



**HAL**  
open science

# LOCAL-CONVEXITY REINFORCEMENT FOR SCENE RECONSTRUCTION FROM SPARSE POINT CLOUDS

Maxime Lhuillier

► **To cite this version:**

Maxime Lhuillier. LOCAL-CONVEXITY REINFORCEMENT FOR SCENE RECONSTRUCTION FROM SPARSE POINT CLOUDS. 2019 International Conference on 3D Immersion (IC3D), Dec 2019, Brussels, France. pp.1-8, 10.1109/IC3D48390.2019.8975900 . hal-02750194

**HAL Id: hal-02750194**

**<https://hal.science/hal-02750194>**

Submitted on 3 Jun 2020

**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.

# LOCAL-CONVEXITY REINFORCEMENT FOR SCENE RECONSTRUCTION FROM SPARSE POINT CLOUDS

Maxime Lhuillier

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000, Clermont Ferrand, France

## ABSTRACT

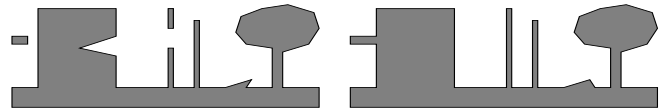
Several methods reconstruct surfaces from sparse point clouds that are estimated from images. Most of them build 3D Delaunay triangulation of the points and compute occupancy labeling of the tetrahedra thanks to visibility information and surface constraints. However their most notable errors are falsely-labeled freespace tetrahedra. We present labeling corrections of these errors based on a new shape constraint: local-convexity. In the simplest case, this means that a freespace tetrahedron of the Delaunay is relabeled matter if its size is small enough and all its vertices are in matter tetrahedra. The allowed corrections are more important in the vertical direction than in the horizontal ones to take into account the anisotropy of usual scenes. In the experiments, our corrections improve the results of previous surface reconstruction methods applied to videos taken by a consumer 360 camera.

**Index Terms**— Environment modeling, sparse features, 3D Delaunay triangulation, visibility, hole filling.

## 1. INTRODUCTION

The surface reconstruction from sparse image features is an interesting challenge and can be useful in several contexts including limited hardware resource, large scale environments, initialization of dense stereo methods, low-textured regions in images, applications that do not need a high level of details. Most methods build 3D Delaunay triangulation of the points reconstructed from images and compute occupancy labeling of the tetrahedra thanks to visibility information and surface constraints. Every tetrahedron is labeled freespace or matter. The surface is the set of the triangle faces between freespace and matter tetrahedra. However tetrahedra can be incorrectly labeled freespace due to noise, bad points and lack of points.

These errors are the most notable ones and take many forms: 3D box-like shapes (e.g. buildings) with spurious concavities, thin 1D-structures (e.g. posts) that are disconnected, thin 2D-structures (e.g. traffic signs) with tunnels connecting both sides. We introduce methods to remove such errors by using a new constraint: local-convexity (LC). Let  $M$  be the matter, i.e. the union of the matter tetrahedra. It is locally-convex at a point  $\mathbf{x} \in M$  if  $\mathbf{x}$  has a small neighborhood



**Fig. 1.** Anisotropic local-convexity for a reconstructed scene. The matter  $M$  is gray and the freespace is white. Left: a balcony and a post are disconnected, a building has a spurious concavity. Right: we would like (1) to fill the building concavity and the gaps of balcony and post and (2) to fill neither the gap between the two posts nor the gap between the ground surface and the tree foliage. The local convexity of  $M$  is used for (1), e.g. take  $\mathbf{y}$  and  $\mathbf{z}$  in different components of a post to connect it. Both locality and anisotropy are useful for (2).

$N \subseteq \mathbb{R}^3$  such that  $N \cap M$  is convex, i.e.  $\mathbf{yz} \subset N \cap M$  if  $\mathbf{y}, \mathbf{z} \in N \cap M$ . The LC of  $M$  is reinforced anisotropically by favoring the vertical direction due to the predominance of vertical structures in usual scenes. Fig. 1 shows what we would like: we complete the set of the matter tetrahedra at locations where it is not LC. Our methods are used as preprocessing or postprocessing of previous surface reconstructions methods.

## 2. PREVIOUS WORK

Shape priors or constraints are useful not only to take into account of noise and bad points, but also to deal with sparsity. They can be inspired by human perception [11] (e.g. group shape parts by convexity) and Gestalt theory [3] (e.g. group points by alignment or proximity laws). A method [14] uses proximity and interpolates a sparse set of points by a connected manifold surface. Starting by a 3D Delaunay triangulation of the points, the points are connected by Delaunay triangles whose edges are as small as possible. Other methods are based on  $\alpha$ -shapes [6], a generalization of convexity that replace half-spaces outside the shape by balls with radius equals to  $\alpha$ . However [14] is not robust to outliers and the visibility (lines-of-sight of the points) is unused in [6, 14].

We now consider surface reconstruction methods used in Computer Vision which assume a sparse point cloud input, e.g. acquired through Structure-from-Motion. They also start from a 3D Delaunay triangulation. A graph-cut method in [18] estimates a closed surface as a set of Delaunay triangles that

Thanks to CNRS, Institut Pascal and UCA for funding.

minimizes a cost function including a visibility term and a surface quality term. The former tries to avoid triangle if it is crossed by line-of-sight. The latter tries to avoid triangle that is face of “too flat” tetrahedron. In [10], the surface is computed as the boundary of an evolving set of tetrahedra that is maintained manifold: the set mostly grows in the tetrahedra crossed by lines-of-sight to meet the visibility constraint. Our experiments show that our corrections based on LC improve the results of both [18] and [10]. Our corrections can also be used with most surface reconstruction methods, which are based on graph-cut optimization or manifold constraint.

The original definition of LC is in [9] and uses Euclidean balls as neighborhoods. In our case, the neighborhoods are larger in the vertical direction than in the horizontal ones due to the anisotropy of usual scenes. Furthermore, LC is ignored in the previous works on surface reconstruction from 3D points reconstructed from images. Thus we briefly overview the usage of LC and its variants (convexity, weak-convexity, star-shapeness) in other problems. LC is expressed in terms of normals and vectors between two close surfaces patches [12, 16]. It is used for segmentation of surface regions into ground/object [12] and convex parts [16]. A method [1] segments an oriented point clouds into weakly convex regions. It is based on the percentage of point pairs whose connecting segment does not leave the inner volume of the shape. Star-shapeness generalizes convexity and can be efficiently integrated in a graph-cut optimization to segment an object in an image [17], if the user provides the star-shape center. Enforcing convexity [7] is more difficult than enforcing star-shapeness [17].

Previous works also complete the shape during its reconstruction but need strong assumptions to estimate primitives (in contrast to LC). In [2], planes are estimated from a point cloud which is denser than ours. Then a simplified surface is generated using priors: the prevalence of vertical structures and orthogonal intersections. A method [13] focuses on indoor scenes and builds a different 3D space partition (from scanner points) assuming that the surface is only horizontal or vertical. In [19], swept surfaces (a generalization of ruled surfaces) are estimated from sparse Structure-from-Motion points assuming that the scene is 100% architectural.

### 3. EXPLICIT USE OF LOCAL-CONVEXITY

Let’s introduce a method (named Method 1) that explicitly uses the LC definition in Sec. 1. This requires neighborhoods that are built using a norm:

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

where  $\mathbf{x} \in \mathbb{R}^3$ ,  $(\mathbf{i}, \mathbf{j}, \mathbf{k})$  is an orthonormal basis of  $\mathbb{R}^3$  such that  $\mathbf{k}$  is vertical,  $s_h$  and  $s_v$  are horizontal and vertical scale factors. The unit ball  $\{\mathbf{x} \in \mathbb{R}^3, \|\mathbf{x}\| \leq 1\}$  of  $\|\cdot\|$  is a cylinder with vertical axis, (horizontal) diameter  $2s_h$  and length  $2s_v$ . The choice of  $s_h$  and  $s_v$  is given in the Appendix.

Let  $T$  be the 3D Delaunay triangulation of the input points. It is a set of tetrahedra and their faces (vertices, edges, triangles) such that the tetrahedra are labeled freespace or matter, e.g. by ray-casting or graph-cut. There are several ways to apply LC. Method 1 does this: first check whether edges of  $T$  should be in matter, then relabel matter a tetrahedron of  $T$  if all its edges should be in matter. In the first step, booleans  $b(\mathbf{v}_i)$  and  $b(\mathbf{v}_i\mathbf{v}_j)$  are defined for every vertex  $\mathbf{v}_i \in T$  and every edge  $\mathbf{v}_i\mathbf{v}_j \in T$ . In the second step, every freespace tetrahedron  $\mathbf{v}_i\mathbf{v}_j\mathbf{v}_k\mathbf{v}_l \in T$  is relabeled matter iff (if and only if) the 10 booleans of its 4 vertices and 6 edges are true. For ease of understanding, we start from a simplified  $b$  in Sec. 3.1, then Sec. 3.2 provides the definition of  $b$  used by Method 1.

#### 3.1. Use only matter vertices

First we use LC in the neighborhood  $N$  defined by the unit ball of  $\|\cdot\|$  centered at  $\mathbf{v}_i$ : if the vertices of an edge  $\mathbf{v}_i\mathbf{v}_j \in T$  are in both matter and  $N$ , then the edge  $\mathbf{v}_i\mathbf{v}_j$  should be matter too. This is done by the following setting of the booleans. Let  $b(\mathbf{v}_i)$  be true iff  $\mathbf{v}_i$  is a vertex of a matter tetrahedron of  $T$ . (i.e. iff  $\mathbf{v}_i$  is a “matter vertex”.) Let  $b(\mathbf{v}_i\mathbf{v}_j)$  be true iff (1)  $\|\mathbf{v}_i - \mathbf{v}_j\| \leq 1$  or (2)  $\mathbf{v}_i\mathbf{v}_j$  is an edge of a matter tetrahedron in  $T$ . Thus every edge  $\mathbf{v}_i\mathbf{v}_j \in T$  should be in matter if  $b(\mathbf{v}_i) = \text{true}$  and  $b(\mathbf{v}_j) = \text{true}$  and  $b(\mathbf{v}_i\mathbf{v}_j) = \text{true}$ . Note that case (1) uses LC and case (2) is straightforward.

Second we explain the relabeling of a freespace tetrahedron  $\Delta \in T$ . If  $\Delta$  is included in an unit ball of  $\|\cdot\|$  and its 4 vertices are matter, then  $\Delta$  should be relabeled matter by LC. If  $\Delta$  is too large to be included in an unit ball, it could be subdivided into parts such that every part (1) is small enough to be included in an unit ball and (2) is the convex hull of matter points. Then  $\Delta$  should be relabeled matter since it is the union of parts which should be matter. The parts could be slices of  $\Delta$ , which are convex hulls of points in the  $\Delta$  edges with true booleans. For efficiency and simplicity, we try to find neither an enclosing unit ball nor a subdivision of  $\Delta$ , we directly relabel matter  $\Delta$  iff its 10 booleans are true.

#### 3.2. Extend the set of matter vertices

Here is the idea: initialize  $b$  (Sec. 3.1), generate a lot of tetrahedra that are not in  $T$  but should be labeled matter following Sec. 3.1, then vertices of  $T$  that are in such a tetrahedron  $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$  should be matter. The vertices  $\mathbf{a}_i$  are in  $T$  and meet  $b(\mathbf{a}_i) = \text{true}$ . The edges  $\mathbf{a}_i\mathbf{a}_j$  meet  $b(\mathbf{a}_i\mathbf{a}_j) = \text{true}$ . Note that  $b(\mathbf{a}_i\mathbf{a}_j)$  is not yet defined if  $\mathbf{a}_i\mathbf{a}_j \notin T$ . In this case, let  $b(\mathbf{a}_i\mathbf{a}_j)$  be true iff  $\|\mathbf{a}_i - \mathbf{a}_j\| \leq 1$ . Then we reset  $b(\mathbf{v}) = \text{true}$  if a vertex  $\mathbf{v} \in T$  is in the convex hull of  $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$  and  $\mathbf{a}_3$ .

In practice we successively consider every vertex  $\mathbf{v} \in T$  such that  $b(\mathbf{v}) = \text{false}$  and only try tetrahedra whose vertices are in the immediate neighborhood of  $\mathbf{v}$ : we enumerate every tetrahedron  $\mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$  such that  $\mathbf{v}\mathbf{a}_i \in T$  and  $b(\mathbf{a}_i) = \text{true}$  and  $\|\mathbf{a}_i - \mathbf{v}\| \leq 1$  where  $0 \leq i \leq 3$ . If the 6 booleans  $b(\mathbf{a}_i\mathbf{a}_j)$  are true and  $\mathbf{v} \in \mathbf{a}_0\mathbf{a}_1\mathbf{a}_2\mathbf{a}_3$ , we reset  $b(\mathbf{v})$  to true.

## 4. IMPLICIT USE OF LOCAL-CONVEXITY

Here we introduce two methods whose principle is simple: generate a lot of sets of freespace tetrahedra included in the 3D Delaunay triangulation  $T$ , then all tetrahedra in a set  $S$  are relabeled matter if there are enough matter tetrahedra that surround  $S$ . Sec. 4.1 presents criteria to evaluate the surrounding and explains the implicit use of LC. Then the two methods are described in Secs. 4.2 and 4.3.

We enforce constraints on the sets such that they are neither too large nor too numerous. All tetrahedra in  $S$  share a common vertex and  $S$  is *strongly connected*: there is a walk in the adjacency graph of tetrahedra in  $S$  linking every tetrahedron in  $S$ . (Tetrahedra are adjacent iff they have a common triangle face.) Thus the expected size of  $S$  is less than 27 [4] and strongly connected components are separately tried.

### 4.1. Surrounding criteria and local-convexity

We need to measure the matter that surrounds  $S$ . Let  $\partial S$  be the *boundary* of  $S$ , i.e. the set of every triangle that is a face of exactly one tetrahedron in  $S$ . Let  $\partial_m S$  be the triangles in  $\partial S$  that are faces of matter tetrahedra (in  $T \setminus S$ ). Let  $c(S)$  be the ratio between the area of  $\partial_m S$  and the area of  $\partial S$ . The area of  $\partial S$  is the sum for every  $\mathbf{abc} \in \partial S$  of the Euclidean norm of  $\mathbf{s} = \frac{1}{2}(\mathbf{b} - \mathbf{a}) \wedge (\mathbf{c} - \mathbf{a})$ . We have  $0 \leq c \leq 1$ . If  $c(S)$  is close to 1,  $S$  is well surrounded by the matter tetrahedra. ( $S$  is a freespace cavity of the matter if  $c(S) = 1$ .)

We define another criterion  $c_{\mathbf{k}}$  that takes into account the scene anisotropy. ( $c$  does not.) The area of the triangle  $\mathbf{abc}$  is generalized by  $\sqrt{\mathbf{s}^T \mathbf{Q} \mathbf{s}}$  where  $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$  is a symmetric positive semidefinite matrix. If  $\mathbf{Q} = \mathbf{k} \mathbf{k}^T$ ,  $\sqrt{\mathbf{s}^T \mathbf{Q} \mathbf{s}}$  is the projected area of  $\mathbf{abc}$  onto a horizontal plane (whose normal is  $\mathbf{k}$ ). We define  $c_{\mathbf{k}}(S)$  by replacing every Euclidean norm of  $\mathbf{s}$  in the expression of  $c(S)$  by  $\sqrt{\mathbf{s}^T \mathbf{k} \mathbf{k}^T \mathbf{s}}$ . We have  $0 \leq c_{\mathbf{k}} \leq 1$ . If  $c_{\mathbf{k}}(S)$  is close to 1,  $S$  is well surrounded by the matter tetrahedra above and below (relatively to the vertical direction  $\mathbf{k}$ ).

In both cases (if  $c(S)$  or  $c_{\mathbf{k}}(S)$  is close to 1), we can use the matter LC in a (non-explicit) neighborhood that includes  $S$ : most tetrahedra in  $S$  are in the convex hull of  $\partial_m S$  and  $S$  should be relabeled matter. These relabeled tetrahedra can be larger than those in Sec. 3 since  $s_h$  and  $s_v$  are unset here.

### 4.2. Method 2

Method 2 extends the Peak Removal operation in [10] to non-manifold surface separating matter and freespace tetrahedra. A set  $S$  is a strongly connected component of the freespace tetrahedra in  $T$  sharing a vertex  $\mathbf{v}$ . Furthermore,  $\mathbf{v}$  is a “peak of  $S$ ”, i.e.  $S$  has a small solid angle with apex  $\mathbf{v}$ . (This angle is the sum for every  $\mathbf{vabc} \in S$  of the solid angle at the vertex  $\mathbf{v}$  of the tetrahedron  $\mathbf{vabc}$ .) In contrast to [10], we can have several sets  $S$  for a same  $\mathbf{v}$  if  $\mathbf{v}$  is a non-manifold vertex.

The relabeling of tetrahedra sharing every vertex  $\mathbf{v} \in T$  is done as follows. First we compute the set(s)  $S$  sharing  $\mathbf{v}$

by a graph traversal in the adjacency graph of the tetrahedra that include  $\mathbf{v}$ . Then every set  $S$  is relabeled matter if its solid angle with apex  $\mathbf{v}$  is less than a threshold and  $c(S)$  is greater than another threshold. We respectively use thresholds  $\pi/2$  and 0.5 in the experiments. The value 0.5 implies that the area between freespace and matter tetrahedra decreases.

### 4.3. Method 3

In contrast to Method 2, Method 3 takes into account the scene anisotropy and tries many more sets  $S$ . For ease of understanding, we start from a simplified version of Method 3 without strong connectivity in Sec. 4.3.1, then we update it with strong connectivity in Sec. 4.3.2. Here we use a confidence measuring that a tetrahedron is freespace: the number of lines-of-sight intersecting the tetrahedron. Other confidence choices are possible but they are not in the paper topic.

#### 4.3.1. Ignore strong connectivity

The tetrahedra in a set  $S$  share a same vertex and have the lowest confidences. There are two reasons for this choice. First it is easy to enumerate all sets: for each vertex  $\mathbf{v} \in T$ , store in a table the freespace tetrahedra in  $T$  including  $\mathbf{v}$  in the increasing order of the freespace confidences, initialize  $S = \emptyset$ , add to  $S$  the  $i$ -th element of the table and increment  $i$  in the table range. Second we want to relabel in priority the tetrahedra that are the most probably falsely-labeled freespace.

Now we detail how to relabel tetrahedra sharing a common vertex  $\mathbf{v} \in T$ . Let  $S_i = S_{i-1} \cup \{\Delta_i\}$  where  $\Delta_i$  is the  $i$ -th tetrahedron of the ordered table and  $S_0 = \emptyset$ . Let  $i_* = \operatorname{argmax}_i c_{\mathbf{k}}(S_i)$ . If  $c_{\mathbf{k}}(S_{i_*})$  is greater than a threshold  $c_0$ , the tetrahedra in  $S_{i_*}$  are relabeled matter. In practice, this process is done for all vertices three times using  $c_0 = 0.6$ .

#### 4.3.2. Use strong connectivity

In Sec. 4.3.2, we use “component” as a shortening of “strongly connected component”. The difference between Method 3 and the simplified method in Sec. 4.3.1 is the following: we compute  $i_* = \operatorname{argmax}_i c_{\mathbf{k}}(S'_i)$  where  $S'_i$  is the component of  $S_i$  that includes  $\Delta_i$ . By doing this, the components of the  $S_i$  are separately tried. We now explain how to compute  $S'_i$ .

The computation of  $S'_i$  is fast thanks to the incremental computation of the components of  $S_i$  from those of  $S_{i-1}$ . Let  $\Delta' \in T$  be one of the four adjacent tetrahedra of  $\Delta_i$ . If  $\Delta' \in S_{i-1}$ ,  $\Delta'$  and  $\Delta_i$  must be in the same component. Thus  $S'_i$  is the union of the set  $\{\Delta_i\}$  and every component  $S'$  of  $S_{i-1}$  including such a  $\Delta'$  (if any). Furthermore, the components of  $S_i$  are  $S'_i$  and those of  $S_{i-1}$  except the set(s)  $S'$ . This is done efficiently thanks to an union-find data structure [15].

## 5. RELABELING FOR PREVIOUS METHODS

We combine our methods and use them to improve [18, 8, 10].

### 5.1. Combine several relabeling methods

We try several combinations of our relabeling Methods 1, 2 and 3 described in Secs. 3, 4.2 and 4.3, respectively. In practice we found that it is always interesting to use 2 as follows: first apply 2, then apply 1 or 3. Indeed, 2 is faster than 1 and 3 and does a part of the job of 1 and 3.

Up to now, falsely-labeled freespace tetrahedra are corrected and falsely-labeled matter tetrahedra are ignored. The latter occurs if the tetrahedra are too acute to be intersected by lines-of-sight. We simply correct this by swapping freespace and matter in Method 2: relabeling freespace the sets of matter tetrahedra (instead of relabeling matter the sets of freespace tetrahedra) with small solid angle if the area between freespace and matter tetrahedra decreases. This method is named  $\tilde{2}$ . We use a more conservative upper limit of the solid angles in this case ( $\pi/100$ ) so that scene thin structures are not removed.

Our experiments needs notations: 2+1 means that we first use 2, then 1, last  $\tilde{2}$ . Similarly, 2+3 means that we first use 2, then 3, last  $\tilde{2}$ . Method  $\tilde{2}$  is used as the end since 1 or 3 can sometimes create falsely-labeled matter tetrahedra. Another combination 2+1+3 is possible but is not in the paper due to space limitation, e.g. if we define the matter tetrahedra of 2+1+3 by the union of those of 2+1 and 2+3. We also write  $2+\tilde{2}$  if only 2 and  $\tilde{2}$  are used. If 2+1 or 2+3 is used to improve a previous surface reconstruction method named X, we write X(2+1) or X(2+3). Here are examples: X=M in Sec. 5.2 and X= $G_i$  in Sec. 5.3.

### 5.2. Preprocessing for a manifold method M

M is the manifold method named M3 in [10]. Its input is  $T$  and a labeling obtained by ray-casting: a tetrahedron is labeled freespace iff it is crossed by a line-of-sight. Its output is another labeling with improved freespace and matter tetrahedra (named outside and inside in [10]) whose boundary is a manifold surface. Our corrections are straightforward in a preprocessing of M, but they remove the manifold property if they are in a postprocessing of M. Since the efforts of M are useless in the latter, we only consider the former in the paper.

### 5.3. Postprocessing for graph-cut methods $G_1$ and $G_2$

$G_1$  is the graph-cut method in [18]. First it builds  $T$  and its adjacency graph, whose weights are computed from the lines-of-sight and their intersections with the Delaunay triangles and the tetrahedron geometry (using  $\lambda_{qual} = 1$  and  $\alpha_{vis} = 1$ ). Then the final tetrahedron labeling is computed by a minimum  $s$ - $t$  cut of the graph.

Our corrections can be a preprocessing or a postprocessing of  $G_1$ . In the former, the corrections are applied to an initial labeling (obtained by ray-casting in  $T$ ) and  $t$ -weights of the graph increase by a value if we relabel their corresponding tetrahedra. This modifies the cost function, which is minimized by graph-cut. The latter is straightforward. The former

requires an additional parameter (the value) and is left as future work. Thus we only consider the latter in the paper.

$G_2$  is the graph-cut method in [8], which improves thin structures compared to  $G_1$ . (We use  $\sigma = 0$ ,  $\delta = 0.5$  and  $\beta/\alpha_{vis} = 32$ .) Similarly, our corrections are done in a post-processing of  $G_2$ .

## 6. EXPERIMENTS

Sec. 6.1 experiments our corrections alone and Sec. 6.2 shows improvements that they provide if they are used in a postprocessing of graph-cut methods or in a preprocessing of a manifold method. These processing choices and the notations of the methods are detailed in Sec. 5. All methods start from the same 3D Delaunay triangulation  $T$  and end by the same Laplacian smoothing of the surface for fair comparisons.

The sparse input point cloud is computed as in Sec. 7.1 of [10] from two  $2496 \times 2496$  videos at 30Hz taken by a Garmin Virb 360 camera. This camera is mounted on the top of a car using a small mast. The trajectory is 6.7km long in an urban scene. There are 5.5M vertices in  $T$ , which are reconstructed from 6.5k keyframes selected in the videos by Structure-from-Motion (using central and global shutter approximations). Fig. 2 shows the 360 camera, a keyframe and a bottom view of the vertices in  $T$ .

### 6.1. Only use corrections

We compare the initial labeling by ray-casting and its relabeling by our methods  $2 + \tilde{2}$ ,  $2 + 1$  and  $2 + 3$ . The comparison is interesting since it helps to understand the preprocessing of method M (which is this relabeling). Furthermore, this shows the performance of our relabeling in a context that is more difficult than that of the postprocessing of methods  $G_x$ . (The initial labeling is more corrupted than that computed by  $G_x$ .)

Fig. 3 shows labeling differences near a 5-way crossroad by projecting the freespace tetrahedra (in black) to the horizontal plane. We see that  $2 + \tilde{2}$  removes the largest freespace tetrahedra that are inconsistent with the geometry of the urban corridors. (Note that the camera moves in the 5 main roads). At this large scale, we also see that 1 and 3 (involved in methods  $2 + 1$  and  $2 + 3$ ) remove other freespace tetrahedra. Fig. 4 shows differences at small scale near two scene components: both  $2 + 1$  and  $2 + 3$  complete thin structures such as trunks and traffic signs in comparison to  $2 + \tilde{2}$ .

We see that our methods change topology: trunk components are connected, tunnels (i.e. holes) in a traffic sign are filled. So we evaluate the topology changes. Let  $\beta_0$  and  $\beta_1$  be the numbers of connected components and tunnels of the matter [5]. Tab. 1 shows that our corrections have advantages: they greatly reduce  $\beta_1$  and improve the surface manifoldness. Other experiments (not in the paper) show that  $\beta_0$  always decreases if  $\tilde{2}$  is unused, but this generates large spurious matter tetrahedra and the  $\beta_1$  reduction is smaller than before.



Fig. 2. One keyframe, the Garmin Virb 360 camera, and a bottom view of the vertices of  $T$ .



Fig. 3. Large scale view of our corrections. From left to right: vertices of  $T$ , initial labeling by ray-casting, results of the correction methods  $2 + \tilde{2}$ ,  $2 + 1$  and  $2 + 3$ . Matter is white and freespace is black.

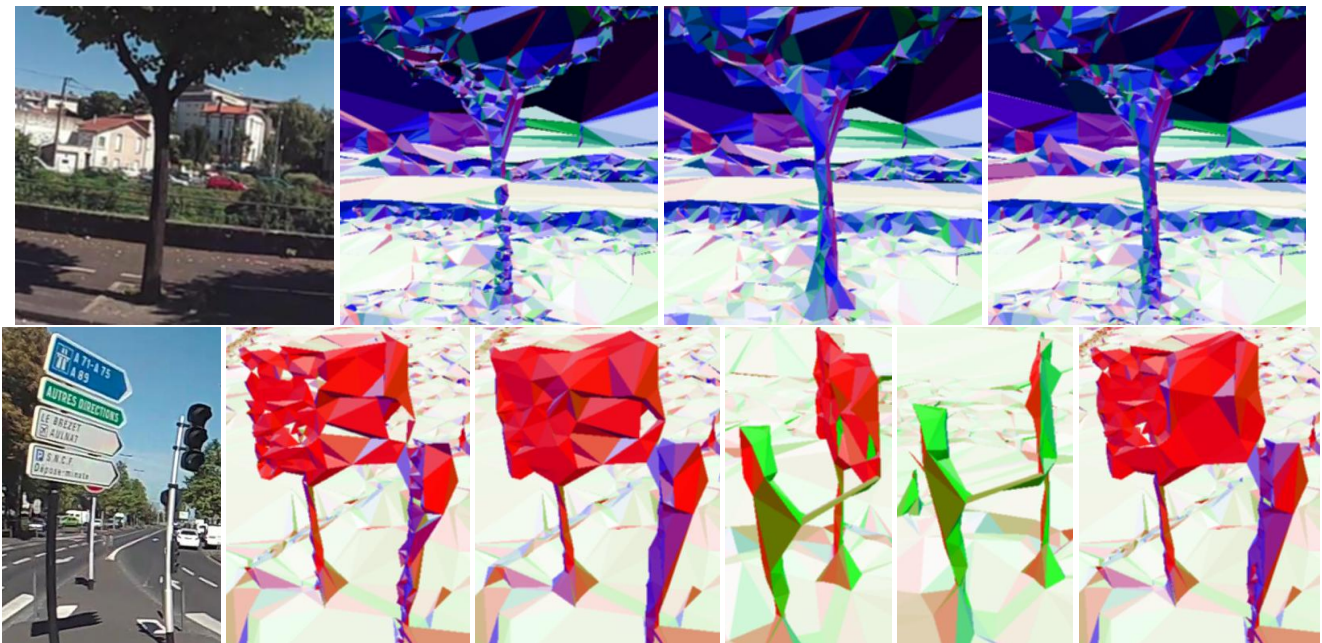
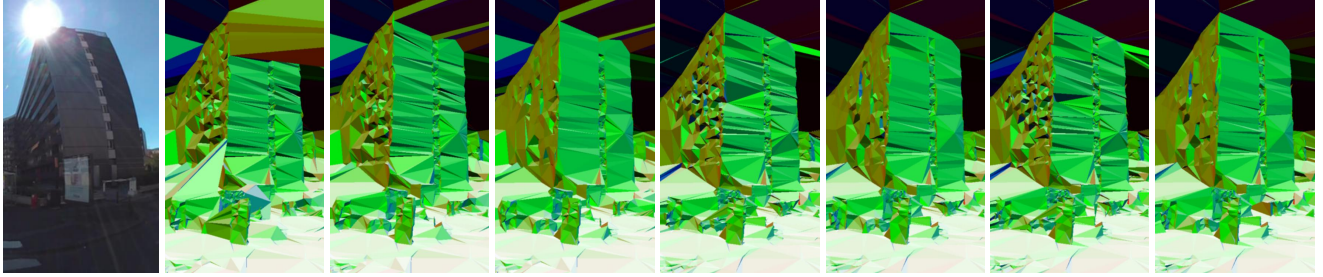


Fig. 4. Small scale views of our corrections. From left to right: input image, results of the correction methods  $2 + \tilde{2}$ ,  $2 + 1$  and  $2 + 3$  (best viewed in colors and by zooming in). Vertical triangles are red-green-blue and horizontal triangles are white/black. Note that  $2 + 1$  has 3 successive views at the bottom line to show the thickness of the traffic sign with few vertices.



**Fig. 5.** Apply our corrections with previous methods to a building (best viewed in colors and by zooming in). From left to right: input image, results of methods M, M(2+1), M(2+3),  $G_1$ ,  $G_1(2+3)$ ,  $G_2$ ,  $G_2(2+3)$ .

Method	ray-casting	2+2	2+1	2+3
comp. number ( $\beta_0$ )	4.8k	5.5k	1.5k	5.1k
tunnel number ( $\beta_1$ )	185k	105k	18k	31k
non-manifold vertex	31%	9.8%	5.0%	4.4%
freespace	57%	51%	38%	49%
computation time	102s	22s	38s	62s

**Table 1.** Topology changes provided by our corrections. See also the percentage of the freespace tetrahedra in  $T$  and the computation time (using one core of a standard laptop).

## 6.2. Use corrections with surface reconstruction methods

Fig. 5 compares the methods M,  $G_1$ ,  $G_2$  and their corrected versions on a building of our scene. The surface normal of the low textured face of the building is always noisy without our corrections. The 2+3 correction improves the normal of all methods, which should be constant in this face. In contrast to this, the normal improvement of the 2+1 correction is negligible in all cases. (The figure only shows M(2+1).) There are two reasons. First 2+1 uses an upper limit on the edge size of tetrahedra and 2+3 does not. Second the edge size is large since the building is in the background and has a low number of points (due to low texture). We also note that both 2+1 and 2+3 remove two blunders of M.

In Fig. 6, we focus the comparisons on sharp edges of a facade of our scene. Here a sharp edge is an intersection between two adjacent and orthogonal planar regions of the scene. This often occurs in urban scenes: we mostly observe  $2 \times 2$  kinds of sharp edges: horizontal or vertical, convex or concave. We say that an edge in a surface is convex (respectively, concave) if its neighborhood in the matter side is convex (respectively, concave). Fig. 6 shows that both 2+1 and 2+3 corrections improve the reconstructions of the vertical sharp edges if they are convex. If the sharp edges are concave, the sharpness is smoothed by 2+1 which tends to fill the concavity in the freespace side. This sharpness smoothing can also occur for 2+3 but it is weaker than that of 2+1. These comments hold for M,  $G_1$  and  $G_2$ .

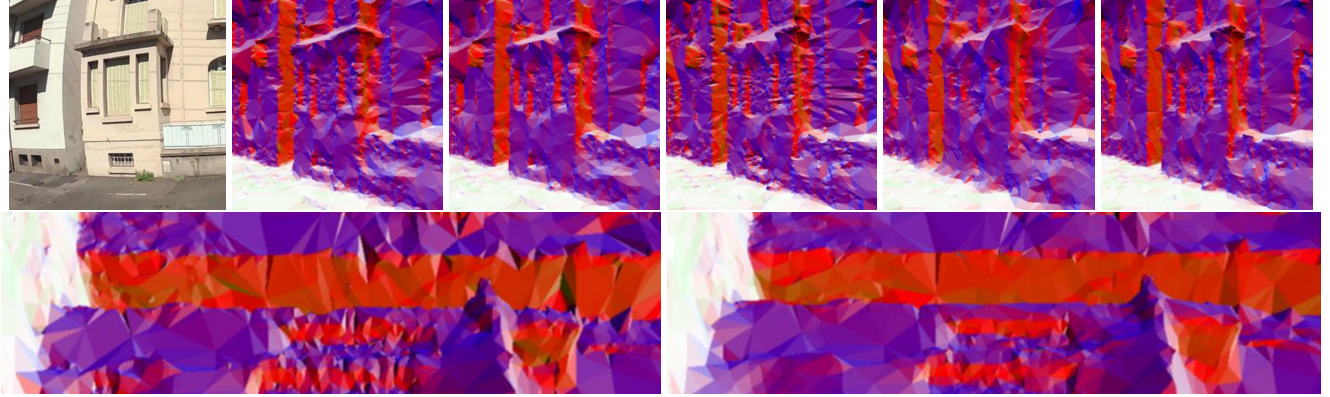
Fig. 7 compares M,  $G_1$ ,  $G_2$  and their corrected versions on thin 1D structures of our scene. In the top row, the result of M(2+1) is better than that of M and M(2+3), since M(2+1) provides the most complete thin 1D structures. In the bottom row, both  $G_1(2+1)$  and  $G_1(2+3)$  complete the thin 1D structures of  $G_1$ . However we see that  $G_1(2+1)$  has a growing bias: the thickness of two thin 1D structures near the central post is larger than the true one, according to the original image of the scene in the top row.

We also do quantitative evaluations of all methods applied to the synthetic urban dataset in [10] (a piecewise planar scene with real textures). The  $T$  vertices and the lines-of-sight are estimated by Structure-from-Motion applied to synthetic videos taken by a 360 camera whose trajectory is a 621m long closed loop (Fig. 8). Tab. 2 shows that our corrections improve the previous surface reconstruction methods using both geometric and topological criteria. Geometric error and triangle number decrease thanks to our corrections. Furthermore a lot of spurious tunnels are filled (topological noise decreases) and the GC surface manifoldness is improved in most cases.

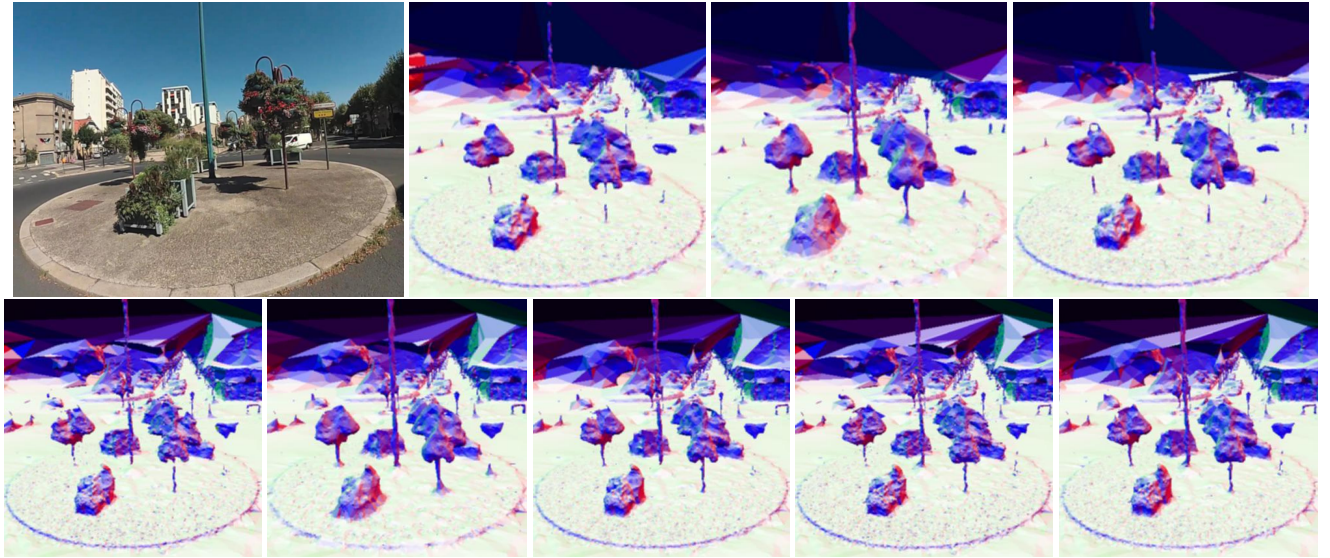
Last we do experiments similar to those of Tab. 2 for our real scene, except that there is no ground truth here. The surface topology of the GC methods is globally improved by our corrections since the percentage of non-manifold vertices in the surface always decreases (from 12% to 67%) and the tunnel number  $\beta_1$  is divided by 3-5. The  $\beta_1$  of method M decreases by about 7%. The triangle number decreases by about 30% using the 2+1 correction or 9-15% using the 2+3 correction. (This is intuitive: a surface with a lot of tunnels and small concavities has more triangles than a correction of this surface such that most tunnels and concavities are filled by LC.) The component number  $\beta_0$  decreases up to 73% (2+1 correction) and increases up to 25% (2+3 correction).

## 7. CONCLUSION

We introduce a new shape constraint, local-convexity, to improve previous surface reconstruction methods based on 3D Delaunay triangulation. This constraint is reinforced by our corrections in a postprocessing or a preprocessing depending



**Fig. 6.** Apply our corrections with previous methods to a facade (best viewed in colors and by zooming in). Top (from left to right): input image, M, M(2+3),  $G_1$ ,  $G_1(2+1)$ ,  $G_1(2+3)$ . Bottom:  $G_2$  and  $G_2(2+3)$ .

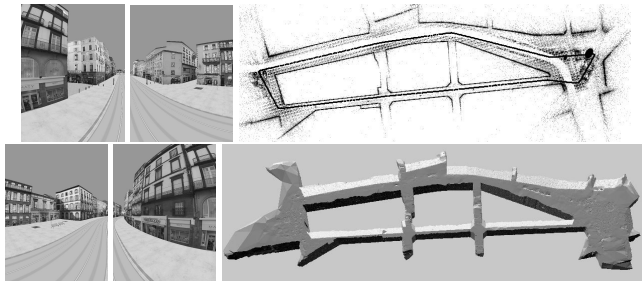


**Fig. 7.** Apply our corrections with previous methods to a traffic roundabout with urban vegetation and post (best viewed by zooming in). Top (from left to right): input image, M, M(2+1), M(2+3). Bottom:  $G_1$ ,  $G_1(2+1)$ ,  $G_1(2+3)$ ,  $G_2$  and  $G_2(2+3)$ .

Method	$G_1$	$G_1(2+1)$	$G_1(2+3)$	$G_2$	$G_2(2+1)$	$G_2(2+3)$	M	M(2+1)	M(2+3)
80% fractile of error (cm)	28	26	26	28	26	26	24	22	22
90% fractile of error (cm)	94	84	88	94	82	88	68	58	56
triangle number	1.26M	0.86M	1.01M	1.27M	0.82M	0.98M	1.09M	0.72M	0.922M
component number ( $\beta_0$ )	46	28	45	27	18	29	2	3	4
tunnel number ( $\beta_1$ )	834	187	205	1737	247	351	32	17	18
non-manifold vertex	0.98%	1.24%	0.27%	1.58%	1.18%	0.34%	0%	0%	0%

**Table 2.** Quantitative improvements provided by our corrections of previous surface reconstruction methods applied to a synthetic urban dataset. There are not only accuracy improvements (error fractiles decrease) but also topology improvements (the number of spurious tunnels decreases) and mesh simplification. The true scene has  $\beta_1 = 3$  tunnels and  $\beta_0 = 1$  component.





**Fig. 8.** Synthetic urban dataset. Left: images at a location. Right: top views of Structure-from-Motion and M surface.

on the kind of surface reconstruction method to which it is added. Since the input point cloud is sparse, we do not try to fit primitives such as planes but enumerate and select (local packs of) tetrahedra for relabeling, mostly from freespace to matter. Experiments show improvements for an urban scene acquired by terrestrial imagery: remove freespace concavities and tunnels of the matter tetrahedra, improve sharp edges if they are convex, complete thin structures, reduce both geometric error and surface complexity. Future work should avoid oversmoothing of sharp edges if they are concave, vary the privileged direction along which we obtain most improvements, and investigate other ways to reinforce local-convexity.

## 8. REFERENCES

- [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 tietze-nakajima - 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] Z. Liu, D.W. Jacobs, and R. Basri. The role of convexity in perceptual completion: beyond good continuation. *Vision Research*, 3, 1999.
- [12] 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.
- [13] S. Oesau, F. Lafarge, and P. Alliez. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS journal of photogrammetry and remote sensing*, 90, 2014.
- [14] S. Ohrhallinger, S. Mundur, and M. Wimmer. Minimizing edge length to connect sparsely sampled unstructured point sets. *Computer and Graphics*, 37(6), 2013.
- [15] R. Sedgewick and K. Wayne. *Algorithms (fourth edition deluxe)*. Addison-Wesley, 2016.
- [16] S.C. Stein, F. Forgotter, M. Schoeler, J. Papon, and T. Kulvicius. Convexity based objet partitioning for robot applications. In *ICRA*, 2014.
- [17] O. Veksler. Star shape prior for graph-cut image segmentation. In *ECCV*, 2008.
- [18] H. Vu, P. Labatut, J. Pons, and R. Keriven. High accuracy and visibility consistent dense multiview stereo. *IEEE PAMI*, 34, 2012.
- [19] C. Wu, S. Agarwal, B. Curless, and S.M. Seitz. Schematic surface reconstruction. In *CVPR*, 2012.

## A. APPENDIX

The choice of  $s_v$  and  $s_h$  (in Sec. 3) is a trade-off: the relabeling is negligible if they are too small, shape details can be lost if they are too large. We first estimate a size  $\epsilon$  for the level of detail that is expected for the reconstructed shape, then multiply it by numbers about 1 to obtain  $s_v$  and  $s_h$ . Let  $s_h = 2\epsilon$  and  $s_v = 3\epsilon$  where  $\epsilon$  is the median of the horizontal lengths (used in Eq. 1) of the edges of the input matter tetrahedra.