Simplification of 2D Polygonal Partitions via Point-line Projective Duality, and Application to Urban Reconstruction

Julien Vuillamy, André Lieutier, Florent Lafarge, Pierre Alliez

DOI: 10.1111/cgf.14511

COMPUTER GRAPHICS *forum*
Volume 0 (2022), number 0 pp. 1–15

Simplification of 2D Polygonal Partitions via Point-line Projective Duality, and Application to Urban Reconstruction

J. Vuillamy,[1,2,3] A. Lieutier,[3] F. Lafarge[1,2] and P. Alliez[1,2]

[1]Université Côte d'Azur, Nice, France
florent.lafarge@inria.fr,pierre.alliez@inria.fr
[2]Inria Sophia Antipolis—Méditerranée, Valbonne, France
andre.lieutier@gmail.com
[3]Dassault Systèmes Provence, Aix-en-Provence, France
andre.lieutier@gmail.com

Abstract

We address the problem of simplifying two-dimensional polygonal partitions that exhibit strong regularities. Such partitions are relevant for reconstructing urban scenes in a concise way. Preserving long linear structures spanning several partition cells motivates a point-line projective duality approach in which points represent line intersections, and lines possibly carry multiple points. We propose a simplification algorithm that seeks a balance between the fidelity to the input partition, the enforcement of canonical relationships between lines (orthogonality or parallelism) and a low complexity output. Our methodology alternates continuous optimization by Riemannian gradient descent with combinatorial reduction, resulting in a progressive simplification scheme. Our experiments show that preserving canonical relationships helps gracefully degrade partitions of urban scenes, and yields more concise and regularity-preserving meshes than common mesh-based simplification approaches.

Keywords: polygonal partition, simplification, projective duality, optimization, urban reconstruction

CCS Concepts: • Computing methodologies → Mesh models; Reconstruction

1. Introduction

Space partitioning data structures, such as adaptive grids, triangulations or polyhedral partitions, often play a central role in geometric modelling by bridging the gap between unorganized data measurements and standardized algorithms. In many applications, however, they are often overly complex and cannot properly capture the geometric regularities contained in the observed scenes. These limitations are particularly relevant in urban reconstruction where expected output models must be both concise and regular. Indeed, a standard representation for urban scenes in Geographic Information Systems (GIS) uses the support of two-dimensional piecewise linear functions to model buildings. To achieve this expected modelling simplicity, existing methods commonly operate by either consolidating the data measurements with resampling and filtering approaches, or by simplifying the output models via remeshing. The former approach often relies upon heuristics and lacks generality, whereas the latter has difficulty generating concise meshes without altering the geometric accuracy. To our knowledge, none of the existing methods study the possibility of simplifying the space partitioning data structure itself.

We address the problem of simplifying an input 2D partition while keeping a satisfactory tradeoff between low complexity, high regularity and high fidelity to the input partition. We deal with 2D partitions in which the cells are convex polygons and the vertices are located where line segments either intersect or end. Sets of collinear line segments are frequent as they originate from long linear structures detected in urban scenes, such as facades or roads. In our context, high fidelity refers to preserving the input polygons and collinearities between line segments. As the partitions are defined by points and line segments, this justifies the proposed duality-based approach. Figure 1 illustrates the goal of our work.

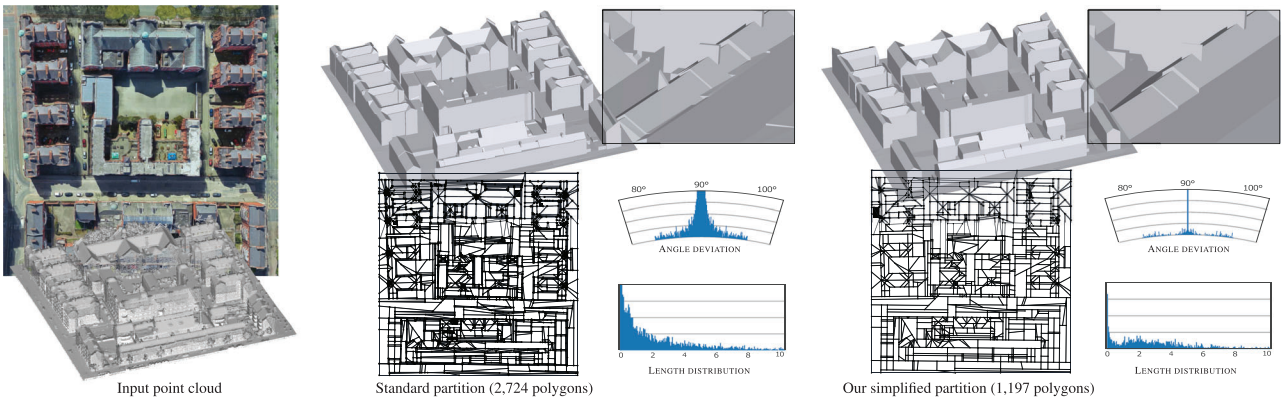

Figure 1: *Decomposing scenes into planimetric partitions of 2D polygons is a popular approach in urban reconstruction. However, common partitioning schemes produce overly complex and unregularized partitions. Our approach simplifies such partitions while preserving their fidelity to data and enforcing some geometric regularities contained in the scene (the histograms depict the angle deviation around* 90° *and the distribution of edge lengths in the partitions). The output model, reconstructed by lifting the partition in 3D, is both more concise and regular (see closeups). Data from the 2015 LiDAR survey of Dublin [LAV*17].*

2. Related Work

Our review of previous work covers urban reconstruction methods based on 2D space partitions, as well as algorithms based on mesh simplification and regularity detection.

2D space partitions and urban reconstruction. 2D space partitioning data structures have been used routinely for urban reconstruction [MWA*13]. Commonly constructed from unorganized measurement data such as point clouds or images, they are often lifted in 3D to represent urban objects such as buildings or facades with a simple disk-like topology.

Popular Delaunay triangulations yield dense partitions that must be simplified in order to produce concise output meshes. Reducing the size of the triangulation is achieved, for instance, by contracting edges with an optimal transport approach [DGCSAD11], splitting and merging triangles [GS97] or inserting and removing vertices within a spatial point process framework [FLBA20]. However, such operations alter the alignment of the triangulation with the input data and prevent the subsequent reconstruction step from producing meshes that are both concise and geometrically accurate [BRG15]. Voronoi diagrams [DL16] or adaptive grids [ZN10] also exhibit this weakness. The former does not allow complex buildings to be accurately reconstructed by a few Voronoi cells lifted in 3D. The latter can only reconstruct buildings accurately with dense grids.

2D partitions of polygons offer a better tradeoff between data fidelity and conciseness on urban objects. For instance, the roof section of a building can be ideally abstracted by a single polygon. Such partitions can be created by kinetic simulations that propagate line-segments aligned with the data [Gui04, BL18] or by the vectorization of region maps [AS17]. However, their use to reconstruct buildings [ZBKB08, BL19] or facade objects [RBDD18] is effective only when the partitions are simple and preserve the geometric regularities inside objects and scenes, such as parallelism, orthogonality or symmetry. In their simplest forms, the problems of simplifying such partitions while controlling the deviation from the initial configuration are similar to NP-hard problems of 2D point cloud simplification such as geometric unit disk cover or minimum dominating set on unit disk graphs [MBHI*95]. For this reason, as is already the case for these simpler problems, we do not expect the existence of an efficient algorithm for a global minimum formulation of the problem of simplifying these 2D partitions. Discrete approaches have been proposed, based on merging and splitting polygons [LLM20]. However, they do not make it possible to simplify a 2D polygon partition while also improving its geometric regularity.

Mesh simplification. An alternative approach consists of simplifying the output (dense) mesh instead of the underlying space partition. Common algorithms contract edges until a target mesh complexity is reached [GH97, Lin00]. Such mesh decimation approaches can be extended to preserve piecewise-planar structures of objects by guiding edge contractions with pre-detected planar shapes [SLA15]. Such planar shapes can also be directly assembled into a concise mesh when their adjacency graph is correctly extracted [CAD04]. Among specialized approaches for simplifying buildings, Kada [Kad07] proposes a method for decomposing a building into structural parts before replacing them with 3D idealized primitives. Bredif *et al*. [BBPDM08] utilize a kinetic model to make facets more consistent with the data. Verdie *et al*. [VLA15] assemble planar shapes that are filtered to produce output meshes with different levels of detail. These approaches yield good results when dense meshes are geometrically and topologically accurate. However, this condition is rarely met in practice when dealing with defect-laden data.

Detection of geometric regularities. Our work is also closely related to the problem of regularity detection. Geometric regularities typically include parallelism, orthogonality, coplanarity [LWC*11] and symmetry [PMW*08, MPWC13]. Their detection is usually operated from input 3D data through global analysis of local geometric features. In contrast, our goal is to regularize polygon partitions without a preliminary detection from the input data. This task

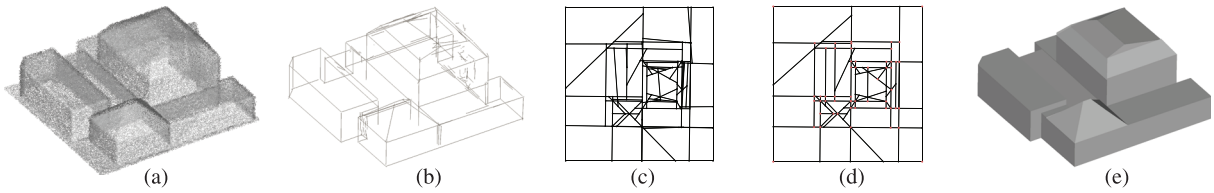

Figure 2: *Overview of our method. After extracting 3D line-segments (b) from an input point cloud (a) and projecting them onto the horizontal plane to form a polygonal partition (c), our method simplifies the partition while capturing some geometric regularities inside (d). The resulting partition is then lifted in 3D to create a concise polygonal surface mesh (e).*

typically requires an energy minimization framework with priors that encourage regularities, as proposed by Monszpart *et al*. for extracting planar primitives from point clouds [MMBM15]. For reconstructing buildings, the most common regularities are orthogonality and parallelism [ZN12, HK12]. These two regularities are often imposed by construction with a Manhattan-World (or Polycube) geometry in which all elements are either orthogonal or parallel [CY00, SRF*14]. Unfortunately, such a geometry is only relevant for some specific urban scenes.

Homogeneous coordinates and point-line duality. Projective geometry has become the *de facto* standard representation for computer graphics and vision. The use of homogeneous coordinates [Her92, NDW93] allows a common representation 4×4 matrices of general 3D isometries, as well as projective transformations induced by an eye-centred perspective. It has found widespread applications [PW09] in the form of epipolar geometry for multi-view reconstruction problems [HZ06] or NURBS for computer-aided design [Far99]. In the latter case, projective representation allows an exact modelling of parametric conics curves and derived parametric surfaces, such as cylinders, spheres and, more generally, surfaces of revolution through (piecewise) polynomial patches in homogeneous coordinates.

The point-line duality is also a common tool for theoretical analysis of geometric problems [CGL85, Gun17]. However, to the best of our knowledge, using this duality for a continuous partition simplification scheme has not been attempted in previous work.

3. Overview

The proposed method generates models compatible with GIS from point clouds typically generated from photogrammetry or laser scanning. It uses an intermediate 2D arrangement to reconstruct buildings as two-dimensional piecewise-linear functions over this arrangement. This assumption of 2.5D geometry, very common in urban reconstruction [MWA*13], can be applied to the large majority of buildings and is a good compromise between fidelity and simplicity of a building's representation. Indeed, this modelling paradigm does not apply to architecturally complex buildings, which require free-form reconstruction methods, but we expect these buildings to form a very small fraction of the total number of buildings in large urban scenes. In addition, some details such as overhangs cannot be represented. However, this level of detail is often not required for GIS applications. In fact, the proposed output reconstruction complements the LOD2 formalism of CityGML [Sta12] widely used in GIS, which models buildings as prismatic blocks with differentiated roof structures.

The next sections of this paper follow the main steps of our reconstruction method, illustrated in Figure 2. Sections 4 and 5, which describes the extraction of 3D line segments from the input point cloud and the creation of a 2D polygonal partition from these line segments, highlight important properties of the arrangements that have guided the conception of the simplification method. Section 6 is the key contribution of our work, formulating a partition simplification scheme that maintains a satisfactory balance between low complexity, high regularity and high fidelity. The last step of the reconstruction process consists of lifting the resulting partition in 3D to create a concise polygonal surface mesh (Section 7). Finally, Section 8 is dedicated to describing the results and comparing them with other simplification methods.

4. Extraction of 3D Line-segments

Planar primitives and their adjacency relationships are first detected in the original point cloud using a k-nearest neighbours algorithm and a region-growing approach [FTK14, HB12, RVDHV06]. For all adjacent planes, intersection lines are computed and 3D segments are generated from the points located near each intersection line. Because missing segments are a lot more detrimental to the reconstruction pipeline than adding duplicate or unnecessary segments, detection parameters are chosen to perform over-detection. Note that, when available, this detection can also be augmented by existing GIS information such as building footprints.

5. Kinetic Cell Arrangement

In order to obtain a parsimonious two-dimensional polygonal partition from projected segments, we now describe a method coined kinetic framework [Gui04, BL18]. By extending all line segments, intersection events are used to decide whether to stop one of the two lines forming the intersection. More specifically, for a segment S, any point I on its supporting line can be assigned a *Time of Arrival*, which equates to the distance to the segment for a constant unit speed movement:

$$T_S(I) = d(I, S)$$

The kinetic framework can be seen as a set of rules dictating the segment's progression. In its simplest form, the progression of a segment in one direction is stopped at an intersection event when

it arrives last on the intersection. For instance, a segment S_1 will be stopped at its intersection I with S_2 if $T_{S_1}(I) > T_{S_2}(I)$. All intersection events I' concerning S_1 in the same direction as I and verifying $T_{S_1}(I') > T_{S_1}(I)$ can then be discarded, as segment S_1 will not cross segment S_2. Additional rules can be designed to improve the connectivity of the arrangement, such as setting a custom speed for each segment, allowing a fixed number of crossings or even allowing a crossing if the time of arrival to the next intersection is lesser than a threshold value.

Arrangements resulting from the kinetic framework exhibit substantially fewer intersections than the complete line arrangement containing $\mathcal{O}(n^2)$ intersections. They also preserve details, as small segments are kept in the arrangement but are likely stopped by larger features and therefore will only have a local impact on the arrangement. Finally, initial segments spanning large distances are often decomposed into multiple smaller segments in the arrangement, while still sharing the same supporting line. As will be shown later, this is a key insight for the design of our proposed simplification method.

6. Simplification of 2D Partitions

The following section introduces an optimization method for simplifying 2D partitions. We start the discussion by justifying our choice of using homogeneous line coordinates within the optimization scheme (Section 6.1). An energy on these line coordinates is then described as a trade-off between fidelity to the initial partition and simplification objectives (Section 6.2). We then detail the computation required to perform a gradient descent step (Section 6.3). In particular, we choose to derive a Riemannian metric between line coordinates that better describes the distance between lines for our problem. Finally, a global algorithm is presented (Section 6.4), which interlaces continuous gradient descent iterations with discrete simplification operations.

6.1. Projective duality: line movement versus point movement

We use the point-line projective duality to describe a partition. Each element is described as a 3D vector representing:

- The coefficients (a, b, c) of an oriented line $ax + by + c = 0$;
- The homogeneous coordinates (x, y, w) for a 2D point $(\frac{x}{w}, \frac{y}{w})$.

This duality is especially visible in the symmetry of the roles played by points and lines in the line equation written as an inner product:

$$\begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} = 0$$

A comprehensive description of projective duality can be found in Berger [Ber09].

We summarize two possible representations of 2D partitions, dual to one another in projective geometry:

- *Point representation*: points of the partition are given explicit coordinates and lines are implicitly defined from two of those points. This representation makes it possible to encode in the combinatorial structure the situation where multiple lines share the same point, but not the situation where a single line supports more than two points.
- *Line representation*: lines of the partition are given explicit coordinates, and points are implicitly defined as the intersection of two of those lines. This dual representation, symmetrically, encodes in the combinatorial structure the situation where a line carries multiple points but not the dual configuration for which more than two lines intersect at the same point.

In terms of optimization, dealing with point coordinates is simpler than line coordinates, as the Euclidean metric associated with the distance between 2D points is natural. Optimizing via line coordinates is slightly more involved as no natural metric describes a distance between lines, which is invariant under arbitrary Euclidean isometries. Drawn to this apparent simplicity, we tried using the point representation in a first attempt. However, two arguments justify the choice to optimize lines instead of points in our specific context and were confirmed by this first experiment.

First, the initial data for our problem consist of a set of detected line segments, while points are only secondary data constructed from the initial segments by the kinetic framework. Measuring data fidelity with respect to the initial segments seems therefore more natural to our problem than data fidelity with respect to point coordinates.

Second, by construction, many segments of the partition exhibit collinearity relationships that ought to be preserved for the sake of keeping the partition as simple as possible. In a point-based representation, this requires optimizing under collinearity constraints. However, as depicted in the inset, configurations close to edge collapses, which would be our objectives in a point-based representation, are also configurations where collinearity constraints are unstable: constraining three points to be aligned when two of them are indistinguishable from one another is not a well-posed problem. In our first experiments, these frequent configurations resulted in numerical instabilities whose robust treatment was problematic. In contrast, in the line representation model, these collinearities are preserved in the structure and the robustness issues can be soundly managed.

The homogeneous line coordinates are initialized via Euclidean normalization, also referred to as its normal form: $a^2 + b^2 = 1$. Although the *spherical* normalization ($a^2 + b^2 + c^2 = 1$) is often used for its ability to represent all projective lines, our problem does not require representing the line at infinity, so we adhere to Euclidean normalization. We will further justify this choice when giving a geometrical meaning to the regularity term of our optimization energy. Figure 3 depicts the unit cylinder structure of the line coordinates with Euclidean normalization.

6.2. Energy formulation

We denote by $\mathcal{L} = (L_1, \ldots, L_n) \in \mathbb{R}^{3n}$ the set of lines, represented in their normal forms. Our simplification problem is formulated as a trade-off between a fidelity term E_{fidelity}, describing the attachment of the partition to the initial configuration, and complexity

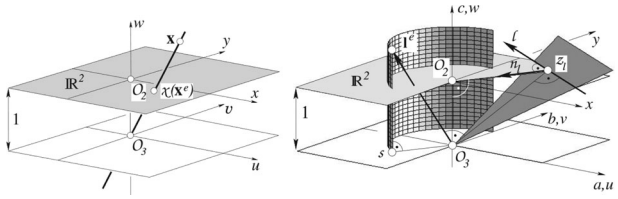

Figure 3: *Illustration of Euclidean normalizations. Left: the Euclidean normalized coordinates of a projective point* $\mathbf{X}$ *correspond to the intersection of the* $w = 1$ *plane and the line passing through* O_3 *and* $\mathbf{X}$*. Right: the Euclidean normalized coordinates* l^e *of the projective line* l *lies on the cylinder verifying* $a^2 + b^2 = 1$*. Image taken from Gerke [Ger17].*

terms $E_{concurrent}$ and $E_{orthogonality}$, respectively, measuring edge collapses and orthogonality objectives on the lines of the partition:

$$E(\mathcal{L}) = E_{fidelity}(\mathcal{L}) + \lambda_1 E_{concurrent}(\mathcal{L}) + \lambda_2 E_{orthogonality}(\mathcal{L}) \quad (1)$$

We now provide details on each term of this objective function.

Fidelity term. As mentioned previously, no metric defines a distance between lines, which is invariant under Euclidean isometries. Indeed, the only discrepancy measures on lines that are preserved by Euclidean isometries are the angle between two lines and the distance between two parallel lines.

In our problem, we have the additional information that each line L is associated with an initial detected segment $S = ((x_0, y_0), (x_1, y_1))$. This can be used to define a distance from the initial configuration. Consider a point $P = (x, y)$ on a line L parameterized by (a, b, c).

$$ax + by + c = 0 \quad (2)$$

Under line movement $\delta L = (\delta a, \delta b, \delta c)$, the squared distance of the point P to the line $L + \delta L$ can be written as:

$$\begin{aligned} d(P, L + \delta L)^2 &= (\delta a \cdot x + \delta b \cdot y + \delta c)^2 \\ &= \delta L^t \begin{pmatrix} x^2 & xy & x \\ yx & y^2 & y \\ x & y & 1 \end{pmatrix} \delta L \end{aligned}$$

This quantity measures the first-order approximation of the squared distance between point P and line $L + \delta L$, which is exact only when $L + \delta L$ satisfies the Euclidean normalization $(a + \delta a)^2 + (b + \delta b)^2 = 1$.

For each line L associated with a segment $S = ((x_0, y_0), (x_1, y_1))$, we consider the sum of these approximated squared distances between the line $L + \delta L$ with the two initial segment endpoints:

$$\frac{1}{2} \begin{bmatrix} x_0^2+x_1^2 & x_0y_0+x_1y_1 & x_0+x_1 \\ x_0y_0+x_1y_1 & y_0^2+y_1^2 & y_0+y_1 \\ x_0+x_1 & y_0+y_1 & 2 \end{bmatrix}$$

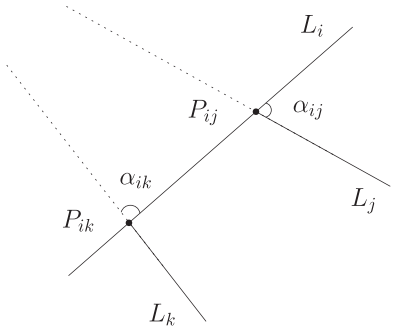

Figure 4: *Illustration of a detected regularization configuration: three lines* L_i, L_j, L_k *form two intersections in the partition.*

Note that by definition, the kernel of this matrix contains all homogeneous coordinates of the line passing through the points (x_0, y_0) and (x_1, y_1).

The quadratic form associated with this positive semi-definite matrix offers a good balance between the translation and rotation movements depending on the segment vertex locations: the rotation of a line associated with a large segment will be more penalized than the rotation of the same line associated with a shorter segment. However, in the extreme case where the segment used for the definition of the quadratic form has zero length, the kernel of the matrix contains all homogeneous coordinates of lines that pass through this unique point $(x_0, y_0) = (x_1, y_1)$. In optimization applications, this means lines associated with very small segments can freely rotate, which can cause numerical instabilities. We stabilize our quadratic form by adding a small penalty $l_{\min}^2$ to the rotation part of all matrices and finally define the matrix M_i for each line L_i in $\mathcal{L}$:

$$M_i = \frac{1}{2} \begin{bmatrix} x_0^2+x_1^2+l_{\min}^2 & x_0y_0+x_1y_1 & x_0+x_1 \\ x_0y_0+x_1y_1 & y_0^2+y_1^2+l_{\min}^2 & y_0+y_1 \\ x_0+x_1 & y_0+y_1 & 2 \end{bmatrix} \quad (3)$$

where $((x_0, y_0), (x_1, y_1))$ denotes the segment associated with the line L_i, and $l_{\min}$ is a minimal length at which we consider segments to be relevant.

The fidelity term on all lines L_i of the arrangement is then defined as follows:

$$E_{fidelity}(\mathcal{L}) = \frac{1}{2} \sum_{i=1}^{n} L_i^t M_i L_i \quad (4)$$

Regularization terms. We consider triplets of lines (L_i, L_j, L_k) where L_i, L_j and L_i, L_k form intersection points in the partition, respectively, denoted by P_{ij} and P_{ik}. We define the following quantity D_{ijk}:

$$D_{ijk} = |\det(L_i, L_j, L_k)| = |(L_i \times L_j) \cdot L_k|$$

We illustrate such a configuration in Figure 4 and provide a geometrical meaning to this quantity, when L_i, L_j, L_k are (Euclidean) normalized line coordinates. The point-line projective duality gives

that the cross product $L_i \times L_j$ is a homogeneous vector representing the intersection point $P_{ij} = (x_{ij}, y_{ij})$ between two lines. Therefore,

$$L_i \times L_j = w_{ij} \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix}$$

with, thanks to the choice of a Euclidean normalization for lines,

$$w_{ij} = \begin{vmatrix} a_i & a_j \\ b_i & b_j \end{vmatrix} = \sin \alpha_{ij}$$

where α_{ij} denotes the angle between L_i and L_j.

The Euclidean normalization gives also that the inner product of $L_i \times L_j$ with the vector L_k yields the (two-dimensional) distance from the point P_{ij} to the line L_k, multiplied by w_{ij}:

$$D_{ijk} = w_{ij} \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} \cdot L_k = w_{ij}\, d(P_{ij}, L_k).$$

The distance from P_{ij} to line L_k can be expressed with the angle α_{ik} between the lines L_i and L_k:

$$d(P_{ij}, L_k) = ||P_{ij} - P_{ik}|| \sin \alpha_{ik}.$$

We get finally:

$$D_{ijk} = ||P_{ij} - P_{ik}|| \sin \alpha_{ij} \ \sin \alpha_{ik} \tag{5}$$

where α_{ij} and α_{ik}, respectively, denote the angles $\angle(L_i, L_j)$ and $\angle(L_i, L_k)$.

The invariance under isometries follows from Equation (5) while the invariance under permutation is inherited from the determinant expression, so that

Lemma 1. *Under Euclidean normalization of the line coordinates, D_{ijk} is invariant under Euclidean isometries and permutations.*

In the specific case of two parallel lines intersected perpendicularly by a third line, this determinant is exactly the distance separating the two parallel lines. We can therefore define a largest allowed distance ϵ to consider edge collapses. By denoting $\mathcal{T}$ the set of all triplets (i, j, k) corresponding to lines (L_i, L_j, L_k) forming at least two intersections in the partition, we define the following regularity objective:

$$E_{concurrent} = \sum_{(i,j,k)\in\mathcal{T}} \min\left(\epsilon, |\det(L_i, L_j, L_k)|\right) \tag{6}$$

We also favour orthogonality by adding an objective on the set $\mathcal{P}$ of all pairs (i, j) corresponding to intersecting lines (L_i, L_j) in the arrangement:

$$E_{orthogonality} = \sum_{(i,j)\in\mathcal{P}} \min(\sin \alpha_{\max}, |\text{dot2d}(L_i, L_j)|) \tag{7}$$

where dot2d denotes the inner product on the first two components of the vectors L_i and L_j, and $\alpha_{\max}$ is a tolerance angle below which we want to encourage lines to be orthogonal.

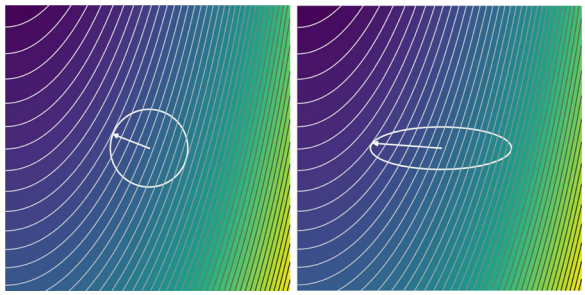

Figure 5: *Impact of the metric on the gradient descent direction. The same objective function is drawn in both images (high values in yellow to low values in purple, with grey isolines) but for the Euclidean (left) and a non-Euclidean (right) metric. We represent in Euclidean space a step of a fixed length for both metrics: this corresponds to a circle for the Euclidean metric and an ellipse for our choice of non-Euclidean metric. The direction of steepest descent (white arrow) is orthogonal to the objective function isolines, where orthogonality is defined by the underlying metric.*

The choice of combining an L^2 term for the fidelity objective and L^1 terms for simplification objectives is motivated by the need for exact line concurrency or orthogonality when possible. Indeed, L^1 regularization is known to lead to sparser solutions than L^2 regularization which, in our case, translates to a more efficient simplification scheme.

Note that taking the minimum with some threshold in Equations (6) and (7) makes these objectives active in the minimization only when the configuration is close to the satisfaction of the corresponding exact constraint.

6.3. Gradient descent in a Riemannian manifold

We solve our optimization problem via a gradient descent algorithm. Indeed, we are looking for local minima and, despite its apparent simplicity, the gradient method has proven to be a reliable descent method in particular for non-smooth and non-convex objective functions.

While the gradient is merely the transpose of the objective's first derivative in a Euclidean context, working on the space of lines using homogeneous coordinates and accounting for the data fidelity term lead to several changes in the standard expression of the gradient.

Recall that each iteration of a gradient descent algorithm follows the direction of steepest descent. This direction depends both on the derivative of the objective function to be minimized and the ambient metric. Indeed, as illustrated in Figure 5, the direction orthogonal to the isosurface of the objective function depends on the isosurface and the metric, which defines what 'orthogonal' means. Another way to express the direction of steepest decent consists in finding the direction minimizing the objective for a given (infinitesimal) step length. Again the step length, and therefore the direction of steepest descent, depends on the metric.

In our context, with many local minima, the trajectory of the gradient descent algorithm determines the local minimum reached at convergence. For this reason, we account for the data fidelity term in the metric associated to lines, inducing a steepest descent direction, which favours a trade-off between the objective minimization and the data fidelity term. This explains our choice for the matrix M_i' below, whereas the matrix induced by the ambient Euclidean metric on the embedded cylinder would have simply been $M_i' = U_i^t U_i$.

We recall here a few notions of Riemannian geometry, more details being provided by Lee [Lee13]. Consider a differentiable real-valued function $f : \mathcal{M} \to \mathbb{R}$, where $(\mathcal{M}, g)$ is a Riemannian manifold with metric g. Denote by $\mathcal{T}_p\mathcal{M}$ the tangent space of $\mathcal{M}$ at a point $p \in \mathcal{M}$. The directional derivative df and the gradient $\mathrm{grad} f$ of f at a point $p \in \mathcal{M}$ verify, for $v \in \mathcal{T}_p\mathcal{M}$:

$$df(v) = \langle \mathrm{grad} f, v \rangle_g \tag{8}$$

Given a system of coordinates for $\mathcal{T}_p\mathcal{M}$ at a given point $p \in \mathcal{M}$, denote by G the symmetric matrix associated with the inner product $\langle \cdot, \cdot \rangle_g$ defined by the metric g, and by D and ∇f the matrices associated with the directional derivative df and the gradient $\mathrm{grad} f$ at the given point p. Equation (8) translates to

$$\nabla f = G^{-1} D^t \tag{9}$$

The manifold $\mathcal{M}$ specific to our optimization is the manifold of oriented lines, seen as the submanifold of the space $\mathbb{R}^3$ of homogeneous coordinates with Euclidean normalization. As illustrated in Figure 3, this manifold corresponds to the cylinder with implicit equation $a^2 + b^2 = 1$. The choice of the metric on this manifold remains to be defined: we could for instance use the metric induced by the ambient Euclidean metric of $\mathbb{R}^3$, which corresponds to choosing the identity matrix for G. However, we defined in Section 6.2 meaningful quadratic forms M_i between line coordinates. We, therefore, utilize those quadratic forms to define metrics tailored to the lines of our problem.

The gradient descent is performed in the n-fold Cartesian product manifold $\mathcal{M} \times \cdots \times \mathcal{M}$, where n is the number of lines in $\mathcal{L}$. Each of these copies of $\mathcal{M}$ is associated with a line of $\mathcal{L}$ and a positive semi-definite matrix defined by Equation (3). The gradient of each line can be computed independently, and we consider the copy of $\mathcal{M}$ associated with the i^{th} line. $L_i = (a, b, c)$ and M_i denote, respectively, the current position in normalized homogeneous coordinates and the matrix associated with this line, which depends on the initial configuration of the line. We also denote by D_i the derivative of the objective function E given by Equation (1) with respect to the homogeneous coordinates of line L_i:

$$D_i = \left(\frac{dE}{da}, \frac{dE}{db}, \frac{dE}{dc} \right)$$

The tangent space at L_i corresponds to a two-dimensional vector space written as:

$$\mathcal{T}_{L_i}\mathcal{M} = \{ v \in \mathbb{R}^3, v \cdot (a, b, 0) = 0 \} = U_i \mathbb{R}^2 \tag{10}$$

Algorithm 1. Global simplification algorithm

Algorithm 1: Global simplification algorithm

Inputs : $\mathcal{C}$: combinatorial structure,
$\mathcal{L}$: list of lines coordinates,
α: step.
Output: $\mathcal{C}$, $\mathcal{L}$.
do
  do
    $\mathcal{L} \leftarrow$ **GradientStep**$(\mathcal{C}, \mathcal{L}, \alpha)$
    $\alpha \leftarrow$ **Update**(α)
  while $\alpha > \alpha_{\min}$
  $NumberOfReductions, \mathcal{C}, \mathcal{L} \leftarrow$ **Reductions**$(\mathcal{C}, \mathcal{L})$
while $NumberOfReductions > 0$

with

$$U_i = \begin{pmatrix} -b & 0 \\ a & 0 \\ 0 & 1 \end{pmatrix}$$

The quadratic form associated with the matrix M_i, restricted to this two-dimensional vector space $\mathcal{T}_{L_i}\mathcal{M}$, is non-degenerate—it can be seen as an actual Riemannian metric for the space $\mathcal{M}$ associated with the i^{th} line. We give the expression of the derivative D_i' and the matrix M_i' associated with the quadratic form in $\mathcal{T}_{L_i}\mathcal{M}$:

$$D_i' = D_i U_i$$
$$M_i' = U_i^t M_i U_i$$

Using Equation (9), the expression of the gradient with respect to line L_i in the tangent space $\mathcal{T}_{L_i}\mathcal{M}$ is

$$(\nabla E)_{\mathcal{T}_{L_i}\mathcal{M}} = (M_i')^{-1} D_i'^t$$

In order to update the coordinates of the line L_i, one could in theory apply the exponential map from $\mathcal{T}_{L_i}\mathcal{M}$ to $\mathcal{M}$. We use instead a simpler procedure, which is equivalent at first order. The expression of the gradient $\nabla_i E$ in homogeneous coordinates is obtained as the product of U_i, mapping $\mathcal{T}_{L_i}\mathcal{M}$ to the space $\mathbb{R}^3$ of homogeneous coordinates, with $(\nabla E)_{\mathcal{T}_{L_i}\mathcal{M}}$:

$$\nabla_i E = U_i (M_i')^{-1} D_i'^t \tag{11}$$

For a given step α, the homogeneous coordinates are updated by adding $-\alpha \nabla_i E$ and normalized, which can be seen as an orthogonal projection onto $\mathcal{M}$.

Denoting $normalize((a, b, c)^t)$, the Euclidean normalization of vector $(a, b, c)^t$, a gradient descent step consists of:

$$\forall i, L_i^{(k+1)} = normalize\left(L_i^{(k)} - \alpha \nabla_i E\right) \tag{12}$$

6.4. Global algorithm

The global simplification process is shown on Algorithm 6.4 where $\mathcal{C}$ and $\mathcal{L} = (L_1, \ldots, L_n) \in \mathbb{R}^{3n}$, respectively, represent the combinatorial and numerical components of the line arrangement. Figure 6 illustrates how a polygonal partition typically evolves during

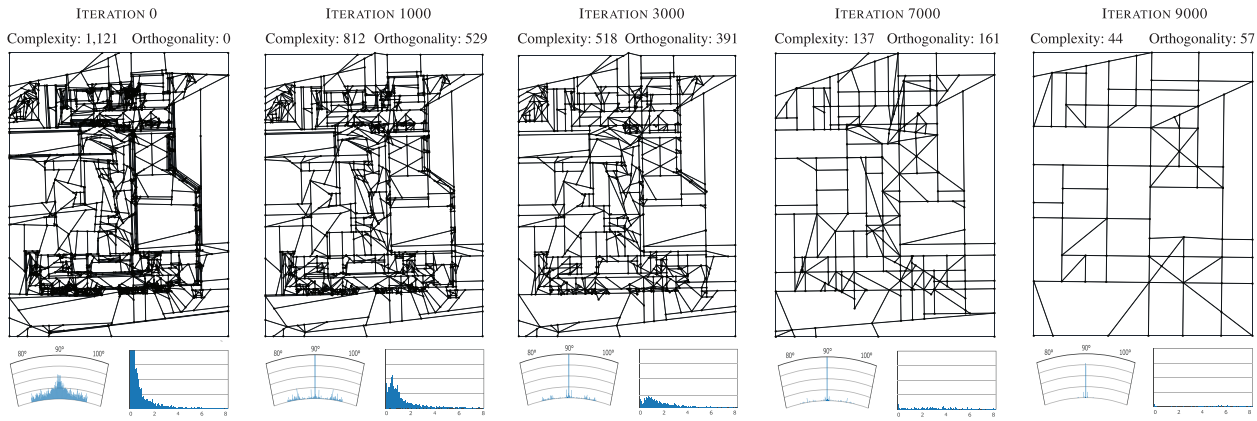

Figure 6: *Evolution of partitions during optimization. The global simplification algorithm progressively reduces the number of lines while increasing the orthogonality between them ('complexity' and 'orthogonality' refer to the number of lines and the number of orthogonal pairs of lines in the partition). The final partition (right) does not contain small or nearly collinear edges anymore and the angles around* 90° *are exactly orthogonal (the histograms depict the angle deviation (left) and the distribution of edge length (right) for each partition).*

the simplification process.

The instruction $\mathcal{L} \leftarrow$ **GradientStep**$(\mathcal{C}, \mathcal{L}, \alpha)$ consists of one Riemannian gradient step described in Equation (12) for each line L_i in which only the numerical data $\mathcal{L}$ is updated.

The instruction $\alpha \leftarrow$ **Update**(α) governs the rate of decrease of α. The analysis of the rate of convergence of gradient methods for non-smooth, non-convex objectives [KW19] is a difficult question that has not been studied in this work. Many techniques for improving the convergence speed [Nes83, Qia99] are taking on renewed importance with the popularity of deep learning. Experimentally, a geometric sequence $\alpha_i = \alpha_0 r^i$, where r is set to a value inferior yet close to 1, is sufficient to successfully converge to local minima. The parameter, $\alpha_{\min}$ of the stopping criterion must be small enough to allow the cancellation of regularization terms below a threshold η, which will be described in Equation (13).

The instruction $NbReductions, \mathcal{C}, \mathcal{L} \leftarrow$ **Reductions**$(\mathcal{C}, \mathcal{L})$ in Algorithm 6.4, applies the combinatorial reduction step that updates both the combinatorial structure $\mathcal{C}$, by decreasing the number n of lines and updating the line intersection relations as well as the list of coordinates $\mathcal{L}$ of new lines resulting from merge decisions. It returns in the integer variable '*NumberOfReductions*' the number of achieved merges, which is used in the termination test.

Robustness of combinatorial reductions. It is well known that optimization with L^1 regularization leads to sparser solutions than its L^2 counterpart, which in our case translates into edge collapses, orthogonalities and line fusions. The line representation does not allow edges to be collapsed, as this would require multiple lines to pass through the same point. However, it is useful to interlace optimization iterations with combinatorial line merges, as shown in Algorithm 6.4, and thus continue optimizing in a simplified partition. Such an entanglement of numerical and combinatorial processes may lead to what is known in computational geometry as robustness problems: combinatorial decisions require perfect consistency, whereas numerical computations produce only approximations.

If the numerical gradient descent was able to perfectly cancel a determinant formed by three lines ($|\det(L_i, L_j, L_k)| = 0$), then at least one of the two following projective situations would occur: (a) two lines are equal or (b) the three lines differ but meet at a single point. In the language of affine geometry, situation (b) splits into the case of three lines that meet at a single affine point (b1) and three lines that meet at infinity, in other words, three parallel lines (b2). The following question now occurs: what happens to this observation when replacing $|\det(L_i, L_j, L_k)| = 0$ by $|\det(L_i, L_j, L_k)| < \eta$ for a small $\eta > 0$?

In order to make the right combinatorial decisions when merging lines, while using the finite accuracy from numerical computations, we need a carefully quantified version of the previous 'exact' implication of one of situations (a), (b1) and (b2) so that, when lines (L_i, L_j, L_k) form a regularization term $|\det(L_i, L_j, L_k)|$ smaller than a threshold value η, we can guarantee that one of the following situations occur:

- Two (or more) lines of the triplet are indistinguishable up to some quantified accuracy d_L, corresponding to exact situation (a).
- Two (or more) points at the line intersections are indistinguishable up to some quantified accuracy d_P corresponding to exact situation (b1).
- The intersection points are outside a disk of radius R corresponding to exact situation (b2).

In Appendix A, we relate this threshold η in the continuous optimization to values used in discrete combinatorial operations: a minimal distance d_P at which points are considered identical, a minimal quantity between line vectors d_L at which they are considered identical and the radius R of a disk centred at 0 containing the partition:

$$\eta = \frac{d_P \, d_L^2}{5(1 + R^2)} \tag{13}$$

When working with 64-bit arithmetic, a typical numerical value in Equation (13) would be $\eta = 10^{-16}$, for $R = 1$ and $d_P = d_L = 10^{-5}$. A discrete step in our optimization consists then in deciding for each regularization term in $E_{\text{concurrent}}$ below η the nature of the simplification: point fusion (b1), line fusion (a) or removal of an intersection point outside the area of interest (b2).

Propagation of reductions. The function **Reductions**$(\mathcal{C}, \mathcal{L})$ must propagate the simplifications in the partition by verifying the following geometric invariant of our structure: two lines cannot have more than one common point and two points cannot have more than one common line.

In a first step, for each triplet (i, j, k) for which $|\det(L_i, L_j, L_k)| < \eta$ one of the mentioned three alternatives (lines merge, points merge or point deletion) is applied. When lines L_i and L_j merge into line L_{ij}, if there is a point indexed by (L_j, L_k), it is inherited by the pair (L_{ij}, L_k). Along successive merges, the aforementioned geometric invariant may be violated, breaking the validity of the arrangement. Each time such a configuration is encountered, the indexes of the faulty pair are pushed on a stack. Once the first step is achieved, one either applies a lines merge or a points merge for each pair popped from the stack. Again, these merges may push new pairs on the stack. Since each pop operation induces a reduction of the structure, the combinatorial simplification eventually ends.

Finally, when merging lines L_i and L_j into L_f, we combine their quadratic forms associated with both lines into a resulting fused quadratic form:

$$M_f = M_i + M_j$$

This process ensures that the fidelity to original segments of the arrangement is propagated throughout discrete operations. Indeed, the quadratic forms are initialized with endpoints of initial segments and compute the sum of squared distances to these endpoints. During the algorithm, quadratic forms still compute the sum of squared distances, but the set of 2D points associated with each line has been inherited by other lines through merges.

7. Piecewise-linear Model

Given a 2D polygonal partition and a point cloud $\mathcal{P}$, the final model could be created by assigning to each cell of the partition the plane best fitting the point cloud of the cell. However, this approach is not resilient to imprecise partitions or missing data and noise in the input points. Instead, using the set of detected planes Π in Section 4, a discrete optimization is performed to obtain a coherent model. Denote by $\mathcal{C}$ the set of cells of the partition. For a cell $\mathcal{C}_i \in \mathcal{C}$ and a plane equation $z = \Pi_k(x, y)$, the following term computes the fidelity of the lifted cell $\mathcal{C}_i$ along Π with the subset of points $P \cap \mathcal{C}_i$ whose projection along z lies in the same cell $\mathcal{C}_i$:

$$E_{\mathcal{C}_i}(\Pi_k) = \frac{\mathcal{A}_{\mathcal{C}_i}}{\text{Card}(P \cap \mathcal{C}_i)} \sum_{p \in P \cap \mathcal{C}_i} \min(\epsilon, |p_z - \Pi_k(p_x, p_y)|) \quad (14)$$

where $\mathcal{A}_{\mathcal{C}_i}$ denotes the cell area of $\mathcal{C}_i$. A threshold ϵ is used on the z-distance to the plane to improve the robustness of the plane fitting in the presence of noise in the point cloud or slight errors in the cell decomposition.

The regularity term between two cells $(\mathcal{C}_i, \mathcal{C}_j)$ intersecting on an edge $e = \mathcal{C}_i \cap \mathcal{C}_j$ needs to penalize the height difference along e.

Given planes $\Pi_k, \Pi_l \in \Pi$ assigned to the cells on each side of the edge e, a regularity term measures a mean vertical area along the edge e:

$$V_e(\Pi_k, \Pi_l) = p_e \text{length}(e) \text{z}_\text{e}(\Pi_k, \Pi_l) \quad (15)$$

where z_e is given by, for $e = ((x_1, y_1), (x_2, y_2))$:

$$\text{z}_\text{e}(\Pi_k, \Pi_l) = \frac{|\Pi_k(x_1, y_1) - \Pi_l(x_1, y_1)| + |\Pi_k(x_2, y_2) - \Pi_l(x_2, y_2)|}{2}$$

For each edge, the parameter p_e in Equation (15) denotes an *a priori* probability that the edge is along a building facade and can, for instance, be found by identifying segments at the intersection of a vertical plane. We call labelling a map $\mathcal{L} : \mathcal{C} \mapsto \Pi$ that associates to each cell of the arrangement a plane in Π. The global energy objective is expressed as follows:

$$E(\mathcal{L}) = \sum_{\mathcal{C}_i \in \mathcal{C}} E_{\mathcal{C}_i}(\mathcal{L}(\mathcal{C}_i)) + \mu \sum_{e = \mathcal{C}_i \cap \mathcal{C}_j} V_e(\mathcal{L}(\mathcal{C}_i), \mathcal{L}(\mathcal{C}_j))$$

The regularity parameter μ, homogeneous to a length, should be understood as the smallest 'feature' size we want to distinguish. This objective function, decomposed as a unary fidelity term and binary regularity interactions on labels, belongs to a family of functions where local minima can be reached efficiently using graph cut optimization [BVZ01].

8. Experiments

Tradeoff between fidelity, complexity and regularity. The two main parameters of our method are λ_1 and λ_2, which balance the three energy terms in the simplification of the partition. As illustrated in Figure 7, these two parameters allow an intuitive control on the accuracy, complexity and level of orthogonality of the partition. Note that, when increasing λ_2, the number of lines usually stays stable and the number of points decreases slightly: this decrease is explained by the situation where two lines become orthogonal to a third line and their intersection point is sent to infinity. The combinatorial reduction will then remove this intersection from the partition.

In Figure 7, as the error is concentrated in small cells, reasonable choices for concurrent lines and orthogonality parameters do not introduce large distortions and therefore the increase in 3D fidelity error is controlled. Further simplifications with larger regularity parameters degrade the partition gracefully: for instance, building contours can be identified in the most simplified example (top right). These large distortions lead, however, to large errors in the 3D fidelity measure.

Flexibility and robustness. We test our reconstruction method on different scales of urban scenes, from individual houses (Figure 10), to building blocks (Figure 8), and larger architectural structures (Figures 1 and 9). Our method produces concise and accurate models as long as observed buildings can be represented by a piecewise planar and disk-topology geometry. We also test our algorithm on input point clouds generated by different acquisition systems. As illustrated in Figure 10, our method returns 3D models of similar

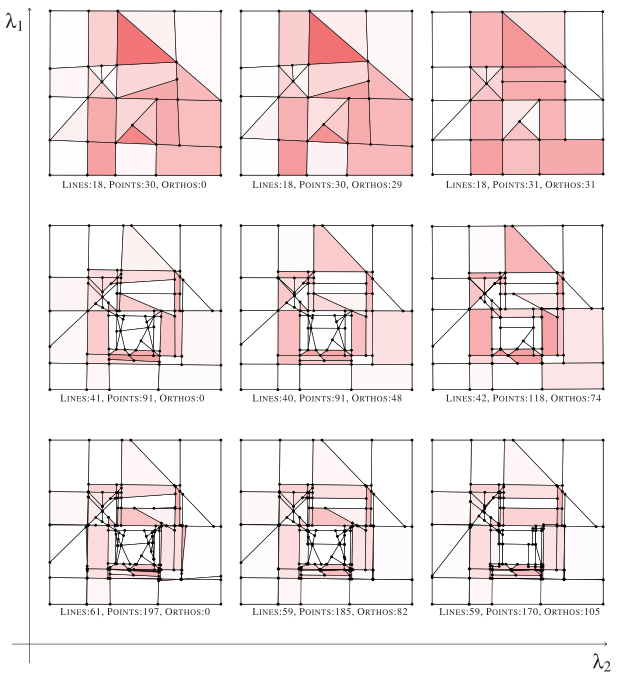

Figure 7: *Impact of parameters λ_1 and λ_2. Increasing λ_1 reduces the complexity of the partition, while increasing λ_2 encourages the presence of orthogonal lines (*Orthos *refers to the number of orthogonalities counted when the angle difference with $\frac{\pi}{2}$ is under 10^{-4°). Increasing λ_1 and λ_2 simultaneously reduces the accuracy of the partition as illustrated by the increasing presence of red polygons towards the top-right diagonal (the root mean squared error of the best-fit plane of the 3D input points projected in each polygon is displayed from white (zero error) to red (high error)).*

quality on both a Laser point cloud with missing data and occlusions and a multi-view stereo point cloud with noise and outliers. The importance of the partition simplification step, our main contribution, is visible in this figure: without simplification, output results are less accurate and less concise.

Performance. Our method is implemented in C++ (single-thread) and uses the Eigen library [GJ*10] for its ease of use when dealing with homogeneous representations as well as solving small linear algebra systems appearing in the computation of the Riemannian gradient. All timings are recorded on a Lenovo 20EQS27P00 laptop with an Intel i7-6820HQ CPU model with 32GB of memory. Table 8 shows the processing times of the different steps of our method on input data of various sizes. The creation and simplification of 2D partitions are fast and scalable, typically a few seconds for large urban scenes. By implementing the main geometric operations in the horizontal 2D plane, we avoid the time- and memory-consuming issues arising from the creation and manipulation of 3D polyhedral arrangements. Adding this simplification method greatly improves the performance of the lift optimization step by reducing the number of cells in the partition.

Metric propagation along combinatorial simplification. A prime example of the usefulness of propagating metric information

Table 1: *Performances of our algorithm on different input data. The processing time for creating and simplifying 2D partitions are negligible compared to those of 3D operations, i.e. line-segment extraction and output mesh.*

	BarnMVS (Figure 10)	BuildingBlock (Figure 8)	Dublin (Figure 1)
#input points	619,472	1000,000	6305,813
Scene area	400 m^2	13,000 m^2	19,000 m^2
Line-segment extraction	6.7 s	8.9 s	37 s
#line-segments	213	813	1681
Kinetic partitioning	3 ms	4 ms	238ms
#polygons	330	1422	2724
Simplification	0.4 s	1.5 s	5.3 s
#polygons	61	288	1197
Lift optimization	0.9 s	3.1 s	87 s
#output faces	24	139	637

is depicted by Figure 6. In this example, the segment detection led to an original partition containing a lot of small segments clustered along the building's facades. During simplification, lines carrying those segments are merged in the first few iterations and their metric information is thus accumulated by lines representing large parts of the facades. When further simplifying, the fidelity term on those lines leads to very little deviation from their original position and they are therefore stabilized by the information of previous simplifications. In the final partition, made of 44 lines—a 96% reduction from its original counterpart—the outline of the building's facades is still visible and lines on this outline can carry the information of up to 100 initial segments.

Comparisons. We compare our method with the piecewise-planar reconstruction method [CLP10] and three mesh simplification pipelines in which input points are first converted into a dense surface mesh by the screened Poisson algorithm [KH13] before being simplified either by the popular Quadrics Error Metrics edge contraction algorithm [GH97], by structure-aware edge decimation [SLA15] or by variational shape approximation [CAD04]. As illustrated in Figure 10, edge contraction-based methods [GH97, SLA15] cannot produce very low complexity models (i.e. with one or two dozen of facets for a standard house) without strongly degrading the geometric accuracy to input data. The mesh simplification pipeline [CAD04] and the reconstruction method [CLP10] can produce more accurate results but with a higher complexity. Our method offers the best accuracy/complexity tradeoff as well as stable results for different acquisition systems (laser and multi-view stereo). Moreover, only our method can deliver meshes in which orthogonalities are preserved. Contrary to previous algorithms [GH97, SLA15], our method does not offer a direct control on the output mesh complexity. This being said, our method has been designed to produce highly concise meshes, departing from other algorithms [CAD04, CLP10].

Limitations. We acknowledge a few limitations of our method. First, the assumption that observed objects can be represented by a 2.5D disk-topology geometry, while true for many buildings, does not hold for other urban objects. Openings and large overhangs are also not correctly captured by this representation. Figure 11

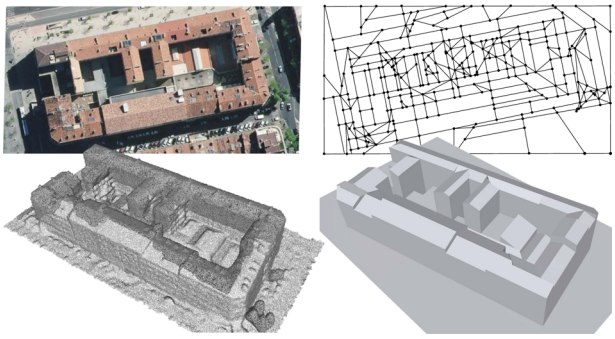

Figure 8: *Result on a building block. The output 3D model (bottom right) obtained from a multi-view stereo point cloud (bottom left) preserves the main roof components while ignoring small elements such as chimneys and dormer-windows.*

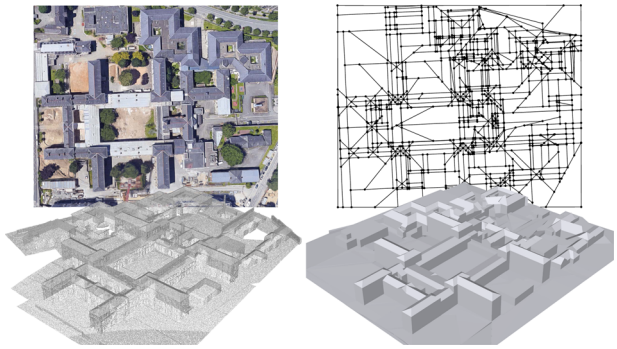

Figure 9: *Result on a large regular building. Our 3D output model (bottom right) preserves the orthogonality existing between facade and rooftop components. Note how the planimetric partition (top right) aligns well with the aerial image (top left, not used).*

illustrates this limitation. Indeed, in this example, points from the roof and the window frames project onto the same 2D cells. The piecewise-linear optimization described in Section 7 is unable to correctly fit a single plane to all these projected points and decides for a plane located at mid-distance between the roof and the window frames. Therefore, this plane produces an undesirable representation.

Second, the geometric regularities taken into account by our method are limited to orthogonalities and, by construction, to collinearities. Considering other regularities such as symmetries would probably improve the quality of our results: we could conceivably extend our optimization formulation to take them into account. Finally, the simplification of partitions can, in some cases, degrade the accuracy of the output model. As illustrated in Figure 12,

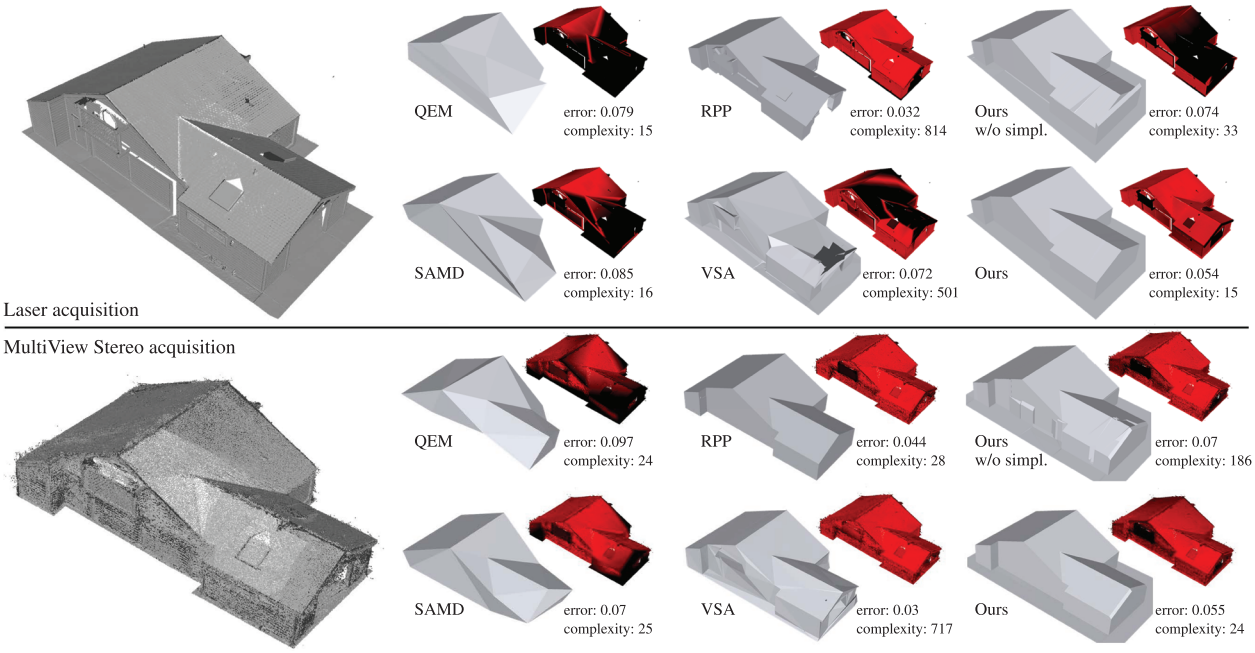

Figure 10: *Comparisons with existing methods. Mesh simplification algorithms QEM [GH97] and SAMD [SLA15] applied from Screened Poisson dense meshes [KH13] cannot produce low complexity models without altering the geometric accuracy ('complexity' and 'error' refer to the number of facets in the output models and to the root mean square Hausdorff distance from input point clouds (left) to the output models, the Hausdorff distance ranging from 0 m (red) to 0.2 m (black)). VSA [CAD04] and the piecewise-planar reconstruction method RPP [CLP10] deliver more accurate results but with a higher complexity. Our method provides the best tradeoff between accuracy and complexity, independently of the acquisition system used to generate the input point cloud. It is also the only one to preserve the orthogonality between the walls of the building. Note how our results degrade when the simplification of the 2D partition is not activated. Input data from Knapitsch et al. [KPZK17].*

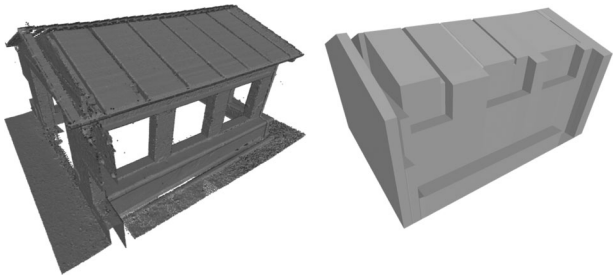

Figure 11: *When running the pipeline on a building that does not verify the 2.5D hypothesis, the piecewise-linear modelling cannot represent the building correctly and leads to overly complex representations, as different heights of the building project on the same 2D cells (roof and opening for example).*

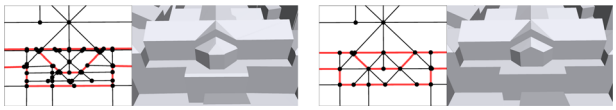

Figure 12: *Failure case. Our simplification process may degrade the accuracy of the output model on curved structures. The left (respectively right) model is obtained without (resp. with) simplification of the 2D partition. Edges highlighted in red, corresponding to facades in the 3D model, show the discrepancy between 2D regularity and 3D fidelity.*

this typically happens on curved structures, where small segments in 2D play a large role in the quality of the facade, and the resulting models can appear overly simplified.

9. Conclusion

We propose an automatic method to reconstruct urban scenes in the form of concise and regular polygonal surface meshes. The key ingredient is a simplification process of 2D polygonal partitions, which seeks a balance between the fidelity to input partition, the enforcement of canonical relationships between lines, and a low complexity output. Lifted in 3D, the simplified partitions allow for scalable generation of 3D models whose quality competes well with existing methods. While our method is designed to reconstruct buildings, the simplification process in itself is general and can be useful in other geometric modelling problems.

In future work, we plan on designing operators able to locally refine partitions *a posteriori*, where the lifted models do not account for the input data, as well as allowing for scalable and iterative improvements of the reconstruction. Finally, investigating how our simplification process can be extended efficiently to 3D polyhedral arrangements could in turn lead to more scalable 3D piecewise-planar reconstruction methods.

Acknowledgements

The work presented in this paper was funded by Dassault Systèmes. Pierre Alliez is supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

Appendix A: From Continuous Optimization to Discrete Combinatorial Operations

By translating the arrangement such that the origin O is at its centre, we assume that all lines as well as all points of the arrangement are at a distance from the origin O bounded by R, where R represents the half-size of the arrangement. We want to determine a value η such that when a regularity criteria D_{ijk} on a triplet of lines (L_i, L_j, L_k) is smaller than η, we can guarantee that either two lines L, L' of the triplet are indistinguishable (i.e. verify $||L - L'|| < d_L$) or that the two points at the intersection of (L_i, L_j) and (L_i, L_k) are indistinguishable (i.e. have distance smaller than a chosen length d_P).

To that effect, we assume that $D_{ijk} < \eta$ and all pair of lines in the triplet verify $||L - L'|| > d_L$ and expect $||P_{ij} - P_{ik}|| < d_P$.

Consider two lines L, L' represented respectively by homogeneous coordinates (a, b, c) and (a', b', c'). Recall that the intersection point P has homogeneous coordinates:

$$P = L \times L' = \left(\begin{vmatrix} b & b' \\ c & c' \end{vmatrix}, \begin{vmatrix} a & a' \\ c & c' \end{vmatrix}, \sin\alpha \right)$$

where α denotes the angle between L and L'. Therefore, using the normalization $a^2 + b^2 = a'^2 + b'^2 = 1$

$$\begin{aligned} OP^2 \sin^2\alpha &= (bc' - b'c)^2 + (ac' - a'c)^2 \\ &= (a^2 + b^2)c'^2 + (a'^2 + b'^2)c^2 - 2cc'(aa' + bb') \\ &= (c' - c)^2 + 2cc'(1 - \cos\alpha) \end{aligned}$$

And finally

$$(c' - c)^2 = OP^2 \sin^2\alpha - 4cc' \sin^2\frac{\alpha}{2} \tag{16}$$

The assumption that the point P as well as the lines L and L' have a distance to the origin upper bounded by R can be translated into:

$$\begin{cases} OP \leq R \\ |c| \leq R \\ |c'| \leq R \end{cases}$$

Combined with Equation (A.1), we get:

$$(c' - c)^2 \leq R^2 \sin^2\alpha - 4cc' \sin^2\frac{\alpha}{2} \leq 5R^2 \sin^2\alpha$$

From the assumption that $||L - L'|| > d_L$,

$$\begin{aligned} d_L^2 &< (a' - a)^2 + (b' - b)^2 + (c' - c)^2 \\ &< 4\sin^2\frac{\alpha}{2} + 5R^2 \sin^2\alpha \\ &< 5(1 + R^2)\sin^2\alpha \end{aligned}$$

The last inequality holds for $\alpha = \alpha_{ij}$ and for $\alpha = \alpha_{ik}$ so that:

$$\sin\alpha_{ij}\sin\alpha_{ik} > \frac{d_L^2}{5(1+R^2)} \tag{17}$$

which is used to upper bound the distance between two points from the regularity criteria. Indeed, if:

$$D_{ijk} = ||P_{ij} - P_{ik}||\sin\alpha_{ij}\sin\alpha_{ik} < \eta,$$

then Equation (A.2) gives:

$$||P_{ij} - P_{ik}|| < \frac{5\eta(1+R^2)}{d_L^2}$$

From the assumptions, we want $||P_{ij} - P_{ik}|| < d_P$, which is verified if the following inequality is verified by η:

$$\eta \leq \frac{d_P\, d_L^2}{5(1+R^2)}$$

References

[AS17] Achanta R., Süsstrunk S.: Superpixels and polygons using simple non-iterative clustering. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (Honolulu, HI, USA, 2017), IEEE Computer Society, pp. 4895–4904.

[BBPDM08] Brédif, M., Boldo D., Pierrot-Deseilligny M., Maître H.: 3D building model fitting using a new kinetic framework. *arXiv:0805.0648* (2008).

[Ber09] Berger M.: *Geometry i*. (Heidelberg, Germany, 1987), Springer Science & Business Media.

[BL18] Bauchet J., Lafarge F.: KIPPI: Kinetic polygonal partitioning of images. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018* (Salt Lake City, UT, USA, 2018), Computer Vision Foundation/IEEE Computer Society, pp. 3146–3154.

[BL19] Bauchet J.-P., Lafarge F.: City reconstruction from airborne lidar: A computational geometry approach. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W8* (2019), 19–26.

[BRG15] Bódis-Szomorú A., Riemensch H., Gool L. V.: Superpixel meshes for fast edge-preserving surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015* (Boston, MA, USA, 2015), IEEE Computer Society, pp. 2011–2020.

[BVZ01] Boykov Y., Veksler O., Zabih R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239.

[CAD04] Cohen-Steiner D., Alliez P., Desbrun M.: Variational shape approximation. *ACM Transactions on Graphics 23*, 3 (2004), 905–914.

[CGL85] Chazelle B., Guibas L. J., Lee D.-T.: The power of geometric duality. *BIT Numerical Mathematics 25*, 1 (1985), 76–90.

[CLP10] Chauve A., Labatut P., Pons J.: Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *Proceedings of the Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010* (San Francisco, CA, USA, 2010), IEEE Computer Society, pp. 1261–1268.

[CY00] Coughlan J. M., Yuille A. L.: The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference. In *Proceedings of the Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000* (Denver, CO, USA, 2000), T. K. Leen, T. G. Dietterich and V. Tresp (Eds.), MIT Press, pp. 845–851.

[DGCSAD11] De Goes F., Cohen-Steiner D., Alliez P., Desbrun M.: An optimal transport approach to robust reconstruction and simplification of 2D shapes. *Computer Graphics Forum 30*, 5 (2011), 1593–1602.

[DL16] Duan L., Lafarge F.: Towards large-scale city reconstruction from satellites. In *Proceedings of the Computer Vision—ECCV 2016—14th European Conference, Proceedings, Part V* (Amsterdam, The Netherlands, 2016), B. Leibe, J. Matas, N. Sebe and M. Welling (Eds.), vol. 9909 of *Lecture Notes in Computer Science*, Springer, pp. 89–104.

[Far99] Farin G.: *NURBS for Curve & Surface Design: From Projective Geometry to Practical Use*. CRC Press, 1999.

[FLBA20] Favreau J., Lafarge F., Bousseau A., Auvolat A.: Extracting geometric structures in images with delaunay point processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 42*, 4 (2020), 837–850.

[FTK14] Feng C., Taguchi Y., Kamat V. R.: Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation, ICRA 2014* (Hong Kong, China, 2014), IEEE, pp. 6218–6225.

[Ger17] Gerke M.: Photogrammetric computer vision: Statistics, geometry, orientation and reconstruction. *The Photogrammetric Record 32*, 158 (2017), 195-220.

[GH97] Garland M., Heckbert P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997* (Los Angeles, CA, USA, 1997), G. S. Owen, T. Whitted and B. Mones-Hattal (Eds.), ACM, pp. 209–216.

[GJ10] Guennebaud G., Jacob B.: Eigen v3.http://eigen.tuxfamily.org (2010).

[GS97] Gevers T., Smeulders A. W. M.: Combining region splitting and edge detection through guided Delaunay image subdivision. In *CVPR'97: Proceedings of the 1997 Conference on Com-*

puter Vision and Pattern Recognition (San Juan, Puerto Rico, 1997), IEEE Computer Society, pp. 1021–1026.

[Gui04] Guibas L. J.: Kinetic data structures. In *Handbook of Data Structures and Applications (Boca Raton, FL, USA, 2004)*, D. P. Mehta and S. Sahni (Eds.). Chapman and Hall/CRC.

[Gun17] Gunn C. G.: Doing Euclidean plane geometry using projective geometric algebra. *Advances in Applied Clifford Algebras 27*, 2 (2017), 1203–1232.

[HB12] Holz D., Behnke S.: Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *Intelligent Autonomous Systems 12—Volume 2 Proceedings of the 12th International Conference IAS-12* (Jeju Island, Korea, 2012), S. Lee, H. Cho, K. Yoon and J. Lee (Eds.), vol. 194 of *Advances in Intelligent Systems and Computing*, Springer, pp. 61–73.

[Her92] Herman I.: *The Use of Projective Geometry in Computer Graphics*. *Lecture Notes in Computer Sciences*. Springer Verlag, Heidelberg, 1992.

[HK12] Habbecke M., Kobbelt L.: Linear analysis of nonlinear constraints for interactive geometric modeling. *Computer Graphics Forum 31*, 2 (2012).

[HZ06] Harltey A., Zisserman A.: *Multiple View Geometry in Computer Vision* (2nd edition) (Cambridge, England, 2006), Cambridge University Press, pp. 65-86.

[Kad07] Kada M.: Scale-dependent simplification of 3D building models based on cell decomposition and primitive instancing. In *Proceedings of the Spatial Information Theory, 8th International Conference, COSIT 2007, Proceedings* (Melbourne, Australia, 2007), S. Winter, M. Duckham, L. Kulik and B. Kuipers (Eds.), vol. 4736 of *Lecture Notes in Computer Science*, Springer, pp. 222–237.

[KH13] Kazhdan M. M., Hoppe H.: Screened Poisson surface reconstruction. *ACM Transactions on Graphics 32*, 3 (2013), 29:1–29:13.

[KPZK17] Knapitsch A., Park J., Zhou Q.-Y., Koltun V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics 36*, 4 (2017), 1–13.

[KW19] Khamaru K., Wainwright M. J.: Convergence guarantees for a class of non-convex and non-smooth optimization problems. *Journal of Machine Learning Research 20*, 154 (2019), 1–52.

[LAV*17] Laefer D. F., Abuwarda S., Vo A.-V., Truong-Hong L., Gharibi H.: 2015 Aerial Laser and Photogrammetry Datasets for Dublin, Ireland's City Center, 2017.

[Lee13] Lee J. M.: Smooth manifolds. In *Introduction to Smooth Manifolds* . (New York, NY, USA, 2013), Springer, pp. 1–31.

[Lin00] Lindstrom P.: Out-of-core simplification of large polygonal models. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000* (New Orleans, LA, USA, 2000), J. R. Brown and K. Akeley (Eds.), ACM, pp. 259–262.

[LLM20] Li M., Lafarge F., Marlet R.: Approximating shapes in images with low-complexity polygons. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020* (Seattle, WA, USA, 2020), Computer Vision Foundation/IEEE, pp. 8630–8638.

[LWC*11] Li Y., Wu X., Chrysathou Y., Sharf A., Cohen-Or D., Mitra N. J.: Globfit: Consistently fitting primitives by discovering global relations. *ACM Transactions on Graphics 30*, 4 (July 2011), 1–12.

[MBHI*95] Marathe M. V., Breu H., Hunt III H. B., Ravi S. S., Rosenkrantz D. J.: Simple heuristics for unit disk graphs. *Networks 25*, 2 (1995), 59–68.

[MMBM15] Monszpart A., Mellado N., Brostow G. J., Mitra N. J.: Rapter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Transactions on Graphics 34*, 4 (2015), 103:1–103:12.

[MPWC13] Mitra N., Pauly ., Wand M., Ceylan D.: Symmetry in 3d geometry: Extraction and applications. *Computer Graphics Forum 32*, 6 (2013), 1–23.

[MWA*13] Musialski P., Wonka P., Aliaga D., Wimmer M., Van Gool L., Purgathofer W.: A survey of urban reconstruction. *Computer Graphics Forum 32*, 6 (2013), 146–177.

[NDW93] Neider J., Davis T., Woo M.: *OpenGL Programming Guide* (vol. *478*). Addison-Wesley, Reading, MA, 1993.

[Nes83] Nesterov Y.: A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Doklady AN USSR* (1983).

[PMW*08] Pauly M., Mitra N., Wallner J., Pottman H., Guibas L.: Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics 27*, 3 (2008), 1–11.

[PW09] Pottmann H., Wallner J.: *Computational Line Geometry (Heidelberg, Germany, 2001)* . Springer.

[Qia99] Qian N.: On the momentum term in gradient descent learning algorithms. *Neural Networks 12*, 1 (1999), 145–151.

[RBDD18] Rodriguez S., Bousseau A., Durand F., Drettakis G.: Exploiting repetitions for image-based rendering of facades. *Computer Graphics Forum 37*, 4 (2018), 119–131.

[RVDHV06] Rabbani T., Van Den Heuvel F., Vosselmann G.: Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36*, 5 (2006), 248–253.

[SLA15] Salinas D., Lafarge F., Alliez P.: Structure-aware mesh decimation. *Computer Graphics Forum 34*, 6 (2015), 211–227.

[SRF*14] Straub J., Rosman G., Freifeld O., Leonard J. J., III J. W. F.: A mixture of manhattan frames: Beyond the Manhattan world. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014* (Columbus, OH, USA, 2014), IEEE Computer Society, pp. 3770–3777.

[Sta12] Standard OGC: OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0, 2012.

[VLA15] Verdie Y., Lafarge F., Alliez P.: LOD generation for urban scenes. *ACM Transactions on Graphics 34*, 3 (2015), 1–14.

[ZBKB08] Zebedin L., Bauer J., Karner K. F., Bischof H.: Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Proceedings of the Computer Vision—ECCV 2008, 10th European Conference on Computer Vision, Proceedings, Part IV* (Marseille, France, 2008), D. A. Forsyth, P. H. S. Torr and A. Zisserman (Eds.), vol. 5305 of *Lecture Notes in Computer Science*, Springer, pp. 873–886.

[ZN10] Zhou Q., Neumann U.: 2.5D dual contouring: A robust approach to creating building models from aerial lidar point clouds. In *Proceedings of the Computer Vision—ECCV 2010, 11th European Conference on Computer Vision, Proceedings, Part III* (Heraklion, Crete, Greece, 2010), K. Daniilidis, P. Maragos and N. Paragios (Eds.), vol. 6313 of *Lecture Notes in Computer Science*, Springer, pp. 115–128.

[ZN12] Zhou Q., Neumann U.: 2.5D building modeling by discovering global regularities. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition* (Providence, RI, USA, 2012), IEEE Computer Society, pp. 326–333.