en un triangle grossier vert.*

3.1. Contruction du maillage grossier

Un algorithme de simplification similaire à [CCM*00] est exécuté en préprocessing. Il construit une forêt d'arbres binaires où les feuilles sont les sommets fins initiaux et les racines les sommets grossiers finaux. L'utilisation de contractions par demi-arêtes (*half-edge collapses*) n'introduisant pas de nouveaux sommets, les sommets grossiers appartiennent au maillage initial. Ainsi, une clusterisation du maillage initial est calculée : les sommets fins qui s'effondrent sur le même sommet grossier sont regroupés en un cluster (cf. Figure 1).

Chaque sommet fin conserve l'identité de son cluster. Chaque tétraèdre fin est associé avec un à quatre clusters en fonction du regroupement de ses sommets. Le maillage grossier est déduit de la clusterisation et du maillage fin. Seuls les tétraèdres appartenant à quatre clusters distincts sont conservés et transformés en projetant leurs sommets fins sur les sommets grossiers correspondants (cf Figure 2).

3.2. Extraction de la Région Locale d'Intérêt

La RLI est construite dans l'espace objet à partir d'une sphère dont le centre et le diamètre sont définis par l'utilisateur. La RLI consiste en l'ensemble des tétraèdres fins contenus à l'intérieur de cette sphère. Pour localiser rapidement le centre d'intérêt au sein du maillage fin, un octree est construit sur les barycentres de tous les tétraèdres. Le tétraèdre le plus proche du centre est alors considéré comme le centre de la RLI et est appelé *racine*. Dans le but de profiter de la cohérence temporelle dans le cas d'une exploration, la recherche recommence toujours à partir du dernier noeud trouvé au sein de l'octree. Dans tous les cas, la localisation du tétraèdre *racine* est négligeable devant le reste du pipeline graphique.

En partant du tétraèdre *racine* et en utilisant les relations d'adjacence, la RLI est agrandie via un parcours en profondeur jusqu'à ce que tous les tétraèdres contenus dans la sphère virtuelle soient ajoutés. Chaque tétraèdre nouvellement ajouté est marqué afin d'éviter des ajouts intempestifs. Parmi ces tétraèdres, on appelle *interne de bord* ceux intersectant la sphère, *interne complet* les autres (cf Figure 3). Les tétraèdres interne complet sont stockés dans une table de hachage et les internes de bord dans une liste dite *de bord* afin de minimiser la complexité de l'algorithme lors de l'utilisation de la cohérence temporelle (cf section 4.1). En effet, la table de hachage garantit un accès, une insertion et une suppression en temps constant $O(1)$. Les tétraèdres de bord seront quant à eux tous parcourus séquentiellement, d'où l'utilisation d'une simple liste chaînée.

Dans le cas où la sphère virtuelle intersecterait plusieurs composantes non connexes, des parcours supplémentaires de l'octree sont réalisés pour trouver les tétraèdres non mar-

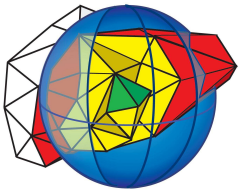

Figure 3: *La RLI est composée de tous les tétraèdres contenus dans la sphère bleue. Tous les tétraèdres colorés sont internes. Les jaunes sont complets, les rouges sont de bord. Le tétraèdre central vert est le tétraèdre racine.*

qués. L'algorithme d'ajout est ensuite appliqué comme précédemment à partir de ces nouveaux tétraèdres racines.

Pour plonger la RLI au sein du contexte, les clusters intersectant la RLI et leurs sommets grossiers associés sont marqués *actifs*. Les sommets et les tétraèdres fins appartenant à un cluster *actif* sont aussi appelés *actifs*. De plus, les tétraèdres fins ayant quatre sommets actifs sont nommés *intérieurs* sinon *de lien*.

Le contexte est défini par tous les tétraèdres grossiers qui n'intersectent pas le focus. Deux composantes connexes sont ainsi obtenues.

3.3. Lien entre les deux résolutions

Pour obtenir un unique maillage valide, une connexion doit être construite entre le focus et le contexte. Cette transition doit être continue entre les deux résolutions. Ainsi, le focus sera entouré par le contexte sans perte d'informations qui pourraient se révéler essentielle à la compréhension globale de la simulation.

Pour raccorder la RLI au maillage grossier, des tétraèdres sont ajoutés. Afin d'assurer une robustesse topologique, les heuristiques reposant sur la géométrie sont évitées. Nous proposons une solution s'appuyant sur la clusterisation, à la fois rapide et valide topologiquement dans toutes les situations.

Un unique maillage est ainsi obtenu, composé de tétraèdres grossiers, fins et étirés :

- Tétraèdres grossiers : tous les tétraèdres ayant quatre sommets inactifs ;
- Tétraèdres fins : tous les tétraèdres actifs intérieurs, incluant ceux contenus dans la RLI (tétraèdres internes) ;
- Tétraèdres étirés : tous les tétraèdres actifs de lien sont étirés en projetant les sommets fins inactifs sur leurs sommets grossiers associés. Cela peut produire des tétraèdres dégénérés qui sont enlevés au cas par cas - sauf pour le DVR.

La Figure 4 illustre la construction de ce maillage unique. Notre solution augmente le nombre de tétraèdres affichés mais en contrepartie assure une construction rapide et simple. De plus, le lien est topologiquement correct.

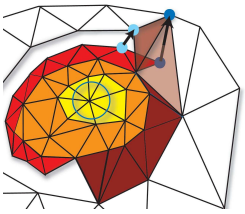

Figure 4: *Le raccord est en cours de construction. Les triangles fins de lien (rouge) sont étirés (carmin) pour relier les deux résolutions et combler le trou. Cela est fait en projetant les sommets fins inactifs sur leur sommet grossier correspondant. Le focus est représenté par le cercle bleu. Les triangles jaunes et orange sont internes et intérieurs, respectivement, alors que les blancs font partie du contexte.*

4. Cohérence Temporelle

Les techniques de rendu telles que les plans de coupe texturés, l'extraction d'isosurfaces ou le DVR sont intégrés au sein de notre approche focus+contexte. Pour accélérer l'ensemble du pipeline graphique, l'extraction de la RLI et les techniques de rendu utilisent la cohérence temporelle lors d'un déplacement du focus.

4.1. Mise à jour de la Région Locale d'Intérêt

Lorsque l'utilisateur déplace le centre d'intérêt lors d'une exploration, un nouveau tétraèdre *racine* est recherché. Si celui-ci n'appartient pas à l'ancienne RLI, la table de hachage et la liste de bord sont effacées et une nouvelle région est reconstruite comme expliqué en section 3.2.

Sinon, la nouvelle RLI est mise à jour par rapport à l'ancienne en trois étapes, comme illustré en Figure 5 :

1. **Mise à jour de la liste de bord :** la liste de bord est parcourue. Tout tétraèdre n'appartenant plus à la nouvelle RLI est enlevé de la liste et est stocké dans une liste dite *extérieure* ;
2. **Enlever tous les tétraèdres n'appartenant plus à la nouvelle RLI :** la liste extérieure créée en 1. est utilisée comme amorce de la recherche. En utilisant les relations d'adjacence, l'ensemble des tétraèdres interne complet n'appartenant plus à la nouvelle RLI sont déterminés. Ceux-ci sont enlevés de la table de hachage. Lors de ce parcours, les tétraèdres déterminés comme de bord sont ajoutés à la liste de bord.
3. **Ajout des nouveaux tétraèdres :** La nouvelle liste de bord est utilisée comme amorce. Les tétraèdres sont visités via les relations d'adjacence. Les tétraèdres complets sont stockés dans la table de hachage, les tétraèdres de bord dans la liste de bord.

La table de hachage assure une insertion et une suppression en temps constant. La complexité globale de la mise à

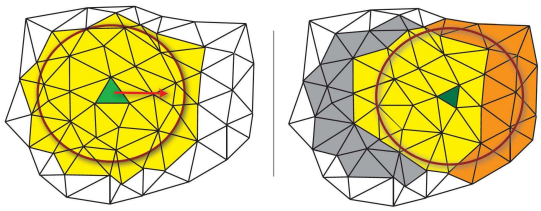

Figure 5: *Application de la cohérence temporelle. La primitive verte centrale est déplacée vers la droite. Les primitives enlevées durant la mise à jour sont grises, alors que les nouvelles sont en orange.*

jour est donc linéaire en nombre de tétraèdres enlevés puis ajoutés. Cette complexité est négligeable quand l'utilisateur déplace le focus avec de petits déplacements, ce qui est le cas lors de l'exploration des alentours des zones remarquables.

4.2. Isosurfaces

Un *Marching Tetrahedra* permet d'extraire les isosurfaces. L'isosurface grossière est calculée une fois par isovaleur. Par contre, l'isosurface fine est extraite des tétraèdres intérieurs et de lien et doit être mise à jour lorsque le focus est déplacé.

Pour cela, la cohérence temporelle minimise le nombre d'extractions. Quand un cluster devient actif, l'isosurface correspondante est calculée une fois pour tous ses tétraèdres (intérieurs et de lien) et est sauvegardée. Ainsi, lorsque la RLI est déplacée mais que le cluster reste actif, aucune nouvelle extraction n'est faite.

La construction du maillage unique au niveau tétraèdrique garantit lors de l'extraction de l'isosurface une continuité entre les deux résolutions de celle-ci.

4.3. Rendu Volumique Direct

Le DVR est intégré au sein de notre structure bi-résolution. Afin d'éviter les limitations dues à la mémoire ou à des opérations de lecture/écriture incertaines des cartes graphiques, un ordre de visibilité sur CPU : *SXMPVO* [CMSW04] et une technique de splatting sur GPU : GATOR [WMFC02] sont implémentés.

Nous rappelons dans cette section les étapes de l'algorithme SXMPVO. Puis, nous détaillons les transformations faites pour intégrer la cohérence temporelle et le DVR au sein de notre approche.

4.3.1. SXMPVO

L'algorithme SXMPVO garantit un ordre total de visibilité pour les maillages non convexes et non connexes sans cycles. Il est composé de quatre étapes :

Étape I : Elle détermine un ordre partiel de visibilité en utilisant les relations d'adjacence dans les zones connexes. Pour cela, le signe du produit scalaire entre la direction de vue et les normales de la face commune entre deux tétraèdres permet de trouver l'occludant et l'occludé. Chaque tétraèdre peut donc aussi se situer dans l'ordre de visibilité par rapport à ses voisins ;

Étape II : Elle trie les faces de bord selon leur profondeur afin de pouvoir déterminer les relations d'occlusions entre les parties non connexes et non convexes dans l'étape suivante ;

Étape III : Elle détermine l'ordre de visibilité des faces de bord dans l'espace objet via un lancer de rayons stockés dans un A-Buffer [Car84]. Le A-Buffer est une structure de données mémorisant pour chaque pixel de l'image une liste ordonnée en fonction de la profondeur, des faces qu'il intersecte. En enlevant pour chaque rayon la première face intersectée, on obtient alors des couples ordonnés de faces, la première cachant la seconde. Cela permet ainsi de construire les relations de visibilité manquantes.

Étape IV : Elle réalise un parcours en profondeur de cet ordre de visibilité total à partir des faces visibles afin d'afficher en utilisant le GPU l'ensemble des primitives de l'arrière vers l'avant.

4.3.2. SXMPVO Cohérent

Dans cette sous-section, nous proposons d'utiliser la mise à jour de la RLI pour accélérer une partie de l'algorithme SXMPVO dans le cas où l'utilisateur a fixé son point de vue. En effet, le rendu volumique étant dépendant du point de vue, l'ordre de visibilité doit être recalculé pour toutes les primitives de la RLI si celui-ci est modifié. Dans le reste de cette section, on supposera donc que l'utilisateur a fixé son point de vue et déplace uniquement le focus.

L'étape I de SXMPVO détermine un ordre partiel en se basant sur les relations d'adjacence. Lors d'une exploration où le point de vue n'est pas modifié, cet ordre de visibilité est inchangé. Ainsi, seuls les tétraèdres nouvellement ajoutés doivent être insérés dans le tri. Cette insertion est faite lors de l'étape 3 de la mise à jour de la RLI - cf section 3.2. De plus, lorsque les tétraèdres sont enlevés de la RLI, leurs relations de visibilité sont effacées pour ne pas être envoyés au GPU lors du parcours en profondeur.

Cette mise en place de la cohérence temporelle permet de réduire la complexité de l'étape I au nombre de tétraèdres ajoutés et enlevés lors de la mise à jour.

L'étape IV est modifiée pour afficher l'ensemble des tétraèdres de la RLI. Durant le parcours en profondeur, les primitives sont marquées pour éviter des rendus multiples ou détecter des cycles de visibilité. Le test de visite a été modifié afin de connaître durant quel rendu le tétraèdre a été

marqué. Si la marque correspond au rendu actuel, le parcours est arrêté, sinon il correspond au rendu précédent et le tétraèdre est parcouru. Cette modification assure que tous les tétraèdres et en particulier les anciens, sont envoyés au GPU.

4.3.3. Intégration du Rendu Volumique Direct

Dans notre structure focus+contexte, le DVR peut être appliqué à la RLI, sur les deux résolutions ou sur le maillage unique valide.

Si le point de vue est fixe, le SXMPVO cohérent est appliqué au sein de la RLI. Dans le cas des deux résolutions, SXMPVO est appliqué sur les deux composantes connexes.

Afficher le maillage unique valide avec le DVR implique un stockage en mémoire de l'ensemble des cellules de lien alors qu'elles sont déduites par projection des sommets inactifs pour les autres techniques. En effet, pour le DVR les relations d'adjacence entre les cellules de lien sont nécessaires pour assurer un ordre partiel correct en étape I, ce qui implique leur mémorisation. Elles sont mises à jour à chaque fois que la RLI est déplacée.

Cette mise à jour se déroule pendant l'étape I :

1. **Trouver les tétraèdres de lien**. Les tétraèdres de lien sont sauvegardés dans une table de hachage. Pour les trouver, l'ensemble des tétraèdres grossiers et des tétraèdres fins actifs sont testés sur leur statut (fins, grossiers ou étirés). Tous les tétraèdres de lien détectés sont stockés, dégénérés ou non. Les relations d'adjacence avec les tétraèdres fins et grossiers sont calculées.
2. **Mettre à jour les relations d'adjacences entre cellules de lien**. Pour cela, les relations d'adjacence entre les tétraèdres fins correspondant sont utilisés. En effet, chaque cellule de lien est en fait une cellule fine étirée. Ces cellules fines sont donc visitées jusqu'à trouver une cellule fine dont la cellule de lien correspondante n'est pas dégénérée. A la fin de cette étape, les cellules de lien dégénérées sont enlevées.

A la fin de l'étape I, un ordre partiel est ainsi obtenu pour le maillage unique valide.

5. Améliorations GPU pour le Rendu Volumique Direct

Les implémentations originales de SXMPVO et de GATOR ont certaines limitations : le A-Buffer (étape III) de SXMPVO est calculé sur le CPU et GATOR réalise des calculs redondants sur le GPU. Nous proposons des améliorations afin de pallier à ces inconvénients.

5.1. Depth-Peeling

Pour accélérer SXMPVO, une implémentation sur GPU de l'algorithme de *depth-peeling* [Eve01, GHLM05] est réalisée pour remplacer le A-Buffer. Cette idée a été citée mais jamais implémentée pour SXMPVO.

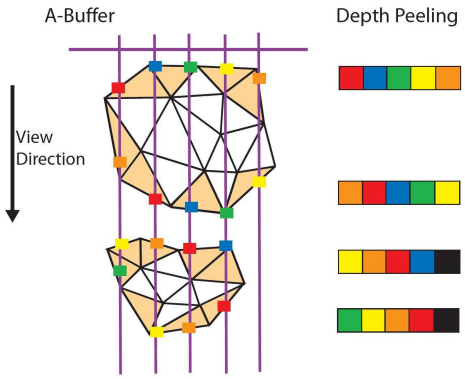

Figure 6: *Correspondance entre A-Buffer et Depth-Peeling. Les tétraèdres de bord sont orangés. Chaque intersection au niveau du A-Buffer est représentée par un carré de couleurs. Les textures correspondantes pour le depth-peeling sont associées.*

En effet, le A-Buffer consiste à émuler sur le CPU une rasterization des faces de bord qui assurerait un tri selon la profondeur de celles-ci en sortie. Cela peut être facilement implémenté sur le GPU en utilisant le depth-peeling. La figure 6 illustre la correspondance entre les deux méthodes. Chaque intersection avec une face de bord est représentée par un carré coloré. Les textures obtenues par depth-peeling sont aussi indiquées en correspondance.

Le depth-peeling réalise des rendus successifs de la scène. Chaque rendu enlève les faces qui étaient visibles au rendu précédent. Si l'on stocke dans des textures chacun de ces rendus successifs, on calcule alors l'ordre suivant : les faces les plus proches, puis les secondes jusqu'aux plus profondes. On obtient ainsi un A-Buffer efficace porté sur le GPU.

En envoyant au GPU uniquement les faces de bord déduites lors de l'étape I, on peut ainsi obtenir au sein de textures l'ensemble des relations de visibilité dans l'espace image. Ces textures sont ensuite rapatriées sur le CPU - exceptée la première qui n'intervient pas dans l'ordre de visibilité ; et assemblées par couple pour retrouver les relations d'occlusions.

Le depth-peeling est implémentée de manière efficace où chaque passe détermine une profondeur. Le rendu off-screen est réalisé via des Frame Buffer Objects. Des Pixel Buffer Objects sont utilisés pour accélérer le rapatriement des textures. De plus, l'espace image est découpé en tiles (4x4) afin de détecter plus rapidement les parties de l'écran qui sont vides et donc rapatrier des textures plus petites.

Cette approche permet ainsi d'éliminer la phase II triant les faces de bord et accélère la phase III qui était le goulot d'étranglement du SXMPVO original.

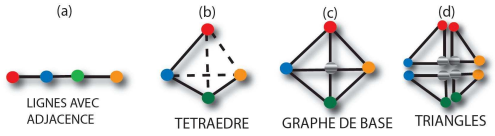

Figure 7: *Implementation du geometry shader. Les tétraèdres (b) sont représentés par des GL_LINES_ADJACENCY_EXT (a). Au sein du shader, ils sont projetés sur le graphe de base (c) puis les triangles correspondants (d) sont envoyés au fragment shader.*

5.2. Geometry Shader

Afin de réaliser le rendu par projection des tétraèdres, il est nécessaire de connaître la projection de ceux-ci dans l'espace image. Pour cela, chaque tétraèdre est projeté sur un graphe de base qui déterminera le nombre de triangles résultant contribuant au rendu final. Ceux-ci peuvent varier de un à quatre en fonction des configurations. Lorsque quatre triangles sont nécessaires, un nouveau sommet doit être introduit, déterminé par une double interpolation linéaire.

L'algorithme GATOR est la première méthode de projection de tétraèdres implémentée sur GPU. Elle est confrontée au problème que la connectivité était inacessible au sein du GPU. Ainsi pour créer des triangles, ceux-ci se devaient d'être envoyés au GPU dès le départ. De plus, chaque tétraèdre se doit d'être connu afin d'être projeté. Afin de pallier à cet inconvénient, GATOR propose d'envoyer pour chaque sommet appartenant aux triangles projetés, l'ensemble des informations concernant le tétraèdre courant. Ainsi, la projection pourra être faite au sein du vertex shader. Chaque tétraèdre est ainsi envoyé cinq fois (via l'utilisation de triangle strips) pour garantir une transmission sans perte au fragment shader. Pour plus de précision, voir [WMFC02]. Cette approche transmet ainsi une grande redondance d'information à la carte graphique et réalise des calculs tout aussi redondants en son sein.

Pour améliorer cette approche, le vertex shader est transformé en geometry shader qui permet d'obtenir l'information de connectivité et de transformer la géométrie - cf figure 7. Cela permet ainsi certaines améliorations :

- Un tétraèdre est envoyé une seule fois au lieu de cinq fois sous la forme d'une ligne avec adjacence de quatre points repérés par leur position et leur couleur. La taille des données envoyée au GPU est donc grandement diminuée.
- Chaque tétraèdre est projeté une seule fois au sein du geometry shader via quelques modifications de l'implémentation originale.
- De plus, aucune primitive dégénérée n'est envoyée au fragment shader. Cette solution évite les deux passes que proposaient Marroquim [MMFE06].

6. Résultats

Nous avons implémenté nos résultats sur un ordinateur Windows XP 32 bits 2,8 GHz, 2 Go de RAM avec une GeForce 8800 GTS de 640 Mo de mémoire vidéo. Nous avons testé notre algorithme sur des ensembles de données listés dans le Tableau 1. L'ensemble DTI est un maillage médical régulier que nous avons converti en tétraèdres. Il représente un cerveau. Le Bucky représente une molécule de carbone composée de 60 atomes. Le Blunt modélise un flux d'air au sein d'une conduite d'aération. Enfin, le Post modélise un flux d'oxygène liquide le long d'un plan percé en son centre par un cylindre.

6.1. Résultats Visuels

Des résultats visuels de notre approche sont visibles en Figures 10, 12 et 13. Pour mettre en évidence la RLI, une fonction de transfert en couleurs est appliquée au sein du focus et en niveau de gris au sein du contexte. De plus, l'aspect découpé de la RLI pour le DVR est adouci via l'utilisation d'un stencil buffer.

Notre méthode peut, dans certains cas, produire des inversions de tétraèdres ou des tétraèdres mal conditionnés dit *slivers* au sein du lien, cause de certains artefacts visuels. Cependant, ils sont vraiment limités puisque sur l'ensemble de nos tests moins de 0,01% des tétraèdres étirés s'inversent et moins de 7% ont un ratio d'aspect supérieur à 20.

La Figure 10 montre l'extraction d'une isosurface au sein du Post. La RLI est déplacée au plus près du tube pour comprendre l'écoulement de l'oxygène liquide. La construction du lien entre les deux résolutions apportent clairement les informations manquantes dues au trou et cela permet une meilleure compréhension du phénomène par rapport à un simple mélange des résolutions au niveau du raccord via un alpha-blending.

La Figure 12 illustre le DVR appliqué au Blunt. La RLI est placée dans la zone de turbulence, où la densité de l'air varie le plus fortement. Déplacer la RLI le long de l'arête permet de comprendre son influence sur les variations du flux d'air. L'analyse peut être affinée grâce à un rendu global.

Les différentes techniques de rendu sont rassemblées en Figure 13. L'utilisateur recherche l'hypotalamus rendu en DVR. Il déplace sa RLI en se localisant grâce au plan de coupe texturé et à l'extraction de l'isosurface. La RLI raffine aussi l'isosurface.

6.2. Construction de la RLI

La figure 8 illustre le gain temporel obtenu lors de la construction de la RLI via l'utilisation de la cohérence temporelle. Un coefficient multiplicateur de l'ordre de 4 est ainsi obtenu. De plus, les étapes de la mise à jour de la RLI se répartissent de la manière suivante : l'étape 1 est négligeable,

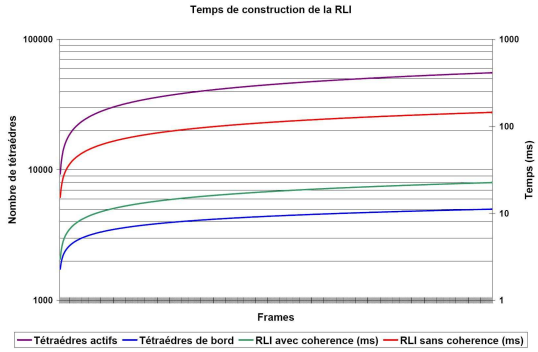

Figure 8: *Cohérence Temporelle pour la construction de la RLI. Les courbes violette et bleue indiquent le nombre de tétraèdres actifs et de bord. Mise en correspondance, les courbes rouge et verte illustrent l'apport de l'utilisation de la cohérence temporelle lors de la construction de la RLI.*

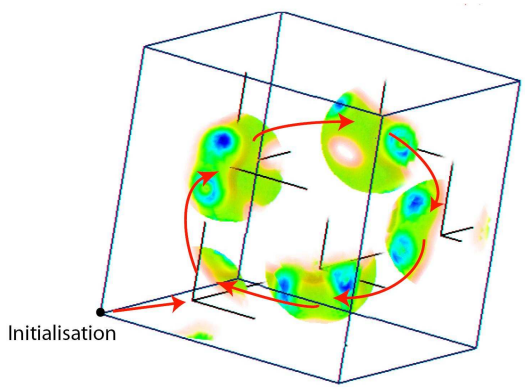

Figure 9: *Déplacement de la RLI au sein du Bucky réalisé pour nos tests.*

l'étape 2 utilise 19% du temps de mise à jour et l'étape 3 81%.

6.3. Temps de Rendu

Pour valider nos améliorations de l'étape III de SXMPVO et du pipeline de GATOR, une RLI composée de 121 159 tétraèdres est affichée au sein du Blunt afin de founir une complexité pertinente. Les résultats pour les isosurfaces et le DVR sont calculés avec le même déplacement du centre d'intérêt au sein du Bucky correspondant à l'exploration des atomes de carbone (cf figure 9). Nous avons testé notre approche sur les autres ensembles et les temps de rendu sont similaires.

6.3.1. Depth-Peeling et Geometry Shader

La Table 2 illustre l'amélioration en temps obtenue grâce à l'utilisation du depth-peeling (III) et du geometry shader

Nom	sommets	tétraèdres	champ scalaire
DTI	950,272	4,596,765	$[1, 203]$
Bucky	262,144	1,250,235	$[0, 254]$
Post	108,300	616,050	$[-0.54, 4.4]$
Blunt	31,858	222,414	$[0.19, 4.98]$

Table 1: *Ensembles de données utilisés par les résultats.*

Taille Image	SXMPVO Original				SXMPVO Amélioré			
680x840	I+ II	III	IV	R	I	III	IV	R
0,3%	195	24	139	2,75	188	31	54	3,5
2,3%	257	87	138	2,1	185	33	55	3,5
31,1%	381	476	138	1	185	42	55	3,4
99,9%	242	1629	139	0,5	187	66	56	3,1
945x1210	I+ II	III	IV	R	I	III	IV	R
2,1%	296	122	139	1,9	187	64	55	3,2
29,3%	338	1173	137	0,7	189	78	56	3,0
99,9%	204	2962	134	0,3	187	126	55	2,6

Table 2: *Comparaisons du temps de rendu pour le DVR pour deux résolutions d'image. 121 259 tétraèdres sont affichés. Les temps sont exprimés en millisecondes pour les étapes (I,II,III,IV) de SXMPVO. Le rendu global (R) est en fps. Le pourcentage de l'image occupé par la projection de la RLI est indiqué.*

	Sans	Avec
Extraction	250.000 tet/s	650.000 tet/s
Rendu	500.000 tet/s	500.000 tet/s

Table 3: *Nombres de tétraèdres extraits avec ou sans la cohérence temporelle, moyenne sur plusieurs tailles de RLI.*

(IV). Aucune cohérence temporelle n'est utilisée, différentes résolutions d'images sont essayées.

Alors que le nombre de pixels augmente, notre nouvelle phase III devient de plus en plus rapide par rapport à l'ancienne. Zoomer sur la RLI ralentit peu cette phase.

Le geometry shader est implémenté avec la table de projection de GATOR. Nous avons aussi essayé celle de Marroquim mais les temps de rendu sont plus lents (63 au lieu de 55 ms) sans différences visuelles remarquables. Les temps globaux de rendu sont divisés en moyenne par 2,5 par rapport à l'approche originale.

6.3.2. Isosurfaces

La Table 3 compare l'extraction d'isosurfaces avec ou sans la cohérence temporelle moyennant plusieurs parcours avec différentes tailles de RLI. Avec la cohérence temporelle, ce débit est multiplié par 2,6.

6.3.3. Rendu Volumique Direct

La Figure 11 montre le nombre d'images par seconde pour le DVR en utilisant le depth-peeling et le geometry shader dans une fenêtre 680x840.

Dans un souci de clarté, nous distinguons les algorithmes original et cohérent. Nous séparons aussi le rendu avec et

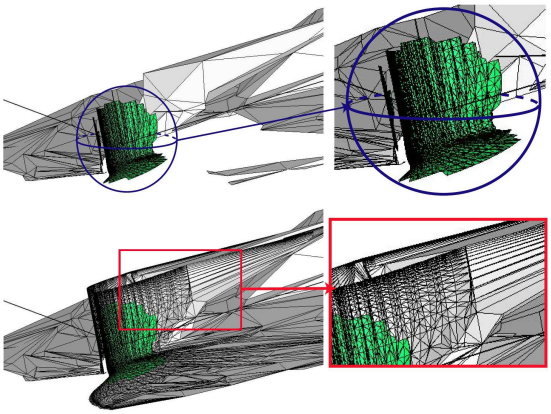

Figure 10: *Extraction d'isosurfaces au sein du Post pour la RLI seulement (en haut) et avec un unique maillage valide (en bas). La sphère d'intérêt est dessinée en haut à gauche, zoomée en haut à droite . La RLI met en évidence l'écoulement de l'oxygène liquide.*

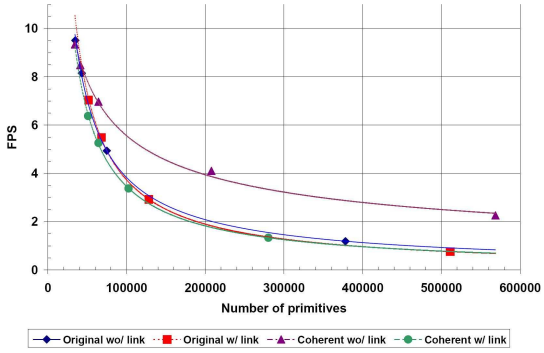

Figure 11: *Images par seconde obtenues via notre approche par rapport au nombre de tétraèdres affichés via un DVR.*

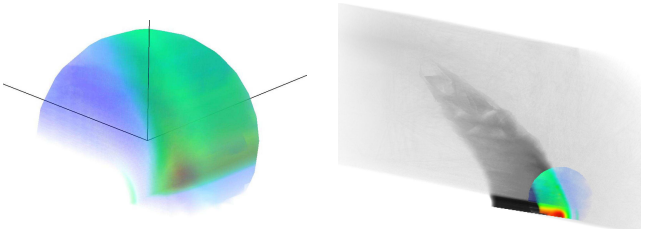

Figure 12: *Rendu Volumique Direct de l'ensemble de données Blunt Fin. La RLI est placée à proximité du choc où la densité de l'air varie fortement (à gauche). Un point de vue différent montre l'intégration du focus au sein du contexte et permet d'obtenir une compréhension globale (à droite).*

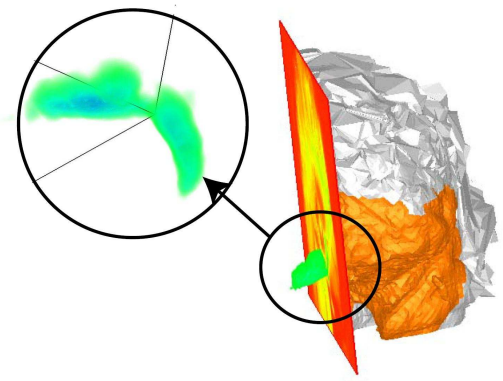

Figure 13: *Integration de plusieurs techniques de rendu pour analyser l'ensemble de données DTI. Un plan de coupe texturé permet de localiser les zones remarquables (en vert). La forme globale du cerveau est affichée via une extraction d'isosurface. Un Rendu Volumique Direct de l'hypothalamus (zoom) et une extraction d'isosurfaces sont réalisés au sein de la RLI.*

sans le lien. Un rendu avec le maillage unique est plus approprié pour une analyse des zones locales remarquables alors que le rendu de la RLI en DVR avec d'autres rendus appliqués au contexte l'est plus pour une localisation ou une exploration globale.

Nos temps de rendu quand le maillage unique est utilisé sont similaires voire meilleurs aux dernières approches multirésolution, mais notre technique assure un temps de préprocessing moins important, des structures de données plus simples et une extraction de la zone de focus à n'importe quel endroit dans l'ensemble de données. On peut de plus remarquer que :

– La cohérence temporelle accélère le temps d'affichage global quand le focus est rendu sans le lien. Cette situation est bien adaptée à une exploration rapide des maillages. Ainsi, un déplacement interactif est garanti pour localiser les zones remarquables.
– La contruction et la mise à jour du lien introduisent de légers surcoûts en étape I mais qui sont sans réels impacts sur les temps globaux de rendu.

Notre approche focus+contexte assure des temps de rendu interactifs si la RLI conserve une taille raisonnable par rapport à la taille de l'ensemble de données. Ce n'est pas une contrainte forte car la majorité des informations remarquables se trouvent dans des zones compactes. Une RLI de diamètre égale à 10-20% de la diagonale de la boîte englobante du maillage semble un bon choix qui garantit 3 à 9 images par seconde avec ou sans la construction du lien. Un plus grand focus peut être même défini interactivement sans la construction du lien.

7. Conclusion et Travaux futurs

Nous venons de présenter une approche focus+contexte permettant d'explorer interactivement de grands ensembles

de données tétraédriques reposant sur une structure bi-résolution. Une RLI est extraite du maillage fin dans l'espace objet et est entourée par une représentation grossière qui définit le contexte. La clusterisation construite lors d'un traitement en préprocessing permet d'extraire rapidement un maillage unique lors de l'exécution en unifiant la RLI avec son contexte. Les techniques de rendu ont été intégrées dans notre approche. La cohérence temporelle ainsi que les dernières fonctionnalités des cartes graphiques sont pleinement exploitées pour assurer une interactivité de l'exploration. Bien que l'approche propose une simplification algorithmique des précalculs et de la construction d'une structure multi-résolution pour la définition de zones d'intérêts locales, plusieurs problèmes restent à résoudre.

Notre algorithme de simplification reste assez complexe et coûteux bien que se basant sur les algorithmes de simplification usuels garantissant la conservation scalaire et topologique. De plus, la clusterisation issue de cette simplification ne résout pas les éventuels problèmes d'invertion de tétraèdres dans des cas rares critiques. Les tétraèdres grossiers obtenus ne sont pas géométriquement "jolis" et par conséquent des tétraèdres de lien de qualité ne peuvent être assurés. D'autres algorithmes de simplification pourraient donc être envisagés garantissant ces deux critères : rapidité et qualité visuelle du maillage. On pourrait par exemple faire appel à des tétraédralisations de Delaunay pour répondre à ces problèmes.

De plus, l'utilisation plus importante des cartes graphiques et de leurs fonctionnalités pourraient être possibles dans certaines parties de notre algorithme afin d'extraire le schéma bi-résolution et d'appliquer les techniques de visualisation. L'avènement du geometry shader abonde dans ce sens et par exemple, une extraction d'isosurface entièrement sur GPU pourrait être envisageable. L'utilisation du GPU augmenterait ainsi l'interactivité de notre approche.

Enfin, le temps de calcul du rendu volumique reste assez coûteux. Malgré cela, la technique de projection de cellules est adaptée à notre situation dans le sens où notre maillage est modifié de manière dynamique ce qui nous empêche d'utiliser les méthodes de Ray-Casting ou HAVS qui sont adaptées à des maillages statiques. A l'opposée, notre nouvelle implémentation de PT assure un envoi direct des données au GPU sans modifications préalables des données. Il serait donc intéressant d'accélérer cette méthode de projection de cellules en proposant une approche entièrement programmable sur la carte graphique.

Références

[Car84] CARPENTER L. : The a-buffer, an antialiased hidden surface method. *SIGGRAPH Comput. Graph.. Vol. 18*, Num. 3 (1984), 103–108.

[CCM*00] CIGNONI P., CONSTANZA D., MONTANI C., ROCCHINI C., SCOPIGNO R. : Simplification of tetrahedral meshes with accurate error evaluation. In *VIS '00 : Proceedings of the conference on Visualization '00* (2000), pp. 85–92.

[CCSS05] CALLAHAN S., COMBA J., SHIRLEY P., SILVA C. : Interactive rendering of large unstructured grids using dynamic level-of-detail. In *IEEE Visualization '05* (2005), pp. 199–206.

[CFM*04] CIGNONI P., FLORIANI L. D., MAGILLO P., PUPPO E., SCOPIGNO R. : Selective refinement queries for volume visualization of unstructured tetrahedral meshes. *IEEE Transactions on Visualization and Computer Graphics. Vol. 10*, Num. 1 (2004), 29–45.

[CICS05] CALLAHAN S., IKITS M., COMBA J., SILVA C. : Hardware-assisted visibility ordering for unstructured volume rendering. *IEEE Transactions on Visualization and Computer Graphics. Vol. 11*, Num. 3 (2005), 285–295.

[CMS94] CIGNONI P., MONTANI C., SCOPIGNO R. : Magicsphere : an insight tool for 3d data visualization. *Computer Graphics Forum. Vol. 13*, Num. 3 (1994), 317–328.

[CMSW04] COOK R., MAX N., SILVA C. T., WILLIAMS P. L. : Image-space visibility ordering for cell projection volume rendering of unstructured data. *IEEE Transactions on Visualization and Computer Graphics. Vol. 10*, Num. 6 (2004), 695–707.

[DGH03] DOLEISCH H., GASSER M., HAUSER H. : Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03 : Proceedings of the symposium on Data visualisation 2003* (2003), pp. 239–248.

[Eve01] EVERITT C. : Interactive order-independent transparency. In *Technical report, NVIDIA Corporation* (may 2001).

[GHLM05] GOVINDARAJU N. K., HENSON M., LIN M. C., MANOCHA D. : Interactive visibility ordering and transparency computations among geometric primitives in complex environments. In *SI3D '05 : Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM Press, pp. 49–56.

[GVW99] GELDER A. V., VERMA V., WILHELMS J. : Volume decimation of irregular tetrahedral grids. *Computer Graphics International* (1999), 222.

[HBH03] HADWIGER M., BERGER C., HAUSER H. : High-quality two-level volume rendering of segmented

data sets on consumer graphics hardware. In *VIS '03 : Proceedings of the 14th IEEE Visualization 2003* (2003), p. 40.

[Lue98] LUEBKE D. P. : *View-Dependent Simplification of Arbitrary Polygonal Environnements*. PhD thesis, Chapel Hill, University of North California, 1998.

[Max95] MAX N. : Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics. Vol. 1*, Num. 2 (1995), 99–108.

[MMFE06] MARROQUIM R., MAXIMO A., FARIAS R., ESPERANCA C. : Gpu-based cell projection for interactive volume rendering. *SIBGRAPI : Brazilian Symposium on Computer Graphics and Image Processing. Vol. 0* (2006), 147–154.

[NE04] NATARAJAN V., EDELSBRUNNER H. : Simplification of three-dimensional density maps. *IEEE Transactions on Visualization and Computer Graphics. Vol. 10*, Num. 5 (2004), 587–597.

[PKH04] PIRINGER H., KOSARA R., HAUSER H. : Interactive focus+context visualization with linked 2d/3d scatterplots. In *2nd International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2004)* (July 2004).

[SG98] STAADT O. G., GROSS M. H. : Progressive tetrahedralizations. In *VIS '98 : Proceedings of the conference on Visualization '98* (1998), pp. 397–402.

[SS05] SONDERSHAUS R., STRASSER W. : View-dependent tetrahedral meshing and rendering. In *GRAPHITE '05 : Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (2005), pp. 23–30.

[SS06] SONDERSHAUS R., STRASSER W. : Segment-based tetrahedral meshing and rendering. In *GRAPHITE '06 : Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia* (2006), pp. 469–477.

[ST90] SHIRLEY P., TUCHMAN A. A. : Polygonal approximation to direct scalar volume rendering. In *Proceedings San Diego Workshop on Volume Visualization, Computer Graphics* (1990), vol. 24, pp. 63–70.

[THJ99] TROTTS I. J., HAMANN B., JOY K. I. : Simplification of tetrahedral meshes with error bounds. *IEEE Transactions on Visualization and Computer Graphics. Vol. 5*, Num. 3 (1999), 224–237.

[VFSG06] VIOLA I., FEIXAS M., SBERT M., GRÖLLER M. E. : Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics. Vol. 12*, Num. 5 (oct 2006), 933–940.

[Wil92] WILLIAMS P. L. : Visibility-ordering meshed polyhedra. *ACM Trans. Graph.. Vol. 11*, Num. 2 (1992), 103–126.

[WKME03] WEILER M., KRAUS M., MERZ M., ERTL T. : Hardware-based ray casting for tetrahedral meshes. In *Proc. Visualization '03* (2003), pp. 333–340.

[WMFC02] WYLIE B., MORELAND K., FISK L. A., CROSSNO P. : Tetrahedral projection using vertex shaders. In *VVS '02 : Proceedings of the 2002 IEEE symposium on Volume visualization and graphics* (2002), pp. 7–12.

[WZMK05] WANG L., ZHAO Y., MUELLER K., KAUFMAN A. : The magic volume lens : An interactive focus+context technique for volume rendering. *VIS '05 : Proceedings of the 16th IEEE Visualization 2005. Vol. 0* (2005), 47.

[XV96] XIA J., VARSHNEY A. : Dynamic view-dependent simplification for polygonal models. In *IEEE Visualization '96* (1996).