HAL
open science

# Estimation incrémentale de surface à partir d'un nuage de point épars reconstruit à partir d'images omnidirectionnelles

Vadim Litvinov, Maxime Lhuillier

## To cite this version:

Vadim Litvinov, Maxime Lhuillier. Estimation incrémentale de surface à partir d'un nuage de point épars reconstruit à partir d'images omnidirectionnelles. Reconnaissance de formes et intelligence artificielle (RFIA) 2014, Jun 2014, France. hal-00988591

**HAL Id: hal-00988591**

**https://hal.science/hal-00988591v1**

Submitted on 8 May 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Estimation incrémentale de surface à partir d'un nuage de point épars reconstruit à partir d'images omnidirectionnelles

Vadim Litvinov Maxime Lhuillier

Institut Pascal, UMR 6602 CNRS/UBP/IFMA, Campus des Cézeaux, 63171 Aubière Cedex
http://maxime.lhuillier.free.fr

## Résumé

*Cet article introduit une méthode incrémentale de reconstruction de surface (une 2-variété). Elle prend en entrée un nuage de points 3D épars reconstruit à partir d'une séquence d'images, par opposition aux algorithmes habituels denses. De plus, notre méthode est incrémentale : la surface est mise à jour à chaque nouvelle pose de caméra donnée en entrée, et la mise à jour a lieu dans un voisinage restreint de la nouvelle pose. Comparée aux autres méthodes de reconstruction de surface, notre méthode a l'avantage de cumuler toutes ces propriétés (nuage épars en entrée, 2-variété en sortie, calcul incrémental et local). La qualité et le temps d'exécution sont évalués sur une séquence d'images omnidirectionnelles (longue de* $2.5$ *km) prise en environnement urbain, et la méthode est quantitativement évaluée sur une séquence urbaine synthétique.*

## Mots Clef

2-variété, reconstruction de surface, Delaunay 3D, sculpture, estimation de géométrie à partir d'images

## Abstract

*This paper introduces an incremental (2-manifold) surface reconstruction method. Its input is a sparse 3D point cloud generated by a Structure-from-Motion algorithm, as opposed to the more common dense algorithms. Furthermore, our method is incremental : the surface is updated for every new camera pose given as input, and the update occurs in a small neighborhood of the new camera pose. Compared to the other surface reconstruction methods, our method has the advantage to have all these properties at the same time (sparse, 2-manifold, incremental, local). Quality and execution time are evaluated on a* $2.5$ *km. long omnidirectional sequence taken in an urban environment. The method is quantitatively evaluated on a synthetic urban scene.*

## Keywords

2-manifold, surface reconstruction, 3D Delaunay triangulation, sculpting, structure-from-motion

# 1 Introduction

La majorité des méthodes qui reconstruisent la surface d'un environnement à partir d'images ont une information dense en entrée (nuage de points dense ou quasiment tous les pixels des images). Par opposition à ces méthodes denses, l'estimation directe d'une surface à partir du nuage de points épars estimé par Structure-from-Motion (SfM) est sous-explorée. En dépis du faible nombre de points, une telle méthode éparse a des avantages. D'abord, les points sont de bonne qualité (robustesse, précision) grâce à la sélection des points d'intérêts et le raffinement par ajustement de faisceaux. Ensuite, le nuage de points épars permet d'obtenir une surface simplifiée avec une complexité de calcul faible, les applications potentielles étant la localisation de robots, la (pré)visualisation avec du matériel peu performant, l'initialisation d'une méthode de stéréo dense.
Dans cet article, on introduit une méthode incrémentale qui reconstruit une surface à partir d'un nuage de points épars généré par SfM. La méthode est incrémentale car la surface est localement modifiée à chaque nouvelle pose fournie par SfM. Ceci est intéressant pour les applications qui ont besoin d'une surface pendant la lecture d'une vidéo. De plus, la surface est une 2-variété, c'est à dire une liste de triangles dans l'espace tels que tout point de la surface a un voisinage homéomorphe à un disque. Cette propriété est nécéssaire pour définir la normale et la courbure de la surface [4], et est donc utilisée par de nombreux algorithmes comme le raffinement de surface impliquant une régularisation (lissage [8], stéréo dense [9], ...) et d'autres [4, 11].
La plupart des méthodes *éparses* sont basées sur la sculpture dans une triangulation de Delaunay 3D. La sec. 2 présente et discute ces méthodes, puis notre méthode est décrite dans les sec. 3 (pré-requis) et 4 (algorithme). Enfin, la sec. 5 donne les expériences et nous concluons en sec. 6.

# 2 Méthodes de sculpture

Soit $P$ un ensemble de points 3D échantillonnés sur une surface inconnue. La triangulation de Delaunay 3D $T$ de $P$ est une liste de tétrahèdres telle que (1) les tétrahèdres partitionnent l'enveloppe convexe de $P$, (2) l'ensemble de leurs sommets est $P$, (3) la sphère circonscrite de tout tétrahèdre ne contient pas de sommet dans son intérieur. Dans cet article, un sommet/arête/triangle est une face d'un tétrahèdre de $T$, et on note $|L|$ pour l'union des triangles (ou tétrahèdres) dans la liste $L$.
Dans une méthode de sculpture, $V$ est une liste de tétra-

hèdres (de $T$) qui représente l'objet reconstruit et dont le volume est $|V|$. Le bord $\partial V$ est la liste de triangles qui sont inclus dans exactement un seul tétrahèdre de $V$. La surface $|\partial V|$ que l'on veut calculer est une 2-variété. Dans le travail ancien [3] où $P$ n'a pas de point erroné, $P \subseteq |V|$. Cette méthode initialise $V = T$, puis sélectionne et retire un tétrahèdre $\Delta$ de $V$ tant que les conditions suivantes sont satisfaites : $\Delta$ a un triangle dans $\partial V$, $|\partial(V \setminus \{\Delta\})|$ est une 2-variété, et $P \subset |V \setminus \{\Delta\}|$. Un test efficace vérifie que $|\partial(V \setminus \{\Delta\})|$ est une 2-variété. Cependant, le genre de $|\partial V|$ est toujours zéro, c'est à dire $|\partial V|$ est toujours homéomorphe à une 2-sphère sans poignée/trou. Le genre peut changer si plusieurs tétrahèdres à la fois sont retirés de $V$ [6].

Dans notre contexte, on a une connaissance supplémentaire de visibilité $R_i$ fournie par SfM : tout point $\mathbf{p}_i \in P$ est calculé à partir d'images prises aux points de vues $\mathbf{c}_j$ avec $j \in R_i$. Ceci implique que $|\partial V|$ ne devrait pas intersecter les rayons (segments de droite) $\mathbf{c}_j\mathbf{p}_i$, $j \in R_i$ à l'exception de $\mathbf{p}_i$. Les tétrahèdres intersectés par un rayon sont étiquetés "vide", les autres sont "plein". Ainsi, les triangles entre vide et plein sont de bon candidats pour être dans $\partial V$. Grâce à l'information de visibilité, on reconstruit des surfaces avec moins de points que dans les méthodes citées précédemment. Cependant, on n'impose plus $P \subset |V|$ car des faux points SfM sont toujours possibles.

Plusieurs méthodes de sculpture [13, 19, 16, 5, 21, 12] utilisent l'information de visibilité et un nuage de point épars $P$ estimé à partir d'images, mais [13, 19, 16, 12] n'imposent pas la contrainte de 2-variété sur $|\partial V|$. Les méthodes [16, 12] sont incrémentales ; [16] définit directement $V$ comme la liste des tétrahèdres pleins, [12] y ajoute une contrainte de régularisation, mais cela ne garantie pas la propriété de 2-variété. La méthode [19] est temps réel (pour les petits objets) et gère le bruit, mais n'est pas incrémentale. Dans [13], une méthode non-incrémentale estime une surface minimisant un coût combinant visibilité et photoconsistence. La méthode [5] est limitée à une surface de genre 0. La méthode [21] est incrémentale mais la complexité en temps d'une itération est parfois trop grande : si la trajectoire de la caméra est une boucle, cette complexité est au moins linéaire en le nombre de poses dans la boucle. Notre contribution (traduction française et amélioration de [15]) est une méthode de sculpture incrémentale calculant une 2-variété à partir de points épars SfM, sans restriction de genre et sans complexité prohibitive en présence de boucles dans la trajectoire. Notons que d'autres méthodes éparses [17, 10, 20] existent, mais elles ne sont pas du type sculpture et n'estiment pas de 2-variétés incrémentalement.

# 3 Pré-requis

## 3.1 Conditions/tests de 2-variété

La surface $|\partial V|$ est une 2-variété si pour tout sommet $\mathbf{v}$ de $|\partial V|$, le graphe des arêtes opposées à $\mathbf{v}$ dans les triangles de $\partial V$ est un cycle [7]. Dans notre cas où la triangulation de Delaunay 3D est représentée par le graphe $g$ d'adjacence des tétrahèdres [1], on utilise une condition "duale" sur les tétrahèdres [14]. Soit $g(\mathbf{v}, V)$ le sous graphe de $g$ dont les tétrahèdres contiennent $\mathbf{v}$, et dont on a retiré les arêtes reliant $V$ à $T \setminus V$. La condition (dite **condition générale**) utilisée est : chaque $g(\mathbf{v}, V)$ a exactement deux composantes connexes (la première dans $V$ et la seconde dans $T \setminus V$, et ici $T$ doit être augmenté des tétrahèdres infinis [1]).

Si on veut ajouter un seul tétrahèdre $\Delta$ à $V$ tel que $|\partial V|$ reste une 2-variété, on peut utiliser cette condition sur chacun des sommets dans $\Delta \cap |\partial V|$. Dans ce cas particulier, on préfère cependant la **condition d'addition** dans [21] qui est plus rapide. Celle-ci s'effectue sur les tétrahèdres ayant un sommet commun avec $\Delta$ et avant d'ajouter $\Delta$ dans $V$. Enfin, si on veut retirer à la place d'ajouter, on utilise la **condition de soustraction** qui est similaire à la condition d'addition (retirer $\Delta$ de $V$ revient à ajouter $\Delta$ à $T \setminus V$).

## 3.2 Notations

A l'image (ou instant) $t+1$, notre méthode prend en entrée :

- une triangulation de Delaunay 3D $T_t$ ;
- une liste $F_t$ de tétrahèdres vides telle que $F_t \subseteq T_t$ ;
- une liste $O_t$ telle que $O_t \subseteq F_t$ et $|\partial O_t|$ est une 2-variété ;
- une liste $P_{t+1}$ de nouveaux points SfM, les positions $\mathbf{c}_{t'} \in \mathbb{R}^3$ de la caméra avec $t' \leq t+1$.

A $t+1$, elle a en sortie :

- une triangulation de Delaunay 3D $T_{t+1}$, en ajoutant $P_{t+1}$ dans $T_t$ ;
- une liste $F_{t+1}$ de tétrahèdres vides (par lancer de rayons dans $T_{t+1}$) telle que $F_{t+1} \subseteq T_{t+1}$ ;
- une liste $O_{t+1}$ telle que $|\partial O_{t+1}|$ est une 2-variété et $O_{t+1}$ est la plus grande possible dans $F_{t+1}$.

Les notations $T, O, F$ sont ces listes si $t$ n'a pas besoin d'être précisé. La boule 3D centrée en $\mathbf{x} \in \mathbb{R}^3$ et de rayon $r > 0$ est $B_r(\mathbf{x})$. On remarque que $P_0 \cup P_1 \cup \cdots \cup P_t$ est l'ensemble des sommets de $T_t$.

## 3.3 Hypothèses et conséquences

On fait les hypothèses suivantes pour des raisons de complexité. D'abord on suppose que $\mathbf{p}_i \in P_t$ et $t' \in R_i$ impliquent $t \in R_i$ et $t' \leq t$ : $\mathbf{p}_i$ est reconstruit à partir de l'image $t$ et les précédantes, mais pas à partir des suivantes. Donc le rayon $\mathbf{p}_i\mathbf{c}_t$ existe si $\mathbf{p}_i \in P_t$, et le SfM ne modifie plus un point reconstruit dès qu'il est sommet de $T$.

Ensuite on suppose que les longueurs des rayons $\mathbf{p}_i\mathbf{c}_t$ sont bornées par $r > 0$. Donc $P_t$ est inclus dans $B_r(\mathbf{c}_t)$. Ceci permet aussi d'éliminer les points d'incertitudes grandes.

Enfin, on suppose que les diamètres des tétrahèdres sont bornés par $l > 0$. Ceci est vérifié si $T$ est initialisé par insertion des sommets d'une grande grille régulière : le diamètre de la sphère circonscrite d'un tétrahèdre est inférieure à la longueur $l$ de la diagonale d'une cellule de la grille implicite [14]. Comme tout tétrahèdre créé par ajout d'un point a ce point comme sommet [5], l'addition de $P_{t+1}$ est une mise à jour locale de $T$ incluse dans $B_{r+l}(\mathbf{c}_{t+1})$.

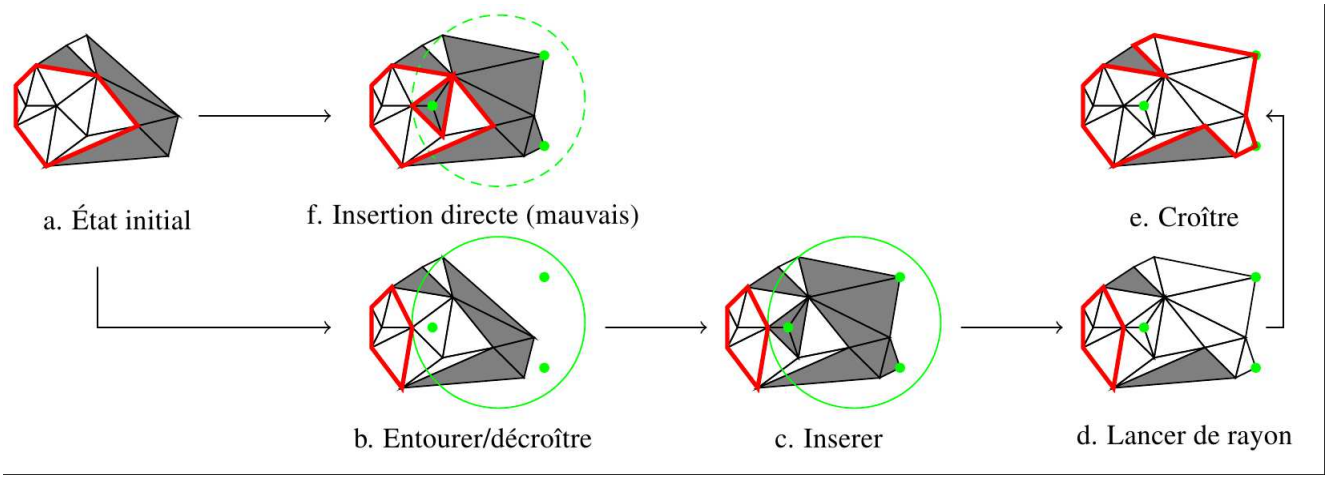


FIG. 1 – Vue générale de la méthode de sculpture (cas 2D). Les triangles blancs sont vides (dans $F$), les gris sont pleins, la ligne rouge est le bord $\partial O$, les points verts sont ceux qui sont ajoutés, et le cercle vert entoure les tétrahèdres qui sont supprimés par ajout des points verts.

# 4 Notre méthode de sculpture

Supposons que l'on ajoute $P_{t+1}$ dans $T_t$ dès que $O_t$ est calculé. Ceci détruit une liste $D$ de tétrahèdres, qui peuvent contenir des tétrahèdres de $O_t$ (Fig. 1.f). Le problème est le suivant : en initialisant $O_{t+1} = O_t \setminus D$, $|\partial O_{t+1}|$ n'est plus forcement une 2-variété.

Voici donc notre idée. On calcule $D$ *sans* ajouter $P_{t+1}$ dans $T_t$. Puis on initialise $O_{t+1} = O_t$, fait décroître $O_{t+1}$ de sorte que $|\partial O_{t+1}|$ reste une 2-variété en retirant progressivement des tétrahèdres de $O_{t+1}$ jusqu'à ce que $O_{t+1} \cap D = \emptyset$. On peut maintenant ajouter $P_{t+1}$ dans $T_t$ sans modifier $O_{t+1}$. Si $O_{t+1} \cap D = \emptyset$ n'est pas exactement satisfaite, une minorité de points dans $P_{t+1}$ ne sont pas ajoutés dans $T_t$ pour être sur que $|\partial O_{t+1}|$ reste une 2-variété. Enfin, on lance des rayons pour obtenir $F_{t+1}$, et on fait croître $O_{t+1}$ dans $F_{t+1}$ de sorte que $|\partial O_{t+1}|$ reste une 2-variété.

## 4.1 Étape 1 : entourer la zone à mettre à jour

Il faudrait calculer $D$ sans modifier $T$. En pratique, on remplace $D$ par $E$ tel que $D \subseteq E \subseteq T_t$ et la décroissance sera stoppée lorsque $O_{t+1} \cap E = \emptyset$ (ce qui implique ce que l'on veut : $O_{t+1} \cap D = \emptyset$). On choisit $E$ comme la liste des tétrahèdres de $T_t$ inclus dans une boule $B$ contenant $|D|$ (Fig. 1.b). On prend $B = B_{r+l}(\mathbf{c}_{t+1})$ (sec. 3.3).

## 4.2 Étape 2 : décroissance de $O$

Il faut trouver $O_{t+1}$ tel que $O_{t+1} \subseteq O_t \setminus E$ et $|\partial O_{t+1}|$ est 2-variété. On initialise $O_{t+1} = O_t$ (Fig. 1.a) et retire de $O_{t+1}$ des tétrahèdres un par un (condition de soustraction) ou plusieurs à la fois (condition générale). Pendant ce processus de décroissance, $|\partial O_{t+1}|$ est 2-variété et $O_{t+1} \subseteq F_t$ (Fig. 1.b). Les détails sont dans la sec. 4.7. En pratique, l'ensemble $O_{t+1} \cap D$ ainsi obtenu est presque toujours vide.

## 4.3 Étape 3 : ajout de points sans changer $O$

On initialise $T_{t+1} = T_t$ et $F_{t+1} = F_t$. Pour tout point $\mathbf{p}$ dans $P_{t+1}$, on effectue le traitement suivant.

Grâce à CGAL [1], on calcule la liste $D(\mathbf{p})$ de tétrahèdres de $T_{t+1}$ qui seraient détruits si on ajoute $\mathbf{p}$ dans $T_{t+1}$.

Si $D(\mathbf{p}) \cap O_{t+1} = \emptyset$, on ajoute $\mathbf{p}$ dans $T_{t+1}$ (Fig. 1.c). Ceci ne modifie pas $O_{t+1}$ et donc $|\partial O_{t+1}|$ reste une 2-variété. On applique aussi $F_{t+1} \leftarrow F_{t+1} \setminus D(\mathbf{p})$, i.e. les tétrahèdres créés par l'ajout de $\mathbf{p}$ sont étiquetés pleins. On leur ajoute une date de création $t+1$. On a toujours $O_{t+1} \subseteq F_{t+1}$.

Si $D(\mathbf{p}) \cap O_{t+1} \neq \emptyset$, il est difficile de mettre à jour $O_{t+1}$ de sorte que $|\partial O_{t+1}|$ reste une 2-variété et $T_{t+1}$ reste une triangulation de Delaunay 3D. On n'ajoute pas $\mathbf{p}$ dans ce cas. On applique alors $P_{t+1} \leftarrow P_{t+1} \setminus \{\mathbf{p}\}$.

## 4.4 Étape 4 : lancer de rayon

Des tétrahèdres de $T_{t+1}$ sont étiquetés vides par lancer de rayons, i.e. on complète $F_{t+1}$ (Fig. 1.d). On ne lance pas tous les rayons pour des raisons de complexité.

D'abord, on trace les rayons de $P_{t+1}$, i.e. tout rayon $\mathbf{p}_i\mathbf{c}_j, j \in R_i$ de tout point $\mathbf{p}_i$ dans $P_{t+1}$, puisque ce calcul n'a pas été fait avant. Lancer $\mathbf{p}_i\mathbf{c}_j$ revient à cheminer dans le graphe d'adjacence des tétrahèdres : on commence par un tétrahèdre ayant pour sommet $\mathbf{p}_i$, et on passe d'un tétrahèdre à l'un de ses 4 voisins si le triangle commun (face) est intersecté par $\mathbf{p}_i\mathbf{c}_j$. Chaque tétrahèdre a un compteur d'intersections qui est incrémenté s'il est dans le parcours.

Ensuite, on collecte dans une liste $R$ tout rayon de $P_{t'}$ (avec $t' \leq t$) qui *peut* intersecter un tétrahèdre de $N$. $N$ est la liste des nouveaux tétrahèdres, i.e. les tétrahèdres qui ont pour date de création $t+1$ (Sec. 4.3). On trace alors chaque rayon de $R$ comme cela vient d'être expliqué. Comme ces rayons ont déjà été tracés, on incrémente seulement le compteur des tétrahèdres de $N$ (chaque rayon est compté une seule fois).

On calcule $R$ ainsi. Soit $X_N$ une boite englobante des tétrahèdres de $N$. Soit $X_{t'}$ une boite englobante des rayons de $P_{t'}$. Pour tout $t'$ tel que $t' \leq t$ et $X_N \cap X_{t'} \neq \emptyset$, on met dans $R$ les rayons de $P_{t'}$ qui intersectent $X_N$.

### 4.5 Étape 5 : croissance de $O$

On fait croître $O_{t+1}$ en lui ajoutant des tétrahèdres un par un (condition d'addition) ou plusieurs à la fois (condition générale). On obtient $O_{t+1} \subseteq F_{t+1}$ et $|\partial O_{t+1}|$ est 2-variété (Fig. 1.e). Cette étape est similaire à celle hors-ligne dans [21], et en voici un bref aperçu.
D'abord, on applique une croissance par ajout d'un tétrahèdre à la fois. Une file de priorité $Q$ stocke les tétrahèdres dans $F_{t+1} \setminus O_{t+1}$ qui on un triangle dans $\partial O_{t+1}$ (on initialise $Q$ avec un tétrahèdre dans $F_{t+1} \cap E$). A chaque étape, on prend dans $Q$ le tétrahèdre $\Delta$ avec le plus grand compteur d'intersections et essaye d'ajouter $\Delta$ à $O_{t+1}$. Si la condition d'addition est satisfaite, on ajoute $\Delta$ dans $O_{t+1}$ et les tétrahèdres de $F_{t+1} \setminus O_{t+1}$ adjacents à $\Delta$ sont ajoutés dans $Q$. La croissance est stoppée lorsque $Q$ est vide. Celle ci est rapide grâce à la condition d'addition, mais le genre de $|\partial O|$ reste toujours le même.
Ensuite, on applique des ajouts de plusieurs tétrahèdres à la fois pour permettre un changement de genre. On trouve un sommet $\mathbf{v}$ dans $|\partial O_{t+1}| \cap |E|$ tel que tous les tétrahèdres ayant $\mathbf{v}$ pour sommet sont dans $F_{t+1}$, et on essaye d'ajouter ces tétrahèdres dans $O_{t+1}$. Si la condition générale est satisfaite, on démarre des croissances un-par-un (ce qui vient d'être expliqué) à partir de ces tétrahèdres. Le processus s'arrête lorsqu'il n'y a plus de $\mathbf{v}$ possible.

### 4.6 Étape 6 : post-traitement

On applique d'abord une méthode de raffinement à $O_{t+1} \cap E$ comme cela est décrit dans [22], à ceci près que l'on n'ajoute pas de point de Steiner au milieu d'arêtes. De cette façon, $T$ reste une triangulation de Delaunay, et on profite de ses bonnes propriétés d'approximation [2] ainsi que de l'implémentation efficace [1]. Ensuite, on applique une étape de lissage incrémental [21].

### 4.7 Détails sur l'étape de décroissance

On initialise $O_{t+1} = O_t$ et retire progressivement des tétrahèdres dans $O_{t+1}$ de sorte que $|\partial O_{t+1}|$ reste une 2-variété jusqu'à ce que $O_{t+1} \cap E = \emptyset$. La décroissance de $O_{t+1}$ est un processus inverse de la croissance (sec. 4.5). Soit $Q$ une liste (file de priorité) de tétrahèdres dans $O_{t+1} \cap E$ qui ont un triangle dans $\partial O_{t+1}$.
D'abord, $O_{t+1}$ décroît en retirant les tétrahèdres un par un. On retire de $Q$ le tétrahèdre $\Delta$ qui a le compteur d'intersection le plus petit. Si $\Delta \notin O_{t+1}$ ou $\Delta$ n'a pas de triangle dans $\partial O_{t+1}$, on passe au $\Delta$ suivant dans $Q$. Autrement, on essaye de retirer $\Delta$ de $O_{t+1}$ de sorte que $|\partial O_{t+1}|$ reste une 2-variété. Si la condition de soustraction est vérifiée, on ajoute à $Q$ les tétrahèdres de $O_{t+1} \cap E$ qui sont adjacent a $\Delta$. On continue jusqu'à ce que $Q = \emptyset$. Cette décroissance est rapide grâce à la condition de soustraction, mais, comme le genre ne change pas, on peut avoir $E \cap O_{t+1} \neq \emptyset$.

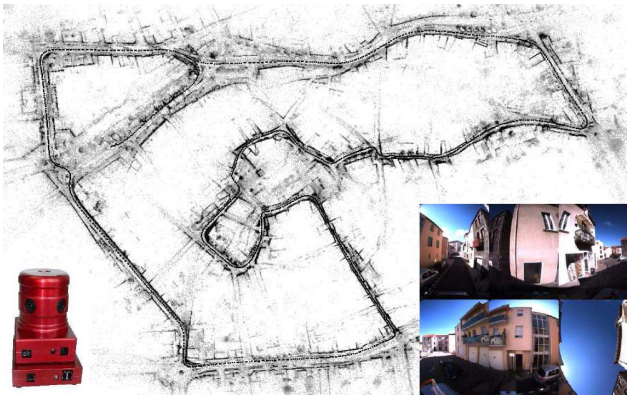

FIG. 2 – Caméra, trajectoire, nuage de points épars par SfM, et images de la séquence.

Ensuite, $O_{t+1}$ décroît en retirant plusieurs tétrahèdres à la fois pour permettre le changement de genre. On trouve un sommet $\mathbf{v}$ qui est à la fois dans un triangle de $\partial O_{t+1}$ et dans un tétrahèdre de $E$, on définit $L$ par la liste de tétrahèdres dans $O_{t+1} \cap E$ qui ont $\mathbf{v}$ pour sommet, et on essaye de retirer $L$ de $O_{t+1}$. Si la condition générale pour $\partial O_{t+1}$ est satisfaite en tout sommet de $L$, on redéfinit une liste $Q$ qui sont les tétrahèdres adjacents à ceux de $L$, et démarre une décroissance un-par-un (qui vient d'être décrite). Le processus s'arrête lorsqu'il n'y a plus de $\mathbf{v}$ possible.

## 5 Expériences

### 5.1 Séquence d'images réelles

On utilise une caméra omnidirectionnelle Ladybug, qui est un système rigide (non central) de 6 caméras synchronisées, chacune prenant des images $1024 \times 768$ à 15 Hz. Elle est fixée sur une voiture et est placée à environ 4 m du sol. La trajectoire fait 2.5 km de long et a une boucle de 2.3 km.
D'abord, on donne des détails sur l'étape SfM. Une approximation centrale simplifie les calculs (on force les centres des 6 cameras à être identiques). De plus, la taille des images en entrée est divisée par deux pour accélerer le calcul. On adapte aussi le SfM dans [18] basé sur l'ajustement de faisceaux local. La boucle est fermée grâce à un ajustement de faisceaux global. A la fin du SfM, 1306 images clefs sont sélectionnées parmi les 7735 6-uplets d'images, 483k points sont reconstruits à partir de 2.64M points de Harris détectés et appariés dans les images. Ainsi, la surface doit être reconstruite avec un nombre de points peu élevé (environ 193 par mètre parcouru par la caméra). La fig. 2 montre une vue de dessus du résultat de SfM.
Ensuite, on explique comment initialiser notre méthode incrémentale. Soit $t$ l'index d'image clef (même notation que dans les sec. 3 et 4). On construit la triangulation de Delaunay 3D $T_{40}$ a partir des points dans $P_t$ avec $t \leq 40$, estime $F_{40}$ par lancer de tous les rayons de ces $P_t$ (Sec. 4.4), et obtenons $O_{40}$ par l'étape de croissance (Sec. 4.5) et le raffinement de [22].
Les points de $P_t$ sont filtrés de la façon suivante. Soit $\mathbf{v}$

la direction verticale ($||\mathbf{v}|| = 1$) et $s = \sum_{t=0}^{39} ||\mathbf{c}_{t+1} - \mathbf{c}_t||/40$. Tout point $\mathbf{p}_i$ de $P_t$ vérifie $-3s \leq \mathbf{v}^\top(\mathbf{p}_i - \mathbf{c}_t)$ (suppression des points sous le sol), $||\mathbf{p}_i - \mathbf{c}_t|| \leq 15s$ (Sec. 3.3) et $\mathbf{p}_i$ doit avoir deux rayons formant un angle supérieur à $10°$ (filtrage standard par angle aperture). Le pas de la grille (Sec. 3.3) est $10s$.

Enfin, on applique la méthode incrémentale pour $40 \leq t \leq 1306$. La fig. 3 montre la surface finale (574k triangles) ainsi que des vues locales de la surface à plusieurs instants $t$. La **vidéo jointe** montre la reconstruction de surface progressive et la surface finale.

Le temps d'exécution pour chaque image est à gauche de la fig. 4. On utilise un 4xIntel Xeon W3530 à 2.8 GHz (le multi-threading est utilisé pour le lancer de rayons, et pour les tests d'ajouts/suppression de tétrahèdres par paquets pour la décroissance/croissance de $O$). L'étape la plus coûteuse en rouge est le raffinement (Sec. 4.6). Cette étape et la plus grande partie des autres sont seulement appliquées dans $E$, qui est la liste des tétrahèdres dans la boule $B$ de rayon borné (sec. 4.1). D'après la fig. 4 (à droite), $E$ contient toujours moins de $95k$ tétrahèdres, ce qui est beaucoup moins que $2089k$, le nombre de tétrahèdres total dans la triangulation de Delaunay finale.

On vérifie qu'une minorité de points SfM ne sont pas ajoutés dans $T$ (sec. 4.3) à cause de $O_{t+1} \cap D \neq \emptyset$ : un seul point dans toute la séquence ne peut être ajouté.

Les différences avec la version précédante [15] sont les suivantes : correction d'un bug dans l'étape de décroissance (qui est maintenant de bien meilleure qualité, mais plus lente), réduction de la boule englobante $B$ (le pas de la grille passe de $15s$ à $10s$). Ce dernier point et l'utilisation plus générale du multi-threading font que les temps de calculs sont inférieurs à ceux de [15].

### 5.2 Séquence de synthèse avec vérité terrain

On donne ici une erreur de reconstruction de notre méthode éparse (SfM central + reconstruction de surface incrémentale) pour une scène de synthèse urbaine, dont les textures proviennent d'images réelles. La séquence de synthèse a 1553 images générées par lancer de rayon et la caméra (non-centrale) de la sec. 5.1. La trajectoire est une boucle fermée de longueur 621m. Le SfM reconstruit 346 images clefs et 145k points à partir de 775k points de Harris détectés et appariés dans les images. Puis on reconstruit incrémentalement la surface et estime l'erreur entre la surface finale (198k triangles) et la surface vérité terrain.

On explique maintenant comment estimer l'erreur de reconstruction. D'abord un recalage basé-caméra est utilisé pour mettre les deux surfaces dans le même repère. L'image clef $i$ a une position estimée $\mathbf{c}_i$ de vérité terrain $\mathbf{c}_i^g$ ; la seconde est la moyenne des 6 centres des caméras dans le repère vérité terrain. Puis on estime la similarité $S$ minimisant $E(S) = \sum_{i=0}^{345} ||S(\mathbf{c}_i) - \mathbf{c}_i^g||^2$, et utilise $S$ pour mettre la surface estimée dans le repère vérité terrain. Ensuite, une approche de type lancer de rayon est utilisée pour calculer l'erreur. Soit $\mathbf{p}$ un pixel dans l'image clef $i$.

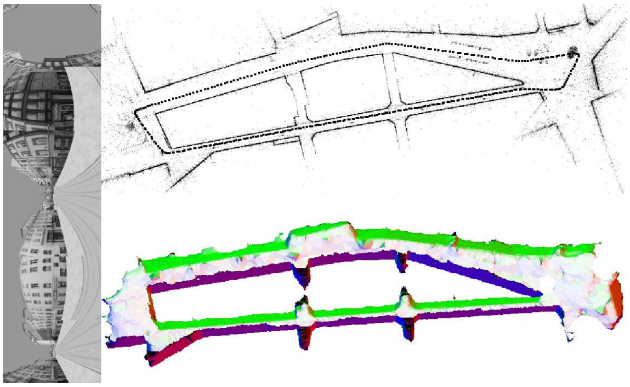

FIG. 5 – A gauche : image de la scène synthétique. A droite : résultats de SfM et la surface finalement estimée.

Soit $\mathbf{p}_e$ (resp. $\mathbf{p}_g$) le premier point d'intersection entre la surface estimée (resp. vérité terrain) et le rayon défini par $\mathbf{p}$ et la $i$-ième pose estimée (resp. vérité terrain). Si $\mathbf{p}_e$ et $\mathbf{p}_g$ existent tout les deux, on définit l'erreur $e(\mathbf{p}) = ||\mathbf{p}_e - \mathbf{p}_g||$.

Le recalage donne $\sqrt{E(S)/346} = 14$ cm. Ensuite on échantillonne uniformément dans la séquence d'images clefs et examine la distribution de $e(\mathbf{p})$ pour 769k pixels. Le $x$-quantile $q_x$ est le réel tel que $x$ pourcents des valeurs de $e$ sont inférieures à $q_x$. On a $q_{10} = 12$, $q_{20} = 14$, $q_{30} = 18$, $q_{40} = 2$, $q_{50} = 28$, $q_{60} = 48$, $q_{70} = 76$, $q_{80} = 114$, $q_{90} = 184$, et 4% des valeurs de $e$ sont supérieurs à 600 (tous les chiffres sont en centimètres).

## 6 Conclusion

Cet article introduit une méthode incrémentale de reconstruction de surface qui prend en entrée un nuage de point épars obtenu par SfM. Comparée avec les méthodes utilisant des hypothèses similaires, elle a l'avantage de produire une 2-variété sans restriction de genre et avec une complexité non prohibitive (la complexité d'une itération ne dépend pas de la longueur des boucles de la trajectoire, contrairement à [21]). Elle a été expérimentée sur une séquence d'images omnidirectionnelles urbaines de 2.5 km ainsi que sur une séquence synthétique avec vérité terrain.

Des améliorations sont possibles sur plusieurs points : le raffinement dans l'étape de post-traitement (trop long actuellement), la suppression des triangles aberrants comme ceux du ciel, la mise en correspondance. D'autres pistes existent pour améliorer la qualité de la surface reconstruite, comme utiliser d'autres types de primitives que les points ou plus d'information a priori pendant le lissage. Enfin, on travaille l'implémentation pour réduire les temps de calcul.

## Remerciements

Ce travail est financé par le CNRS, UBP, FEDER et la région Auvergne. Le modèle 3D synthétique (vérité terrain) provient du projet CRISTAL.

FIG. 3 – Vues de la surface reconstruite. En haut : vues globales et locales de la surface finale (les triangles au ciel sont retirés pour faciliter la visualisation). En bas : vues locales de la surface pendant son calcul. Les normales des triangles sont colorées : le sol en blanc, les murs verticaux sont rouges-verts-bleus, le ciel est noir.

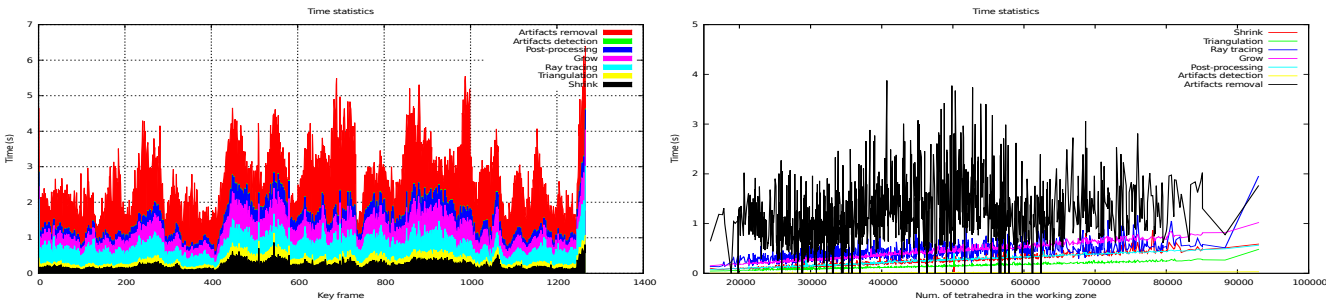


FIG. 4 – Temps de reconstruction pour la séquence d'images réelles

## Références


[1] CGAL, *Computational Geometry Algorithm Library*, www.cgal.org

[2] N. Amenta and M. Bern, *Surface reconstruction by voronoi filtering*, Discrete Computational Geometry, vol. 22, num. 4, pp. 481-504, 1999.

[3] J.D. Boissonnat, *Geometric structures for three-dimensional shape representation*, ACM Transaction on Graphics, vol. 3, num. 4, 1984.

[4] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez and B. Levy, *Polygon mesh processing*, AK Peters, 2010.

[5] O. Faugeras, E. Le Bras Mehlman and J.D. Boissonnat, *Representing stereo data with the Delaunay triangulation*, Artificial Intelligence, 1990.

[6] A. Gezahegne, *Surface reconstruction with constrained sculpting*, Master of Science Thesis, University of California Davis, 2005.

[7] P. Giblin, *Graphs, surfaces and homology*, Cambridge University Press, Third Edition, 2010.

[8] A. Gueziec, G. Taubin, F. Lazarus and B. Horn, *Cutting and stitching : converting sets of polygons to manifold surfaces*, IEEE Transactions on Visualization and Computer Graphics, vol. 7, num. 2, 2001.

[9] C. Hernandez Esteban and F. Schmitt, *Silhouette and stereo fusion 3D object modeling*, Computer Vision and Image Understanding, vol. 96, num. 3, 2004.

[10] A. Hilton, *Scene modelling from sparse 3D data*, Image and Vision Computing, vol. 23, 2005.

[11] C.M. Hoffmann, *Geometric and solid modeling : an introduction*, Morgan Kaufmann Publishers Inc., 1989.

[12] C. Hoppe, M. Klopschitz, M. Donoser and H. Bischof, *Incremental Surface Extraction from Sparse Structure-from-Motion Point Clouds*, BMVC'13.

[13] P. Labatut, J.P. Pons and R. Keriven, *Efficient multiview reconstruction of large-scale scenes using interest points, Delaunay triangulation, and graph cut*, ICCV'07.

[14] M. Lhuillier and S. Yu, *Manifold surface reconstruction of an environment from sparse structure-from-motion data*, Computer Vision and Image Understanding, vol. 117, num. 11, 2013.

[15] V. Litvinov and M. Lhuillier, *Incremental solid modeling from sparse and omnidirectional structure-from-motion data*, BMVC'13.

[16] D. Lovi, N. Birkbeck, D. Cobzas and M. Jagersand, *Incremental free-space carving for real-time 3D reconstruction*, 3DIMPVT'12.

[17] D.D. Morris and T. Kanade, *Image-consistent surface triangulation*, CVPR'00.

[18] E. Mouragnon, M.Lhuillier, M. Dhome, F. Dekeyer and P. Sayd, *Generic and real-time structure from motion*, BMVC'07.

[19] Q. Pan, G. Reitmayr and T. Drummond, *ProFORMA : probabilistic feature-based on-line rapid model acquisition*, BMVC'09.

[20] C.J. Taylor, *Surface reconstruction from feature based stereo*, ICCV'03.

[21] S. Yu and M. Lhuillier, *Incremental reconstruction of a manifold surface from sparse visual mapping*, 3DIMPVT'12.

[22] S. Yu and M. Lhuillier, *Genus refinement of a manifold surface reconstructed by sculpting the 3D-Delaunay triangulation of Structure-from-Motion points*, ICPR'12.