

# Modèles 3d à partir d'un nuage épars de points et une caméra catadioptrique

Shuda Yu et Maxime Lhuillier

LASMEA UMR 6602, UBP/CNRS, 24 avenue des Landais, 63177 Aubière Cedex

<http://maxime.lhuillier.free.fr>

## Résumé

La majorité des méthodes automatiques de modélisation 3d à partir d'images reconstruisent un ensemble dense de points. Nous proposons ici de reconstruire une surface directement à partir de l'ensemble épars de points reconstruits en estimant la géométrie. Deux méthodes voisines sont présentées, l'une est progressive (la surface est progressivement complétée au fur et à mesure que les images sont lues) et l'autre pas. Il s'agit aussi d'obtenir une variété topologique de dimension 2, pas une simple soupe de triangles mal connectés les uns aux autres. Malgré le faible nombre de points reconstruits, l'expérience montre que l'on peut obtenir des résultats intéressants pour des environnements extérieurs avec une caméra catadioptrique.

## Mots Clef

Reconstruction de surface, variété topologique de dimension 2, nuage de points épars, caméra catadioptrique.

## Abstract

The majority of automatic 3d modeling methods from images reconstruct a dense cloud of points. Here we propose to reconstruct a surface directly from the sparse cloud of points provided by geometry estimation. Two related methods are presented : one is progressive (the surface is progressively updated while reading the images) and the other is not. We also would like to obtain a 2d topological manifold, not a simple soup of partially connected triangles. Although the number of reconstructed points is low, experiments show that it is possible to obtain interesting results for outdoor environments using a catadioptric camera.

## Keywords

Surface reconstruction, 2d topological manifold, sparse cloud of points, catadioptric camera.

## 1 Introduction

La reconstruction automatique de scène à partir d'une séquence d'images est toujours un domaine de recherche actif. Dans notre contexte où la scène est un environnement et les points de vues dans un voisinage du sol (imagerie non aérienne), l'utilisation d'une caméra à grand champs de vue est un choix naturel car on doit reconstruire tout ce qui est visible autour des points de vue.

Il y a deux étapes dans le calcul. D'abord, les paramètres de caméra (poses successives, paramètres intrinsèques) et

un nuage de points épars sont estimés simultanément. Ensuite, une liste de triangles texturés en 3d sont estimés à partir des points et/ou des images elles mêmes.

Dans le cas idéal (c'est le cas dans cet article), chaque triangle est exactement connecté à trois autres triangles par leurs arêtes, la surface obtenue n'a ni trou ni auto-intersection, et elle découpe l'espace en régions "extérieures" et "intérieures". Chaque point de la surface a un voisinage (dans la surface) qui a la topologie d'un disque : la surface est une variété topologique de dimension 2, ce que l'on abrégera dans la suite par **2-variété**.

De plus, contrairement à la majorité des systèmes de reconstruction existants, nous n'essayons pas de reconstruire un ensemble de points dense avec une méthode de stéréo dense. Nous pensons qu'un ensemble épars de points devrait être suffisant pour des applications de visite virtuelle ou de localisation de véhicules.

Cet article présente deux méthodes qui génèrent une 2-variété à partir d'un nuage épars de points reconstruits avec une caméra catadioptrique. La première suppose que tous les points ont déjà été calculés. La seconde est plus adaptée aux applications en ligne et suppose que les points sont reconstruits progressivement lors de la lecture de la séquence d'images : la 2-variété est progressivement mise à jour avec les points reconstruits très récemment. On compare maintenant nos méthodes avec les plus proches travaux.

### 1.1 Lancers de rayons dans un Delaunay 3D

Beaucoup de méthodes de reconstruction de surface [5] sont basées sur la propriété suivante [3] : si  $P$  est un échantillonnage de points suffisamment dense sur la surface (lisse) de la scène, alors une bonne approximation de cette surface est donnée par une liste de triangles de la triangulation  $T$  de Delaunay 3D de  $P$ . Rappelons que  $T$  est une liste de tétraèdres qui partitionnent l'enveloppe convexe de  $P$ , telle que  $P$  est l'ensemble des sommets de tous les tétraèdres. Les triangles sont les facettes (i.e. les bords) des tétraèdres. La fig. 1 illustre cette propriété dans le cas 2d. Ici, les points sont reconstruits à partir d'images et on connaît la liste  $V_i$  d'indices des images qui reconstruisent le point 3d  $p_i \in P$ . On appelle "rayon" un segment qui relie  $p_i$  à la  $j$ -ième position du centre optique de la caméra  $c_j$  tel que  $j \in V_i$ . Les méthodes de sculpture par lancer de rayon (incluant les nôtres) utilisent ces rayons comme contraintes de visibilité pour étiqueter les tétraèdres : un tétraèdre est "vide" s'il est intersecté par un rayon. Les

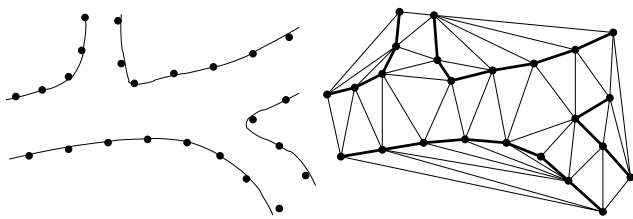


FIGURE 1 – En 2d, les surfaces/tétraèdres/triangles sont remplacés par des courbes/triangles/arêtes, respectivement. À gauche : courbes à reconstruire et points qui les échantillonnent. À droite : triangulation de Delaunay des points. Les arêtes de Delaunay en gras approximent les courbes.

autres tétraèdres sont “pleins”. Les méthodes de reconstruction par sculpture dans une triangulation de Delaunay 3d sont [7, 9, 15, 12, 18]. Elles s’appliquent sur un nuage de points (ou d’arêtes [7]) éparés, et seule [12] est une méthode progressive.

Toutes ces méthodes (à l’exception de [7] et [18]) estiment directement la surface par la liste des triangles séparant les tétraèdres vides et pleins (voir la fig. 2 dans le cas 2d). Malheureusement, la surface ainsi obtenue peut ne pas être

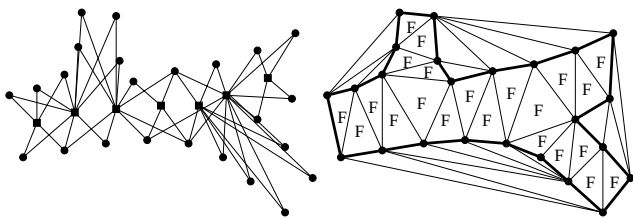


FIGURE 2 – À gauche : points (cercles) et centre optiques (carrés) reliés par des rayons (segments). À droite : les triangles de Delaunay intersectés par des rayons sont vides et notés par “F”. Les autres triangles sont pleins. Les arêtes de Delaunay séparant vide et plein sont en gras.

une 2-variété (voir la fig. 3 dans le cas 2d). Par exemple, il y a une singularité sur le sommet  $v$  dans la surface si tous les tétraèdres sont pleins à l’exception de deux tétraèdres  $t_1$  et  $t_2$  tels que  $t_1 \cap t_2 = \{v\}$ . Dans [7], une méthode de crois-

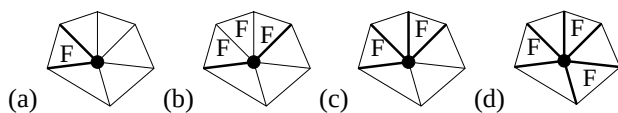


FIGURE 3 – 4 configurations de triangles d’une triangulation de Delaunay 2d au voisinage d’un sommet  $v$  (cercle). La courbe est la liste d’arêtes en gras séparant les triangles vides (notés “F”) et pleins. Les courbes sont des variétés topologiques de dimension 1 dans (a),(b), mais pas (c),(d).

sance de région dans la liste des tétraèdres pleins supprime toutes les singularités et génère une 2-variété. Une paire stéréo est utilisée pour reconstruire des scènes très simples. Dans [18], la croissance de région est améliorée

pour traiter le cas de topologies plus complexes : les régions pleines et vides peuvent être des toroïdes à plusieurs trous. Ensuite, la surface d’une église complète (avec son voisinage) est obtenue à partir de seulement 76033 points reconstruits avec une caméra catadioptrique.

Cet article est une version française et améliorée de [18], à laquelle on a ajouté une seconde méthode progressive.

## 1.2 Autres méthodes voisines

Une méthode progressive [2] utilise aussi une triangulation de Delaunay 3d. Elle donne en général une 2-variété, et ne stocke en mémoire centrale que la partie la plus récente et localement modifiée du Delaunay 3d. Il n’y a pas de contraintes de visibilité, mais il est nécessaire d’avoir un nuage de points plus dense et de connaître une direction principale de la distributions des points.

Les méthodes [13, 17, 8] utilisent des triangulations 2d (pas 3d) dans les images et traitent le cas de nuages éparés de points. Seule [8] est progressive, mais la surface obtenue n’est pas une 2-variété (il y a des trous et des redondances) et les expériences sont faites sur des séquences d’images réelles très courtes. Dans [13], la surface est raffinée en commutant des arêtes et elle doit minimiser la somme pour tous les triangles de la covariance des luminances dans les images. Dans [17], des arêtes reconstruites sont ajoutées dans des Delaunay 2d contraints, puis les triangulations 2.5d résultantes sont fusionnées à l’aide de la méthode “marching cube”. On obtient une 2-variété, mais une subdivision régulière de l’espace est nécessaire. Cependant, une subdivision non régulière de l’espace avec des tétraèdres est plus adéquate pour les scènes de grande échelle.

## 2 Méthode non progressive

On décrit maintenant chacune des étapes de cette méthode.

### 2.1 Triangulation de Delaunay 3D

On utilise [11] pour le calcul automatique de la géométrie : on estime simultanément les poses successives de la caméra, des paramètres intrinsèques (coefficients de distortion radiale supposés constants), et un nuage éparés de points. Le modèle de caméra est central avec un axe de symétrie, et ces points sont les reconstructions de points de Harris appariés dans les images. Cette étape donne également l’information de visibilité  $V_i$  (partie 1.1).

Notons qu’un point  $p_i$  a une mauvaise précision s’il est reconstruit dans une configuration dégénérée : si  $p_i$  et tous les  $c_j, j \in V_i$  sont presque collinéaires (notations dans la partie 1.1). Ce cas arrive dans une ligne droite de la trajectoire de caméra et lorsque des points reconstruits pour cette partie sont proches de cette droite. Ainsi,  $p_i$  est éliminé de tout traitement ultérieur si tous les angles  $c_j p_i c_k$  ( $j, k \in V_i$ ) sont inférieurs à un seuil  $\epsilon$ . Les points restants sont les sommets de la triangulation  $T$  de Delaunay 3d.

### 2.2 Lancers de rayons

Tous les tétraèdres sont initialisés à plein, et on effectue un lancer de rayon sur chaque rayon pour marquer à vide

tout tétraèdre intersecté par un rayon.  $T$  est un graphe [6] : un sommet du graphe est un tétraèdre, un arc du graphe est un triangle séparant deux tétraèdres. Tracer un rayon  $p_i c_j$  est un simple parcours de graphe, partant du tétraèdre contenant  $c_j$ , se déplaçant sur un des 4 tétraèdres voisins par le triangle intersecté par le segment  $p_i c_j$ , se terminant dans le tétraèdre dont un des sommets est  $p_i$ .

On connaît l'état (vide ou plein) de tous les tétraèdres, ceux-ci partitionnant l'enveloppe convexe  $C$  des points filtrés. On cherche maintenant l'état de  $\mathbb{R}^3 \setminus C$ . Dans notre contexte catadioptrique où les points  $p_i$  sont reconstruits dans quasiment toutes les directions, les points de vues  $c_j$  sont dans  $C$ . Donc les rayons sont aussi dans  $C$ . Un rayon ne peut intersecter  $\mathbb{R}^3 \setminus C$ , et alors  $\mathbb{R}^3 \setminus C$  est plein.

### 2.3 Définition et test d'une 2-variété

La surface  $S$  à obtenir est une liste de triangles de  $T$ , qui doit à la fois être une 2-variété et approcher la surface séparant les tétraèdres vides et pleins.

Soit  $o$  un point sur la surface  $S$ . On dit que  $o$  est régulier s'il a un voisinage dans  $S$  qui a la topologie d'un disque, c'est à dire s'il y a un homéomorphisme entre ce voisinage et  $\{x \in \mathbb{R}^2, ||x|| \leq 1\}$ . Sinon  $o$  est singulier. Par définition,  $S$  est une 2-variété si tous ses points sont réguliers. Dans notre contexte où  $S$  est une liste de triangles de  $T$ , il suffit de vérifier que chacun des sommets  $o$  de  $S$  est régulier en utilisant un voisinage particulier de  $o$  : la liste des triangles de  $S$  qui ont pour sommet  $o$  [4]. Ainsi,  $o$  est régulier si et seulement si les arêtes opposées à  $o$  dans les triangles de  $S$  ayant  $o$  comme sommet forment un polygone simple.

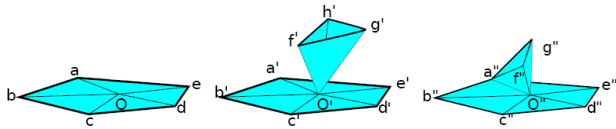


FIGURE 4 –  $O$  est régulier car les arêtes opposées à  $O$  forment un polygone simple sur la surface (bleue).  $O'$  et  $O''$  ne sont pas réguliers car les polygones  $a'b'c'd'e'a' - f'g'h'f'$  et  $a''b''c''d''e''a'' - f''g''h''f''$  ne sont pas simples (le premier est non connexe, le second a un sommet multiple  $a''$ ).

### 2.4 Principe de la croissance de région

Une 2-variété découpe  $\mathbb{R}^3$  en régions étiquetées “intérieur” et “extérieur”. Dans notre approche, la région extérieure  $O$  de  $S$  contient un maximum de tétraèdres vides et aucun tétraèdre plein. On utilise une méthode de croissance de région : on ajoute un par un des tétraèdres vides dans  $O$  de sorte que le bord de  $O$  soit toujours une 2-variété. Le bord du  $O$  final est la surface recherchée.

Grâce à la partie 2.3, le bord de  $O$  est une 2-variété si et seulement si tous ses sommets sont réguliers. Comme on ajoute les tétraèdres un par un, les voisinages d'au plus 4 sommets (ceux du tétraèdre ajouté) sont modifiés. Il suffit donc de vérifier que ces sommets sont réguliers après

l'ajout du tétraèdre. Si ce n'est pas le cas, on retire ce tétraèdre de  $O$  et on en essaye un autre.

Le  $O$  final dépend de l'ordre d'ajout des tétraèdres dans  $O$ . On choisit d'ajouter dans  $O$  un tétraèdre qui a au moins une facette (triangle) inclus dans  $S$ , i.e. qui est dans le voisinage immédiat de  $O$ . On définit également un ordre de priorité pour l'ajout en associant à chaque tétraèdre vide un score de priorité : le nombre de rayons qui intersectent le tétraèdre. Les tétraèdres dans le voisinage de  $O$  sont stockés dans un tas (file de priorité) pour une sélection rapide du tétraèdre ayant la plus grande priorité.

### 2.5 Algorithme de croissance de région

Les entrées sont le  $O$  initial, la liste  $F_0$  de tétraèdres dans lequel la croissance est possible, une liste  $Q_0$  de tétraèdres qui inclue l'initialisation du tas, et la fonction  $r$  qui associe à chaque tétraèdre  $\Delta$  le nombre de rayons qui l'intersecte. Dans le cas non progressif,  $O = \emptyset$ ,  $Q_0 = T$  et  $F_0$  est la liste de tous les tétraèdres vides. La sortie est  $O$ . Voici un algorithme rédigé avec le style du langage C.

```
// **** initialisation du tas Q ****
Q = ∅;
if (O==∅) { // pour version non progressive
    soit Δ ∈ F0 tel que r(Δ) est maximum;
    Q ← Q ∪ {Δ};
} else { // pour version progressive
    pour chaque tetrahedre Δ dans Q0
        if (Δ ∉ O et un de ses 4-voisins Δ' ∈ O)
            Q ← Q ∪ {Δ};
}
// **** croissance de region de O ****
while (Q != ∅) {
    retirer de Q le Δ qui a le plus grand r(Δ);
    if (Δ ∈ O) continue;
    O ← O ∪ {Δ};
    if (tous les sommets de Δ sont reguliers) {
        pour chaque tetrahedre Δ' qui est 4-voisin de Δ
            if (Δ' ∈ F0 and Δ' ∉ O) Q ← Q ∪ {Δ'};
    } else O ← O \ {Δ};
}
```

Dans la boucle while, on insère au plus 4 tétraèdres dans  $Q$  à chaque fois qu'un tétraèdre est ajouté définitivement dans  $O$  avec  $O \subseteq T$ . Il n'y a donc pas de boucle infinie, et on peut montrer que la complexité en temps est  $O(|T| \log(|T|))$  avec  $|T|$  le nombre de tétraèdres dans  $T$ .

### 2.6 Fermeture de boucle

Notons que la croissance de région ainsi définie ne change pas la topologie initiale de  $O$ . Ici  $O$  est initialisé par un tétraèdre qui a la topologie d'une boule. Ceci est un problème si la véritable région extérieure n'a pas cette topologie, par exemple lorsque la trajectoire de la caméra forme des boucles autour d'immeubles. Dans le cas le plus simple d'une seule boucle, la région extérieure devrait avoir la to-

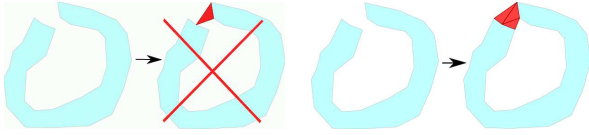


FIGURE 5 – À gauche : on ne peut pas fermer une boucle en ajoutant un seul tétraèdre (en rouge) dans  $O$  (en bleu clair) à cause d'un sommet singulier. À droite : ajouter plusieurs tétraèdres à la fois permet de le faire.

pologie d'un toroïde, mais la région extérieure calculée ne peut pas fermer la boucle (fig. 5).

On corrige ce problème de la façon suivante. D'abord on trouve un sommet sur  $S$  tel que tous les tétraèdres intérieurs incident à ce sommet sont vides. Ensuite, on force tous ces tétraèdres à être extérieurs. Si tous les sommets de tous ces tétraèdres sont réguliers, la méthode réussit et  $O$  est gardé tel quel. Sinon, on remet ces tétraèdres à intérieurs (i.e. on les retire de  $O$ ). Enfin, on alterne ce schéma avec la méthode de la partie 2.5 jusqu'à ce que l'on ne puisse plus faire croître  $O$ . Ici  $F_0$  est inchangé et  $Q_0$  est la liste de tétraèdres voisins aux tétraèdres forcés ci-dessus.

## 2.7 Post-traitements

Bien que la surface ainsi obtenue soit une 2-variété, plusieurs défauts apparaissent pendant la visualisation. On liste maintenant ces défauts et explique comment les réduire en utilisant de l'information a priori sur la scène.

**Suppression des pics.** Un pic est un sommet de  $S$  tel que l'anneau de ses triangles adjacents dans  $S$  définit un angle solide  $w$  qui est trop petit (ou trop grand) pour être physiquement plausible, i.e.  $w < w_0$  ou  $w > 4\pi - w_0$  avec  $w_0$  un seuil. Nous considérons maintenant le coté aigu du pic (les tétraèdres extérieurs ou intérieurs) et inversons leur état (intérieur devient extérieur, et vice versa). La suppression du pic réussit si tout les sommets des tétraèdres inversés sont réguliers. Autrement, on inverse une seconde fois pour annuler l'inversion. Puis on essaye de supprimer un autre pic. En pratique, on parcourt plusieurs fois la liste des sommets de  $S$  pour détecter et supprimer les pics.

**Lissage de surface.** Afin de réduire le bruit de reconstruction, on applique sur la surface  $S$  un filtre  $\mathbf{p}' = \mathbf{p} + \Delta\mathbf{p}$  avec  $\mathbf{p}$  un sommet de  $S$  et  $\Delta\mathbf{p}$  un laplacien discret défini sur les sommets de  $S$  [16]. Le  $\mathbf{p}'$  lissé est rangé dans un tableau distinct de celui de  $\mathbf{p}$ . On n'effectue pas  $\mathbf{p} \leftarrow \mathbf{p}'$  pour éviter le surplus de calculs dû au déplacement d'un sommet dans  $T$  (ce lissage est idempotent).

**Suppression du ciel.** Jusqu'à maintenant,  $S$  est fermée et contient des triangles correspondant au ciel (si les images sont prises dehors). Ces triangles doivent être supprimés car ils ne correspondent pas à une vraie surface. Ils compliquent aussi la visualisation du modèle 3d d'un point de vue aérien. D'abord, la direction verticale  $\mathbf{v}$  est estimée de façon robuste en supposant que le mouvement de la caméra est (en gros) sur un plan horizontal. Ensuite, on considère

des rectangles "ouverts" définis par une arête finie  $\mathbf{c}_i\mathbf{c}_{i+1}$  et deux arêtes infinies partant de  $\mathbf{c}_i$  (ou  $\mathbf{c}_{i+1}$ ) et de direction  $\mathbf{v}$ . Un triangle de  $S$  intersecté par un rectangle ouvert est un triangle du ciel et est supprimé de  $S$ . Maintenant  $S$  est trouée au ciel. Ce trou est agrandi en propageant son bord, de triangle à triangle, de sorte que l'angle entre la normale du triangle et  $\mathbf{v}$  soit inférieur à un seuil  $\alpha_0$ .

**Placage de texture.** On définit la texture pour chaque triangle de  $S$  grâce à une version simplifiée de [10], qui prend telle quelle la texture d'un point de vue bien choisi  $\mathbf{c}_j$  pour chaque triangle  $tr$  de  $S$ . Dans notre cas où la séquence comporte plusieurs centaines d'images (pas des dizaines), on pré-sélectionne une liste de candidats  $\mathbf{c}_j$  pour  $tr$  de la façon suivante :  $tr$  est entièrement projeté dans l'image de  $\mathbf{c}_j$  (les grands  $tr$  sont subdivisés),  $tr$  n'est pas occulté par un autre triangle de  $S$ ,  $\mathbf{c}_j$  donne l'un des  $u$  plus grands angles solides pour le triangle  $tr$  (avec  $u$  un seuil).

## 3 Méthode progressive

On décrit maintenant chacune des étapes de cette méthode.

### 3.1 Estimation de la géométrie

Cette estimation est elle aussi progressive : il s'agit de [14] dans le cas particulier d'une caméra centrale catadioptrique. On alterne les étapes suivantes : (1) une nouvelle image clef est sélectionnée de la vidéo d'entrée et des points de Harris sont appariés avec ceux de l'image clef précédente par corrélation (2) la nouvelle pose est estimée avec la méthode de Grunert et RANSAC (3) de nouveaux points 3d sont reconstruits avec les nouveaux appariements (4) un ajustement de faisceaux local (AFL) raffine la géométrie des  $l$  images clefs les plus récentes. La partie 3 va ajouter les étapes (5), (6) ... à cette boucle principale. Cette méthode distingue les images clefs et non-clefs pour obtenir un résultat stable. Comme les calculs 3d ne se font que sur les images clefs, on ignore les images non-clefs dans cet article et on utilise  $t$  à la fois comme index de temps et d'images clefs. Avec  $l = 3$  (comme dans [14]), les  $l$  images les plus récentes sont  $t - 2$ ,  $t - 1$  et  $t$ .

### 3.2 Ajout de points dans la triangulation $T$

Afin d'éviter les mises à jour de  $T$  coûteuses en calculs, on n'insère un point dans  $T$  que lorsque celui-ci a atteint sa valeur finale. Plus précisément, on ajoute dans  $T$  à l'instant  $t$  (i.e. après l'AFL du temps  $t$ ) tous les points 3d tels que leurs dernier point tracké est dans l'image  $t - 2$ . Comme dans la partie 2.1, on vérifie d'abord que ces points ne sont pas dans une configuration dégénérée.

La méthode progressive a besoin d'associer une date de création aux points et tétraèdres de  $T$ . De plus, on utilise l'implémentation de Delaunay 3d de CGAL [6] qui ajoute les points un par un dans  $T$ . L'ajout de  $\mathbf{p}$  détruit une liste  $L_d(\mathbf{p})$  de tétraèdres et crée une liste  $L_c(\mathbf{p})$  de tétraèdres différents. On associe la date  $t$  à  $\mathbf{p}$  et aux tétraèdres de  $L_c(\mathbf{p})$ . On a aussi besoin de la plus petite date  $d_t$  de tous les tétraèdres extérieurs dans  $L_d(\mathbf{p})$  détruits à la date  $t$



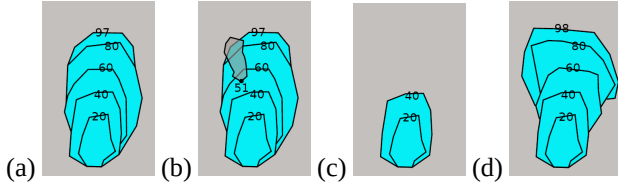


FIGURE 6 – Croissance de région pour l'image  $t = 98$  avec  $n = 20$ . (a) : avant la croissance de région pour l'image 98,  $O_{20}, O_{40}, O_{60}, O_{80}$  et  $O_{97}$  sont déjà calculés. (b) : l'ajout de points dans  $T$  détruit des tétraèdres. La plus petite date de création des tétraèdres détruits lors de cet ajout est  $d_t = 51$ . (c) :  $O_{60}, O_{80}$  et  $O_{97}$  sont invalides et supprimés. (d) : croissances de régions de  $O_{40}$  à  $O_{60}$ , de  $O_{60}$  à  $O_{80}$ , et de  $O_{80}$  à  $O_{98}$ .

(pour tout  $\mathbf{p}$  ajouté à l'instant  $t$ ). Chaque tétraèdre a été étiqueté extérieur ou intérieur dans l'étape de croissance de région (partie 3.4) réalisée à un instant précédant. On calcule facilement  $L_c(\mathbf{p})$  et  $L_d(\mathbf{p})$  grâce à CGAL.

### 3.3 Lancers de rayons

Cette étape doit être progressive, et il est trop coûteux de lancer tous les rayons connus jusqu'à la date  $t$ . Notons que les points 3d et les tétraèdres ont tous une date de création. Un rayon a pour date de création celle de son point 3d. On utilise ici une hypothèse de localité spatiale : un rayon ne peut intersecter un tétraèdre que si leurs dates de création sont similaires. On en déduit que l'état (vide ou plein) d'un tétraèdre ancien est connu depuis longtemps, et qu'il ne reste plus qu'à connaître l'état des tétraèdres les plus récents. De plus, les tétraèdres les plus récents ne sont intersectés que par les rayons les plus récents. On utilise donc la méthode suivante : à la date  $t$ , on applique un lancer de rayons pour les rayons dont les dates de créations sont dans  $\{t - k, \dots, t - 1, t\}$  avec  $k$  un seuil.

### 3.4 Croissance de région

Dans le cas progressif, il est naïf de démarrer la croissance de région à  $O = \emptyset$  comme dans la partie 2.5 : il est mieux de partir d'une liste  $O$  obtenue à une date antérieure. Ceci implique le stockage de plusieurs listes  $O$  à tout instant. Pour simplifier les notations dans cette partie, le temps  $t$  part de 1 (pas 0). Soit  $\mathcal{L}_t$  l'ensemble des tétraèdres créés à la date  $t'$ . A tout instant  $t$ , l'algorithme gère plusieurs listes de tétraèdres extérieurs qui correspondent à des dates multiples d'un entier  $n$  :  $O_n, O_{2n}, \dots, O_{i_t n}$  avec  $i_t$  le plus grand entier tel que  $i_t n \leq t$ . Ces listes et une autre liste  $O_t$  de tétraèdres extérieurs vérifient

$$\begin{aligned} O_n \subseteq O_{2n} \subseteq \dots \subseteq O_{(i_t-1)n} \subseteq O_{i_t n} \subseteq O_t \\ \forall t' \in \{n, 2n, \dots, i_t n, t\}, O_{t'} \in \mathcal{L}_1 \cup \mathcal{L}_2 \cup \dots \cup \mathcal{L}_{t'} \\ \text{et le bord de } O_{t'} \text{ est une 2-variété.} \end{aligned} \quad (1)$$

La surface  $S$  à la date  $t$  est le bord de  $O_t$ .

Si  $t \leq n$ , on applique la croissance de région comme dans la partie 2.5 pour obtenir  $O_t$  à partir de  $O_0 = \emptyset$ .

Supposons maintenant que  $t > n$ . L'algorithme est une récurrence sur  $t$  pour l'éq. 1. La fig. 6 illustre la récurrence

pour  $t = 98, n = 20, d_t = 51$ . D'après la partie 3.2, l'ajout de points à l'instant  $t$  détruit des tétraèdres extérieurs et on connaît leur plus petite date  $d_t$ . Soit  $i_0$  le plus grand entier tel que  $i_0 n < d_t$ . Si  $i \leq i_0$ ,  $O_{i n}$  est inchangé et son bord reste une 2-variété. Si  $i > i_0$ ,  $O_{i n}$  peut avoir des tétraèdres détruits, son bord n'est plus nécessairement une 2-variété, et  $O_{i n}$  doit être recalculé. Le temps  $t$  part de 1, donc  $1 \leq d_t, 0 \leq i_0$  et  $O_{i_0 n}$  existe. On applique alors la croissance de région (partie 2.5) de  $O_{i_0 n}$  pour obtenir  $O_{(i_0+1)n}$ . On l'applique aussi de  $O_{(i_0+1)n}$  pour obtenir  $O_{(i_0+2)n}$  et ainsi de suite jusqu'à obtenir  $O_{i_t n}$ . Enfin, on l'applique de  $O_{i_t n}$  pour obtenir  $O_t$ .

Notons que  $F_0$  et  $Q_0$  doivent être définis pour la croissance de région de  $O_{i n}$  pour obtenir  $O_{(i+1)n}$  (ou  $O_t$ ), comme cela est dit dans la partie 2.5. Pour une raison de complexité en temps, on ne prend pas  $Q_0 = T$  mais les tétraèdres les plus récents  $Q_0 = \mathcal{L}_{i n - b_0} \cup \dots \cup \mathcal{L}_{(i+1)n}$  avec  $b_0 \in \mathbb{N}$  une constante. On définit aussi  $F_0$  par les tétraèdres vides de  $\mathcal{L}_{(i_0-1)n} \cup \dots \cup \mathcal{L}_{(i+1)n}$  (pas ceux de  $\mathcal{L}_0 \cup \dots \cup \mathcal{L}_{(i+1)n}$  qui est trop grand).

### 3.5 Fermeture de boucle

On modifie légèrement la méthode non progressive de la partie 2.6 pour obtenir une méthode progressive. D'abord, la méthode est seulement appliquée sur les sommets les plus récents créés aux instants  $t - b_1, \dots, t - 1, t$  avec  $b_1 \in \mathbb{N}$  une constante. Ensuite, chaque sommet n'est essayé qu'une seule fois. Pour les croissances de région, on utilise le  $Q_0$  de la partie 2.6 et  $F_0$  est la liste des tétraèdres vides de  $\mathcal{L}_{(i_0-1)n} \cup \dots \cup \mathcal{L}_t$ .

### 3.6 Lissage de surface

A l'instant  $t$ , on n'utilise la méthode de lissage du cas non progressif (partie 2.7) que sur des sommets  $\mathbf{p}$  de  $S$  dont le résultat du lissage peut changer entre  $t - 1$  et  $t$ . Le laplacien  $\Delta \mathbf{p}$  ne dépend que de  $\mathbf{p}$  et du voisinage immédiat de  $\mathbf{p}$  dans  $S$ , i.e. l'ensemble des sommets connectés à  $\mathbf{p}$  par une arête de  $S$ . Donc  $\mathbf{p}$  n'est lissé au temps  $t$  que si sa date de création, ou celle de l'un des sommets dans son voisinage immédiat, est supérieure à  $i_0 n$  (partie 3.4).

## 4 Expériences

La caméra catadioptrique est tenue à la main et montée sur un monopied tel que l'axe de symétrie est (en gros) vertical. Le miroir [1] donne un très grand champs de vue :  $360$  degrés dans le plan horizontal et environ  $60$ - $50$  degrés en dessous et au dessus. En pratique, l'utilisateur alterne un pas dans la scène et la prise d'une photographie. Pour les calculs, on utilise un Core 2 Duo E8500 à  $3.16$  GHz.

### 4.1 Méthode non progressive

**Géométrie.** La séquence a 208 images prises tout autour d'une église. La longueur de la trajectoire est d'environ  $(25 \pm 5\text{cm}) \times 208 = 52 \pm 10\text{m}$  (les longueurs exactes des pas entre images successives sont inconnues). Le calcul automatique de la géométrie [11] a reconstruit 76033 points

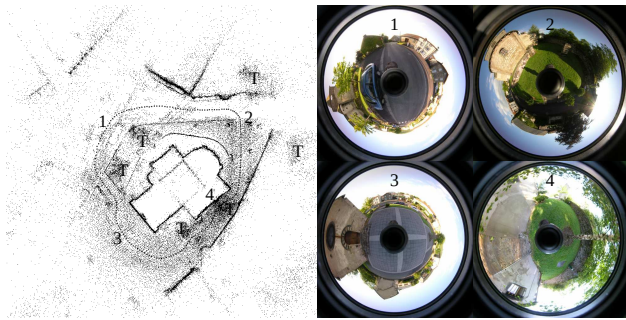


FIGURE 7 – À droite : 4 images de la séquence église. À gauche : vue de dessus de la géométrie calculée, avec 76033 points, 208 positions de caméra, des numéros où les 4 images sont prises, des “T” pour indiquer la présence d’arbres.

3d à partir de 477744 points (inliers) de Harris détectés et appariés dans les images. La fig. 7 montre des images de la séquence et le résultat de ce calcul.

**Différentes étapes.** Ensuite la surface est estimée avec  $\epsilon = 5$  degrés,  $w_0 = \pi/2$  stéradians,  $\alpha_0 = 45$  degrés et  $u = 10$ . La fig. 8 explique l’intérêt des étapes de la méthode. La colonne 1 montre que les points reconstruits ne peuvent être utilisés tels quels comme sommets de la surface, et que la suppression des pics et le lissage de surface sont utiles. La colonne 2 montre des vues de dessus de la surface avant (en haut) et après (en bas) la suppression du ciel. En haut, la surface est fermée et on voit les triangles du ciel. En bas, on peut visualiser le modèle 3d d’un point de vue aérien. Notons que la version actuelle de suppression du ciel est très simple, et devrait être améliorée pour supprimer les grands triangles en haut du modèle (notamment ceux qui connectent des arbres à l’église). En haut de la colonne 3, on voit que la surface forme un cul-de-sac si la fermeture de boucle n’est pas utilisée (le piéton ne peut pas aller du lieu noté 3 au lieu noté 4). La colonne 4 montre que la fermeture de boucle améliore également l’espace extérieur  $O$  en dehors de la zone du cul-de-sac (en bas de colonne 3).

**Points artificiels.** Cependant, tout ceci n’est pas suffisant pour éviter un problème montré en haut de la colonne 5 : on voit un arche qui connecte une partie verticale de la surface avec le sol, bien que tous les tétraèdres inclus dans cet arche soient vides. On a essayé de résoudre cela en changeant l’ordre de priorité des tétraèdres pour la croissance de région, mais des arches apparaissent toujours (au même endroit ou ailleurs). Ici, nous réduisons grandement le risque d’obtenir ces arches avec une méthode simple. Dans l’étape de triangulation de Delaunay (partie 2.1), on ajoute des points dits “artificiels” tels que les long tétraèdres potentiellement impliqués dans les arches sont subdivisés en tétraèdres plus petits. Ces points sont ajoutés dans une zone critique pour la visualisation : le voisinage immédiat de la trajectoire de la caméra. Le bas de la colonne 5 montre que ces points éliminent l’arche. Plus précisément, les points artificiels ont des listes de visibilité vides, leur nombre est égal à 0.5% du nombre de

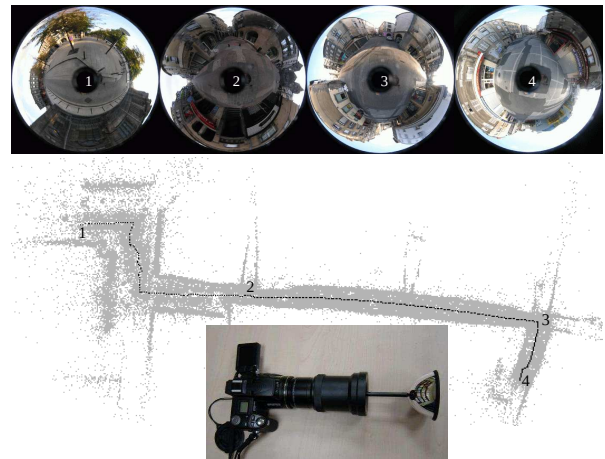


FIGURE 9 – En haut : 4 images de la séquence ville. Au milieu : une vue de dessus de la géométrie complètement calculée, avec 119206 points, 354 positions de caméra, et des numéros où les 4 images sont prises. En bas : Nikon Coolpix 8700 et miroir 0-360.

points reconstruits, et ils sont uniformément et aléatoirement ajoutés dans le voisinage immédiat  $N$  de la trajectoire de la caméra. On définit  $N$  par l’union des demi-boules (hémisphère sud tel que la direction du nord est  $\mathbf{v}$ ) centrées sur les  $\mathbf{c}_j$  avec un rayon  $r = 10 * \text{moyenne}_j ||\mathbf{c}_{j+1} - \mathbf{c}_j||$ .

**Résultat global.** La colonne 6 de la fig. 8 montre d’autres vues de l’église obtenue avec toutes les étapes de la méthode non progressive. La triangulation de Delaunay 3d a 59780 sommets. 191947 tétraèdres ont été déclarés vides après les lancers de 398956 rayons. La croissance de région et la fermeture de boucle génèrent 162174 tétraèdres extérieurs et une surface de seulement 94228 triangles pour modéliser l’église et son voisinage. Le rapport entre les nombres de tétraèdres extérieurs et tétraèdres vides est de 86% (la valeur idéale est 100%). Enfin, 898 triangles sont retirés du ciel. Le calcul de surface (sans la géométrie et le placage de texture) prend environ 30 s. Une vidéo HD accessible sur <http://maxime.lhuillier.free.fr> montre le tour complet de l’église.

## 4.2 Méthode progressive

**Géométrie.** La séquence a 354 images prises dans la partie ancienne d’une ville. La longueur de la trajectoire est d’environ  $(35 \pm 5\text{cm}) \times 353 = 122 \pm 17\text{m}$ . Le calcul automatique de la géométrie [14] reconstruit 119206 points 3d à partir de 819087 points de Harris détectés et appariés dans les images. La fig. 9 montre des images de la séquence et le résultat de ce calcul. Dans ce contexte, toutes les images sont forcées à être des images clefs (vu le procédé d’acquisition) et la calibration est donnée par [11]. De plus, la mise en correspondance de points de Harris dans [14] n’est pas très efficace et est remplacée par la mise en correspondance de points de Harris dans [11].

**Résultats.** On utilise ici  $\epsilon = 10$ ,  $k = 40$ ,  $n = 60$ . La fig. 12 montre la surface obtenue à 4 instants  $t$  diffé-

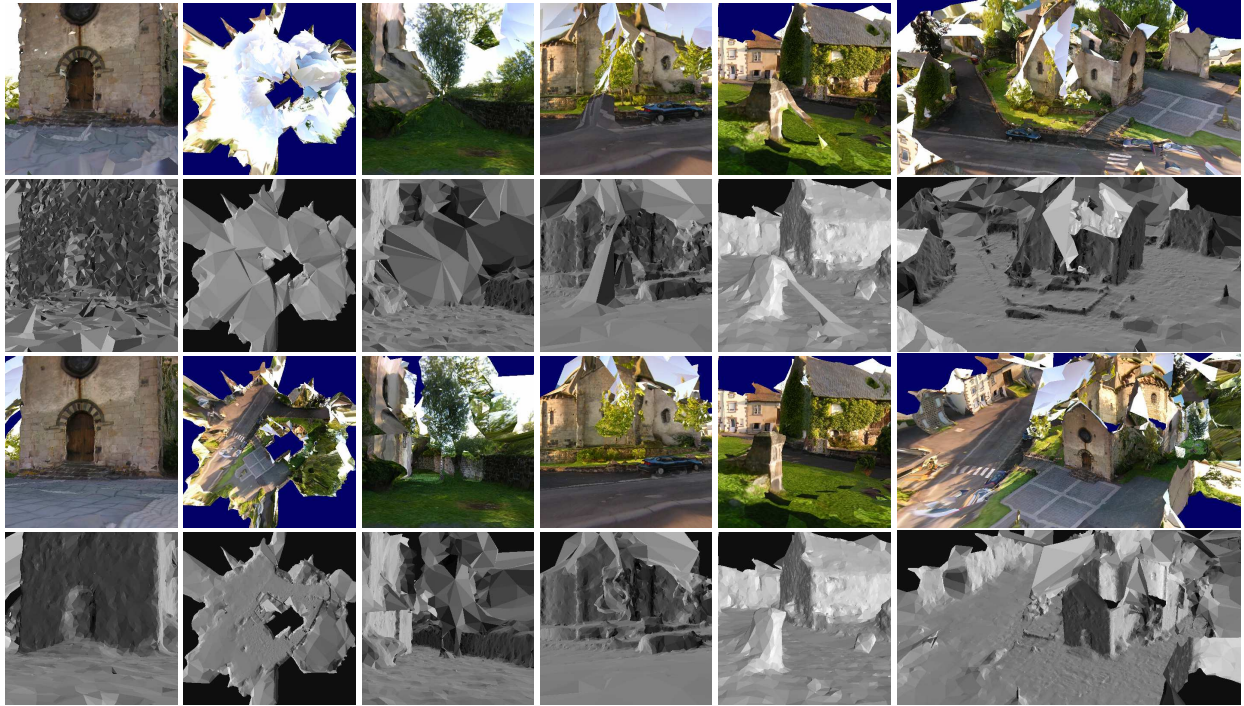


FIGURE 8 – Vues de modèles 3d de l'église obtenus avec la méthode non progressive. Colonne 1 : avec (bas) et sans (haut) la suppression des pics et le lissage de surface (partie 2.7). Colonne 2 : avec (bas) et sans (haut) la suppression du ciel (partie 2.7). Colonnes 3 et 4 : avec (bas) et sans (haut) la fermeture de boucle (partie 2.6). Colonne 5 : avec (bas) et sans (haut) les points artificiels (partie 4.1). Colonne 6 : avec toutes les étapes de la méthode. Dans tous les cas, les niveaux de gris indiquent les directions des normales des triangles.

rents. L'observateur se déplace dans la scène de sorte à observer la partie la plus récente de la surface au cours de sa construction. A l'instant  $t$ , cette partie se situe dans un voisinage immédiat de  $c_{t-2}$ , l'observateur est en  $c_{t-10}$  et observe en direction de  $c_{t-2}$ . Le bout de surface le plus récent et l'observateur avancent donc simultanément de sorte que leur éloignement est à peu près constant. La fig. 10 montre des vues de la dernière surface obtenue (145506 triangles).

La fig. 11 montre les temps de calcul des différentes étapes à chaque instant/image  $t$  : Delaunay (ajouts de points), Carving (les lancers de rayons), Manifold (croissance de région + fermeture de boucle), et Post-processing (lissage de surface). Environ 209 points sont ajoutés à chaque instant dans  $T$ . Delaunay et Post-processing sont quasiment négligeables par rapport aux autres étapes. Carving et Manifold prennent toujours moins de 290 et 280 millisecondes, respectivement. On explique maintenant la forme de Manifold. Supposons d'abord  $t - d_t$  constant ;  $t - d_t$  est environ égal à  $30 = n/2$  en pratique. Si  $t - i_t n > 30$ ,  $i_0 = i_t$  et on a une croissance de région de  $O_{i_t n}$  à  $O_t$ . Sinon,  $i_0 = i_t - 1$  et on a deux croissances de régions : de  $O_{(i_t-1)n}$  à  $O_{i_t n}$ , puis de  $O_{i_t n}$  à  $O_t$ . Dans les deux cas, le nombre de tétraèdres dans  $O_t$  mais pas dans  $O_{i_t n}$  croît avec  $t$  (ces tétraèdres sont datés  $i_t n + 1, \dots, t - 1, t$ ). Cela explique la forme en dent de scie de Manifold. Cette forme est perturbée par des variations de  $t - d_t$  autour de sa valeur moyenne, notamment quand celles-ci font passer de 1 à 2 (ou de 2 à 1) croissances de région.

D'autres expériences ont été faites avec d'autres valeurs de paramètres. Premièrement, on a essayé  $\epsilon = 5$ . Cela ajoute plus de points d'arrière plan, augmente la longueur moyenne des rayons, mais multiplie le temps de calcul par 1.6 sans pour autant améliorer la qualité de la surface. Deuxièmement, plus  $k$  est grand, plus le nombre de rayons à lancer est grand, plus l'état (vide ou plein) des tétraèdres est précis, meilleure est la qualité de surface. On a cependant observé que  $k > 40$  n'apporte rien et augmente le temps de calcul. Enfin, il faut éviter  $n < 10$  à la fois pour la qualité et le temps de calcul.

## 5 Conclusion

Dans cet article, des surfaces sont générées à partir de nuages de points épars reconstruits à partir d'images, chacune d'elles apportant de l'ordre de 300 points en plus. Les travaux précédents avec des chiffres aussi faibles étaient limités à des cas d'école (au plus quelques dizaine d'images, points parfois appariés manuellement, soupes de triangles). Ici, on reconstruit des scènes extérieures non triviales avec plusieurs centaines d'images catadioptriques, le calcul est automatique et inclus celui de la géométrie, enfin une variété topologique de dimension 2 (ou 2-variété) est obtenue. On utilise une triangulation de Delaunay 3d des points, des lancers de rayons pour segmenter l'espace en deux, et des croissances de régions pour obtenir une 2-variété. Les deux méthodes (progressive ou non) peuvent être améliorées de plusieurs façons, notamment pour la détection du ciel, la



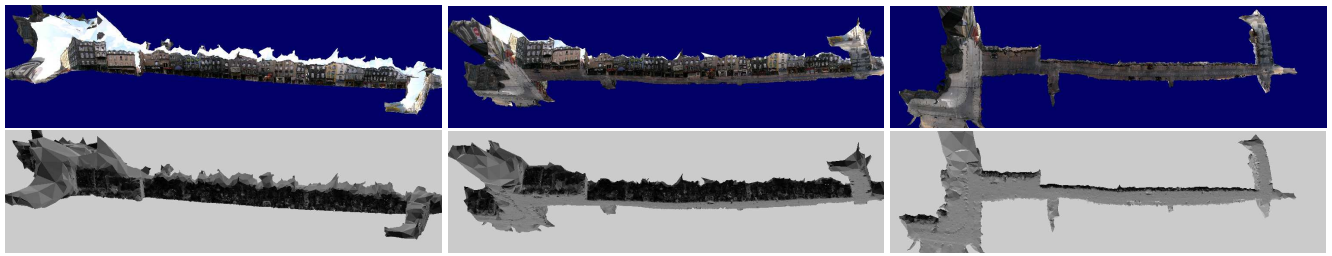


FIGURE 10 – Surface obtenue à la fin de la méthode progressive.

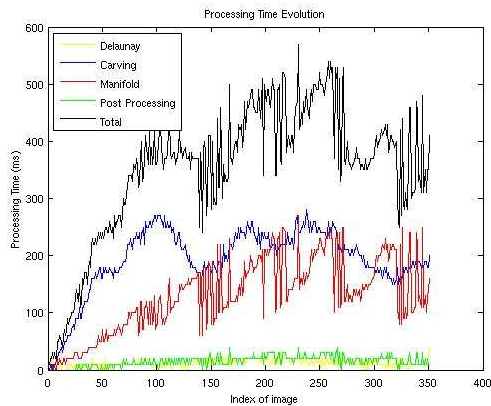


FIGURE 11 – Temps de calcul pour la méthode progressive.

suppression des arches, le placage de texture, et le lissage de surface. Les reconstructions et intégrations d'arêtes dans la surface devraient améliorer la qualité. Enfin, on peut réduire les temps de calculs en améliorant l'implémentation.

## Références

- [1] [www.0-360.com](http://www.0-360.com), [www.kaidan.com](http://www.kaidan.com)
- [2] R.Allegre, R.Chaine, S.Akkouche. A streaming algorithm for surface reconstruction. *Eurographic symposium on geometry processing*, 2007.
- [3] N.Amenta, M.Bern. Surface reconstruction by voronoi filtering. *Discrete Computational Geometry*, 22, 1998.
- [4] J.D.Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transaction on Graphics*, 3(4), p. 266-286, 1984.
- [5] F.Cazals, J.Giesen. Delaunay triangulation based surface reconstruction : ideas and algorithms. INRIA technical report 5394, 2004.
- [6] [www.cgal.org](http://www.cgal.org)
- [7] O.Faugeras, E.Le Bras-Mehlman, J.D.Boissonnat. Representing stereo data with the delaunay triangulation. *Artificial Intelligence*, p. 41-47, 1990.
- [8] A.Hilton. Scene modelling from sparse 3d data. *Image and Vision Computing*, 23, p. 900-920, 2005.
- [9] P.Labatut, J.P.Pons, R.Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and Graph-Cuts. *CVPR'07*.

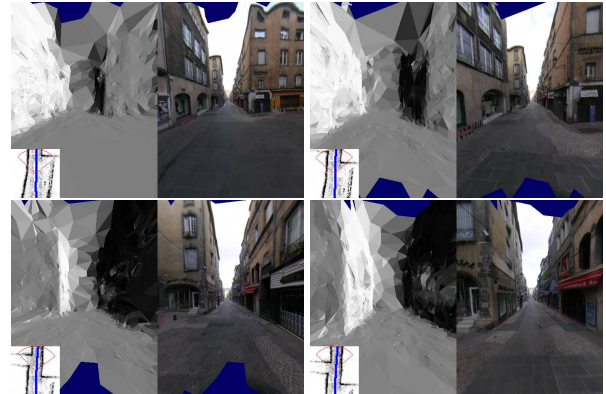


FIGURE 12 – 4 images montrant la reconstruction progressive de la surface (dans le sens de la lecture). À gauche de chaque image : les niveaux de gris pour les normales de triangles. À droite de chaque image : une image catadioptrique est utilisée pour le placage de texture (les trous en bleu sont dus aux triangles sans texture dans cette image).

- [10] V.Lempitsky, D.Ivanov. Seamless Mosaicing of Image-Based Texture Maps. *CVPR'07*.
- [11] M.Lhuillier. Automatic scene structure and camera motion using a catadioptric system. *Computer Vision and Image Understanding*, 109(2), p. 186-203, 2008.
- [12] D.Lovi, N.Birkbeck, D.Cobzas, M.Jagersand. Incremental free-space carving for real-time 3d reconstruction. *3DPVT'10*.
- [13] D.D.Morris, T.Kanade. Image-consistent surface triangulation. *CVPR'00*.
- [14] E.Mouragnon, M.Lhuillier, M.Dhome, F.Dekeyser, P.Sayd. Generic and real-time structure from motion. *BMVC'07*.
- [15] Q.Pan, G.Reitmayr, T.Drummond. ProFORMA : probabilistic feature-based on-line rapid model acquisition. *BMVC'09*.
- [16] G.Taubin. A signal processing approach to fair surface design. *SIGGRAPH'95*.
- [17] C.J.Taylor. Surface reconstruction from feature based stereo. *ICCV'03*.
- [18] S.Yu, M.Lhuillier. Surface reconstruction of scenes using a catadioptric camera. *Lecture notes in computer sciences 6930*, p.145-156, 2011 (also in *Mirage'11*).