

3D Reconstruction and LOD Generation for Urban Scenes

Abstract

We introduce a novel method that reconstructs 3D urban scenes in the form of levels of detail (LODs). Starting from raw data sets such as surface meshes generated by multi-view stereo systems, our algorithm proceeds in three main steps: classification, abstraction and reconstruction. From geometric attributes and a set of semantic rules combined with a Markov random field, we classify the scene into four meaningful classes. The abstraction step detects and regularizes planar structures on buildings, fits icons on trees and roof super-structures, and performs filtering and simplification for LOD generation. The abstracted data support the reconstruction step using a min-cut formulation on a set of 3D arrangements to generate watertight buildings. Our experiments on complex buildings and large scale urban scenes show that our approach generates meaningful LODs while being robust and scalable. By combining semantization and abstraction it also outperforms general mesh approximation approaches at preserving urban structures.

CR Categories: I.3.5 [Computer Graphics]: Three-Dimensional Graphics and Realism—Computational Geometry and Object Modeling I.4.5 [Image Processing and Computer Vision]: Reconstruction—;

Keywords: urban reconstruction, levels of detail, abstraction, iconization, Markov random field, min-cut formulation, arrangement of planes.

1 Introduction

The quest for automated modeling of large scale urban scenes has received increasing interest in recent years. A first class of approaches apply *procedural modeling* from grammatical rules and a fair amount of user interaction to generate detailed 3D models that are highly semantized [Vanegas et al. 2010b]. Another class of approaches, referred to as *urban reconstruction* and focus of the present work, aims at the automated generation of accurate 3D models from physical measurements [Musalski et al. 2012].

The availability of massive airborne photos or LIDAR data sets at the scale of entire cities has stimulated research on automated methods for urban reconstruction. With recent advances on fully automated multi-view stereo (MVS) workflows it is now possible to reconstruct from high resolution images an entire city with centimeter-grade resolution [Acute3D 2013; Pix4D 2013]. The quality of the reconstruction may be evaluated through visual inspection, faithfulness to the ground truth when available, or complexity/distortion tradeoff. Common outputs of MVS workflows are complex surface triangle meshes, enriched with high resolution textures. Contrary to depth maps and airborne LIDAR scans, MVS meshes offer true 3D representations with unprecedented amount of details. However, despite exquisite resolution, such output meshes are considered raw as contain many defects and no semantic (Fig.2). In addition, they contain no information required to generate LODs that are controllable via intuitive parameters, and *meaningful* for applications such as interactive navigation, urban planning, computational engineering and video games. Meaningful herein relates to LODs that are coherent across the entire scene, allow for incremental refinement, and provide some level of abstraction. This explains our will to go beyond simplification through semantic- and structure-aware reconstruction with LODs (see Fig.1).

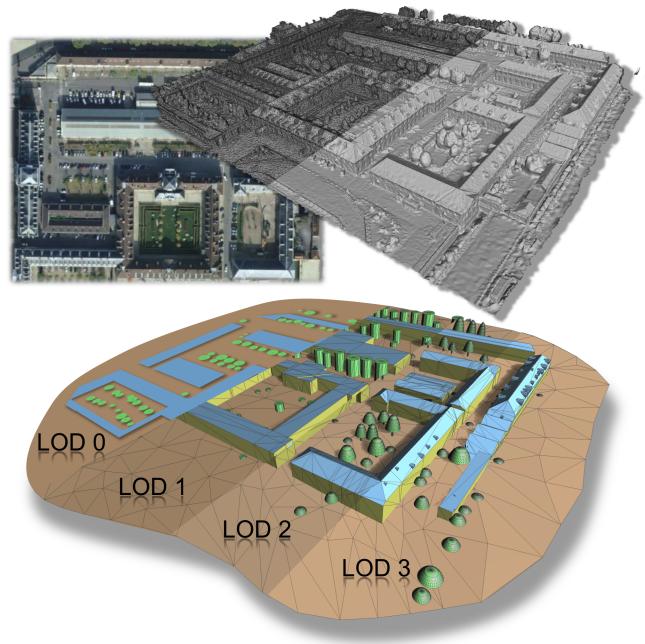


Figure 1: Reconstruction and LOD Generation. Starting from a raw surface mesh generated by a multi-view stereo workflow, our approach generates four compact levels of detail that are meaningful, abstracted and enriched with urban semantic.

1.1 Related Work

Our review of previous work covers the four main facets of our problem statement: reconstruction, abstraction, LOD generation and semantic specific to urban scenes.

Urban reconstruction. In our setup reconstruction amounts to turn the raw input data into LODs of a 3D urban scene composed of watertight buildings and an environment composed of ground and trees. This problem being ill-posed, the state-of-the art ranges from interactive [Arikan et al. 2013] to automated [Poullis and You 2009] through semi-automated approaches [Sinha et al. 2008]. A possible taxonomy of the literature is to distinguish between two types of input data: depth maps and LIDAR point sets. Depth maps are commonly generated from MVS images. The approaches proposed for generating compact 3D-models of buildings from depth maps proceed, e.g., in 2D through space partitioning [Zebedin et al. 2008] or in 3D through assemblies of cuboids generated by Monte Carlo sampling [Ortner et al. 2008]. In addition to being often hampered with high noise and providing only 2.5D view-dependent inputs, a major limitation of depth maps for our setup is that they prevent distinguishing buildings from high vegetation. LIDAR point set data became popular from the mid-2000 mostly for their accuracy, despite the fact that they are geometrically less structured than depth maps and do not contain any radiometric information. Urban LIDAR data stimulated a series of work mainly focused on parsing building components and extracting building contours. For LIDAR as well as for depth map data, a popular methodology consists of relying on 3D planar primitives for roofs and facades [Poullis and You 2009; Lafarge and Mallet 2011], with advances on parsing planes [Toshev et al. 2010] and discovering global regularities among planes [Zhou and Neumann 2012]. LIDAR data are both accurate and sufficiently dense when acquired from the street level,

but our setup requires airborne acquisition for large scale scenes. Airborne LIDAR data are thus insufficiently dense for our needs, which hampers classification and reconstruction of facades and prevents reconstructing roof super structures.

Some approaches address the urban reconstruction problem by inverse procedural modeling. Based on a grammar and related semantic rules, forward procedural modeling has no equivalent in terms of control over geometric complexity, structure and semantic [Bao et al. 2013]. However, inverse procedural modeling applied to airborne measurement data is an enduring scientific challenge: state-of-the-art approaches rely on simple grammars and require assumptions such as axis-aligned geometry that do not match our objective [Vanegas et al. 2010a].

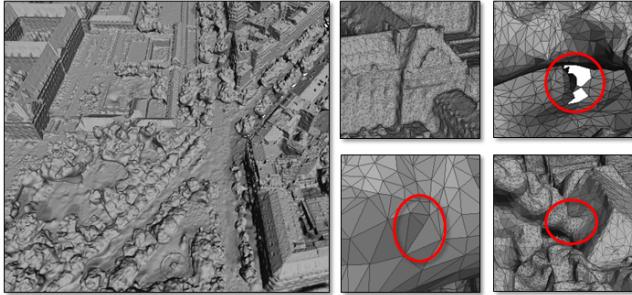


Figure 2: Raw surface mesh generated by a MVS workflow. The mesh is dense (in the order of 10M triangle facets per city block), semantic-free and defect-laden. It contains many geometric and topological defects such as holes, islands, self-intersections and merging of urban components from distinct classes such as trees and facades.

vegetation and ground. Existing approaches are however not directly applicable to our problem as our input data do not contain any additional properties such as echo number or signal magnitude which help distinguishing the classes of interest. Lin et al. [2013] decompose the elements of residential buildings through supervised learning, with however no abstraction nor LODs. In addition, our setup differs as targets unsupervised learning and dense urban scenes with global regularities.

In summary, and in spite of the variety of methods currently available to address each facet of our problem individually, there is a dire need for an automated reconstruction and LOD generation method applicable to dense MVS data measured on large scale urban scenes.

1.2 Positioning and Contributions

In the literature the process of semantization or abstraction is often an end. In this paper we view the semantized and abstracted LODs both as an end and a means to improve robustness, quality as well as efficiency of the reconstruction. Our positioning is inspired by recent work on joint reconstruction and semantization [Hane et al. 2013; Lin et al. 2013]. In our framework the main objects of interest are buildings and trees, and the structure to discover corresponds to the LODs used by CityGML [Groger and Plumer 2012], see Fig.3. A given LOD describes both the lower LODs and additional details enriching its structure and geometry. The generation of LOD is a crucial advantage as it allows us to target a large range of urban applications. Another positioning is efficiency: all design choices are made with an eye on performances, crucial for processing urban scenes ranging from a single building to entire city blocks or even districts spanning hundreds of buildings.

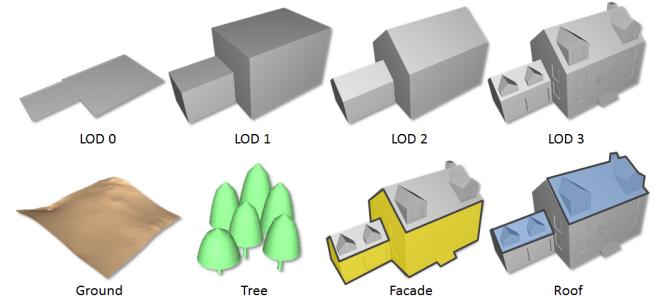


Figure 3: LODs used by cityGML and semantic. Top: LOD0 delineates the footprint of buildings and trees. LOD1 represents the building volume with flat roofs and trees as cylinder icons. LOD2 provides additional details with piecewise-planar roofs and half-ellipsoid icons for trees. LOD3 provides further details such as roof superstructures, doors and windows. Bottom: urban semantic used for reconstruction and LOD generation.

Our main contributions are as follows:

- A fully automated reconstruction pipeline that departs from existing work by the ability to (i) generate multiple coherent LODs, and (ii) take as input raw surface meshes generated by multi-view stereo workflows;
- Three new technical ingredients: (i) a feature-preserving Markov Random Field used for classification, (ii) a greedy process for the global regularization of planes with a hierarchical organization of the canonical geometric relationships, and (iii) a min-cut formulation applied to a discrete approximation of a 3D arrangement of planes for robust reconstruction.

2 Algorithm

The reconstruction algorithm proceeds with three main steps: classification (§2.1), abstraction (§2.2) and reconstruction (§2.3), see Fig.4.

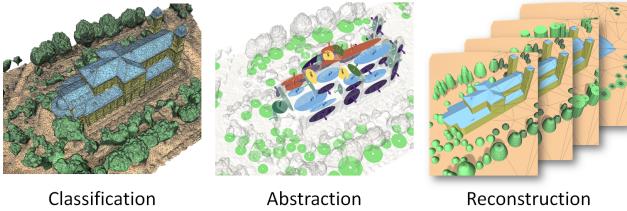


Figure 4: Main steps of our algorithm.

2.1 Classification

The classification step relies on a Markov Random Field (MRF) in order to distinguish between four classes of urban objects: *ground*, *tree*, *facade* and *roof*. As the classification is unsupervised we rely solely on geometric attributes using the following rationale: (i) *ground* is characterized by locally planar surfaces located below the other classes, (ii) *trees* have curved surfaces, (iii) *facades* are vertical surfaces adjacent to *roofs* and (iv) *roof* are mostly composed of piecewise-planar surfaces.

2.1.1 Superfacet Clustering

As the input meshes are very dense, classifying each triangle facet through the MRF would lead to unpractical computation times. In a pre-processing step we thus over-segment the input mesh into *superfacets*: sets of connected facets, similar in spirit to the notion of superpixels used for image analysis. Superfacets are obtained by clustering, through region growing, the facets with similar shape operator matrices. More specifically, we estimate the shape operator matrix [Cohen-Steiner and Morvan 2003] for each facet on a local spherical mesh neighborhood (radius 2 yards by default), and compare during clustering these matrices via the Frobenius distance. Fig.5(left) depicts how this clustering procedure identifies the nearly planar components and preserves the sharp features.

2.1.2 Geometric Attributes

Three geometric attributes are computed for each facet f_i of the input mesh:

- The **elevation** attribute a_e is defined as a function of the relative height (z coordinate) of the facet centroid, denoted by z_i :

$$a_e(f_i) = \sqrt{\frac{z_i - z_{min}}{z_{max} - z_{min}}}, \quad (1)$$

where $(z_{min}; z_{max})$ denote the height range of all facet centroids located within a local spatial neighborhood. The square root ensures that small values of relative height get a larger elevation attribute. The size of the neighborhood, set by default to 40 yards, must be sufficiently large to meet ground components and sufficiently small to gain resilience to hilly environments.

- The **planarity** attribute a_p denotes the planarity of the superfacet containing f_i , derived from the surface variation [Pauly et al. 2002]:

$$a_p(f_i) = 1 - \frac{3 \lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \quad (2)$$

where λ_0 denotes the minimum eigenvalues of the covariance matrix computed in closed form over all facets of the superfacet containing f_i [CGAL 2013].

- The **horizontality** attribute a_h measures the deviation of the unit normal \mathbf{n}_i to facet f_i with respect to the vertical axis:

$$a_h(f_i) = |\mathbf{n}_i \cdot \mathbf{n}_z|, \quad (3)$$

where \mathbf{n}_z denotes a unit vector along the Z coordinate axis.

From these per-facet geometric attributes, all taking values within $[0, 1]$, we compute the geometric attribute for each superfacet as the area-weighted sum of the geometric attributes of its facets. We compute similarly the normals of superfacets. Figure 5 illustrates the superfacet clustering and geometric attributes on a part of a urban scene.

2.1.3 Markov Random Field

From the geometric attributes computed per superfacet, a Markov Random Field is used to label each superfacet with one of the four classes: $\{\text{ground}, \text{tree}, \text{facade}, \text{roof}\}$. The defects of the raw input mesh require a regularized global optimization process offered by MRF which adds contextual as well as spatial consistency to the classification. More specifically, we use a MRF with pairwise superfacet interactions. The quality of a label configuration l is measured by energy U :

$$U(l) = \sum_{i \in S} D_i(l_i) + \gamma \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j), \quad (4)$$

where D_i and V_{ij} denote the unary data term and propagation constraints respectively, balanced by parameter $\gamma > 0$. S denotes the set of superfacets. E denotes all pairs of adjacent superfacets, two superfacets being adjacent if they share at least one edge in the input mesh. The data term combines the above-described attributes weighted by the area A_i of the superfacet i :

$$D_i(l_i) = A_i \times \begin{cases} 1 - a_p \cdot a_h \cdot \bar{a}_e & \text{if } l_i = \text{ground} \\ 1 - \bar{a}_p \cdot a_h & \text{if } l_i = \text{tree} \\ 1 - a_p \cdot \bar{a}_h & \text{if } l_i = \text{facade} \\ 1 - a_p \cdot a_h \cdot a_e & \text{if } l_i = \text{roof} \end{cases} \quad (5)$$

where $\bar{a}_e = 1 - a_e$. The pairwise interaction V_{ij} between two adjacent superfacets i and j favors label smoothness away from sharp creases:

$$V_{ij}(l_i, l_j) = C_{ij} \cdot w_{ij} \cdot 1_{\{l_i \neq l_j\}}, \quad (6)$$

where $1_{\{\cdot\}}$ denotes the characteristic function, and C_{ij} denotes the length of the interface between superfacets i and j (sum of interface edge lengths). Weight w_{ij} is introduced to lower the label propagation over sharp creases that often appear when two classes meet (e.g., for trees adjacent to facades, see Fig. 6). w_{ij} is defined as the angle cosine between the estimated normals of two superfacets. As the unary data term and pairwise potential are weighted by the superfacet areas and interface lengths, this energy formulation behaves similarly to a facet-based energy with grouping constraints.

An approximate solution to this energy minimization problem is solved through the $\alpha - \beta$ swap algorithm [Boykov et al. 2001], while setting by default γ to 0.5 in all shown experiments.

2.1.4 Semantic Rules

The aforementioned geometric rationale alone is not sufficient to solve the ill-posed classification problem. Two types of errors frequently occur when dealing with complex urban scenes: (i) roof superstructures such as chimneys or dormer-widows may be wrongly

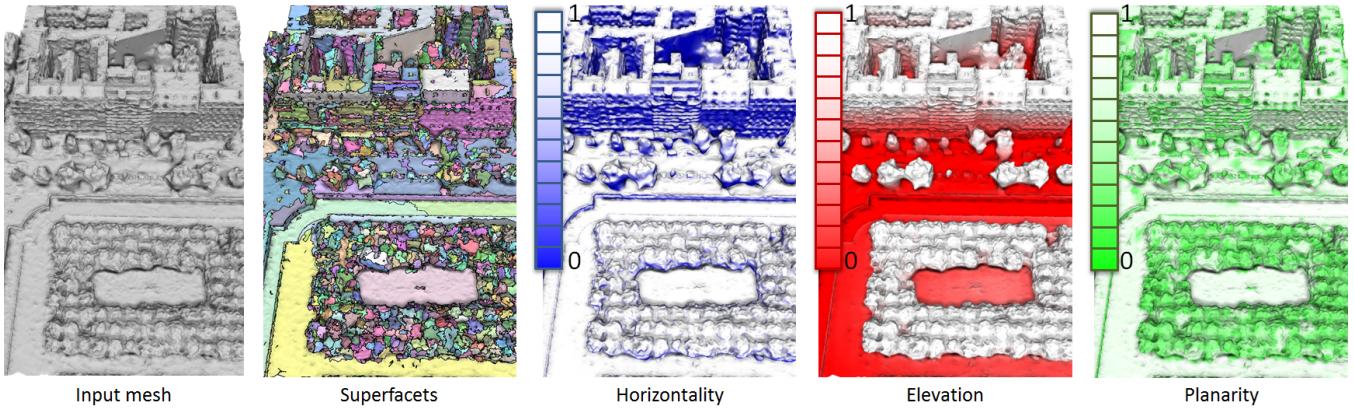


Figure 5: Superfacet clustering and geometric attributes used for classification.

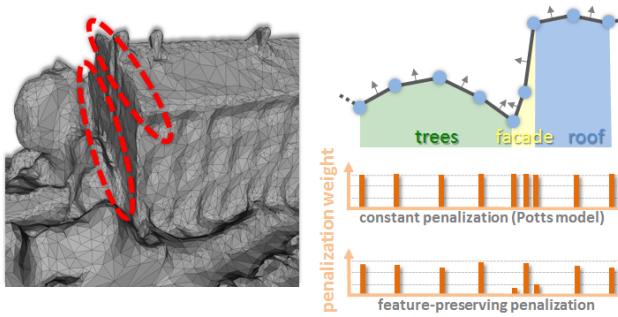


Figure 6: Soft feature-preserving regularization. Departing from the common uniform Potts model, our pairwise interaction term penalizes label propagation over sharp creases by taking into account the normal variation of the superfacets.

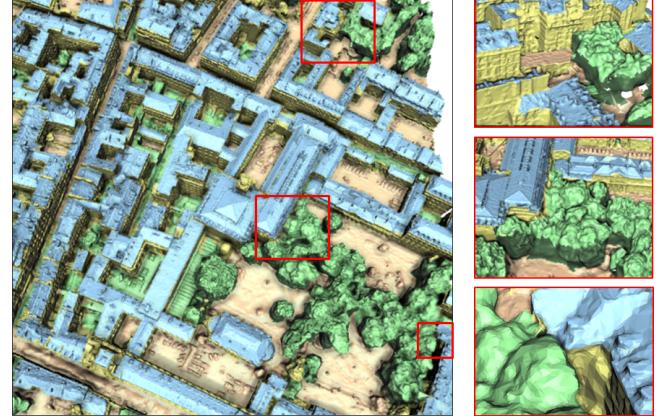


Figure 7: Classification into four classes of urban elements: roof (blue), facade (yellow), ground (brown) and trees (green). The regularizing term of the energy as well as the semantic rules improve spatial consistency. The close-ups depict how roofs and facades, as well as trees adjacent to facades, are adequately separated.

labeled as *tree*, these elements being too small and irregular to be considered locally planar, and (ii) vertical components of large trees may be labeled as *facade*. We thus add the following semantic rules:

- Rule 1. superfacets labeled as *tree* and adjacent to only superfacets labeled as *roof* are re-labeled *roof*. This rule relies on the common assumption that large trees are not located on top of roofs.
- Rule 2. superfacets labeled as *facade* and adjacent to superfacets labeled as *tree* and *ground* are turned to *tree*.

As illustrated by Fig. 7, these two rules bring more contextual coherence to the semantic labeling in presence of small irregular roof superstructures and trees with cylindrical shapes. Finally, after classification we decompose the scene into connected components: isolated buildings or blocks of connected buildings are extracted by searching for connected superfacets labeled as *roof* and *facade*. Isolated trees or forest are extracted using a similar process. Such decomposition greatly reduces the complexity of the reconstruction step as each connected block of buildings or trees can be reconstructed independently.

ture simplification through planar shape approximation and regularization, iconization and LOD-based filtering. Remind that superfacets are classified into four classes: *ground*, *tree*, *facade* and *roof*, the latter containing roof superstructures. A key idea behind our approach is to specialize the abstraction and reconstruction steps to these classes as well as to the LODs, such that, e.g., trees are represented by icons instead of attempting to reconstruct them with planar proxies. In addition, regularization is our means to improve scalability and robustness of the reconstruction step.

The classes are abstracted as follows:

- **Ground** is simply represented by a regular 2D triangulation lifted in 3D with a natural neighbor interpolation of the maximum elevation attribute of the input mesh superfacets labeled as *ground* (see §2.1).
- **Facades and roofs** are approximated by a set of planar proxies with reinforced regularities, these proxies being used as input to the final watertight reconstruction step (§2.3). We restrict ourselves to planar proxies as planar surfaces cover on average 80% of urban areas, and are amenable to effective abstraction and reconstruction for LOD generation.
- **Roof super-structures and trees** are abstracted through

262 labeled as *tree*, these elements being too small and irregular to be
263 considered locally planar, and (ii) vertical components of large trees
264 may be labeled as *facade*. We thus add the following semantic
265 rules:

- 266 • Rule 1. superfacets labeled as *tree* and adjacent to only superfacets
267 labeled as *roof* are re-labeled *roof*. This rule relies on
268 the common assumption that large trees are not located on top
269 of roofs.
- 270 • Rule 2. superfacets labeled as *facade* and adjacent to superfacets
271 labeled as *tree* and *ground* are turned to *tree*.

272 As illustrated by Fig. 7, these two rules bring more contextual
273 coherence to the semantic labeling in presence of small irregular
274 roof superstructures and trees with cylindrical shapes. Finally, after
275 classification we decompose the scene into connected components:
276 isolated buildings or blocks of connected buildings are extracted by
277 searching for connected superfacets labeled as *roof* and *facade*.
278 Isolated trees or forest are extracted using a similar process. Such
279 decomposition greatly reduces the complexity of the reconstruction
280 step as each connected block of buildings or trees can be recon-
281 structed independently.

2.2 Abstraction

283 From the superfacets classified in §2.1 the abstraction step cre-
284 ates compact descriptions involving characteristic icons and planar
285 proxies. Compact descriptions are obtained by geometric and struc-

307 ionization on two distinct height maps. We do not approxi-
 308 mate roof superstructures with planar proxies due to the lim-
 309 ited resolution of airborne MVS meshes (typically only a few
 310 triangles for a chimney). For a similar reason there is no spe-
 311 cific superstructure class, and, e.g., chimneys are re-labeled
 312 from *tree* to *roof* during classification (see §2.1).

313 The icons and regularized proxies are then filtered through LOD
 314 generation, before reconstruction.

315 2.2.1 Planar Proxies

316 We first identify a set of near-planar superfacets by selecting the
 317 ones labeled as *roof* or *facade*, with a high planarity attribute a_p
 318 and a minimum large area (by default we set $a_p > 0.7$ and area
 319 larger than 10 square yards). For each near-planar superfacet we
 320 compute its least-squares fitting plane [CGAL 2013], referred to as
 321 superfacet proxy. We then improve globally the regularity of these
 322 proxies by altering their orientation and position so as to reinforce
 323 their canonical geometric relationships. Departing from existing
 324 approaches such as GlobFit [Li et al. 2011] or LIDAR-specific al-
 325 gorithms [Zhou and Neumann 2012], we perform a greedy process
 326 in which the geometric relationships are hierarchically organized.
 327 We also introduce a novel Z-symmetry geometric relationship relevant
 328 for abstracting building roofs.

329 Denote by P_1 and P_2 , two proxies having respective unit normals
 330 \mathbf{n}_1 and \mathbf{n}_2 and centroids c_1 and c_2 . We define four canonical rela-
 331 tionships under an orientation tolerance ϵ and an Euclidean distance
 332 tolerance d :

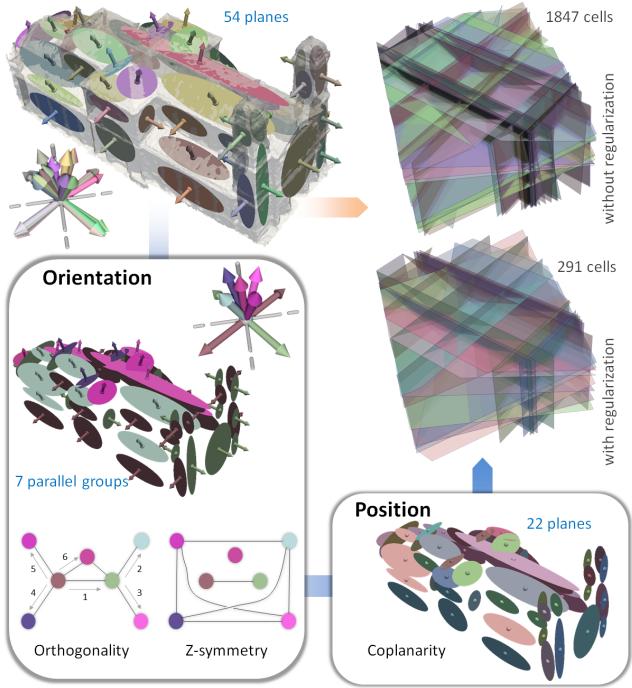
- 333 • *Parallelism.* P_1 and P_2 are ϵ -parallel if $|\mathbf{n}_1 \cdot \mathbf{n}_2| \geq 1 - \epsilon$;
- 334 • *Orthogonality.* P_1 and P_2 are ϵ -orthogonal if $|\mathbf{n}_1 \cdot \mathbf{n}_2| \leq \epsilon$;
- 335 • *Z-symmetry.* P_1 and P_2 are ϵ -Z-symmetric if $||\mathbf{n}_1 \cdot \mathbf{n}_z| - |\mathbf{n}_2 \cdot \mathbf{n}_z|| \leq \epsilon$, where \mathbf{n}_z is the unit vector along the vertical axis;
- 336 • *Coplanarity.* P_1 and P_2 are d - ϵ -coplanar if they are ϵ -parallel and $|d_{\perp}(c_1, P_2) + d_{\perp}(c_2, P_1)| < 2d$, where $d_{\perp}(c, P)$ denotes the orthogonal distance between point c and proxy P .

340 The first three relationships are relating to the proxy orientations,
 341 and coplanarity is a particular case of parallelism with an additional
 342 relative positioning constraint. The notion of Z-symmetry matches
 343 the common assumption that connected components of roofs tend
 344 to share similar slope values.

345 Regularization proceeds as follows. We first cluster the proxies
 346 which are ϵ -parallel proxies into parallel clusters, and compute the
 347 average orientation of each cluster. For efficiency the next two steps
 348 act on clusters instead of individual proxy. We then construct an or-
 349 thogonality graph with one node per parallel cluster, and one edge
 350 between two nodes when they are ϵ -orthogonal. We proceed sim-
 351ilarly with a Z-symmetry graph. Without altering the structure of
 352 the graphs and the centroid of each proxy, we then alter the orienta-
 353 tion of the nodes by propagating the orthogonality and Z-symmetry
 354 constraints greedily along the edges of the orthogonality graph (in
 355 general with a larger number of edges), from large to small nodes
 356 (see numbers on the orthogonality graph depicted by Fig.8).

357 The size of a node refers to the total area of its associated su-
 358 perfacets. Denote respectively by source and target node a pair
 359 of nodes altered by the propagation. The initial orientation of the
 360 target node is altered by constraining its normal to match the con-
 361 straints (orthogonal and sometimes also Z-symmetry) with respect
 362 to the source node. When only one constraint is propagated we
 363 choose the constrained orientation that best aligns to the initial ori-
 364 entation. When the two constraints are propagated there is in gen-
 365 eral a unique orientation that is both orthogonal and Z-symmetric.

366 When no solution exists due to relationships contradictions along
 367 edges of the graph or when the constrained orientation deviates too
 368 much from the initial orientation (dot product between initial and
 369 altered normals lower than $1 - \epsilon$) we undo the alteration and restore
 370 the initial normal. Finally, we also detect d - ϵ -coplanarity among
 371 parallel clusters and compute new proxy positions by clustering,
 372 in 1D along the normal of each parallel cluster, the area-weighted
 373 centroid of each proxy after projection onto the said normal. Fig.8
 374 illustrates the global regularization process.



375 **Figure 8: Regularization.** Partitioning the space from the initial
 376 set of proxies (top left) leads to a complex and irregular set of vol-
 377 umetric cells (top right). Each color in the orientation step denotes
 378 a parallel cluster. Similarly, each color in the position step de-
 379 notes a coplanar cluster. The graphs depict the orthogonality and
 380 Z-symmetry relationships among the 7 parallel clusters. Numbers
 381 on edges of the orthogonality graph denote the greedy propagation
 382 order of orthogonality constraints. The space partition generated
 383 from the regularized proxies is less complex and more regular (mid-
 384 dle right).

385 2.2.2 Iconization

386 The iconization step is devised to abstract trees and roof superstruc-
 387 tures from height maps. For trees we construct a height map by
 388 rasterizing in an image the input mesh in XY coordinate plane and
 389 taking as height value the maximum elevation attribute restricted to
 390 superfacets labeled as *tree*. We greedily extract the local maxima of
 391 this map by a watershed algorithm in order to locate the center of
 392 each tree icon in the XY plane, and fit the best half ellipsoid to the
 393 map while keeping the center fixed, similar to [Lafarge and Mallet
 394 2011]. For roof superstructures, which are used only for LOD3,
 395 we construct a height map by rasterizing in an image the difference
 396 between the maximum elevation attribute restricted to superfacets
 397 labeled as *roof*, and the LOD2 model generated by the watertight
 398 reconstruction process described in Section §2.3. We then locate the
 399 center of each superstructure icon similarly to the tree icons, and fit
 a 3D template icon made of two superimposed parallelepipeds.

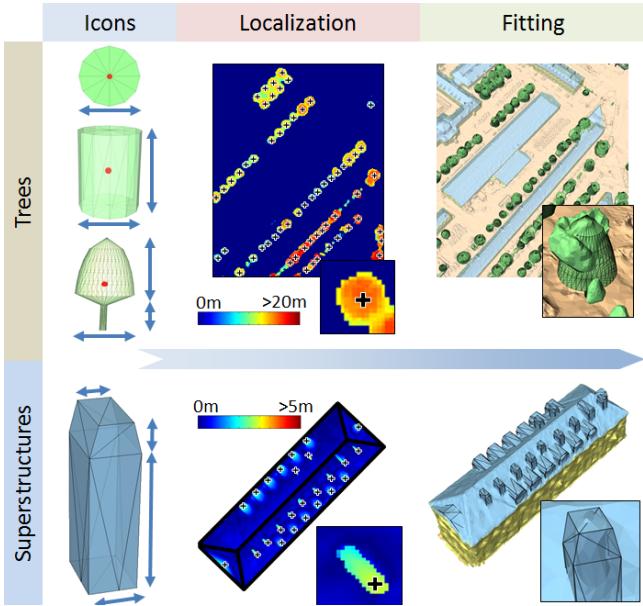


Figure 9: Iconization of trees and roof superstructures.

423 meshes that preserve the structural components of the scene at the
424 chosen LOD, and completes missing parts of the scene in a plausible
425 manner.

426 Even when restricting it to each building component, computing
427 the complete, exact arrangement leads to very high computational
428 complexity (we experimented with scenes containing hundreds of
429 components, each containing on average hundred planes). Previous
430 work based on arrangements attempted to reduce such complexity
431 by restricting to axis-aligned planes [Furukawa et al. 2009] or by
432 computing a two-level hierarchy made up of a rectilinear volumetric
433 grid combined with a convex polyhedral cell decomposition
434 [Chauve et al. 2010]. The former work is too restrictive and the
435 latter approach exceeds half an hour when dealing with more than
436 few hundred planes. Observing that only a very small subset of the
437 faces of the arrangement contribute to the output after solving for a
438 min-cut surface, we postpone the exact geometric computation oper-
439 ations to the final surface extraction step after min-cut solve, and
440 rely instead on a transient discrete approximation of the arrange-
441 ment so as to avoid the compute-intensive exact geometric opera-
442 tions required to insert each plane into the arrangement.

443 For each sub-part of the input MVS corresponding to a building
444 component we first compute an object-oriented bounding box B .
445 We then sample uniformly B by placing sample points at the cor-
446 ners of a uniform grid aligned to B . Each of these sample points,
447 referred to as *anchors*, is associated to (i) a Boolean localiza-
448 tion flag specifying whether the anchor is estimated to be inside or out-
449 side the inferred building, and (ii) an integer index denoting the
450 cell of the arrangement containing the anchor. We iterate over each
451 anchor and guess its inside/outside flag with respect to the inferred
452 building by casting rays and counting the intersection parity of these
453 rays against the input MVS mesh using an AABB tree data struc-
454 ture. Five rays have shown sufficient in all experiments: four to-
455 wards the upper corners of B and one towards its barycenter. We
456 then compute an approximate arrangement of planes by inserting
457 iteratively all planes of B , then the planar proxies, while refining
458 an arrangement tree. Instead of computing the exact geometry of
459 the arrangement cells for each plane insertion we update the anchor
460 cell indices, see Fig.10. The anchor are used as quadrature points
461 to approximate geometric information of cells (volume, facet areas
462 and adjacency).

463 2.3.2 Min-Cut Formulation

464 For each arrangement a min-cut formulation is used to find an in-
465 side/outside labeling of the cells, the output surface being defined
466 as the interface facets between inside and outside. Consider a graph
467 $(\mathcal{C}, \mathcal{F})$ where $\mathcal{C} = \{c_1, \dots, c_n\}$ denotes the nodes relating to the
468 cells induced by the space partition, and $\mathcal{F} = \{f_1, \dots, f_m\}$ de-
469 notes edges relating to the facets separating all pairs of adjacent
470 cells. A cut in the graph consists of separating the cells \mathcal{C} into two
471 disjoint sets \mathcal{C}_{in} and \mathcal{C}_{out} . The edges between \mathcal{C}_{in} and \mathcal{C}_{out} corre-
472 spond to a set of facets forming a surface $\mathcal{S} \subset \mathcal{F}$.

473 In order to quantize the quality of the solution, i.e., the surface \mathcal{S} in-
474 duced by the cut $(\mathcal{C}_{in}, \mathcal{C}_{out})$, we introduce the following cost func-
475 tion C :

$$476 C(\mathcal{S}) = \sum_{c_k \in \mathcal{C}_{out}} V_{c_k} g(c_k) + \sum_{c_k \in \mathcal{C}_{in}} V_{c_k} (1 - g(c_k)) + \beta \sum_{f_i \in \mathcal{S}} A_{f_i}, \quad (7)$$

477 where V_{c_k} denotes the volume of cell c_k , $g(c_k)$ denotes the function
478 estimating the label likelihood of cell c_k with respect to the ratio
479 of its inside/outside anchors, and A_{f_i} denotes the discrete area of
480 facet f_i . The first two terms of the cost function C are data terms
481 whereas the third term weighted by parameter $\beta \geq 0$ acts as a
482 regularization term in order to favor solutions with small area. The

412 **2.2.3 LOD Generation**

413 The LOD generation step proceeds by filtering the regularized prox-
414 ies and abstracting the icons, in accordance to the urban LODs used
415 by CityGML:
416

- 417 • LOD0: ground mesh is not used as the representation is planar. Trees are depicted as discs computed as vertical projec-
418 tion of tree icons, and buildings are depicted by 2D regions
419 bounded by polylines computed only from the abstracted
420 proxies labeled as *facade* using a 2D instance of the min-cut
421 formulation (§2.3). Superstructures are omitted.
- 422 • LOD1: ground mesh, enriched with vertical cylinders for
423 trees and a LOD0-building elevated in 3D with horizontal
424 proxies as roofs whose height is computed by median of cor-
425 responding superfacet height.
- 426 • LOD2: ground mesh, enriched with tree icons, and building
427 reconstructed (§2.3) with all proxies to generate piecewise-
428 planar roofs.
- 429 • LOD3: LOD2 enriched with roof superstructure. We do not
430 attempt at reconstructing details on facades such as doors and
431 windows as we are dealing with airborne measurement data
432 with insufficient resolution on vertical structures (see Fig.2).

433 **2.3 Reconstruction**

434 The final reconstruction step turns the proxies regularized and fil-
435 tered in previous step into watertight buildings. We instantiate a se-
436 ries of 3D arrangements, and label the cells of these arrangements
437 as inside or outside via a min-cut formulation.

438 **2.3.1 3D Arrangements**

439 For each connected component of buildings identified in §2.1, a 3D
440 arrangement of planes provides us with a means to assemble the
441 planar proxies into well-behaved surfaces: watertight and free of
442 self-intersection. When combined with global regularization, LOD
443 filtering and min-cut, it furthermore generates lightweight polygon

444 meshes that preserve the structural components of the scene at the
445 chosen LOD, and completes missing parts of the scene in a plausible
446 manner.

447 Even when restricting it to each building component, computing
448 the complete, exact arrangement leads to very high computational
449 complexity (we experimented with scenes containing hundreds of
450 components, each containing on average hundred planes). Previous
451 work based on arrangements attempted to reduce such complexity
452 by restricting to axis-aligned planes [Furukawa et al. 2009] or by
453 computing a two-level hierarchy made up of a rectilinear volumetric
454 grid combined with a convex polyhedral cell decomposition
455 [Chauve et al. 2010]. The former work is too restrictive and the
456 latter approach exceeds half an hour when dealing with more than
457 few hundred planes. Observing that only a very small subset of the
458 faces of the arrangement contribute to the output after solving for a
459 min-cut surface, we postpone the exact geometric computation oper-
460 ations to the final surface extraction step after min-cut solve, and
461 rely instead on a transient discrete approximation of the arrange-
462 ment so as to avoid the compute-intensive exact geometric opera-
463 tions required to insert each plane into the arrangement.

464 For each sub-part of the input MVS corresponding to a building
465 component we first compute an object-oriented bounding box B .
466 We then sample uniformly B by placing sample points at the cor-
467 ners of a uniform grid aligned to B . Each of these sample points,
468 referred to as *anchors*, is associated to (i) a Boolean localiza-
469 tion flag specifying whether the anchor is estimated to be inside or out-
470 side the inferred building, and (ii) an integer index denoting the
471 cell of the arrangement containing the anchor. We iterate over each
472 anchor and guess its inside/outside flag with respect to the inferred
473 building by casting rays and counting the intersection parity of these
474 rays against the input MVS mesh using an AABB tree data struc-
475 ture. Five rays have shown sufficient in all experiments: four to-
476 wards the upper corners of B and one towards its barycenter. We
477 then compute an approximate arrangement of planes by inserting
478 iteratively all planes of B , then the planar proxies, while refining
479 an arrangement tree. Instead of computing the exact geometry of
480 the arrangement cells for each plane insertion we update the anchor
481 cell indices, see Fig.10. The anchor are used as quadrature points
482 to approximate geometric information of cells (volume, facet areas
483 and adjacency).

484 **2.3.2 Min-Cut Formulation**

485 For each arrangement a min-cut formulation is used to find an in-
486 side/outside labeling of the cells, the output surface being defined
487 as the interface facets between inside and outside. Consider a graph
488 $(\mathcal{C}, \mathcal{F})$ where $\mathcal{C} = \{c_1, \dots, c_n\}$ denotes the nodes relating to the
489 cells induced by the space partition, and $\mathcal{F} = \{f_1, \dots, f_m\}$ de-
490 notes edges relating to the facets separating all pairs of adjacent
491 cells. A cut in the graph consists of separating the cells \mathcal{C} into two
492 disjoint sets \mathcal{C}_{in} and \mathcal{C}_{out} . The edges between \mathcal{C}_{in} and \mathcal{C}_{out} corre-
493 spond to a set of facets forming a surface $\mathcal{S} \subset \mathcal{F}$.

494 In order to quantize the quality of the solution, i.e., the surface \mathcal{S} in-
495 duced by the cut $(\mathcal{C}_{in}, \mathcal{C}_{out})$, we introduce the following cost func-
496 tion C :

497
$$498 C(\mathcal{S}) = \sum_{c_k \in \mathcal{C}_{out}} V_{c_k} g(c_k) + \sum_{c_k \in \mathcal{C}_{in}} V_{c_k} (1 - g(c_k)) + \beta \sum_{f_i \in \mathcal{S}} A_{f_i}, \quad (7)$$

499 where V_{c_k} denotes the volume of cell c_k , $g(c_k)$ denotes the function
500 estimating the label likelihood of cell c_k with respect to the ratio
501 of its inside/outside anchors, and A_{f_i} denotes the discrete area of
502 facet f_i . The first two terms of the cost function C are data terms
503 whereas the third term weighted by parameter $\beta \geq 0$ acts as a
504 regularization term in order to favor solutions with small area. The

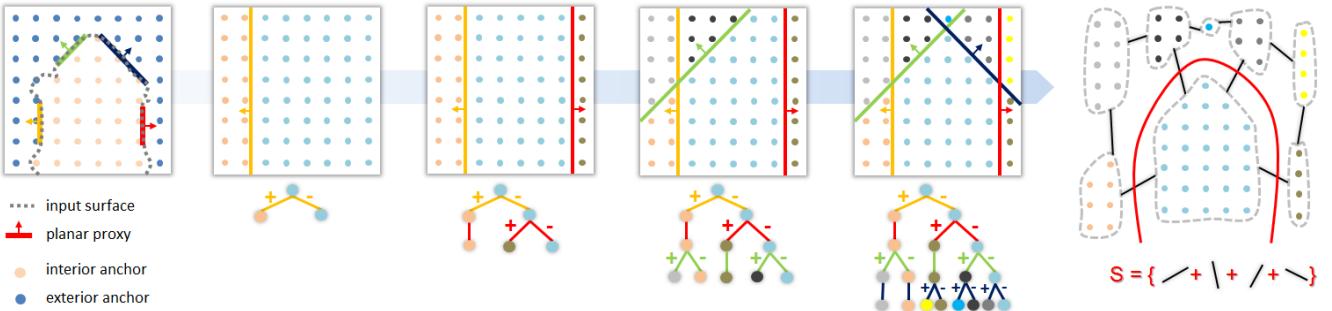


Figure 10: Discrete arrangement. We guess the inside/outside flag of each anchor by ray casting the input mesh (left). At each plane insertion, the anchors (top) as well as the arrangement tree (bottom) are updated. The anchors are then partitioned into discrete cells (right, colored points) from which we compute a discrete volume or the ratio of inside/outside anchors, and identify the adjacent cells. After solving for optimal cut (red) we collect all cut edges of the graph (black) and compute the output surface by plane intersection.

optimal cut minimizing the cost $C(\mathcal{S})$ is found via the max-flow algorithm [Boykov and Kolmogorov 2004].

Smart3DCapture and Autodesk 123DCatch) as well as with defect-free meshes used to evaluate robustness and accuracy. Fig.12 illustrates our algorithm at work on various types of urban scenes ranging from residential houses to dense urban blocks through architectural buildings. The reconstructions match our initial goal to generate meaningful levels of details: the semantic is recovered, the structures are preserved and the details are coherent across the scene.

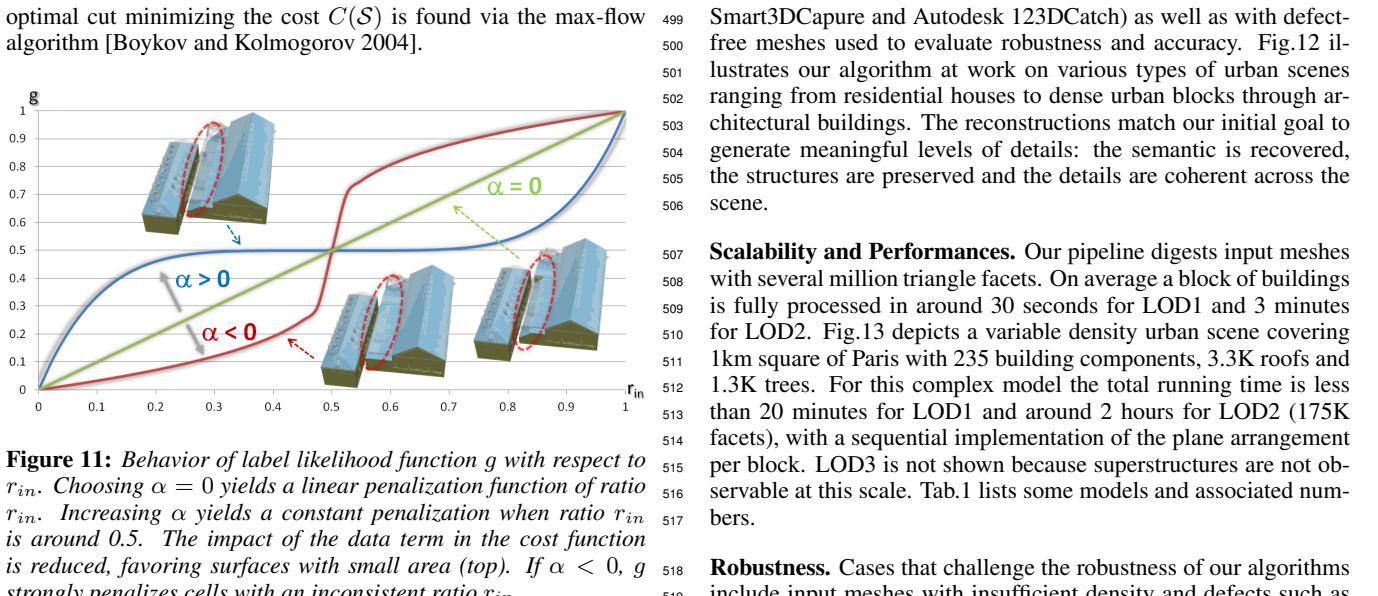


Figure 11: Behavior of label likelihood function g with respect to r_{in} . Choosing $\alpha = 0$ yields a linear penalization function of ratio r_{in} . Increasing α yields a constant penalization when ratio r_{in} is around 0.5. The impact of the data term in the cost function is reduced, favoring surfaces with small area (top). If $\alpha < 0$, g strongly penalizes cells with an inconsistent ratio r_{in} .

Function $g(c_k)$, defined in the interval $[0, 1]$, quantizes the coherence of assigning label *inside* to cell c_k with ratio r_{in} of inside anchors contained in c_k :

$$g(c_k) = \frac{(2r_{in} - 1) \times |2r_{in} - 1|^\alpha + 1}{2}, \quad (8)$$

where α denotes a parameter tuning the data sensitivity of function g , as illustrated by Fig.11. The optimal cut corresponds to a subset of facets separating the inside and outside cells, as depicted by Fig.10. The final geometry of these interface facets is then computed with exact arithmetic by intersecting the set of corresponding planes. By construction each interface facet is thus a planar convex polygon. For LOD0 and LOD1 we create a 2D instance of such discrete arrangement and min-cut formulation by sampling a single horizontal layer of anchors.

Scalability and Performances. Our pipeline digests input meshes with several million triangle facets. On average a block of buildings is fully processed in around 30 seconds for LOD1 and 3 minutes for LOD2. Fig.13 depicts a variable density urban scene covering 1km square of Paris with 235 building components, 3.3K roofs and 1.3K trees. For this complex model the total running time is less than 20 minutes for LOD1 and around 2 hours for LOD2 (175K facets), with a sequential implementation of the plane arrangement per block. LOD3 is not shown because superstructures are not observable at this scale. Tab.1 lists some models and associated numbers.

Robustness. Cases that challenge the robustness of our algorithms include input meshes with insufficient density and defects such as noise, holes and overlaps. Fig.12(fourth row) shows that small scale roofs may not be reconstructed in LOD2 but are recovered in LOD3 as roof superstructures. Imperfect input data often lead to over- or under-detected planar proxies. We observe that over-detection is often compensated by the proxy regularization procedure that merges nearly-coplanar proxies. Under-detection however leads to very few proxies as observed on free-form architectural buildings, and hence to an overly abstracted reconstruction. Nevertheless, in the worst case where no proxies are detected for a building component, the output LOD is abstracted as its bounding box. Data that challenge the classification step include merged objects such as a tree touching a facade, and clutter elements such as cars or hedges digested by the four classes of interest. The regularization term of the energy together with the semantic rules improve spatial consistency and reduce the number of classification errors. Fig.14 evaluates the robustness of the proxy detection and abstraction on an input mesh with variable scale features, noise and smoothed features.

3 Results

Our algorithm is implemented in C++ using the CGAL library, an $\alpha - \beta$ swap and a max-flow library [Boykov and Kolmogorov 2004; Boykov et al. 2001]. All timings are measured on an Intel Core i7 clocked at 2GHz. We experiment with real-world meshes generated by state-of-the-art multi-view stereo workflows (Acute3D

537 Accuracy and abstraction. Fig.15 evaluates the accuracy of the
538 reconstructed LODs against the input meshes, albeit our approach
539 is designed to provide a tradeoff between faithfulness to input
540 data and structure-aware abstraction. The comparisons against
541 two mesh approximation approaches [Garland and Heckbert 1997;
542 Cohen-Steiner et al. 2004], referred to as QEM and VSA respec-
543 tively, show comparable approximation errors, better resilience to

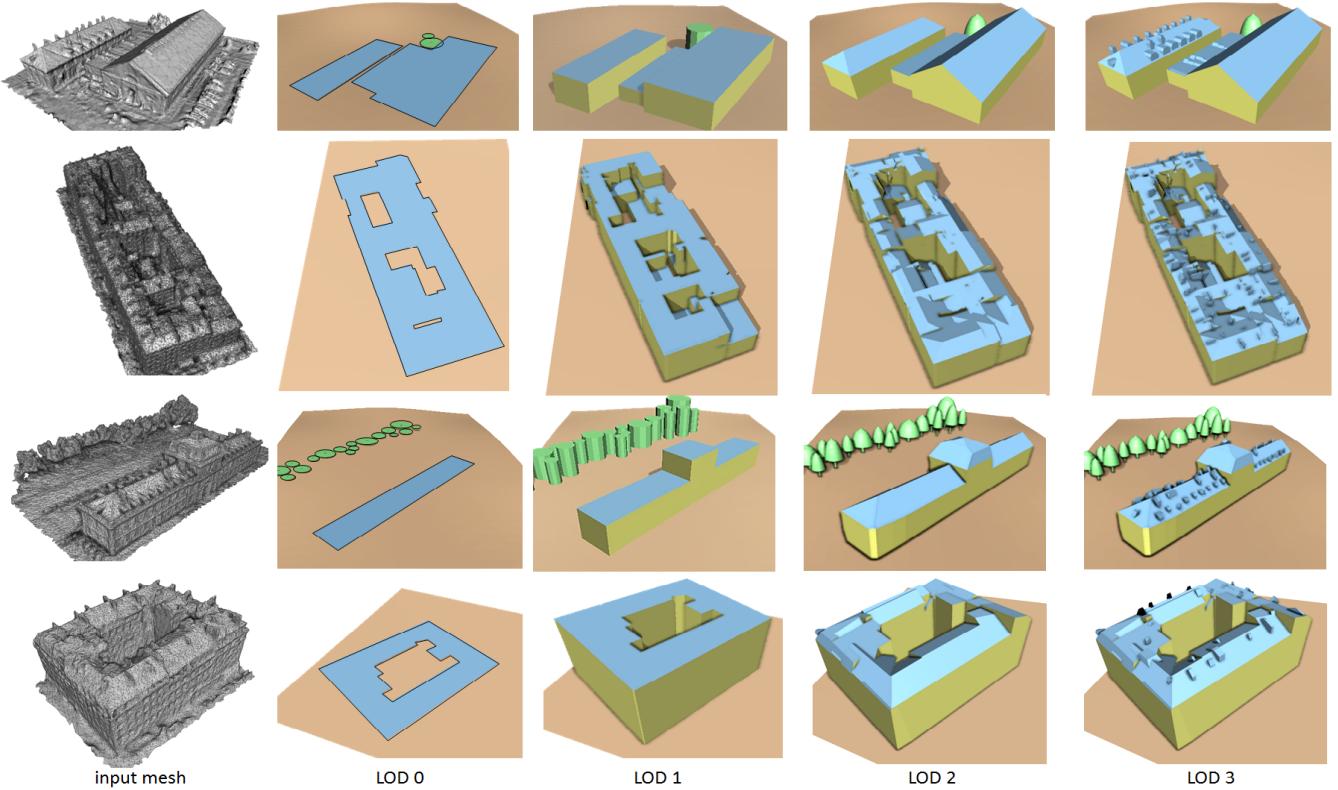


Figure 12: Reconstruction and LOD Generation. First row: on this simple residential scene all facades and roofs are well classified and the Z-symmetry relationships between the two types of roof (2 and 4 slopes) enables abstraction. Second row: on this dense urban component each roof is simple but all roofs form a complex arrangement as the buildings have been built at different times with little coherence. Third row: on this architectural building both Z-symmetry and orthogonal relationships cooperate to abstract the central part of the roof. Fourth row: this building contains complex and thin roof superstructures. Despite a limited accuracy of the input MVS mesh our method recovers the main facades and roofs, and most superstructures.

Input mesh	Classification	Planar proxies	Iconization	LOD1 reconstruction	LOD2 reconstruction	Model complexity
Church (59K facets, Fig.15)	5s	1.5s	2s	41s	198s	190 facets
Building block (170K facets, Fig. 12, 2nd row)	7s	1.1s	1.1s	21s	137s	456 facets
Invalides district (11M facets, Fig. 13)	55s	95s	36s	17min	112min	175K facets

Table 1: Running times and output complexity. The time required for reconstruction LOD2 and LOD3, as well as for extracting LOD0 and LOD1 are similar as the time required for iconization and height filtering is negligible. The complexity refers only to the number of polygon facets of the building models at LOD2, the trees and superstructures being omitted.

holes and topological artifacts of the input mesh through the arrangement of planes, and better coherence and preservation of thin structures across LODs such as the square church towers. Notice how LOD3 represents roof details such as chimneys and dormer-windows while keeping a low polygon count.

Input data. Deciding upon the best type of input data for urban reconstruction is a recurrent dilemma [Leberl et al. 2010]. While most existing approaches take point clouds as input, in particular LIDAR scans, we argue that dense meshes generated by MVS workflows offer significant advantages. Contrary to point clouds, MVS meshes have richer geometric and topological description derived from photo-consistency principles. To our knowledge, none of the existing surface reconstruction methods from point clouds are able to combine semantization and photo-consistency in order to generate abstracted LODs. Fig.16 describes a quantitative comparison of our approach against two specialized urban reconstruction methods based on planimetric arrangement from LIDAR scans [Lafarge and Mallet 2011] and point set structuring [Lafarge and

Alliez 2013].

Limitations. We limited the classification to four common classes of urban objects as they provide a satisfactory tradeoff between robustness and quality of the reconstruction (we found only few errors during visual inspection of the large scale scene at LOD1 and LOD2 against the airborne tiled image, see Fig. 13). Such limitation however hampers the reconstruction of other urban structures such as bridges or elevated roads. The use of planar proxies is also a limitation when dealing with freeform architecture buildings such as the dome of *Les Invalides* depicted by Fig.13.

4 Conclusions

Our work on reconstruction of urban scenes provides an automated framework to generate semantic-aware LODs from raw MVS meshes. The four LODs generated are both meaningful and refinable thanks to a coherent design of the abstraction and reconstruc-

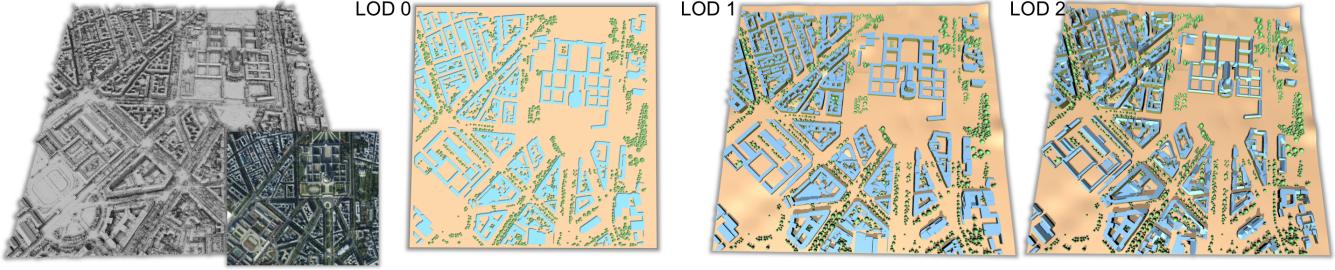


Figure 13: Reconstruction on large-scale urban scene. The input mesh (11M triangle facets) was generated from 600 airborne images. LOD1 and LOD2 comprise 10K and 175K polygon facets respectively, excluding tree and ground meshes.

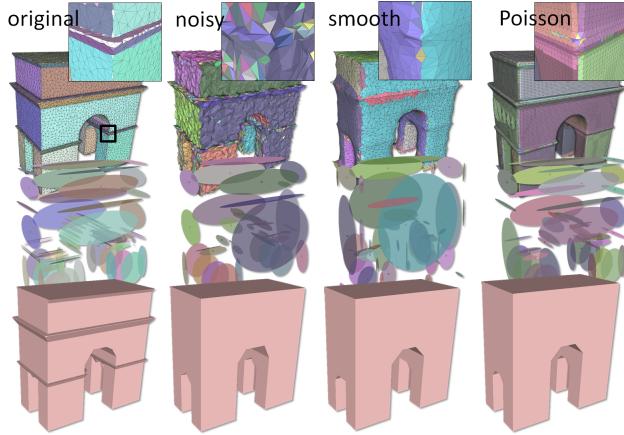


Figure 14: Robustness. Left: “defect-free” input mesh colored by superfacets (top) and its planar proxies (middle). Our reconstruction algorithm (applied here with no classification to evaluate only the proxy detection and abstraction steps) recovers most features (bottom). Notice the curved area reconstructed by planar polygons. Middle left: when noise is added the small scale features are filtered out and the vault is overly simplified. Right: when fed with the output of the Poisson reconstruction method the behavior of the algorithm is similar to the one on the smoothed mesh (middle right).

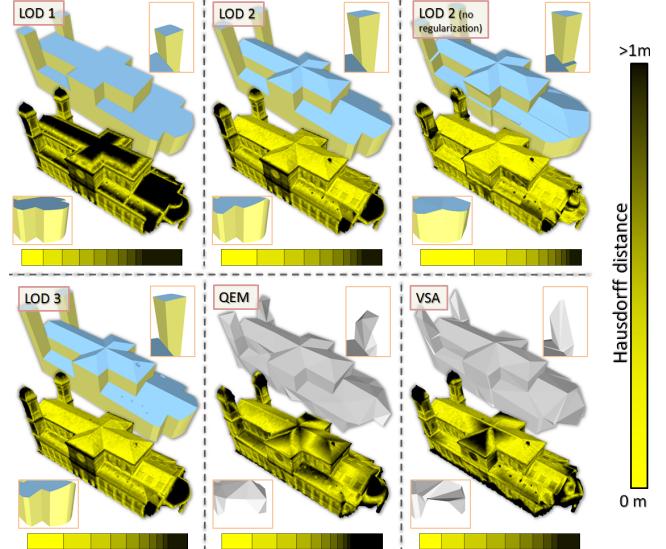


Figure 15: Geometric accuracy and structure-awareness. We compare the LODs to two mesh approximation algorithms by measuring the Hausdorff distance (color scale from yellow to black) to the input mesh. The complexity of the LOD2, QEM and VSA models is identical (190 facets). The root mean square error (RMS) of LOD1 is higher than for QEM and VSA (0.47 vs 0.4 and 0.43 respectively), the roof being poorly approximated. LOD2 without plane regularization has a lower RMS than LOD2 with planar regularization but is less abstracted and consumes less time to reconstruct (0.33s vs 0.39s). In terms of structure-awareness, thin components such as the church towers are correctly preserved in the different LODs, which is not the case for mesh approximation algorithms (see top close-ups). In addition, QEM and VSA do not fill the holes contained in the input mesh (see bottom close-ups).

601 structures as buildings. Methods specialized to depth maps reveal
602 more appropriate.

603 As future work we wish to devise a photo-consistent framework by
604 exploiting in all steps of our approach the color attributes of the
605 multi-view stereo images. Where data sets are available it may also
606 be interesting to investigate the fusion of airborne and ground-based
607 measurements in order to reconstruct facade elements such as doors
608 or windows.

609 References

- 610 ACUTE3D, 2013. <http://www.acute3d.com/>.
- 611 AREFI, H., ENGELS, J., HAHN, M., AND MAYER, H. 2008. Levels of Detail in 3D
612 Building Reconstruction from LIDAR Data. In *Proc. of ISPRS conference*.
- 613 ARIKAN, M., SCHWARZLER, M., FLORY, S., WIMMER, M., AND MAIERHOFER,
614 S. 2013. O-snap: Optimization-based snapping for modeling architecture. *Transac-*
615 *tions on Graphics* 32, 1.

577 tion steps for the LODs (e.g., roof superstructures of LOD3 are
578 reconstructed from height maps derived from the roofs computed
579 in LOD2, and LOD0 is reconstructed by instantiating a 2D min-cut
580 algorithm applied to a subset of abstracted proxies used in other
581 LODs).

582 Our initial aim at devising a fully automated pipeline translates into
583 a unsupervised classification method relying solely on geometric
584 attributes and few semantic rules. The classification is performing
585 well with only few relevant local geometric attributes and global
586 solve. Our approach is shown to exhibit robustness to defect-laden
587 meshes through regularized optimizations combined with 3D ar-
588 rangements which by construction generate well-behaved surfaces.
589 Our initial goal to devise a scalable work flow is also met by exploit-
590 ing both the classification and abstraction steps in order to instanti-
591 ate one reconstruction process per building or tree component, and
592 to reduce the combinatorial complexity of the 3D arrangements.
593 At first glance our approach may be seen as a complex assem-
594 bly: iconization on depth maps for trees and superstructures, 3D
595 arrangements of primitives for buildings. A closer look however
596 reveals that our methodological choices are specialized to the scale,
597 structure and semantic of data. In addition, they are matching the
598 limitations and constraints of real-world measurement data: despite
599 recent advances on acquisition and reconstruction [Acute3D 2013]
600 the resolution of MVS meshes is too limited to handle roof super-

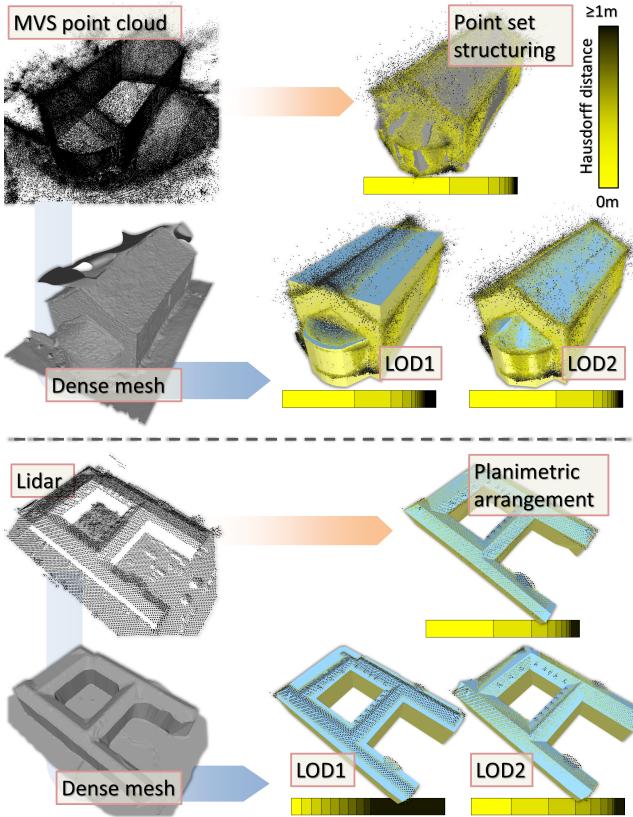


Figure 16: Comparisons with urban reconstruction from point clouds. Starting from a point cloud (left: Airborne LIDAR scan, right: raw point cloud generated by MVS), we generate a dense mesh from which LOD1 and LOD2 are extracted. Our algorithm compares favorably against two reconstruction methods taking raw point clouds as input: the geometric accuracy of LOD2 to the input point cloud (see distribution of errors for all points, from yellow to black) is similar for airborne LIDAR scan (left) and higher for MVS imagery (right).

- 616 ATTENE, M., KATZ, S., MORTARA, M., PATANE, G., SPAGNUOLO, M., AND
617 A. TAL, A. 2006. Mesh segmentation - a comparative study. In *Proc. of Shape
618 Modeling International*.
- 619 BAO, F., YAN, D.-M., MITRA, N., AND WONKA, P. 2013. Generating and exploring
620 good building layouts. *ACM Transactions on Graphics*. Proc. of SIGGRAPH.
- 621 BOYKOV, Y., AND KOLMOGOROV, V. 2004. An experimental comparison of min-
622 cut/max-flow algorithms for energy minimization in vision. *PAMI* 26, 9.
- 623 BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy mini-
624 mization via graph cuts. *PAMI* 23, 11.
- 625 CGAL, 2013. Computational Geometry Algorithms Library (www.cgal.org).
- 626 CHAUVE, A.-L., LABATUT, P., AND PONS, J.-P. 2010. Robust piecewise-planar
627 3D reconstruction and completion from large-scale unstructured point data. In
628 *Conference on Computer Vision and Pattern Recognition*, IEEE.
- 629 CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A Benchmark for 3D
630 Mesh Segmentation. *ACM Transactions on Graphics*. Proc. of SIGGRAPH.
- 631 COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted delaunay triangulations
632 and normal cycle. In *Proc. of ACM Conf. on Computational Geometry*, 312–321.
- 633 COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape
634 approximation. In *Proc. of SIGGRAPH Conference*, ACM.
- 635 FALCIDIENO, B., AND SPAGNUOLO, M. 1998. A Shape Abstraction Paradigm
636 for Modeling Geometry and Semantics. In *Proc. of the Conference on Computer
637 Graphics International*, 646–657.
- 638 FURUKAWA, Y., CURLESS, B., SEITZ, S., AND SZELISKI, R. 2009. Manhattan-
639 world stereo. In *IEEE Conf. on Computer Vision and Pattern Recognition*.
- 640 GARLAND, M., AND HECKBERT, P. 1997. Surface simplification using quadric error
641 metrics. In *ACM SIGGRAPH Conference*.
- 642 GROGER, G., AND PLUMER, L. 2012. Citygml interoperable semantic 3d city
643 models. *Journal of Photogrammetry and Remote Sensing* 71.
- 644 HANE, C., ZACH, C., COHEN, A., ANGST, R., AND POLLEFEYS, M. 2013. Joint 3D
645 scene reconstruction and class segmentation. In *IEEE Conf. on Computer Vision
646 and Pattern Recognition*.
- 647 KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3d mesh
648 segmentation and labeling. *ACM Transactions on Graphics*. Proc. of SIGGRAPH.
- 649 LAFARGE, F., AND ALLIEZ, P. 2013. Surface reconstruction through point set struc-
650 turing. *Computer Graphics Forum* 32, 2, 225–234. Proc. of EUROGRAPHICS.
- 651 LAFARGE, F., AND MALLET, C. 2011. Building large urban environments from
652 unstructured point data. In *ICCV*.
- 653 LEBERL, F., IRSCHARA, A., POCK, T., MEIXNER, P., GRUBER, M., SCHOLZ, S.,
654 AND WIECHERT, A. 2010. Point clouds: Lidar versus 3d vision. *Photogrammetric
655 Engineering and Remote Sensing* 76, 10.
- 656 LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J.
657 2011. Globfit: Consistently fitting primitives by discovering global relations. *ACM
658 Transactions on Graphics*. Proc. of SIGGRAPH.
- 659 LIN, H., GAO, J., ZHOU, Y., LU, G., YE, M., ZHANG, C., LIU, L., AND YANG, R.
660 2013. Semantic decomposition and reconstruction of residential scenes from lidar
661 data. *ACM Transactions on Graphics*. Proc. of SIGGRAPH.
- 662 LUEBKE, D., WATSON, B., COHEN, J. D., REDDY, M., AND VARSHNEY, A. 2002.
663 *Level of Detail for 3D Graphics*. Morgan Kaufmann Editions.
- 664 MEHRA, R., ZHOU, Q., LONG, J., SHEFFER, A., GOOCH, A., AND MITRA, N.
665 2009. Abstraction of Man-Made Shapes. *ACM Transactions on Graphics*. Proc. of
666 SIGGRAPH.
- 667 MITRA, N., WAND, M., ZHANG, H., COHEN-OR, D., AND BOKELOH, M. 2013.
668 Structure-aware shape processing. In *EUROGRAPHICS State-of-the-art Report*.
- 669 MUSIALSKI, P., WONKA, P., ALIAGA, D., VAN GOOL, L., AND PURGATHOFER,
670 W. 2012. A survey of urban reconstruction. In *EUROGRAPHICS State-of-the-art
671 Report*.
- 672 ORTNER, M., DESCOMBES, X., AND ZERUBIA, J. 2008. A marked point process of
673 rectangles and segments for automatic analysis of digital elevation models. *PAMI*
674 30, 1.
- 675 PAULY, M., GROSS, M. H., AND KOBBELT, L. 2002. Efficient simplification of
676 point-sampled surfaces. In *Visualization*, IEEE, 163–170.
- 677 PIX4D, 2013. <http://pix4d.com/>.
- 678 POULLIS, C., AND YOU, S. 2009. Automatic reconstruction of cities from remote
679 sensor data. In *IEEE Conf. on Computer Vision and Pattern Recognition*.
- 680 ROTTENSTEINER, F., SOHN, G., JUNG, J., GERKE, M., BAILLARD, C., BENITEZ,
681 S., AND BREITKOPF, U. 2012. The ISPRS benchmark on urban object classifica-
682 tion and 3d building reconstruction. In *Proc. of the ISPRS congress*.
- 683 SHAMIR, A. 2008. A survey on mesh segmentation techniques. *Computer Graphics
684 Forum* 27, 6.
- 685 SINHA, S., STEEDLY, D., SZELISKI, R., AGRAWALA, M., AND POLLEFEYS, M.
686 2008. Interactive 3d architectural modeling from unordered photo collections. *ACM
687 Transactions on Graphics*. Proc. of SIGGRAPH Asia.
- 688 TOSHEV, A., MORDOHAI, P., AND TASKAR, B. 2010. Detecting and parsing ar-
689 chitecture at city scale from range data. In *IEEE Conf. on Computer Vision and
690 Pattern Recognition*.
- 691 VANEGAS, C., ALIAGA, D., AND BENES, B. 2010. Building reconstruction us-
692 ing Manhattan-world grammars. In *IEEE Conf. on Computer Vision and Pattern
693 Recognition*.
- 694 VANEGAS, C., ALIAGA, D., WONKA, P., MULLER, P., WADDELL, P., AND WAT-
695 SON, B. 2010. Modeling the appearance and behavior of urban spaces. In *EURO-
696 GRAPHICS State-of-the-art Report*.
- 697 YUMER, M. E., AND KARA, L. B. 2012. Co-abstraction of shape collections. *ACM
698 Transactions on Graphics*. Proc. of SIGGRAPH Asia.
- 699 ZEBEDIN, L., BAUER, J., KARNER, K., AND BISCHOF, H. 2008. Fusion of feature-
700 and area-based information for urban buildings modeling from aerial imagery. In
701 *ECCV*.
- 702 ZHOU, Q., AND NEUMANN, U. 2012. 2.5D building modeling by discovering global
703 regularities. In *IEEE Conf. on Computer Vision and Pattern Recognition*.