

Renforcer la convexité locale pour reconstruire une scène à partir d'un nuage de points épars

M. Lhuillier

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000, Clermont Ferrand, France
Maxime.Lhuillier@free.fr

Résumé

Plusieurs méthodes reconstruisent des surfaces à partir de nuages de points épars estimés à partir d'images. La plupart génèrent une triangulation de Delaunay 3D des points dont les tétraèdres sont étiquetés "vides" ou "plein" en utilisant l'information de visibilité et des contraintes sur la surface. Cependant, leurs erreurs les plus notables sont des tétraèdres incorrectement étiquetés vides. Nous présentons des corrections de ces erreurs basées sur la contrainte de convexité locale : dans le cas le plus simple, un tétraèdre vide est re-étiqueté plein s'il est suffisamment petit et si tous ses sommets sont dans des tétraèdres pleins. Ces corrections sont plus importantes dans la direction verticale pour tenir compte de l'anisotropie des scènes habituelles. Dans les expériences, les corrections améliorent les résultats de méthodes existantes de reconstruction de surface appliquées à des vidéos prises par une caméra 360 grand public.

Mots Clef

Modélisation d'environnement, primitives éparées, triangulation de Delaunay 3D, visibilité, remplissage de trous

Abstract

Several methods reconstruct surfaces from sparse point clouds that are estimated from images. Most of them build 3D Delaunay triangulation of the points whose tetrahedra are labeled "freespace" or "matter" using visibility information and surface constraints. However their most notable errors are falsely-labeled freespace tetrahedra. We present corrections of these errors based on local-convexity constraint : in the simplest case, a freespace tetrahedron of the Delaunay is relabeled matter if it is small enough and all its vertices are in matter tetrahedra. These corrections are more important in the vertical direction to take into account the anisotropy of usual scenes. In the experiments, the corrections improve the results of existing surface reconstruction methods applied to videos taken by a consumer 360 camera.

Keywords

Environment modeling, sparse features, 3D Delaunay triangulation, visibility, hole filling

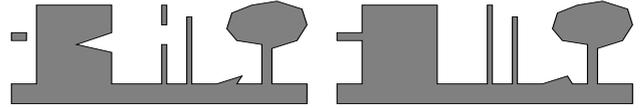


FIGURE 1 – Convexité locale pour reconstruire une scène. Le plein P est gris et le vide est blanc. A gauche : un balcon et un poteau sont non-connexes, un bâtiment a une fausse concavité. A droite : on veut (1) remplir la concavité du bâtiment et les trous du balcon et du poteau et (2) ne pas remplir la zone entre les deux poteaux ni celle entre le sol et le feuillage de l'arbre. Par exemple, on prend y et z dans différentes composantes du poteau pour le rendre connexe.

1 Introduction

La plupart des méthodes éparées génèrent une triangulation de Delaunay 3D des points reconstruits à partir de primitives éparées détectées dans les images, puis calculent un étiquetage des tétraèdres grâce à l'information de visibilité et des contraintes sur la surface. Chaque tétraèdre est étiqueté vide ou plein. La surface est l'ensemble des triangles séparant les tétraèdres vides et pleins. Cependant, des tétraèdres peuvent être incorrectement étiquetés vides à cause du bruit, de mauvais/d'un manque de points.

Ces erreurs sont les plus notables et apparaissent de plusieurs façons : des structures 3D (e.g. immeubles) avec de fausses concavités, des fines structures 1D (e.g. poteaux) non connexes, des fines structures 2D (e.g. panneaux de signalisation) avec des tunnels reliant les deux côtés. Cet article propose des méthodes pour supprimer de telles erreurs en utilisant la contrainte de convexité locale. On utilise l'abréviation LC pour dire à la fois "localement convexe" et "convexité locale" (choix évident en fonction du contexte). Soit P l'ensemble plein, i.e. l'union des tétraèdres pleins. On dit que P est LC en un point $x \in P$ si x a un voisinage $V \subseteq \mathbb{R}^3$ tel que $V \cap P$ est convexe, c'est à dire que l'on a $yz \subset V \cap P$ si $y, z \in V \cap P$. On renforce (anisotropiquement) la LC de P en favorisant la direction verticale à cause de la prédominance de structures verticales dans les scènes habituelles. La figure 1 montre ce que l'on veut : compléter l'ensemble des tétraèdres pleins à des endroits où il n'est pas LC. Nos méthodes sont utilisables en pré-traitement ou post-traitement de méthodes existantes de reconstruction de surface. Cet article est une traduction de [11].

2 Travaux précédants

Des a priori ou des contraintes sur la forme à reconstruire sont utiles non seulement pour tenir compte du bruit et des faux points, mais aussi pour traiter le cas de nuage de points épars. On peut s'inspirer de la perception humaine [12] (e.g. grouper des parties de forme par convexité) et de la théorie Gestalt [3] (e.g. grouper des points par alignement ou proximité). Une méthode [15] utilise la proximité et interpole un ensemble de points épars par une surface ayant les propriétés de connexité et 2-variété. Partant d'une triangulation de Delaunay 3D des points, ces points sont connectés par des triangles de Delaunay dont les arêtes sont les plus petites possibles. D'autres méthodes sont basées sur les α -shapes [6], une généralisation de la convexité qui remplace les demi-espaces hors de la forme par des boules sphériques de rayon égal à α . Cependant [15] n'est pas robuste aux mauvais points, et [6, 15] n'utilisent pas l'information de visibilité (segments reliant points et caméra).

On considère maintenant les méthodes de reconstruction de surfaces en vision par ordinateur, qui prennent en entrée des nuages de points épars estimés à partir des images (e.g. structure-from-motion). Elles partent aussi d'une triangulation de Delaunay 3D. Une méthode de coupe de graphe dans [19] estime une surface fermée comme un ensemble de triangles de Delaunay qui minimise une fonction de coût incluant un terme de visibilité et un terme de qualité de surface. Le premier essaye d'éviter les triangles traversés par les segments de visibilité. Le second essaye d'éviter les triangles qui sont faces de tétraèdres "trop plats". Dans [10], la surface est calculée comme le bord d'un ensemble de tétraèdres qui a la propriété de 2-variété. Cet ensemble croît dans les tétraèdres traversés par les segments de visibilité. L'expérience montre que nos corrections basées sur LC améliorent les résultats de ces deux méthodes. Ces corrections peuvent aussi être utilisées avec la plupart des méthodes de coupe de graphe ou avec contrainte de 2-variété. La définition originale de LC est dans [9] et utilise des boules sphériques comme voisinage. Ici les voisinages sont plus grands dans la direction verticale que dans les directions horizontales à cause de l'anisotropie des scènes habituelles. De plus, les travaux précédants (reconstruction de surface à partir de points calculés à partir d'images) n'utilisent pas LC. Donc on résume brièvement l'usage de LC et ses variantes (convexité, convexité faible, formes étoilées) dans d'autres problèmes. LC est exprimée avec des normales et vecteurs entre 2 patches de surface dans [13, 17]. Elle est utilisée pour la segmentation en sol/objets [13] ou en parties convexes [17]. Une méthode [1] segmente un nuage de points orientés en régions faiblement convexes. Elle est basée sur le pourcentage de paires de points telle que le segment qui les relie reste à l'intérieur du volume de la forme. La contrainte de forme étoilée généralise la convexité, et s'applique assez simplement dans une optimisation par coupe de graphe pour segmenter un objet dans une image [18], si l'utilisateur choisit le centre de la forme étoilée. Pour la convexité, c'est plus difficile [7].

D'autres travaux complètent aussi la forme pendant sa reconstruction mais ont besoin d'hypothèses fortes pour estimer des primitives (en comparaison avec LC). Dans [2], des plans sont estimés à partir d'un nuage de points plus dense qu'ici, puis une surface simplifiée est calculée à l'aide d'a priori : la prévalence des structures verticales et des intersections orthogonales. Une méthode [14] se focalise sur les scènes intérieures et construit une partition de l'espace 3D différente (à partir de points de scanner) en supposant que la surface est soit verticale soit horizontale. Dans [20], des surfaces généralisant les surfaces réglées sont estimées à partir d'un nuage épars de structure-from-motion en supposant que la scène est 100% architecturale.

3 Voisinages explicites

On introduit ici une méthode, appelée méthode 1, qui utilise explicitement la définition de LC dans la partie 1. Afin de définir des voisinages, on introduit une norme

$$\|\mathbf{x}\| = \max\left\{\frac{1}{s_v}|\mathbf{k}^\top \mathbf{x}|, \frac{1}{s_h}\sqrt{(\mathbf{i}^\top \mathbf{x})^2 + (\mathbf{j}^\top \mathbf{x})^2}\right\}, \quad (1)$$

avec $\mathbf{x} \in \mathbb{R}^3$, $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ est une base orthonormée de \mathbb{R}^3 telle que \mathbf{k} est vertical, s_h et s_v sont des facteurs d'échelles horizontale et verticale. La boule unité $\{\mathbf{x} \in \mathbb{R}^3, \|\mathbf{x}\| \leq 1\}$ est un cylindre d'axe vertical, de longueur $2s_v$ et de diamètre $2s_h$. On choisit s_h et s_v dans la partie 3.3.

Soit T la triangulation de Delaunay 3D des points donnés en entrée. C'est un ensemble de tétraèdres avec leurs faces (sommets, arêtes, triangles) tel que les tétraèdres sont étiquetés vides ou plein, par exemple par lancer de rayon ou par coupe de graphe. Il y a plusieurs façons d'appliquer LC. La méthode 1 fait ceci : trouver d'abord les arêtes de T qui devraient être incluses dans le plein, puis re-étiqueter à plein chaque tétraèdre vide de T dont toutes les arêtes devraient être dans le plein. Dans la première étape, des booléens $b(\mathbf{v}_i)$ et $b(\mathbf{v}_i\mathbf{v}_j)$ sont définis pour chaque sommet $\mathbf{v}_i \in T$ et chaque arête $\mathbf{v}_i\mathbf{v}_j \in T$. Dans la seconde, chaque tétraèdre $\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k\mathbf{v}_l \in T$ est re-étiqueté plein ssi (si et seulement si) les 10 booléens de ses 4 sommets et 6 arêtes sont vrais. Pour simplifier l'exposé, on commence par un b simplifié dans la partie 3.1, puis la partie 3.2 donne la définition de b utilisée par la méthode 1.

3.1 Utiliser seulement des sommets pleins

On utilise d'abord LC dans un voisinage V défini par une boule unité de $\|\cdot\|$ centrée en \mathbf{v}_i : si les sommets d'une arête $\mathbf{v}_i\mathbf{v}_j \in T$ sont à la fois dans le plein et dans V , alors $\mathbf{v}_i\mathbf{v}_j$ devrait être entièrement dans le plein aussi. On définit pour cela les booléens de la façon suivante : $b(\mathbf{v}_i) = \text{vrai}$ ssi \mathbf{v}_i est sommet d'un tétraèdre plein de T . (On dit alors que \mathbf{v}_i est un sommet plein.) De plus $b(\mathbf{v}_i\mathbf{v}_j) = \text{vrai}$ ssi (1) $\|\mathbf{v}_i - \mathbf{v}_j\| \leq 1$ ou (2) $\mathbf{v}_i\mathbf{v}_j$ est une arête d'un tétraèdre plein de T . Le cas 1 utilise LC et le cas 2 est évident.

3.2 Étendre l'ensemble des sommets pleins

Le principe est le suivant : initialiser b comme dans la partie 3.1, générer des tétraèdres qui ne sont pas dans

T mais qui devraient être pleins d’après leurs 10 booléens, puis tout sommet de T se trouvant dans un tel tétraèdre $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ devrait être plein. Les sommets \mathbf{a}_i sont dans T et vérifient $b(\mathbf{a}_i) = \text{vrai}$. Les arêtes $\mathbf{a}_i\mathbf{a}_j$ vérifient $b(\mathbf{a}_i\mathbf{a}_j) = \text{vrai}$. On note que $b(\mathbf{a}_i\mathbf{a}_j)$ n’est pas encore défini si $\mathbf{a}_i\mathbf{a}_j \notin T$. Dans ce cas, on prend $b(\mathbf{a}_i\mathbf{a}_j) = \text{vrai}$ ssi $\|\mathbf{a}_i - \mathbf{a}_j\| \leq 1$. Puis on effectue $b(\mathbf{v}) = \text{vrai}$ si le sommet $\mathbf{v} \in T$ est dans l’enveloppe convexe de $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$ et \mathbf{a}_3 .

En pratique, on considère chaque sommet $\mathbf{v} \in T$ tel que $b(\mathbf{v}) = \text{faux}$ et on essaye seulement des tétraèdres dont les sommets sont dans le voisinage immédiat de \mathbf{v} : on énumère chaque tétraèdre $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ tel que $\mathbf{v}\mathbf{a}_i \in T$ et $b(\mathbf{a}_i) = \text{vrai}$ et $\|\mathbf{a}_i - \mathbf{v}\| \leq 1$ avec $0 \leq i \leq 3$. Si les 6 booléens $b(\mathbf{a}_i\mathbf{a}_j) = \text{vrai}$ et $\mathbf{v} \in \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$, on effectue $b(\mathbf{v}) = \text{vrai}$.

3.3 Choix des facteurs d’échelles

Il y a un compromis à faire pour choisir s_h et s_v : la méthode 1 a un effet négligeable s’ils sont trop petits, des détails sont perdus s’ils sont trop grands. On estime d’abord un niveau de détail ϵ attendu pour la forme reconstruite, puis on le multiplie par des nombres proches de l’unité pour obtenir s_v et s_h . On prend ici $s_h = 2\epsilon$ et $s_v = 3\epsilon$ avec ϵ égal à la médiane des distances horizontales (aussi utilisées dans l’Eq. 1) des arêtes des tétraèdres pleins donnés en entrée.

4 Voisinages implicites

On introduit ici deux méthodes dont le principe est simple : générer des ensembles de tétraèdres vides inclus dans la triangulation de Delaunay 3D T , puis re-étiqueter à plein tous les tétraèdres d’un ensemble S s’il y a suffisamment de tétraèdres pleins qui entourent S . La partie 4.1 présente des critères pour évaluer l’entourage de S et expliquer l’usage implicite de LC. Les deux méthodes sont ensuite décrites dans les parties 4.2 et 4.3.

On applique des contraintes sur ces ensembles de façon à ce qu’ils ne soient ni trop grands ni trop nombreux. Tous les tétraèdres dans S partagent un sommet commun et S est *fortement connexe* : il y a un chemin dans le graphe d’adjacence des tétraèdres de S qui passe par tout les tétraèdres de S (Rappel : deux tétraèdres sont adjacents s’ils ont une face triangulaire commune.) Ainsi S n’a pas plus de 27 tétraèdres en moyenne [4] et les composantes fortement connexes sont générées séparément.

4.1 Critères d’entourage et convexité locale

On a besoin d’évaluer la zone pleine qui entoure S . Soit ∂S le bord de S , i.e. l’ensemble de tout triangle face d’exactly un tétraèdre de S . Soit $\partial_p S$ les triangles dans ∂S qui sont faces de tétraèdres pleins (donc dans $T \setminus S$). Soit $c(S)$ le rapport entre l’aire de $\partial_p S$ et l’aire de ∂S . L’aire de ∂S est la somme pour tout triangle $\mathbf{abc} \in \partial S$ de la norme euclidienne de $\mathbf{s} = \frac{1}{2}(\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a})$. On a $0 \leq c \leq 1$. Si $c(S)$ est proche de 1, S est bien entouré de tétraèdres pleins. (S forme une cavité vide dans le plein si $c(S) = 1$.)

On définit un autre critère c_k qui prend en compte l’anisotropie de la scène (ce que c ne fait pas). L’aire du triangle \mathbf{abc} est un cas particulier de $\sqrt{\mathbf{s}^T \mathbf{Q} \mathbf{s}}$ avec $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ une matrice symétrique positive. Si $\mathbf{Q} = \mathbf{k}\mathbf{k}^T$, $\sqrt{\mathbf{s}^T \mathbf{Q} \mathbf{s}}$ est l’aire de \mathbf{abc} projeté sur un plan horizontal. On définit $c_k(S)$ en remplaçant toutes les normes euclidiennes de \mathbf{s} dans l’expression de $c(S)$ par $\sqrt{\mathbf{s}^T \mathbf{k}\mathbf{k}^T \mathbf{s}}$. On a $0 \leq c_k \leq 1$. Si $c_k(S)$ est proche de 1, S est bien entouré de tétraèdres pleins au dessus et en dessous (relativement à la verticale \mathbf{k}).

Dans les deux cas (si $c(S)$ ou $c_k(S)$ est proche de 1), on peut utiliser la LC du plein dans un voisinage (non explicite) qui contient S : la plupart des tétraèdres dans S sont dans l’enveloppe convexe de $\partial_p S$ et donc S devrait être re-étiqueté plein. Notons que ces tétraèdres n’ont pas de limite sur leur taille, contrairement à ceux de la méthode 1.

4.2 Méthode 2

La méthode 2 étend une opération de suppression de pics (“Peak Removal” dans [10]) au cas où la surface séparant vide et plein n’est pas une 2-variété. Un ensemble S est une composante fortement connexe des tétraèdres vides de T qui partagent un sommet \mathbf{v} . De plus, \mathbf{v} est un “pic de S ”, i.e. l’angle solide de S au sommet \mathbf{v} est petit. (Cet angle est la somme pour tout tétraèdre $\mathbf{vabc} \in S$ de l’angle solide de \mathbf{vabc} au sommet \mathbf{v} .) Une différence avec [10] est que l’on peut avoir plusieurs ensembles S pour un même \mathbf{v} si la surface n’est pas une 2-variété en \mathbf{v} .

Le re-étiquetage de tétraèdres partageant tout sommet $\mathbf{v} \in T$ se fait ainsi. D’abord on calcule le(s) ensemble(s) S associés à \mathbf{v} par un parcours dans le graphe d’adjacence des tétraèdres incluant \mathbf{v} . Puis tout ensemble S est re-étiqueté plein si son angle solide au sommet \mathbf{v} est inférieur à un seuil et si $c(S)$ est supérieur à un autre seuil. On utilise $\pi/2$ (comme dans [10]) et 0.5 dans les expériences, respectivement. La valeur 0.5 implique que l’aire de la surface séparant vide et plein décroît.

4.3 Méthode 3

Contrairement à la méthode 2, la méthode 3 tient compte de l’anisotropie de la scène et essaye plus d’ensembles S . De plus, on utilise ici une mesure de confiance pour un tétraèdre vide : le nombre de segments point-caméra traversant le tétraèdre. D’autres mesures de confiance sont possibles mais ce n’est pas le sujet ici. Pour faciliter l’explication, on décrit d’abord une version simplifiée de la méthode 3 sans connexité forte, puis la version réellement utilisée avec connexité forte.

Sans connexité forte. Les tétraèdres de S partagent un même sommet et ont les mesures de confiance les plus faibles. Il y a deux raisons pour ce choix. D’abord on énumère facilement tous les ensembles : pour chaque sommet $\mathbf{v} \in T$, mettre dans une table les tétraèdres vides de T incluant \mathbf{v} dans l’ordre croissant de leur mesure de confiance, ajouter dans S le i -ième élément de la table et incrémenter i pour parcourir la table. Ensuite on veut re-étiqueter en priorité les tétraèdres qui sont les plus probablement

faussement-étiquetés vides.

On détaille maintenant comment re-étiqueter des tétraèdres partageant un sommet $\mathbf{v} \in T$. Soit $S_i = S_{i-1} \cup \{\Delta_i\}$ avec Δ_i le i -ième tétraèdre de la table ordonnée et $S_0 = \emptyset$. Soit $i_* = \operatorname{argmax}_i c_{\mathbf{k}}(S_i)$. Si $c_{\mathbf{k}}(S_{i_*})$ est supérieur à un seuil c_0 , les tétraèdres dans S_{i_*} sont re-étiquetés pleins. En pratique, ce processus est fait pour tout les sommets trois fois avec $c_0 = 0.6$.

Avec connexité forte. Dans ce paragraphe, on utilise le mot “composante” pour dire “composante fortement connexe”. La différence entre la méthode 3 et sa version simplifiée dans le paragraphe précédent est la suivante : on calcule $i_* = \operatorname{argmax}_i c_{\mathbf{k}}(S'_i)$ avec S'_i la composante de S_i qui inclue Δ_i . En faisant cela, toutes les composantes des S_i sont générées séparément. On explique maintenant comment calculer S'_i : grâce au calcul incrémental des composantes de S_i à partir de celles de S_{i-1} . Soit $\Delta' \in T$ l'un des 4 tétraèdres adjacents à Δ_i . Si $\Delta' \in S_{i-1}$, Δ' et Δ_i sont dans la même composante. Donc S'_i est l'union de l'ensemble $\{\Delta_i\}$ et de toute composante S' de S_{i-1} incluant un tel Δ' . De plus, les composantes de S_i sont S'_i et celles de S_{i-1} à l'exception de(s) ensemble(s) S' . On implémente cela efficacement avec la structure union-find [16].

5 Utilisation des re-étiquetages

On combine les re-étiquetages pour améliorer [19, 8, 10].

5.1 Combiner des re-étiquetages

On abrège les méthodes 1, 2 et 3 (respectivement décrites dans les parties 3, 4.2 et 4.3) en écrivant simplement 1, 2 et 3. En pratique, il est intéressant d'utiliser 2 de la façon suivante : appliquer 2, puis appliquer 1 ou 3. En effet, 2 est plus rapide que 1 et 3 et fait une partie du travail de 1 et 3. Jusqu'ici, des tétraèdres faussement étiquetés vides sont corrigés mais rien n'est fait pour les tétraèdres faussement étiquetés pleins. Ces derniers incluent notamment des tétraèdres qui sont trop pointus/fins pour être intersectés par des segments point-caméra. On corrige cela en échangeant les rôles de vide et plein dans 2 : on re-étiquete à vide des tétraèdres pleins (au lieu de re-étiqueter à plein des tétraèdres vides) formant des angles solides faibles si cela fait décroître l'aire de la surface séparant vide et plein. On note cela la méthode $\tilde{2}$, et on utilise ici une limite supérieure d'angle solide plus exigeante ($\pi/100$) pour limiter l'élimination des fines structures.

On utilise les notations suivantes dans les expériences : 2+1 signifie que l'on utilise d'abord 2, puis 1, et enfin $\tilde{2}$. De même, 2+3 signifie que l'on utilise d'abord 2, puis 3, et enfin $\tilde{2}$. On utilise $\tilde{2}$ à la fin car 1 et 3 peuvent créer quelques tétraèdres faussement étiquetés pleins. Une autre combinaison 2+1+3 est possible mais ne figure dans l'article pour une raison de place. (On peut définir les tétraèdres pleins de 2+1+3 par l'union des tétraèdres pleins de 2+1 et 2+3.) On écrit aussi 2+ $\tilde{2}$ si seulement 2 puis $\tilde{2}$ sont utilisés. Si 2+1 ou 2+3 est utilisé pour améliorer une méthode de reconstruction de surface X , on écrit $X(2+1)$ ou $X(2+3)$, e.g.

$X=M$ dans la partie 5.2 et $X=G_i$ dans la partie 5.3.

5.2 Pré-traitement d'une méthode M

M est la méthode calculant une 2-variété nommée $M3$ dans [10] ($M=$ Manifold). Elle prend en entrée T et un étiquetage obtenu par lancer de rayons : un tétraèdre est vide s'il est traversé par un segment point-caméra. Elle calcule en sortie un autre étiquetage vide-plein (nommé extérieur-intérieur dans [10]) qui est une amélioration de celui donné en entrée, telle que la surface séparant vide et plein est une 2-variété. Les re-étiquetages peuvent être en pré-traitement ou en post-traitement de M . Comme ils ne conservent pas la propriété de 2-variété dans le second cas, on choisit le premier cas.

5.3 Post-traitement des méthodes G_1 et G_2

G_1 est la méthode de coupe de graphe dans [19]. D'abord elle génère T et son graphe d'adjacence dont les poids aux arêtes sont calculés à partir des segments point-caméra et de la géométrie des tétraèdres (avec $\lambda_{qual} = 1$ et $\alpha_{vis} = 1$). Puis l'étiquetage final des tétraèdres (vide ou plein) est calculé par *st*-coupe minimale du graphe.

Les re-étiquetages peuvent être en pré-traitement ou en post-traitement de G_1 . Le second cas est simple. Dans le premier cas, ils sont appliqués sur un étiquetage initial (obtenu par lancer de rayon dans T) et on additionne une valeur à des t -poids du graphe si on re-étiquete à plein les tétraèdres correspondants. Ceci nécessite un paramètre supplémentaire et modifie la fonction de coût qui est ensuite minimisée par coupe de graphe. On laisse le premier cas en développement futur et on ne considère ici que le second.

G_2 est la méthode de coupe de graphe dans [8], qui améliore les fines structures par rapport à G_1 . (On utilise $\sigma = 0$, $\delta = 0.5$ et $\beta/\alpha_{vis} = 32$.) De façon similaire, on effectue les re-étiquetages en post-traitement de G_2 .

6 Expérimentations

La partie 6.1 expérimente les corrections (i.e. re-étiquetages) seules et la partie 6.2 montre les améliorations qu'elles produisent lorsqu'elles sont utilisées en post-traitement de méthodes de coupe de graphe ou en pré-traitement d'une méthode de 2-variété. Ces choix et les notations sont détaillés dans la partie 5. Toutes les méthodes partent de la même triangulation de Delaunay T et finissent par le même lissage de surface (Laplace). Le nuage de points épars est calculé comme dans la partie 7.1 de [10] à partir de deux vidéos 2496×2496 à 30Hz prises par une caméra Garmin Virb 360. Elle est fixée au dessus d'une voiture à l'aide d'un petit mat. La trajectoire est longue de 6.7km dans une scène urbaine. Les 5.5M de sommets de T sont reconstruits à partir de 6.5k images clefs, sélectionnées dans la double vidéo par structure-from-motion (avec les approximations centrales et global shutter). La figure 2 montre la caméra 360, une image clef, et une vue de dessous des sommets de T .



FIGURE 2 – Une image clef, la caméra Garmin Virb 360, et une vue de dessous des sommets de T .



FIGURE 3 – Vues à grande échelle de nos corrections. De gauche à droite : sommets de T , étiquetage initial par lancer de rayon, résultats des méthodes de corrections $2+2$, $2+1$ et $2+3$. Le plein est blanc et le vide est noir.

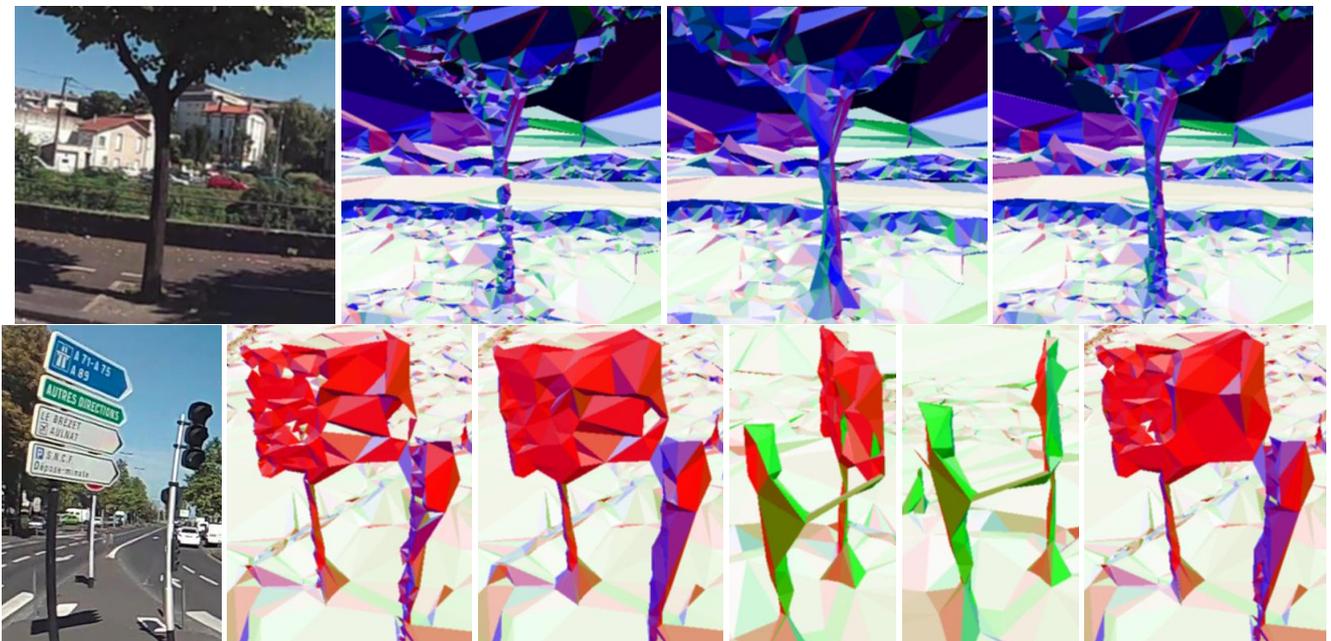


FIGURE 4 – Vues à petite échelle de nos corrections. De gauche à droite : partie d'image en entrée, résultats des méthodes de corrections $2+2$, $2+1$ et $2+3$ (à regarder en couleur et en zoomant). Les triangles verticaux sont rouges-verts-bleus et les triangles horizontaux sont blancs/noirs. On note que $2+1$ a trois vues successives sur la ligne du bas pour montrer l'épaisseur du panneau avec un faible nombre de sommets.

Méthode	lancer rayon	2+2	2+1	2+3
nb compos. (β_0)	4.8k	5.5k	1.5k	5.1k
nb tunnels (β_1)	185k	105k	18k	31k
sommets singul.	31%	9.8%	5.0%	4.4%
tétraèdre vide	57%	51%	38%	49%
temps de calcul	102s	22s	38s	62s

TABLE 1 – Changements de topologie dues aux corrections, pourcentage de tétraèdres vides dans T , temps de calcul (sur un coeur de PC portable standard).

6.1 Corrections seules

On compare l'étiquetage initial par lancer de rayon et ses re-étiquetages par nos méthodes 2+2, 2+1 et 2+3. Cette comparaison est intéressante car elle permet de comprendre les pré-traitements de la méthode M (qui sont ces re-étiquetages). De plus, elle montre la performance des re-étiquetages dans un contexte plus difficile que celui d'un post-traitement des méthodes G_x . (L'étiquetage initial est de plus mauvaise qualité que celui calculé par G_x .)

La figure 3 montre des différences d'étiquetages au voisinage d'un carrefour à 5 voies en projetant les tétraèdres vides (en noir) sur un plan horizontal. On voit que 2+2 supprime les tétraèdres vides les plus grands qui sont inconsistants avec la géométrie des couloirs urbains. (On note que la caméra s'est déplacée dans les 5 rues principales). De plus 1 et 3 (impliquées dans 2+1 et 2+3) suppriment d'autres tétraèdres vides. La figure 4 montre des différences à une échelle plus petite : 2+1 et 2+3 complètent des fines structures comme un tronc et un panneau de signalisation par rapport à 2+2.

On voit que les corrections changent la topologie : le tronc devient connexe, des tunnels (i.e. trous) dans le panneau sont bouchés. On évalue donc ces changements. Soient β_0 et β_1 les nombres de composantes connexes et de tunnels du plein, respectivement. (Ils se calculent comme les nombres de Betti du complexe simplicial engendré par les tétraèdres pleins [5].) La table 1 montre que nos corrections ont des avantages : elles réduisent beaucoup β_1 ainsi que le pourcentage de sommets singuliers. On rappelle qu'un *sommet singulier* est un point de la surface où la surface n'est pas une 2-variété. D'autres expériences (hors article), montrent que β_0 décroît toujours si 2 est inutilisée, mais cela génère de grands et faux tétraèdres pleins et la réduction de β_1 est plus faible.

6.2 Corrections en pré- ou post-traitements

La figure 5 compare les méthodes M, G_1 , G_2 et leurs versions corrigées sur un bâtiment de notre scène. La normale de la surface sur la face faiblement texturée de ce bâtiment est toujours bruitée sans nos corrections. La correction 2+3 améliore la normale pour toutes les méthodes, qui devrait être constante sur cette face. En comparaison, l'amélioration de normale par la correction 2+1 est négligeable dans tous les cas. (La figure ne montre que M(2+1).) Il y a deux

raisons. D'abord 2+1 utilise une limite supérieure sur la longueur d'arête des tétraèdres contrairement à 2+3. Ensuite la longueur d'arête est grande ici car le bâtiment est dans l'arrière plan et a un faible nombre de points (due à la texture faible). On note aussi que 2+1 et 2+3 suppriment deux erreurs de M.

Dans la figure 6, on se concentre sur les arêtes vives d'une façade de notre scène. Ici une arête vive est l'intersection entre deux régions planes adjacentes et orthogonales de la scène. Cela arrive fréquemment dans les scènes urbaines : on observe surtout 2×2 sortes d'arêtes vives : horizontale ou verticale, convexe ou concave. On dit que l'arête d'une surface est convexe (respectivement concave) si son voisinage du côté plein est convexe (respectivement, concave). La figure 6 montre que 2+1 et 2+3 améliorent la reconstruction d'arêtes vives verticales si elles sont convexes. Si les arêtes vives sont concaves, elles sont lissées par 2+1 qui tend à remplir la concavité du côté vide. Ce lissage peut aussi arriver avec 2+3, mais en plus faible. Ces commentaires sont aussi valables pour M, G_1 et G_2 .

La figure 7 compare M, G_1 , G_2 et leur versions corrigées sur de fines structures 1D de notre scène. Dans la ligne de dessus, le résultat de M(2+1) est meilleur que celui de M et M(2+3) dans la mesure où M(2+1) génère les fines structures 1D les plus complètes. Dans la ligne de dessous, $G_1(2+1)$ et $G_1(2+3)$ complète les fines structures 1D de G_1 . Cependant, on voit que $G_1(2+1)$ produit un résultat biaisé : l'épaisseur de deux fines structures 1D proches du poteau central est plus large qu'en vrai, d'après l'image de la scène montrée sur la ligne de dessus.

On donne aussi des évaluations quantitatives de toutes les méthodes en les appliquant sur un jeu de données de synthèse dans [10] (une scène urbaine plane par morceaux avec des textures réelles). Les sommets de T et les segments point-caméra sont estimés par structure-from-motion appliqués à des vidéos de synthèses prises par une caméra 360, dont la trajectoire fait une boucle de 621m de long (Fig. 8). La table 2 montre que nos corrections améliorent des méthodes de reconstruction de surface existantes à la fois pour des critères géométriques et topologiques. L'erreur géométrique et le nombre de triangles décroissent grâce à nos corrections. De plus, de nombreux faux tunnels sont bouchés (le bruit topologique décroît), et le pourcentage de sommets singuliers de la surface décroît la plupart du temps pour la coupe de graphe.

Enfin on fait des expériences similaires à celles de la table 2 pour notre scène réelle, à la différence qu'il n'y a pas de vérité terrain ici. Nos corrections améliorent la topologie des surfaces des méthodes G_i : le pourcentage de sommets singuliers dans la surface décroissent toujours (de 12% à 67%) et le nombre de tunnels β_1 est divisé par 3-5. Le β_1 de la méthode M décroît d'environ 7%. Le nombre de triangles décroît d'environ 30% avec la correction 2+1 ou de 9-15% avec 2+3. Le nombre de composantes connexes β_0 décroît de 73% (2+1) ou s'accroît de 25% (2+3).

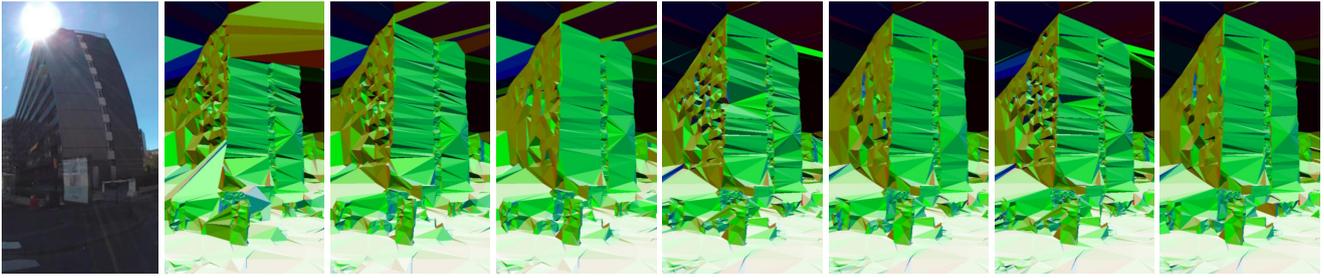


FIGURE 5 – Nos corrections de méthodes existantes appliquées sur un bâtiment (à regarder en couleur et en zoomant). De gauche à droite : partie d’image en entrée, résultats des méthodes M, M(2+1), M(2+3), G_1 , $G_1(2+3)$, G_2 , $G_2(2+3)$.

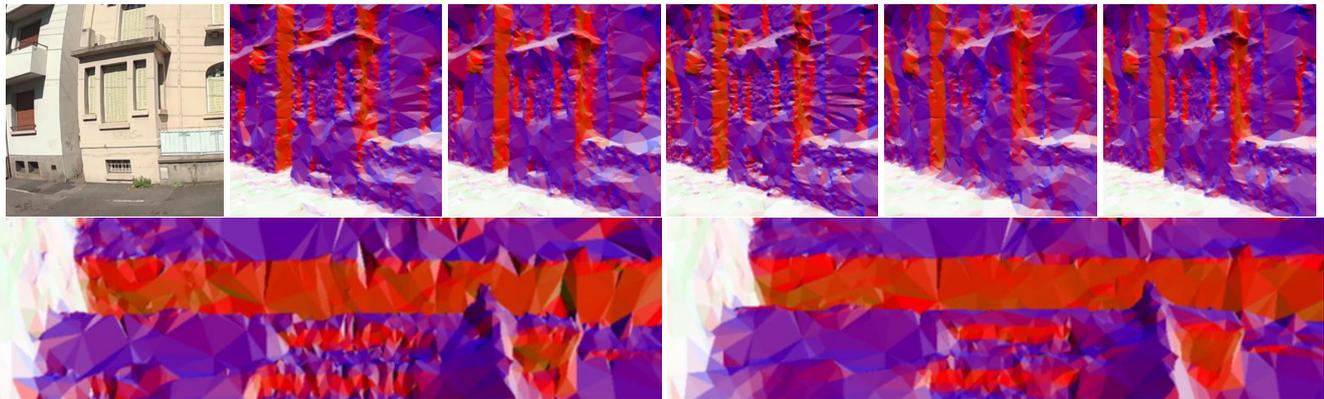


FIGURE 6 – Nos corrections de méthodes existantes appliquées sur une façade (à regarder en couleur et en zoomant). En haut (de gauche à droite) : partie d’image en entrée, M, M(2+3), G_1 , $G_1(2+1)$, $G_1(2+3)$. En bas : G_2 et $G_2(2+3)$.

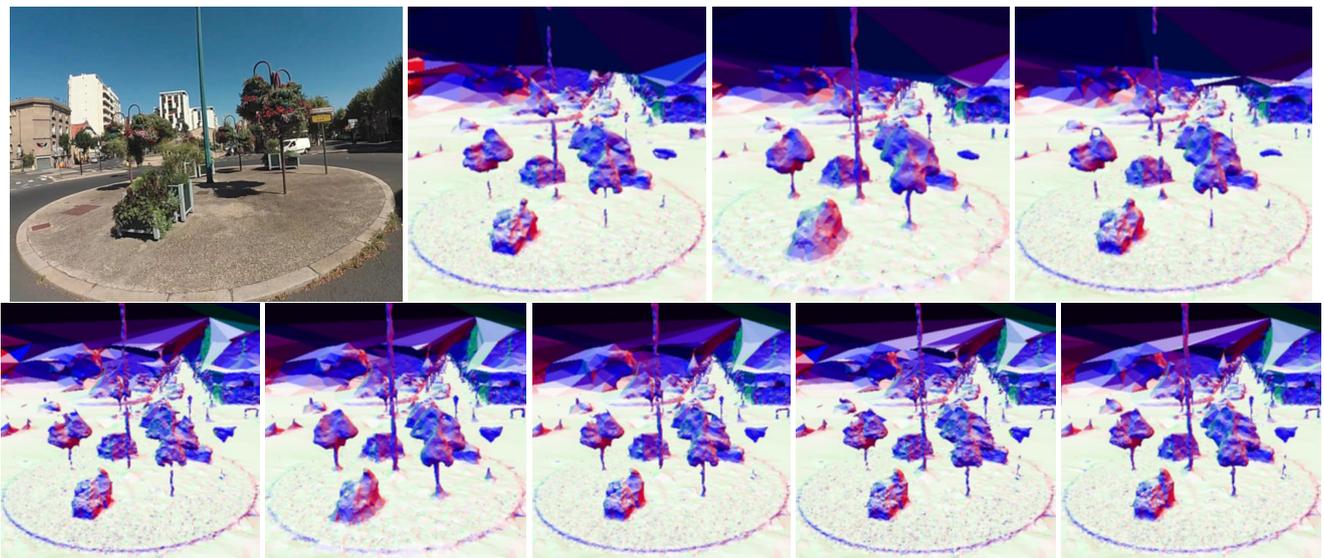


FIGURE 7 – Nos corrections de méthodes existantes appliquées sur un rond-point avec de la végétation urbaine et un poteau (à regarder en couleur et en zoomant). En haut (de gauche à droite) : partie d’image en entrée, M, M(2+1), M(2+3). En bas : G_1 , $G_1(2+1)$, $G_1(2+3)$, G_2 et $G_2(2+3)$.

Méthode	G_1	$G_1(2+1)$	$G_1(2+3)$	G_2	$G_2(2+1)$	$G_2(2+3)$	M	M(2+1)	M(2+3)
fractile 80% erreur (cm)	28	26	26	28	26	26	24	22	22
fractile 90% erreur (cm)	94	84	88	94	82	88	68	58	56
nb triangles	1.26M	0.86M	1.01M	1.27M	0.82M	0.98M	1.09M	0.72M	0.922M
nb comp. connexes (β_0)	46	28	45	27	18	29	2	3	4
nb tunnels (β_1)	834	187	205	1737	247	351	32	17	18
sommets singuliers	0.98%	1.24%	0.27%	1.58%	1.18%	0.34%	0%	0%	0%

TABLE 2 – Améliorations quantitatives données par nos corrections sur des méthodes existantes de reconstruction de surfaces appliquées à un jeu de données de synthèse (scène urbaine plane par morceaux). Il y a des améliorations de précision (des fractiles d’erreurs décroissent) et de topologie (le nombre de faux tunnels décroît), ainsi qu’une simplification de surface modérée. La scène vérité terrain a $\beta_1 = 3$ tunnels et $\beta_0 = 1$ composante connexe.

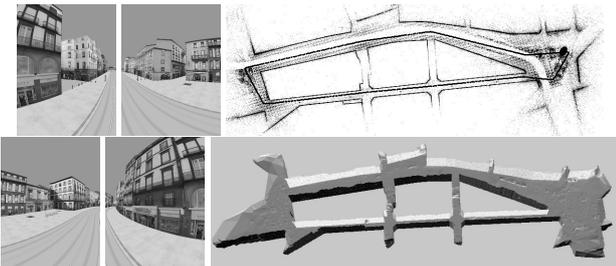


FIGURE 8 – Jeu de données de synthèse (scène urbaine plane par morceaux). A gauche : images autour d’un point de vue. A droite : vues de dessus du structure-from-motion et de la surface calculée par la méthode M.

7 Conclusion

On utilise la contrainte de convexité locale pour améliorer des méthodes existantes de reconstruction de surface basées sur la triangulation de Delaunay 3D. Cette contrainte est renforcée par nos corrections en post-traitement ou en pré-traitement, selon le type de méthode. Comme le nuage de points en entrée est éparé, on ne cherche pas ici à estimer des primitives comme des plans mais on énumère et sélectionne des (ensembles de) tétraèdres à re-étiqueter, principalement de vide à plein. L’expérience montre des améliorations sur une scène urbaine acquise par imagerie terrestre 360 : suppression de concavités et tunnels des tétraèdres pleins, amélioration des arêtes vives si elles sont convexes, complétion de fines structures verticales, réduction de l’erreur géométrique et de la complexité de la scène. Des travaux futurs seraient utiles pour éviter le sur-lissage d’arêtes vives concaves, varier la direction privilégiée, et étudier d’autres moyens de renforcer la convexité locale.

Références

- [1] S. Asafi, A. Goren, and D. Cohen-Or. Weak convex decomposition by lines-of-sight. *Eurographic symposium on geometry processing*, 32(5), 2013.
- [2] A.L. Chauve, P. Labatut, and J.P. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *CVPR*, 2010.
- [3] A. Desolneux, L. Moisan, and J.M. Morel. *Gestalt theory and computer vision*. Springer, Dordrecht, 2000. In : Carsetti A. (eds) Seeing, Thinking and Knowing. Theory and Decision Library A : vol 38.
- [4] O. Devillers. On deletion in delaunay triangulations. *International Journal of Computational Geometry and Applications*, 12, 2012.
- [5] T.K. Dey and S. Guha. Computing homology groups of simplicial complexes in \mathbb{R}^3 . *Journal of the ACM*, 45(2), 1998.
- [6] H. Edelsbrunner and E.P. Mucke. Three-dimensional alpha-shape. *ACM Transactions on Graphics*, 13(1), 1994.
- [7] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis. Convexity shape prior for segmentation. In *ECCV*, 2014.
- [8] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR*, 2011.
- [9] Y. Karshon and C. Bjorndahl. Revisiting tietzenakajima - local and global convexity for maps. *Canadian Journal of Mathematics*, 62(5), 2007.
- [10] M. Lhuillier. Surface reconstruction from a sparse point cloud by enforcing visibility consistency and topology constraints. *CVIU*, 175, 2018.
- [11] M. Lhuillier. Local-convexity reinforcement for scene reconstruction from sparse point clouds. In *International Conference on 3D Immersion*, 2019.
- [12] Z. Liu, D.W. Jacobs, and R. Basri. The role of convexity in perceptual completion : beyond good continuation. *Vision Research*, 3, 1999.
- [13] F. Moosmann, O. Pink, and C. Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. In *Intelligent Vehicle Symposium*, 2009.
- [14] S. Oesau, F. Lafarge, and P. Alliez. Indoor scene reconstruction using feature sensitive primitive extrac-

tion and graph-cut. *ISPRS journal of photogrammetry and remote sensing*, 90, 2014.

- [15] S. Ohrhallinger, S. Mundur, and M. Wimmer. Minimizing edge length to connect sparsely sampled unstructured point sets. *Computer and Graphics*, 37(6), 2013.
- [16] R. Sedgewick and K. Wayne. *Algorithms (fourth edition deluxe)*. Addison-Wesley, 2016.
- [17] S.C. Stein, F. Forgotter, M. Schoeler, J. Papon, and T. Kulvicius. Convexity based object partitioning for robot applications. In *ICRA*, 2014.
- [18] O. Veksler. Star shape prior for graph-cut image segmentation. In *ECCV*, 2008.
- [19] H. Vu, P. Labatut, J. Pons, and R. Keriven. High accuracy and visibility consistent dense multiview stereo. *IEEE PAMI*, 34, 2012.
- [20] C. Wu, S. Agarwal, B. Curless, and S.M. Seitz. Schematic surface reconstruction. In *CVPR*, 2012.