Modeling Urban Scenes From Pointclouds

William Nguatem
Bundeswehr University Munich, Germany
william.nguatem@unibw.de

Helmut Mayer
Bundeswehr University Munich, Germany
helmut.mayer@unibw.de

Abstract

*We present a method for **M**odeling **U**rban **S**cenes from **P**ointclouds (MUSP). In contrast to existing approaches, MUSP is robust, scalable and provides a more complete description by not making a Manhattan-World assumption and modeling both buildings (with polyhedra) as well as the non-planar ground (using NURBS). First, we segment the scene into consistent patches using a divide-and-conquer based algorithm within a nonparametric Bayesian framework (stick-breaking construction). These patches often correspond to meaningful structures, such as the ground, facades, roofs and roof superstructures. We use polygon sweeping to fit predefined templates for buildings, and for the ground, a NURBS surface is fit and uniformly tesselated. Finally, we apply boolean operations to the polygons for buildings, buildings parts and the tesselated ground to clip unnecessary geometry (e.g., facades protrusions below the non-planar ground), leading to the final model. The explicit Bayesian formulation of scene segmentation makes our approach suitable for challenging datasets with varying amounts of noise, outliers, and point density. We demonstrate the robustness of MUSP on 3D pointclouds from image matching as well as LiDAR.*

1. Introduction

Three-dimensional (3D) modeling of urban scenes has received major interest in recent years [15, 18, 30, 41] due to emerging applications in virtual and augmented reality, simulation, etc. The ultimate goal is to generate compact yet rich representations, making available 3D assets readily understandable by other processes (e.g. rendering). Two major technologies have made these developments possible: (1) Image matching, i.e., Structure-from-Motion and Multi-View Stereo (SfM/MVS) and, (2) Light Detection And Ranging (LiDAR). LiDAR is a mature technology producing 3D points with high accuracy.

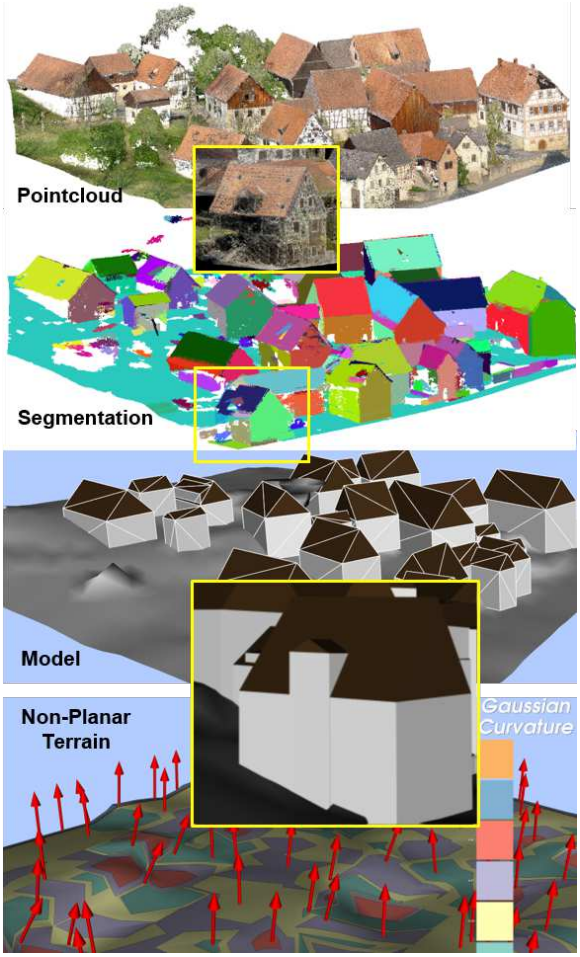

Figure 1. We propose an approach based on segmentation (random colors are assigned to the segmented patches) and model fitting for **M**odeling **U**rban **S**cenes from **P**ointclouds (MUSP). In this example, MUSP transforms 35 million 3D points from a Multi-View Stereo (MVS) reconstruction of a residential scene containing an irregular arrangement of buildings on a non-planar terrain into a semantic watertight polygonal model. In the presence of multiple close objects on a non-planar ground, consistent patch segmentation in noisy pointclouds is itself a highly non-trivial task. MUSP employs a probabilistic framework that addresses this problem without any knowledge about the number of segments. A template-based polygon fitting ensures consistency as well as model completeness.

However, the availability of cheap dedicated hardware (GPUs) combined with the development of new algorithms makes pointcloud acquisition using an SfM/MVS pipeline an interesting alternative [10, 11, 14], amplified by the inherent flexibility due to the widespread of cheap consumer cameras.

2. Related Work

Analysis of 3D data is an active topic of research with focus on semantic segmentation, object detection, automatic modeling and compression [18, 20, 31, 26, 35, 38]. The creation of 3D models from pointclouds using classical meshing algorithms, e.g., Poisson reconstruction or Marching cube, typically overfits the data and produces overly complex triangulated meshes with little or no semantics. Moreover, meshing algorithms often fail in the presence of noise and high point density variation which is often the case for the pointclouds derived by SfM/MVS.

Procedural Modeling: In conjunction with shape grammars and reversible jump Markov Chain Monte Carlo, procedural modeling algorithms have been widely used for man-made structures in urban scenes [5, 12, 22, 27, 34, 39] The idea is to define a set of basic shapes and production rules from which further shapes evolve. However, for large-scale urban scenes with less regular building placement, these algorithms suffer from a lack of scalability. Additionally, the convergence is difficult to ascertain, because the basic shapes might not capture the diversity. It is also difficult to derive good proposal distributions for the Markov chain that generalize well and are suitable for the target distribution, i.e., avoid the phenomenon called persistent rejection.

Geometric Primitive Fitting: This alternative to procedural modeling is often used for urban scene segmentation from LiDAR pointclouds [31, 43, 15, 45, 28]. Here, a cost function is formulated to assess the quality of the fit. The algorithms mostly operate in three different modes: Region growing [32, 24, 30], RANdom SAmple Consensus (RANSAC) [9, 4, 35, 33] and energy-based [13]. RANSAC is fast and robust against noise, but requires a-priori knowledge of K, the number of segments present in the scene. In its vanilla form, the inlier count is used to quantify the fit. However, pointcloud segmentation alone only gives a partial solution, as it does not address the generation of a lightweight model required, e.g., for real-time rendering. A few recent approaches simultaneously address geometric primitive detection and regularization for modeling 3D data [3, 44, 18, 41]. However, these algorithms only handle LiDAR, or 3D data of moderate size and amount of noise as well as (already) triangulated 3D points [41].

Modeling beyond a Manhattan-World: The assumption of a mixture of Manhattan-Frames is widely used both implicitly [20, 1, 17] and explicitly [16, 38] for processing 3D pointclouds. In this line of work, a planar ground is often assumed. However, this is valid only for indoor scenes [23, 1]. A flexible approach fitting NURBS to pre-segmented patches of LiDAR pointclouds is proposed in [6]. A major challenge remain, how to robustly segment and how to infer unique and consistent control points of the NURBS surface in the presence of substantial noise, large point density variations and missing data.

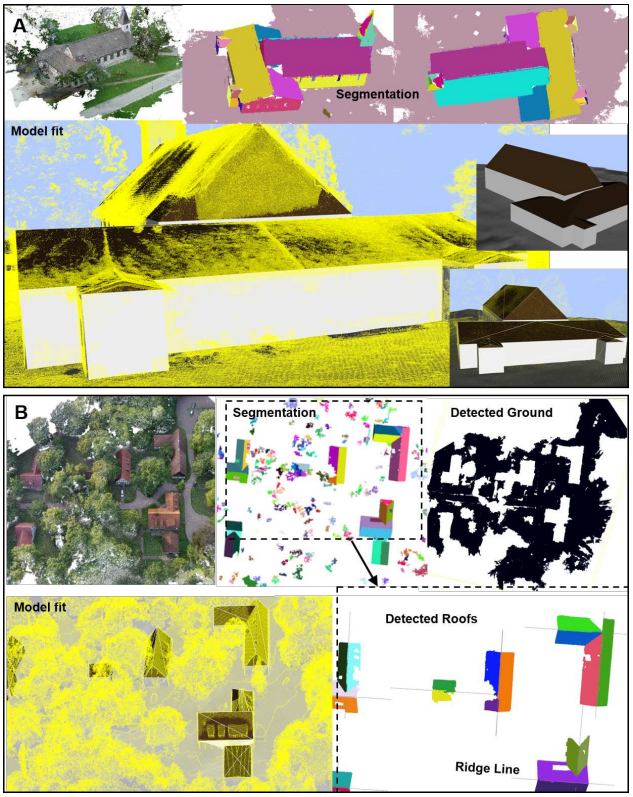

Figure 2. In addition to accurate and fast modeling, every stage of our work flow is robust against noise e.g., trees or reconstruction artifacts from SfM/MVS. Here, two data sets (A: 36M points, B:23M points) from SfM/MVS. Segmented clusters and detected roof segments are randomly colored. Downsampled input data in yellow superimposed on the model to show the quality of the fit on abstracted polygons for buildings and ground.

In summary, the state-of-the-art approaches for modeling from pointclouds have four major deficits—they lack scalability and robustness against substantial noise, they are tailored only for LiDAR or already triangulated 3D points and they cannot capture the natural smoothness of the ground due to underlying Manhattan-World assumptions.

3. Contributions

The major contributions of our method are:

1. A robust, scalable and probabilistic pointcloud segmentation algorithm which uses a nonparametric Bayesian framework for clustering and automatically

infers K, the number of segments present in the scene.

2. A set of basic architectural rules that enables semantic decomposition of scenes into the four meaningful categories—ground, roof elements, facades and rest.

3. Polygon-sweeping as a substitute for the more general plane-sweeping algorithm particularly suited to detect facades in noisy pointclouds.

The input to MUSP is an unstructured 3D pointcloud, $\mathcal{D}$, of an urban scene captured from e.g., an unmanned aerial vehicle. The output is a watertight 3D semantic surface model. We assume that the metric scale for $\mathcal{D}$ and the vertical (up) direction $\boldsymbol{v}$ are known. Furthermore, we assume that buildings with curved facades are not present in the scene. We now present MUSP in its three main stages: Segmentation, Semantic Decomposition as well as Surface Fitting and Regularization.

4. Segmentation

The objective of this section is to find patches in $\mathcal{D}$ which exhibit local planarity, i.e., are planar within a predefined small radius. Region growing, RANSAC and energy optimization have been used to solve similar problems [35, 13, 28], however, these algorithms are not suitable for segmenting noisy pointclouds of large-scale scenes with a possibly unknown number of object instances such as in Figs 1 and 2. We have developed the following divide and conquer based algorithm that divides the scene into voxels, fits a plane within each voxel using RANSAC, and uses a nonparametric Bayesian approach as an alternative to traditional clustering algorithms (e.g., K-Means and spectral clustering) to cluster voxels consistently (conquer). The idea is that normals of voxels from the same patch will belong to the same cluster on the unit sphere. Nonparametric Bayesian has the appealing advantage that K can be unknown and inferred together with the underlying structure from the data.

4.1. Divide

The goal is to estimate consistent unit normals describing the underlying surface geometry using sample space division. We divide the scene into a grid of non-overlapping voxels and compute an adjacency graph of voxel (spatial neighborhood) relations, $\mathcal{G}_{adj}$. The voxel leaf size, l_s, is chosen such that the smallest object of interest, e.g., roof superstructure, is greater than l_s. We use RANSAC to fit a plane within every voxel. The planes have local support limited to the voxel bound, thus clipped planes. Compared to normals of individual 3D points, clipped plane normals (CPN) offer a number of advantages:

- Surfaces in urban scenes exhibit many small local variations which aren't representative of the underlying geometry, thus normals of individual 3D points often overfit. On the other hand, coupled with the robustness of RANSAC, CPN are stable and do not overfit.

- Normals of individual 3D points can wrap-around the unit sphere when estimated with eigenvector analysis of the covariance matrix. This problem is exacerbated in pointclouds with substantial noise, such as those commonly encountered in data from SfM/MVS of urban scenes (see Fig. 3 middle column). Conversely, CPN do not wrap-around (cf. Fig. 3 right column). RANSAC chooses a single "best" plane to the inliers, i.e., without influence of points from adjacent surfaces.

- CPN computation is far less expensive than normal estimation of individual 3D points. The strong planarity within most voxels means that RANSAC converges in a few iterations and also the number of CPN is many magnitudes lower than the number of 3D points.

Next, we introduce the clustering framework for CPN based on Bayesian nonparametrics, hence combining voxels consistently. The algorithm assumes that CPN exhibit a multivariate Gaussian distributions on the unit sphere. Hence, the goal is inference for Gaussian mixture models with an unknown number of components and structure—the parameters of the individual mixture components. Taking a Bayesian setting, we place a Normal-Inverse-Wishart (NIW) prior on the mean and covariance parameters jointly for every mixture component.

MUSP uses Dirichlet Process (DP) [8, 19, 29] mixture of Gaussians as a nonparametric Bayesian representation given by $G{=}DP(\alpha{,}G_0)$, where G_0 is the base distribution, α is a positive scalar known as the concentration parameter representing the strength of belief in G_0. We fully specify G_0 as the NIW distribution, itself parameterized by $G_0 = NIW\,(\mu_0{,}\kappa_0{,}\Psi_0{,}\upsilon_0)$. However, the concentration parameter, α, remains unspecified and is set manually. The choice of α will affect the clustering performance as we present in section 7. Furthermore, to construct the DP, the stick-breaking construction [37] is used to capture the possibly infinite mixture of components. It is defined by the following hierarchy,

$$G{=}\sum_{j=1}^{\infty} w_j \delta_{\theta_j}(\theta) \tag{1}$$

$$\begin{gathered} G{\sim}DP(\alpha{,}G_0), \quad \theta_j{\sim}G \\ \left(x_1{,}\cdots{,}x_{N_{cpn}}\right){\sim}f\left(x{,}\theta_j\right) \end{gathered} \tag{2}$$

where w_j represent the weights or proportions of the various components and $\delta_{\theta_j}(\theta)$ is the component indicator which is zero everywhere, except for $\delta_{\theta_j}(\theta) = 1$. The weights are interpreted as the length of the pieces broken off iteratively

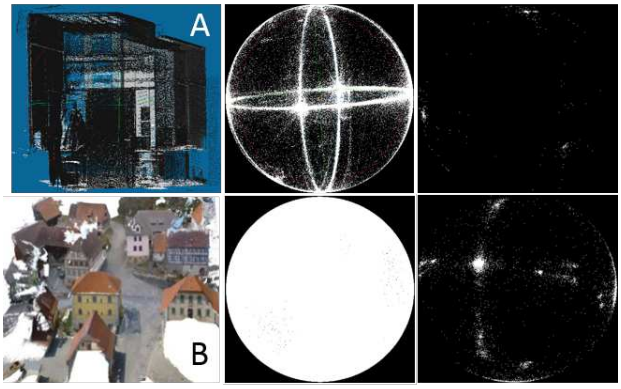

Figure 3. Comparison of normals of pointclouds from the Kinect sensor of an office scene (A), and from SfM/MVS (B) of an urban scene. Normals of individual 3D points can wrap-around the unit sphere, and can over-fit for urban scenes (middle column). On the other hand, normals of clipped planes (3^{rd} column) form distinct clusters.

from a unit length stick (hence the name) given by Equation (3),

$$w_1{=}V_1, w_l{=}V_l \prod_{j<l} (1{-}V_j)\ ,\ l{=}2,3.\cdots \tag{3}$$

At each iteration, the proportion to break-off from the remaining stick, V_l, is sampled from a beta distribution as follows

$$V_l{\sim}Beta\,(1,\alpha) \tag{4}$$

4.2. Conquer

Ensuring proper clustering of unit normals poses several challenges that would require directional statistics, e.g., von Mises Distribution [7]. However, since CPN do not wrap-around, we cluster them without considering the directionality using the Gibbs sampling inference within a nonparametric Bayesian setting depicted in Algorithm 1.

Gibbs sampling is fundamentally sequential, hence (usually) requires a good initialization to guarantee fast convergence. MUSP employs the two-level approach shown in Fig. 4 to overcome this problem. In the first level, we cluster a subset, the "coreset", of the CPN that preserves its salient relationships using random assignments of 30 clusters for initialization. The inferred underlying primary structure, $\boldsymbol{\theta}^0_1, \boldsymbol{\theta}^0_2, \cdots, \boldsymbol{\theta}^0_{K^0}$, is then used as the initialization for clustering the full CPN containing N_{cpn} data items. The "coreset" is obtained using stratified resampling as follows:

1. Divide the unit sphere into polygons of approximately equal size.

2. Sample $N_{coreset} \ll N_{cpn}$ in proportional to the number of CPN per polygon.

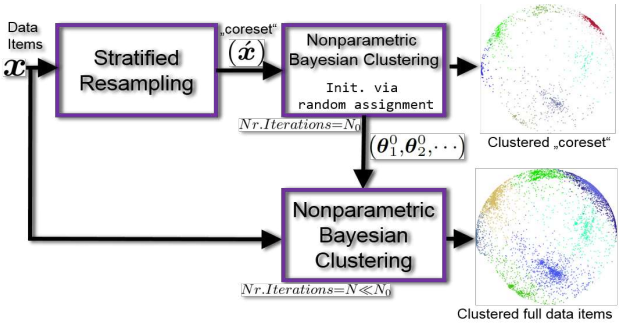

Figure 4. Two-level clustering of CPN. The first level clusters only the "coreset" (a subset of the data capturing the salient structure) using random assignments for initialization. The second level clusters the full data items employing the output of level one for the initial assignments. In both levels, Algorithm 1 is employed. While the clustered "coreset" captures the salient structure, it contains only 10% data as opposed to the full CPN. Colors of clusters are randomly choosen.

This ensures variance reduction in the downsampled CPN as compared to a naive simple uniform random sampling strategy [2, 42]. An example "coreset" for the pointcloud depicted in Fig. 1 is shown in Fig. 4.

When appropriate values for the parameters α, $\theta_0 = (\mu_0, \kappa_0, \Psi_0, \upsilon_0)$ and $G_0 = NIW\,(\theta_0)$ are specified, the output of Algorithm 1 is an approximate posterior distribution which can be used for the data assignment to the clusters, $p\left(z_{i:N_{cpn}} \left| x_{i:N_{cpn}}, \mu, \kappa, \Psi, \upsilon \right.\right)$. It is important to note that, although the stick-breaking construction can capture clusters with infinite structure (number of components), it is usually required to truncate the number of components by an upper bound, K_b, for practical reasons. In section 7, we present the parameters for $\theta_0 = (\mu_0, \kappa_0, \Psi_0, \upsilon_0)$ and default values for K_b, and the number of iterations of the Gibbs sampling algorithm for both the first ($It_{coreset}$) and second (It_{cpn}) levels used for all experiments.

Although the computed approximate posterior distribution defines a set of consistent partitions for CPN on the unit sphere (e.g., Fig. 5B), it poses a further challenge: Two unit normals, $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ sampled from voxels of two parallel facades belong to the same cluster. To address this problem, we use the voxel adjacency graph, $\mathcal{G}_{adj}$, and compute connected components within a cluster while imposing a voxel-neighborhood-parallelism constraint as follows: Any two neighboring voxels, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are connected only if the normals $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ of their clipped planes are such that $|1{-}\boldsymbol{n}_1{\cdot}\boldsymbol{n}_2| < \epsilon_1$.

Connected components suffer from discontinuities (zig-zag effect) at the component boundaries (see Fig. 5C). We remedy this problem in a region growing way by extending the support of the boundary voxels while maintaining their normals. The extent to which the regions are grown is constrained to the pre-defined voxel leaf size, l_s. It should

Input: $G_0{=}NIW\,(\mu_0{,}\kappa_0{,}\Psi_0{,}\upsilon_0)$, data items $x_1,\ldots,x_{N_{cpn}}$, truncation level $K_b=40$
Output: $p\left(z_{i:N_{cpn}}\middle|x_{i:N_{cpn}}{,}\mu{,}\kappa{,}\Psi{,}\upsilon\right)$, a set of new cluster parameters $\hat{\theta}^*$
Get initial cluster assignment $z_1,\ldots,z_{N_{cpn}}$, i.e., assign data items randomly to initial clusters with parameters θ^*
foreach *Gibbs sampling iteration* **do**

1. Sample new assignments $\tilde{z}_i$, (with $i\in\{1,\ldots,N_{cpn}\}$) with $p\,(\tilde{z}_i{=}k)\propto w_k\prod_{j=1}^{K_b-1}(1{-}w_j)\,f\,(x_i{,}\theta_k)\,,k{\leq}K_b$
2. Resample new cluster parameters for underlying structure, θ_k^*, from the posterior $p\left(\tilde{\theta}_k^*\,|x_1{,}\cdots{,}\,x_{N_{cpn}}\right)\propto G_0\left(\tilde{\theta}_k^*\right)\prod_{\tilde{z}_i=k} f\left(x_i{,}\tilde{\theta}_k^*\right)$
3. Resample new stick-breaking weights, $\tilde{w}_k$, $\tilde{w}_k{\sim}Beta\left(1{+}m_k{,}\alpha{+}\sum_{j=k+1}^{K_b} m_j\right)$, where $m_k{=}\sum_{j=k+1}^{N_{cpn}}\delta_k\,(\tilde{z}_j)$ and $\delta()$ is the component indicator as described in Equations (1) and (2).

end
Algorithm 1: Posterior approximation of DP using Gibbs sampling.

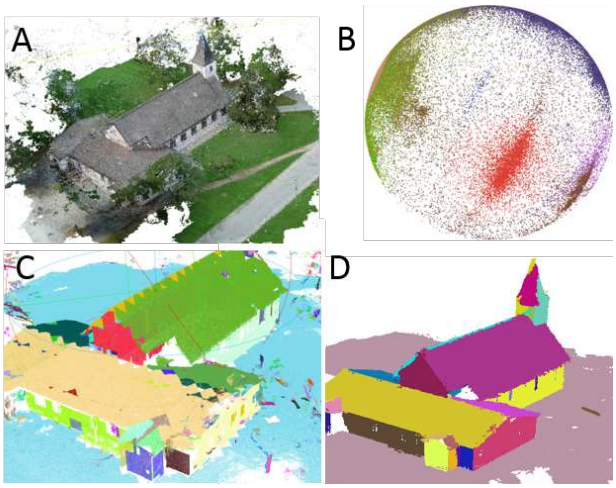

Figure 5. Segmentation: For a dataset from SfM/MVS (A) containing 26 million points, CPN are clustered (B), connected components extracted (C) and smoothed leading to an accurate segmentation (D).

be noted that the probabilistic clustering provides a good initial segmentation for merging neighboring voxels consistently. Without this, it would be difficult to develop good voxel merging criteria that generalize well.

5. Semantic Decomposition

In this section, we assign one of the four distinct labels, ground, facade, roof element and the rest (unspecified) to the segmentation results from Section 4. First, we identify the ground segment, then we use RANSAC and fit planes to the remaining segments. Finally, we identify facades, flat roofs, gable roof segments and discard the unspecified (probably vegetation) based on the segments spatial configuration and orientation.

5.1. Ground Detection

MUSP assumes the input data, $\mathcal{D}$, depicts scenes without extremely tall buildings. Therefore, the ground segment is the segment with the largest convex-hull area (area of the convex-hull of a segment), e.g., turquoise segment in Fig. 5D and green segment in Fig. 1.

5.2. Facades and Flat or Mansard Roof Detection

Since we assume that the vertical (up) direction, $\boldsymbol{v}$, is given and buildings with curved facades are not present in the scene, the geometry of facades can be well approximated by planes. We impose orthogonality of the segment normals to the vertical (up) direction, $\boldsymbol{v}$, as a constraint for facade detection. If for a given segment, S_1, RANSAC fits a plane with normal $\boldsymbol{n}_1$, then, S_1 is a facade if $|\boldsymbol{n}_1{\cdot}\boldsymbol{v}|\leq\epsilon_1$. Similarly, S_1 is regarded as a flat or mansard roof if $|1{-}\boldsymbol{n}_1{\cdot}\boldsymbol{v}|<\epsilon_1$.

5.3. Generic Gable Roof Detection

In [41, 44], a general rotational Z-symmetry for gable roofs is assumed. However, the corresponding constraints are too stringent and allow little or no architectural imperfections and asymmetry. They also do not permit noise or data acquisition artifact as often present in pointclouds derived from SfM/MVS. Moreover, neighboring buildings are very often close to one another so that a Z-symmetry in the opposite direction is present e.g., the multi-gable roofs Fig. 6A. We use the following rules and the template shown in Fig. 6C to detect generic gable roofs. Two segments S_1 and S_2 form a generic gable roof segment pair, with the ridge line, i.e, line of intersection of the two planes formed by RANSAC plane fitting for S_1 and S_2, if:
Proximity: S_1 is close to S_2.
Concavity: Angles $\omega_a{<}0.5\pi{,}\omega_b{<}0.5\pi$ (see Fig. 6C).
Downward Concavity: The centroid of both segments lie below the ridge line.

A major disadvantage of this relaxation as compared to the more restrictive Z-symmetry assumption is that both, symmetric and asymmetric (half)-hipped as well as pavilion roof types are falsely detected as generic gable roofs. We resolve this ambiguity during the regularization and model fitting as described in the next section.

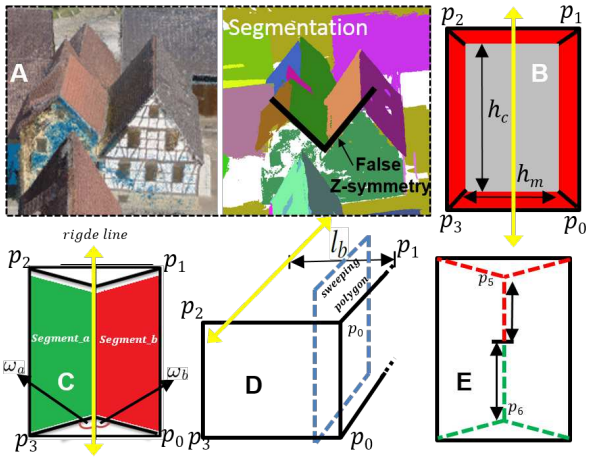

Figure 6. (A) Z-symmetry in the wrong direction because buildings are close to one another. Roof hypothesis defined by intersecting segment pair, $segment_a$ and $segment_b$ (A) for gable roofs, and approximate rectangle of segments for flat roofs (B in grey). Polygon sweeping along and orthogonal to the sweeping line (yellow line in BCD) to determine the true location of the facade. (B,E) Varying locations p_5, p_6 or values h_m, h_c captures other roof models e.g., mansard.

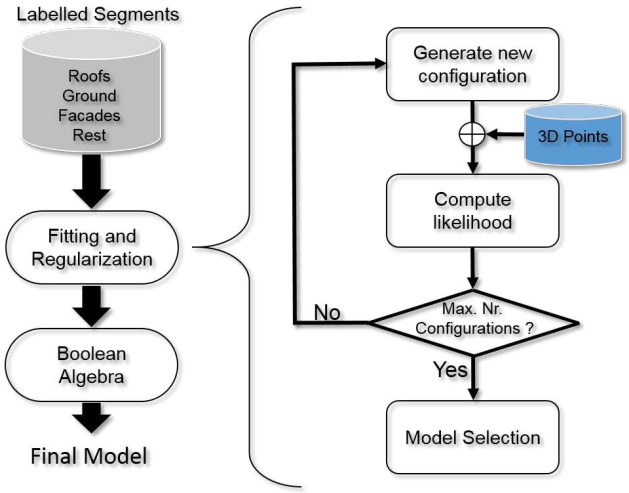

Figure 7. Work flow for generating, scoring and selecting competing configurations.

6. Surface Fitting and Regularization

MUSP proceeds with the work flow shown in Fig. 7, fitting polygons to the labelled segments. The idea is to generate competing configurations, $\boldsymbol{c}_u$, score them consistently with a likelihood function, and finally select the "best". The form of the likelihood function follows directly from the polygon chain constituting a configuration. We use an MSAC (M-Estimator SAmple Consensus) [40] based likelihood to determine if a 3D point from $\mathcal{D}$ is within the close vicinity of the polygon chain defined by a configuration. It is defined as follows:

$$\mathcal{L}(\boldsymbol{c}_u) = \exp\left(-\sum_j \rho(e_j)\right), \rho(e_j) = \begin{cases} e_j & e_j < T \\ T & e_j \geq T \end{cases} \tag{5}$$

where e_j is the shortest Euclidean distance from point p_j in the pointcloud to the surface of the polygon chain defined by configuration $\boldsymbol{c}_u$, and T the inlier threshold.

6.1. Association of Roofs and Facades

The goal here is to search for the facades corresponding to a detected generic gable roof segment pair, or a flat roof, approximated by the bounding rectangle (grey region in Fig. 6B) of its segment. This problem reduces to searching for planar inliers underneath the roof segments. Ideally, these are already labelled as facade segments from section 5. Unfortunately, a well-defined segmentation cannot be guaranteed for all facades, e.g., due to missing data or non-isolated buildings with complex footprints. We thus resort to polygon sweeping, as shown in Fig. 6 and described in the next section.

6.2. Polygon Sweeping

This is a restricted form of the widely used plane sweeping. Polygons are swept along and orthogonal to a sweeping line (yellow line in Figures 6BCD). For gable roofs, the ridge line is used as sweeping line. The approximate bounding rectangle ($h_c \times h_m$ in Figure 6B) defines the sweeping directions, hence, the sweeping lines for flat roofs. The polygon is swept along l_b (see Figure 6D). For every sweeping step, the polygon is scored using MSAC. The process is repeated in the orthogonal direction to the sweeping line. Here, the polygons are swept along l_a. The locations l_a and l_b defining the extent on which to sweep are defined based on the convex-hull polygons of the roof segments along and orthogonal to the sweeping line. For example, in the gable roof case, if h_{max} defines the location of the convex-hull orthogonal to the ridge line and the vertical (up) direction, then l_b is defined as: $l_b = h_{max} + d$, with tolerance d. Similarly, if the maximum extent of the hull along the direction of the ridge-line is r_{max}, then the width of the sweeping polygon orthogonal to the ridge-line direction is: $l_a = r_{max} + d$. The value of the tolerance is determined based on how close the buildings are to one another. Throughout our experiments, the value $d = 1\,\text{m}$ is used (see Table 1). To account for missing data, we have developed the following heuristics: If one facade is not there, we assume symmetry and replicate it from the detected parallel facade. If both parallel facades for a detected roof segment are missing, we use the convex-hull of the roof segment as facade locations.

Next, using the detected facades and the planes derived from the roof segments, we compute the points $p_0, \ldots, p_3$

(see Figure 6C) by the intersection of three planes: two adjacent facade planes and one roof segment plane.

6.3. Model Regularization and Selection

Polygon sweeping leads to models with basic gable and flat roofs. We capture symmetric and asymmetric (half)-hipped as well as pavilion roof models using the work flow in Fig. 7, by varying the locations of points p_5, p_6 along the ridge line (see Figure 6E). Similarly, by varying the size of the approximated rectangle, $h_c \times h_m$ in Figure 6B, yet not beyond the detected facades, we capture other derivatives of the flat roof, e.g., mansard roof (see Fig 9). We score the resulting new configurations, c_u, using the likelihood function $\mathcal{L}(\boldsymbol{c}_u)$. MUSP assume that all configurations are equally likely, and selects the best configuration as the one with the highest likelihood.

6.4. Ground Modeling and Surface Trimming

Modeling the ground with a plane provides the lowest possible complexity in terms of number of polygons used. Yet this often result in lost of the natural smoothness. We solve this problem using Non-Uniform Rational B-Spline (NURBS) surfaces. Surface modeling with NURBS has two major difficulties: (1) Finding appropriate control points and (2) the points that will form a mesh. For the former, we project centroids of voxels belonging to the ground to their clipped planes and use these as control points. The second problem is called tessellation. For complexity and efficiency reasons, MUSP uses uniform tessellation.

To achieve a more compact representation while preserving the semantic, we limit the extent of the facade beyond the tessellated ground. For this, we perform Boolean operations on the geometry by locating collision points of the facade polygons to the tessellated ground mesh and trimmed them off.

7. Experiments and Discussion

We perform experiments with real-world 3D pointclouds generated by state-of-the-art SfM/MVS work flows (Agisoft Photoscan, Pix4Dmapper and [14, 36]). As MUSP inherently assume aerial data acquisition, it may be difficult to model scenes captured from terrestrial sensors. Yet, we also experimented with terrestrial LiDAR scans from [25]. The size ranges from a few million 3D points to very large scale data sets containing billions of 3D points [25]. MUSP is a fully automatic system, but has only a few adjustable parameters summarized alongside their default values in Table. 1.

We use the Normal-Inverse-Wishart (NIW) distribution as base distribution, G_0, in Algorithm 1. For all experiments, we specify this distribution by setting its four parameters to: $\mu_0 = (0,0,0)$, $\kappa_0 = 1$, $\Psi_0 = \mathbf{I}_3$, $\upsilon_0 = 4$, where $\mathbf{I}_3$ is the $\mathbf{R}^3$ identity matrix.

Table 1. Parameters with default values used

MUSP Stage	Parameter	Default
Segmentation	RANSAC inlier thresh.	0.3m
	Voxel size l_s	1.0m
	$It_{coreset}, It_{cpn}$	500, 10
	Upper bound K_b	40
Semantic Decomp.	ϵ_1	0.10
Fitting & Regul.	d	1.0m
	MSAC inlier T	0.3m

7.1. Evaluation

MUSP depends on robust patch segmentation in pointclouds which itself relies on a combination of RANSAC, nonparametric Bayesian clustering and region growing. Under well-defined conditions, the performance of RANSAC and region-growing have widely been studied. Hence, we limit our discussions on the convergence and accuracy of nonparametric Bayesian clustering of CPN. The concentration parameter α determines the number of clusters, K, inferred [21]. The higher the value of α, the more clusters are found since many isolated CPN on the unit sphere will become distinct clusters. Fortunately, the presented divide and conquer framework inherently adapts towards this behaviour since connected component analysis resolves issues that may arise if too many clusters are created e.g., due to noise from vegetation. We study this behaviour by computing two Monte-Carlo approximations of the posterior using Algorithm 1 for the data set shown in Fig. 10(B) containing a single building in a single major Manhattan-Frame. We use the values $\alpha = 5$, and $\alpha = 15$ respectively. The complete analysis is performed on the first level of our two-level hierarchy, i.e., using "coreset" hence initial random assignments. The clustering dynamics—number of clusters, number of CPN per cluster, and number of Gibbs sampling iterations required to convergence—is shown in Fig. 10(B). Although a random initialization is used, the Gibbs sampling converges on average in less than 50 samples to the correct K.

Evaluation of Modeling: Currently, there are no benchmarks in this line of work because the generation of ground-truth for outdoor scenes still requires immense manual labor. We use the setup shown in Fig. 10A to evaluate MUSP in the most likely case of incomplete data e.g., due to occlusion. Our test dataset is from SfM/MVS and contains 20 million 3D points. The scene shows a gable roof building in its immediate surrounding. To simulate missing data, we manually perturb, i.e, segment ten different combinations of the facades from the test data. First we model the scene without perturbation, then we model all perturbed versions of the test data and compute the Jaccard coefficient defined

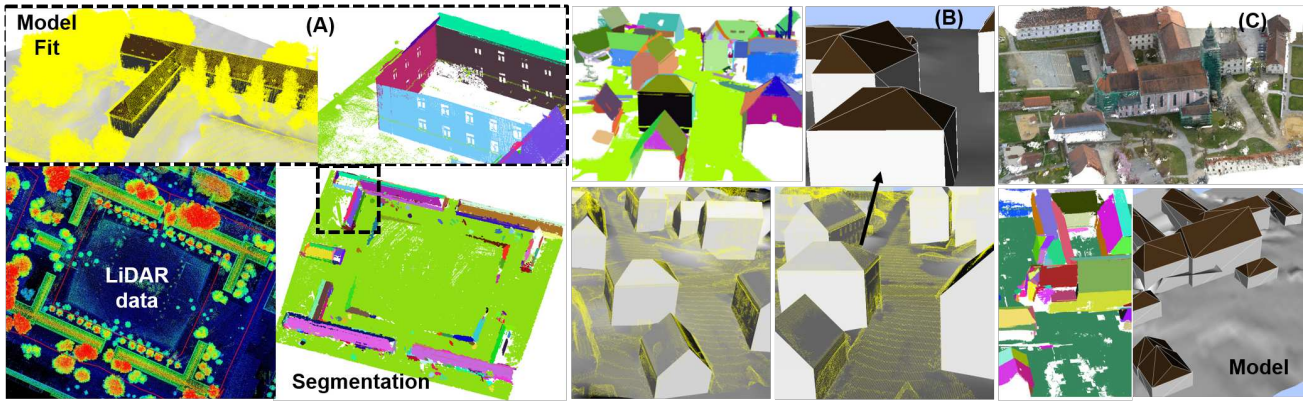

Figure 8. Results of our work flow for three data sets. Terrestrial LiDAR scan (A) from [25] containing 2.2 billion 3D points and data sets from SfM/MVS (B and C). The input pointcloud for data set B is shown in Fig. 3B. Segmented patches are represented with random colors.

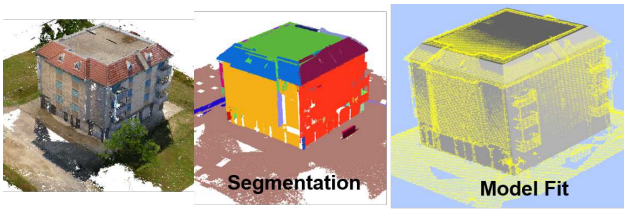

Figure 9. MUSP models a mansard roof building from pointclouds derived from SfM/MVS.

by,

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

where A and B are the volumes of the model with and without perturbations (Fig. 10A, red and green polyhedra respectively). We achieve an average Jaccard coefficient of 0.93 for this gable roof building. Further discussions of evaluation with respect to increasing noise level and incomplete data can be found in the supplementary material.

Due to diversity of buildings and the local planarity assumption, it can be difficult to correctly model buildings with curved facade or roof surfaces exhibiting strong local curvatures with MUSP. Furthermore, voxelized space with fixed voxel size of 1m means lots of finer features in a building such as the chimney or windows will be lost.

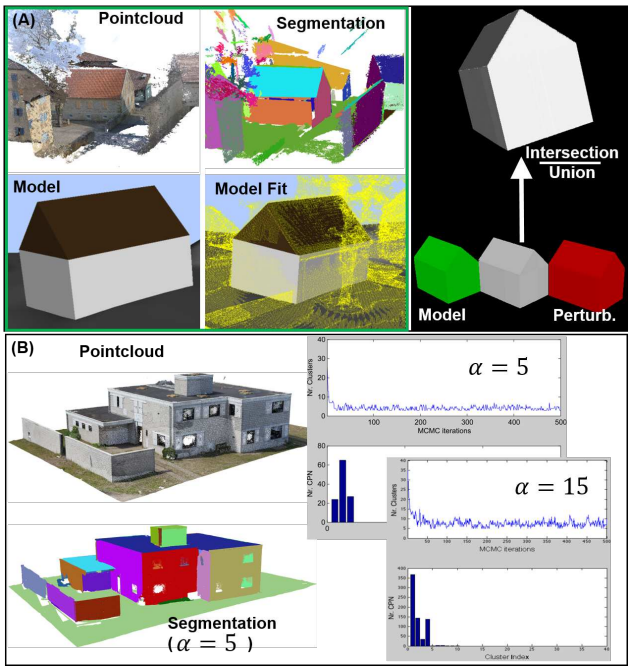

Figure 10. (A) Evaluation of modeling by comparing volumes of polyhedra, (Jaccard Coefficient). Analysis of the convergence of the Gibbs sampling based clustering of CPN for a data set with a single Manhattan-Frame (B).

8. Conclusion

We have proposed an algorithm for modeling from pointclouds. Our work flow is probabilistic, and scales to datasets with billions of noisy 3D points. It is based on an accurate scene segmentation using a combination of RANSAC and nonparametric Bayesian clustering, a set of basic rules for scene decomposition as well as polygon-sweeping and NURBS surface fitting for modeling both natural (the ground) and man-made surfaces (buildings). Besides robustness against substantial noise and scalability, our approach offers several advantages compared to existing approaches such as standard meshing algorithms. First, it abstracts the scene to a compact but accurate representation while maintaining semantics. Second, it retains the natural smoothness inherent in the ground by modeling with NURBS. An obvious next step is to refine the extracted models to include windows and doors. Also, the strongly template-based modeling of buildings can be extended to include free-form surfaces, e.g., by using NURBS for curved facades and roofs.

References

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *Computer Vision and Pattern Recognition*, 2016. 2

[2] J. Arvo. Stratified sampling of spherical triangles. In *In Computer Graphics (SIGGRAPH 95 Proceedings*, pages 437–438, 1995. 4

[3] A. L. Chauve, P. Labatut, and J. P. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *Computer Vision and Pattern Recognition*, 2010. 2

[4] O. Chum and J. Matas. Matching with prosac – progressive sample consensus. In *Computer Vision and Pattern Recognition*, pages 220–226, 2005. 2

[5] A. Dick, P. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134, 2004. 2

[6] A. Dimitrov, R. Gu, and M. Golparvar-Fard. Non-uniform b-spline surface fitting from unordered 3d point clouds for as-built modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(7):483–498, 2016. 2

[7] M. Evans, N. Hastings, and B. Peacock. von mises distribution. *Ch. 41 in Statistical Distributions*, pages 189–191, 2000. 4

[8] T. S. Ferguson. A bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1(2):209–230, 1973. 3

[9] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[10] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a Cloudless Day. In *11th European Conference on Computer Vision*, pages 368–381, 2010. 2

[11] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. 2

[12] F. Hou, H. Qin, and Y. Qi. Procedure-based component and architecture modeling from a single image. *Vis. Comput.*, 32(2):151–166, 2016. 2

[13] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 97(2):123–147, 2011. 2, 3

[14] A. Kuhn, H. Hirschmüller, D. Scharstein, and H. Mayer. A tv prior for high-quality scalable multi-view stereo reconstruction. *International Journal of Computer Vision*, pages 1–16, 2016. 2, 7

[15] F. Lafarge and C. Mallet. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision*, 99(1):69–85, 2012. 1, 2

[16] M. Li, L. Nan, N. Smith, and P. Wonka. Reconstructing building mass models from uav images. *Computers and Graphics*, 54:84 – 93, 2016. Special Issue on CAD/Graphics 2015. 2

[17] M. Li, P. Wonka, and L. Nan. *Manhattan-World Urban Reconstruction from Point Clouds*, pages 54–69. Springer International Publishing, 2016. 2

[18] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang. Semantic decomposition and reconstruction of residential scenes from lidar data. *ACM SIGGRAPH 2013*, 32(4), 2013. 1, 2

[19] R. Mitra and P. Müller. *Nonparametric Bayesian Inference in Biostatistics*. Springer Series in Statistics, ISBN: 9783319195179, 2015. 3

[20] A. Monszpart, N. Mellado, G. Brostow, and N. Mitra. RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM SIGGRAPH 2015*, 2015. 2

[21] P. Mueller, F. Andrs Quintana, A. Jara, and T. Hanson. Bayesian nonparametric data analysis. *Springer Series in Statistics*, 2015. 7

[22] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3):614–623, 2006. 2

[23] C. Mura, O. Mattausch, and R. Pajarola. Piecewise-Planar Reconstruction of Multi-Room Interiors with Arbitrary Wall Arrangements. *Computer Graphics Forum*, 2016. 2

[24] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. Van Gool, and W. Purgathofer. A Survey of Urban Reconstruction. *Computer Graphics Forum*, 32, 2013. 2

[25] A. Nchter. Robotic 3d scan repository. `http://kos.informatik.uni-osnabrueck.de/3Dscans/`, 2016. 7, 8

[26] W. Nguatem and H. Mayer. Contiguous patch segmentation in pointclouds. In *38th German Conference on Pattern Recognition (GCPR)*, pages 131–142, 2016. 2

[27] G. Nishida, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and A. Bousseau. Interactive sketching of urban procedural models. *ACM Trans. Graph.*, 35(4), 2016. 2

[28] S. Oesau, F. Lafarge, and P. Alliez. Planar shape detection and regularization in tandem. *Computer Graphics Forum*, page 14, 2015. 2, 3

[29] E. G. Phadia. *Prior Processes and Their Applications (Nonparametric Bayesian Estimation)*. Springer Series in Statistics, ISBN: 9783319327884, 2016. 3

[30] C. Poullis. A framework for automatic modeling from point cloud data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(11):2563–2575, 2013. 1, 2

[31] C. Poullis and S. You. Photorealistic large-scale urban city model reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 15(4):654–669, 2009. 2

[32] S. Pu and G. Vosselman. Knowledge based reconstruction of building models from terrestrial laser scanning data. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 64(6):575 – 584, 2009. 2

[33] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. Usac: A universal framework for random sample consensus. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038, 2013. 2

[34] N. Ripperda and C. Brenner. Reconstruction of façade structures using a formal grammar and rjmcmc. *Pattern Recognition: 28th DAGM Symposium, Berlin, Germany, September 12-14, 2006. Proceedings*, pages 750–759, 2006. 2

[35] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007. 2, 3

[36] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 7

[37] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994. 3

[38] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher III. The manhattan frame model—manhattan world inference in the space of surface normals. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 2

[39] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Měch, and V. Koltun. Metropolis procedural modeling. *ACM Transactions on Graphics*, 30(2):11:1–11:14, 2011. 2

[40] P. H. S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:2000, 2000. 6

[41] Y. Verdie, F. Lafarge, and P. Alliez. Lod generation for urban scenes. *ACM Transactions on Graphics*, 34(3):30:1–30:14, 2015. 1, 2, 5

[42] A. Yershova and S. M. LaValle. Deterministic sampling methods for spheres and so(3), 2004. 4

[43] Q.-Y. Zhou and U. Neumann. A streaming framework for seamless building reconstruction from large-scale aerial lidar data. In *Computer Vision and Pattern Recognition*, pages 2759–2766. IEEE Computer Society, 2009. 2

[44] Q. Y. Zhou and U. Neumann. 2.5d building modeling by discovering global regularities. In *Computer Vision and Pattern Recognition*, pages 326–333, 2012. 2, 5

[45] Q.-Y. Zhou and U. Neumann. Complete residential urban area reconstruction from dense aerial lidar point clouds. *Graph. Models*, 75(3):118–125, 2013. 2