Reconstructing Building Mass Models from UAV images

Minglei Li[a,b], Liangliang Nan[a,*], Neil Smith[a], Peter Wonka[a]

[a] *Visual Computing Center, KAUST, KSA*
[b] *Institute of Remote Sensing and Digital Earth, CAS, P.R.China*

Abstract

We present an automatic reconstruction pipeline for large scale urban scenes from aerial images captured by a camera mounted on an unmanned aerial vehicle. Using state-of-the-art Structure from Motion and Multi-View Stereo algorithms, we first generate a dense point cloud from the aerial images. Based on the statistical analysis of the footprint grid of the buildings, the point cloud is classified into different categories (i.e., buildings, ground, trees, and others). Roof structures are extracted for each individual building using Markov random field optimization. Then, a contour refinement algorithm based on pivot point detection is utilized to refine the contour of patches. Finally, polygonal mesh models are extracted from the refined contours. Experiments on various scenes as well as comparisons with state-of-the-art reconstruction methods demonstrate the effectiveness and robustness of the proposed method.

Keywords: urban reconstruction, aerial images, point cloud, Markov random field, graph cut

1. Introduction

Digital 3D models of urban scenes are important for a variety of applications such as urban planning, navigation, simulation, virtual reality, and entertainment. However, the digitization of urban scenes with complex architectural structures still remains a challenge [1, 2, 3]. Most of traditional surface reconstruction techniques reconstruct objects with smooth surfaces by exploiting either increasingly sophisticated solvers or better formulation of prior knowledge [4]. For urban scenes, since automatic semantic segmentation is very hard to achieve, the reconstruction process (especially for complex architectural structures) requires tedious manual effort.

In the last two decades, a considerable amount of reconstruction approaches have been developed, aiming at automatically modeling large scale urban scenes. Most of these approaches, however, are designed to deal with Light Detection and Ranging (LiDAR) point clouds obtained from airborne planes or ground level vehicles, which usually face expensive device cost and unavoidable severe occlusions. Most recently, state-of-the-art Structure from Motion (SfM) and Multi-View Stereo (MVS) methods [5, 6, 7] have produced extremely compelling results on a wide variety of scenes. A typical SfM and MVS pipeline starts by automatically matching features among the input image sequences, then it recovers the internal and external camera parameters, and produces a sparse and finally a dense 3D point cloud of the scene. To further enhance the data acquisition, in this work we exploit an Unmanned Aerial Vehicle (UAV) mounted with a camera, which provides more flexibility and significantly improves the efficiency for capturing large scale urban scenes.

Although UAV imagery is more effective in capturing all sides of urban buildings and robust against occlusion, the point clouds computed from SfM and MVS are still noisy and sparse, which hinders automatic processing and reconstruction. To overcome these problems, statistical information from different resolution are extracted to enhance the segmentation and reconstruction. The proposed method manages to classify the point cloud and reconstruct architectural models automatically, and it is robust to a wide range of data qualities.

The contributions of our work include:

- a novel framework for automatic reconstruction of large scale urban scenes from UAV images, which provides realistic reconstruction with semantic information.
- an object level point cloud segmentation algorithm and a roof extraction algorithm based on a regularized MRF formulation, which significantly speeds up the whole reconstruction pipeline.
- an effective contour refinement method based on pivot point detection, which ensures compact final reconstruction.

2. Related Work

The reconstruction of urban scenes has been a hot topic in computer graphics and computer vision in the last two decades with large number of approaches recently developed [2, 4]. In this section, we review the work that are most related to the proposed method. We divide these work into three categories according the data sources they use.

[*] Corresponding author
Email address: `liangliang.nan@gmail.com` (Liangliang Nan)

Image-based reconstruction. Using street level ortho-rectified photographs, Müller et al. [8] devised a procedural modeling strategy that identifies repeated elements in the facade image using mutual information. Sinha et al. [9] proposed an interactive system to recover the 3D structure of buildings by manually drawing outlines overlaid on 2D photographs and calculating their intersections. Enhanced by 3D depth information recovered from SfM, Xiao et al.[10] proposed a semi-automatic image-based approach for facade modeling. Garcia-Dorado et al. [11] first calibrated aerial images and fused them with GIS meta-data to compute a per-building 2.5D volumetric reconstruction using graph cut.

Laser scan-based reconstruction. In the last decades, laser scanners have provided a new type of data source for urban reconstruction. Lin et al. [12] first classified the point clouds of a large scale residential area into different categories, and then performed reconstruction based on segmentation of each building into basic symmetric and convex blocks. Lafarge and Mallet [13] proposed a non-supervised approach for point cloud classification. Then, regular roof sections are represented by basic geometric primitives and irregular roof components are represented by the combination of a set of geometric primitives. By assuming piecewise planar structures, Lafarge and Alliez [14] reconstruct surfaces using a point consolidation strategy that preserves of the buildings' structure at a given scale.

Aiming at 2.5D reconstruction, Zhou and Neumann [15] proposed a data-driven approach to detect a set of principal directions to align roof boundaries. They used these roof boundaries to produce a footprint for the reconstruction. Poullis and You [16] created compact city models from high elevation LiDAR data by simplifying boundaries of fitted planes. Lafarge et al. [17] employed Bayesian decision to assemble simple urban structures as the reconstruction from a single Digital Surface Model (DSM). By extending the traditional dual contouring algorithm into 2.5D, Zhou and Neumann [18] optimized the 2D boundaries for the roofs, which enables the reconstruction of buildings with arbitrarily shaped roofs. In their following work [19, 20], the authors further incorporated topology control and global regularity to improve the dual contouring results, yielding impressive performance.

To reconstruct facade details, Nan et al. [21] proposed an interactive reconstruction method that exploits the repetitive structure of the facades. During the drag-and-drop operation, each facade element is snapped to its proper location based on discrete optimization that balances between a regularity term and a data fitting term. By using Manhattan World assumption, Venegas et al. [22] first segmented the point cloud into walls, edges, corners, and edge-corners. They then organized the classified points into clusters to extract a volumetric description of the buildings.

MVS-based reconstruction. As images of urban scenes becomes easier to acquire from both the internet (e.g., flicker) and cameras (e.g., smart phones), more and more recent research interests have focused on reconstructing urban scenes from a set of images or videos.

Pollefeys et al. [23] designed a real-time system to generate street level city models from video frames captured by onboard cameras. Given a dense point cloud reconstructed from a set of images using SfM and MVS techniques, Arikan et al. [24] proposed O-Snap, an interactive reconstruction system that fits planar primitives along with boundary polygons, and then snaps polygons together to obtain a mesh model of a building through non-linear optimization. Using the same data source, Nan et al. [25] proposed to reconstruct detailed urban models by assembling facade details onto a set of manually extruded coarse models based on linear integer programming. Some other approaches [26, 27] are also proposed for reconstruction of large scale scenes based on MVS. These methods can generate high resolution results, but semantic information are ignored during the reconstruction and usually suffer from data storage difficulties.

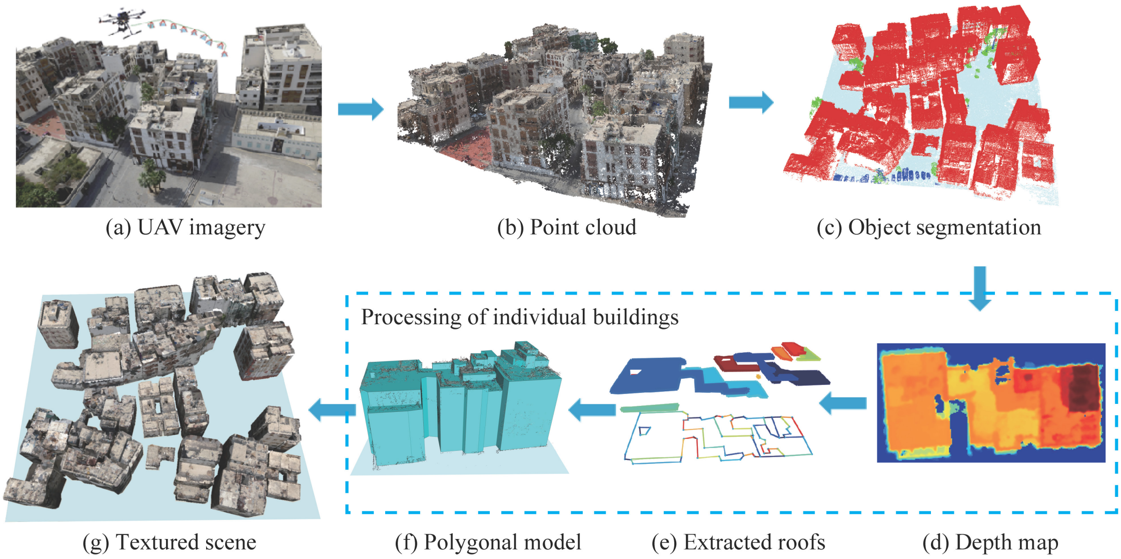
To obtain a level-of-detail representation of urban scenes, Verdie et al. [28] introduced an abstraction step between the classification and reconstruction steps to regularize planar structures from a large set of plane candidates. Finally, a surface model is extracted from a set of 3D arrangements based on a min-cut formulation. Compared with methods using ground level images, airborne-based data sources cover larger area of the scene. Most of existing airborne-based methods [15, 16, 17, 18] describe data in 2.5D due to the data acquisition strategy. This strategy makes quality reconstruction of building faces not possible, since only the roof information is available in the data. In this paper, we focus on the automatic generation of lightweight urban models from airborne point sets reconstructed from UAV images.

3. Overview

Our method takes as input a sequence of images of a scene and outputs 3D polygonal mesh models of the scene. The images are captured by a camera mounted on a UAV. In a pre-processing step, we extract a point cloud from these images using SfM and MVS [29]. Then there are two core steps for automatic generation of the urban models: point cloud classification and roof extraction. The main idea for automating these processes relies on a regularized MRF labeling strategy. An overview of our method is shown in Figure 1.

We first classify the point cloud of a large scene into four different categories, i.e., buildings, ground, trees, and others. We define a set of point features based on a 2D supporting grid by projecting the point set onto the ground. Then the classification is achieved using a regularized MRF formulation. Graph cut [30, 31] is used to solve the labeling problem (see Section 4).

After point cloud classification, the data for each building is processed independently. By projecting the points onto the ground plane, a depth map is first generated to represent the 2.5D structure of a building. Based on the depth map, a higher resolution regularized MRF formulation is used to extract the roof structure of the building, followed by a regularization step for the roof contours. Finally, polygonal mesh models are generated by extruding the roof patches onto the ground (see Section 5).



(a) UAV imagery (b) Point cloud (c) Object segmentation

(g) Textured scene (f) Polygonal model (e) Extracted roofs (d) Depth map

Figure 1: An overview of the proposed reconstruction pipeline. From a sequence of images captured by the camera mounted on an UAV (a), a point cloud (b) is generated using SfM and MVS. Then an object level segmentation is performed to decompose the entire scene into buildings and other objects (c). For each individual building, we extract the roofs (e) from its depth map (d) defined on a grid representation. Then a polygonal model (f) is extracted from the roofs (e). Finally, the entire scene can be textured (g) for various applications.

4. Object Level Segmentation

The goal of the object level segmentation step is to separate each individual building from others. By doing so, the point cloud of each building in the large scene can be processed independently. In our work, this procedure focuses on three categories, i.e., buildings, ground, and trees. We first describe how the statistical information is obtained, then we exploit graph cut to segment the points into the above three categories.

4.1. Point features

Inspired by previous work [13, 32] that exploits geometric features defined on single points to perform classification, our approach relies on a statistical analysis of the neighborhoods of the points.

In order to classify the points into the aforementioned different categories, we first compute the statistical information of the data based on a 2D supporting grid. Specifically, the entire region of the the scene is discretized into a grid defined on the ground plane using predefined grid resolution r_g. We project all the points onto the grid and within each grid cell we compute attributes for the points projected into this cell. In the grid, each cell has the standard 4-connected neighbors. An illustration of a 2D supporting grid is shown in Figure 2. Note in the classification step, our goal is to extract individual buildings and it is not necessary to extract precise contours for buildings, thus we choose to use a larger (compared with the one used for roof extraction described in Section 5.1) grid resolution for the classification. Empirically, the grid resolution is set to 0.35 m.

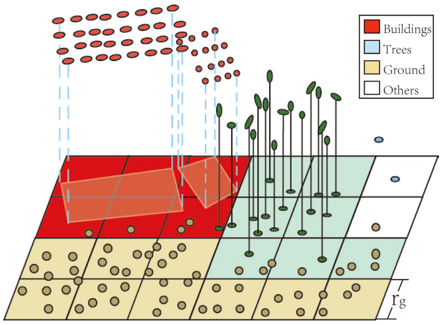


Figure 2: An illustration of the 2D supporting grid for object level segmentation.

To extract discriminative features for classification, we analyze the spatial distribution and structure of the points for each category based on the 2D supporting grid. Considering different objects in an urban area often exhibit strong structural regularities, e.g., buildings often exhibit planar regions, sharp corners, and axis aligned dominant planes; points of trees have more random distribution for both positions and normal directions; the ground plane is usually regarded as a single large segment that is relatively planar and low in height, allowing for it to be identified separately. Our point cloud classification algorithm incorporates features defined by these observations.

We introduce an identification function $F(\cdot)$ that measures the probability of a grid cell $c_i \in C$ belonging to one of these three categories. Similar to [13], our identification function is defined on a set of features extracted from the point set, as below:

- the maximum height of the cell from the ground: $h_i = \max\{p_i \rightarrow z\} - z_{ground}$
- the standard deviation of the absolute value of z compo-

nent of the normal vectors in a cell: σ_{Nz}

- the standard deviation of the height of the points in a cell: σ_H

Then, the normalized identification function $F(\cdot)$ is defined as

$$\begin{aligned} F_{ground} &= \max(1 - h_i/\hbar, 0) \\ F_{building} &= \min(\max(h_i/\hbar - \gamma \cdot \sigma_{Nz}, 0), 1) \ , \\ F_{tree} &= \min(\alpha \cdot \sigma_H + \beta \cdot \sigma_{Nz}, 1) \end{aligned} \tag{1}$$

where z_{ground} denotes the elevation of the ground plane; $\hbar$ is a threshold such that a point is considered belonging to a building if its height from ground is higher than $\hbar$; α and β are weights that balance between the elevation feature and the point distribution. Since the heights of trees in the experimented areas are less than $6m$ and σ_{Nz} ranges from 0.01 to 0.4, we set $\hbar = 6m$, $\gamma = 3$, $\alpha = 0.03$, and $\beta = 3$ through all our experiments. Intuitively, a value of $F(f_c)$ closer to 1 means the cell in the grid has higher possibility to be assigned the label f_c, and vice versa.

4.2. Point classification

As the point cloud is discretized and embedded into a uniform 2D grid, the goal of the classification is to classify the corresponding cells into different categories. This classification is a typical labeling problem. We compute an assignment of labels f_c to elements $c \in C$ such that the joint labeling f minimizes an objective function $E(f)$. Our energy function consists of two terms: data and smoothness costs.

Data cost. The data cost $D(c, f_c)$ measures how well the label assignment fits to the cells C. The normalized identification functions $F(\cdot)$ provide the initial labeling estimation for all the cells. We define the data cost for each category as follows

$$D(c, f_c) = \begin{cases} 1 - F_{ground} & \text{if } f_c = ground \\ 1 - F_{building} & \text{if } f_c = building \\ 1 - F_{tree} & \text{if } f_c = tree \end{cases} . \tag{2}$$

Smoothness cost. The smoothness term measures the spatial correlation of neighboring cells. Given two adjacent elements p and q, the smoothness energy term is defined by

$$V_{p,q} = \frac{1}{\gamma \cdot \left|h_p - h_q\right| + 1} \cdot \mathbb{1}(p, q), \tag{3}$$

where $\mathbb{1}(p, q)$ is an indicator function that has value 0 if p and q are signed the same label, otherwise it has value 1. Intuitively, the smoothness term penalizes assigning different labels to a pair of adjacent cells (p, q) that have smaller difference in their heights, i.e., $\left|h_p - h_q\right|$. For all the examples shown in the paper, γ is set to 10.

Optimization. Thus the overall energy function is

$$E(f) = \sum_{c \in C} D(c, f_c) + \lambda \sum_{p,q \in N} V_{p,q}. \tag{4}$$

Finding a solution to this labeling problem is equivalent to the minimization of the above energy function. In our implementation, we use graph cut [30, 31] to find the optimal labeling assignment. Compared with previous point cloud classification methods [32, 13] that use features defined on local neighborhood of the points, our statistic based classification can obtain more reliable results especially for complex scenes with higher level of noise and is more consistent with human perception.

4.3. Object segmentation

Since our final goal is to reconstruct buildings exhibited in the scene, we perform a segmentation step that aggregates and extracts individual buildings using a simple label based region growing algorithm.

We first extract buildings, trees, and ground by querying and combining neighboring cells that have the same label assigned in the previous classification step. The remaining points are more likely distributed in small regions with irregular geometries, thus are classified into the fourth category (i.e., *others*). Using the features defined in Section 4.1, some tall objects (e.g., wire poles) may be misclassified as *building*. We filter out these false positives using a simple thresholding mechanism. In our implementation, if the 2D area of a projected object labeled as building contains less than 200 grid cells (i.e., 24.5 m^2), the object is then assigned as *others*. A point set of building may still contain some points that may belong to ground or other categories, but these outliers are only restricted within no more than one cell outward of the contour of the building structure. So these outliers will have little effect on the final reconstruction.

5. Polygonal Mesh Extraction

Given the point clouds of individual buildings separated from the scene, our next goal is to reconstruct mesh models from these point clouds. Automatic reconstruction is challenging due to the following two reasons. First, point clouds reconstructed from images using SfM and SVM are usually nonuniform and contain a higher level of noise compared with laser scans. Second, missing data is an unavoidable problem during the data acquisition process due to occlusions, lighting conditions, and the trajectory planing of the UAV. We observe that in the point clouds generated from aerial images the walls of the buildings are extremely sparse and incomplete if the trajectory for the UAV are not carefully designed, while roofs are relatively denser and more complete than the walls. Thus, quite a few previous work mainly utilize only roof information for reconstruction [15, 16, 17]. These methods are either based on region growing for roof extraction [16, 17], or detection of roof contours by measuring certain point features (e.g., [15]), thus they suffer difficulties caused by noise and missing data. In this work, we propose a regularized MRF formulation to extract the roof structure of the building, followed by a refinement step for the roof contours. Finally, building models are extruded from the roof patches.

Compared with previous graph-cut based approaches for surface reconstruction [33, 13, 14, 11], where their formulations are based on either the irregular graph of the Delaunay tetrahedron, or points, or triangulated meshes, our formulation makes use of a graph with a four-neighbor grid structure in 2D space. Thus, our strategy significantly simplifies the roof extraction process, resulting in better stability and efficiency.

5.1. Roof extraction

In order to reliably extract roof structures, we employ another MRF-based segmentation algorithm on a grid with higher resolution defined on the point set of the building. This grid is similar to the one used in the previous classification stage, but with smaller cells that ensure more details of the roof structures can be recovered. Another difference is that in the new grid each cell stores an elevation value of the local points projected into the cell. Thus this grid can also be regarded as a depth map that provides us an effective way to process the data. With the depth map representation, processing can be conducted efficiently and effectively on the depth map despite the imperfections of the data.

Before extracting the roof structures from the point set, it would be helpful to reduce the noise in the data. To this end, we run a median filter on the depth map, since median filters are well known for reducing noise and outliers, and meanwhile preserve features (i.e., edges) of the data.

After the 2D filtering preprocess, we generate a set of plane hypotheses from the depth map using RANSAC [34]. Thus each cell in the depth map is assigned with an initial hypothesis label. Performing RANSAC on the depth map is extraordinarily efficient as the depth map significantly reduces the amount of data and maintains sufficient 2.5D information of the point cloud. We now perform a global optimization over all the cells in the depth map, to consistently segment the depth map into a set of planar regions (including roofs and the ground). This optimization is formulated as a cell-wise labeling problem, which is similar to the one used in the previous classification step (see Section 4.2). Our objective function still has a data cost term and a smoothness cost term.

Data cost. The data cost term $D(p, f_p)$ encodes the likelihood of assigning a label f_p to a cell $p \in P$. It is defined as the distance measured from p to the corresponding plane with label f_p

$$\begin{aligned} D(p, f_p) &= dist(p, f_p) \\ &= \mathbf{x}_p \cdot \mathbf{n}_f + D. \end{aligned} \tag{5}$$

where $\mathbf{x}_p$ is the position of cell p in the grid, $\mathbf{n}_f$ is the normal vector of the plane, and D is the constant coefficient in the plane equation denoted as $Ax + By + Cz + D = 0$.

Smoothness cost. Smoothness cost term $V_{p,q}$ penalizes the assignment of two different labels to adjacent cells p and q, and thus encourages the coherence between neighboring cell pairs:

$$V_{p,q} = \begin{cases} 0 & \text{if } l_p = l_q \\ \delta_1 & \text{if } l_p \neq l_q,\ l_p = l_{ground} \text{ or } l_q = l_{ground}, \\ \delta_2 & \text{otherwise} \end{cases} \tag{6}$$

where l_{ground} denotes the ground plane. The penalty term δ_1 is a constant term that makes the penalty robust to region boundaries. δ_2 is another penalty term defined as the distance between the projected points on different planes

$$\delta_2 = \left\| proj_i(p), proj_j(p) \right\|_2 ,$$

where $proj(p)$ is the projection of the point on the corresponding plane.

Optimization. By combining the above two terms, the overall energy is defined similar to that in Equation 4:

$$E(f) = \sum_{p \in P} D(p, f_p) + \mu \sum_{p,q \in N} V_{p,q}, \tag{7}$$

where P is the cells set and N represents the standard 4-neighborhood. Parameter μ is a weight that balances between the two terms. To optimize the above energy, we use the same graph cut algorithm used in Section 4.2.

5.2. Contour refinement and model extraction

Minimizing the above energy defined in Equation 7 will decompose the depth map of a building into a set of roof patches (see Figure 3). In our experiments, we observed that directly optimizing Equation 7 tends to generate zigzag artifacts in the roof contours.

Considering planar and orthogonal structures are common in architecture (i.e., most building structures are aligned with three dominant principal directions), we add a rotation step before the grid structure is built. Specifically, we detect two dominant directions by analyzing the normals of the original point set, and then transform the point cloud such that these directions are aligned with the X and Y axes of the 2D coordinate system. Experiments show that the alignment of the grid with the 2D coordinate system significantly helps to eliminate the zigzag artifacts in the extracted roof patches. Figure 3 shows the extracted roofs after the rotation step.

To extrude polygonal models from the roof patches, we first extract straight line segments from the roof contours obtained in the previous process. Specifically, we divide each roof contour into the following two categories of small contours according to the contents linked by the contours.

- Building boundaries: a *roof* patch is on one side of the contour and a *ground* patch is on another side;
- Roof boundaries: patches on both sides of the contour are of *roof*.

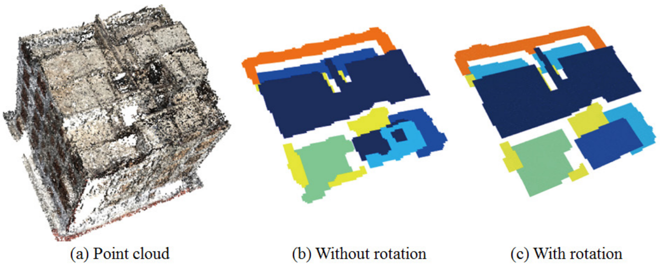

(a) Point cloud (b) Without rotation (c) With rotation

Figure 3: Roof extraction without and with the rotation step. Color denotes different roof patches.

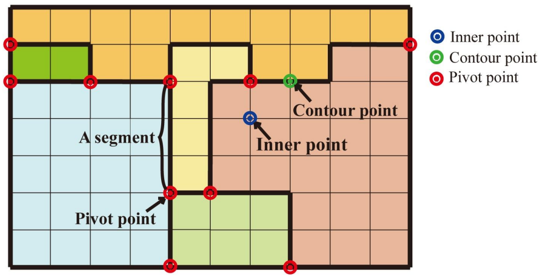
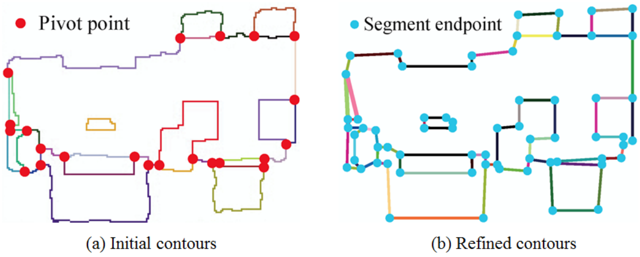


Figure 4: An illustration of pivot points and contour segments. The colors represent different roof patches.



(a) Initial contours (b) Refined contours

Figure 5: Contour simplification using the Douglas-Peucker polygonal approximation algorithm [35].

Since each cell in the grid has been assigned with a roof patch, pivot points are detected by checking the number of roof patches associated with the junctions in the grid. Specifically, a junction in the grid is considered as a pivot point if the cells associated with this junction belongs to at least 3 different roof patches (see Figure 4).

We linearize, and thus simplify the contours of roof patches using the Douglas-Peucker polygonal approximation algorithm [35]. This algorithm decomposes the contours into a sequence of straight line segments by recursively finding a point that has the maximum distance to the simplified segments, and this point is discarded if it is closer than a threshold ε to the approximating segments. The recursion is continuing until no more points can be found that have distances greater than ε to the simplified segments. In our experiment, we set ε to 0.2 m. Figure 5 shows an example of contour simplification results.

In the end, we finish the whole pipeline by constructing a polygonal mesh from the refined contours. Specifically, we construct a 2D polygon for each boundary loop in the contours, and further extrude them to the ground plane by adding vertical walls that are orthogonal to the roofs and the ground plane. The result is 2.5D reconstruction of the building in the scene. By performing the same processing on each individual buildings, then entire scene is reconstructed.

6. Results and Discussion

We have tested our approach on several datasets of large scenes acquired by a high resolution camera mounted on an UAV. After the images are obtained, we generate colored point clouds from these images using SfM and MVS. Since SfM and MVS are based on local image features, the computed point clouds usually suffer from serious noise, occlusions, and nonuniform densities. We then reconstruct polygonal models from the point clouds using our proposed method. Figures 6 and 7 show the reconstruction of theses scenes.

Figure 6 shows a portion of the United Nations Educational Scientific and Culture Organization cultural heritage site of Al-Balad, Jeddah, Saudi Arabia. The area scanned by the UAV consists of many unique 100-300 year old buildings with complex architectural features, cluttered rooftops, lattice shuttered windows, and balconies. In the last fifty years, the cultural heritage site has lost over 600 historical buildings and within even the last several months homes have been destroyed by accidental fire. We were given special permission to scan the area due to its endangerment with the intent to document the remaining buildings and generate a master plan of the area. This study will help in digitizing the remaining 375 buildings that would be too time-consuming to do using manual methods. The dataset consists of 1,518 images captured during three 10-minute autonomous flights with a Sony QX100 camera (20M pixels) and 24mm (equivalent lens) achieving a ground sampling density of $2.5 - 3.0cm$ per pixel. The three flights were repeated over the same area at an elevation of 50m (oblique), 75m (oblique), and 75m (nadir). The total generated point cloud contains 20 million colored points. Although a dense point cloud was generated a higher frequency of noise especially in low-feature surface areas (e.g., windows, white walls, metallic surfaces, etc.) was created. Our method benefits from the statistical analysis of the imperfect point cloud, which compensates the low quality of the data in an excellent way. As can be seen from this figure, although the roofs of the buildings are noisy and have missing regions, our method successfully detected and reconstructed all buildings in these regions, resulting in crack-free models.

Figure 7 shows a portion of a large modern residential area consisting of a mix of two story homes with garage ports and multiple balconies, multi-story apartment buildings, and a residential park. Two 15-minute flights were conducted to capture the entire area (approximately 125×100 m^2) using a larger UAV with a Sony Nex-7 camera (24M pixels) mounted on a gimbal angled at 50° achieving a ground sampling density of 1cm per pixel. The dataset consists of 924 images and the point cloud generated from these images contains approximately 80 million colored points. The reconstructed polygonal models fit the initial point cloud in a precise manner, and significantly reduce the storage. By converting the point cloud representation into polygonal models, the storage of the scene is reduced from 1.2 GB (initial point cloud with color, in binary format) to 530 KB (polygonal model).

Robustness to parameters. Our MRF formulations for the object level point cloud segmentation and roof extraction relies on two key parameters: λ and μ. In our experiments, we found the final reconstruction results are not sensitive to these parameters.

Figure 8 demonstrates the object level segmentation results for a historical downtown scene with increasing value of parameter λ. As can be seen from this figure, smaller values of λ

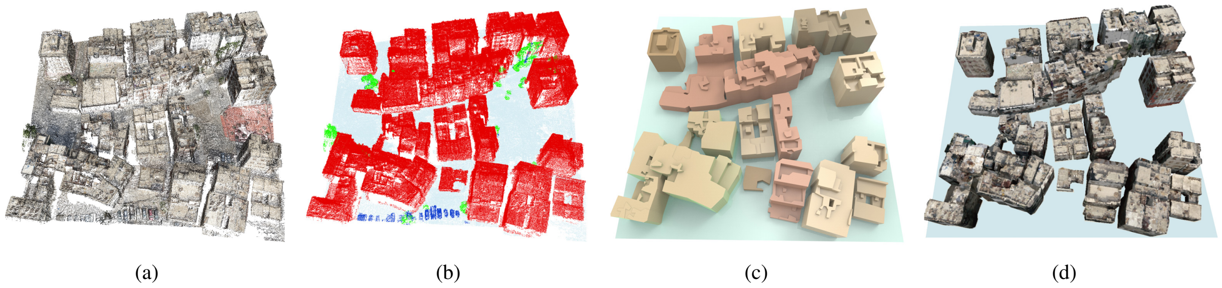


Figure 6: Segmentation and reconstruction of an old downtown area. (a) Initial point cloud; (b) Object level segmentation result; (c) Polygonal models reconstructed by the proposed method; (d) Textured polygonal models.

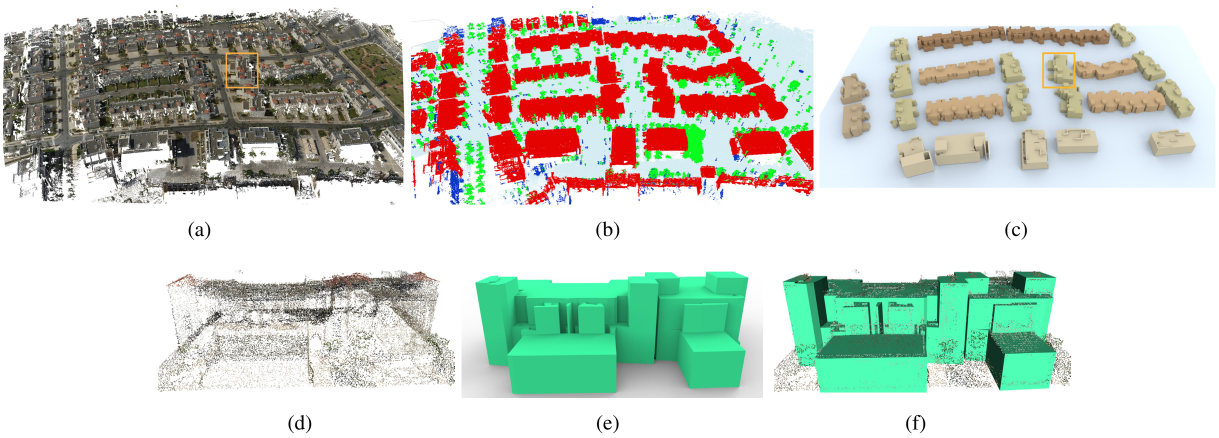


Figure 7: Segmentation and reconstruction of a large modern residential area. (a) Initial point cloud; (b) Object level segmentation result; (c) Polygonal models reconstructed by the proposed method; (d), (e), and (f) are the zoomins of the marked building in the scene.

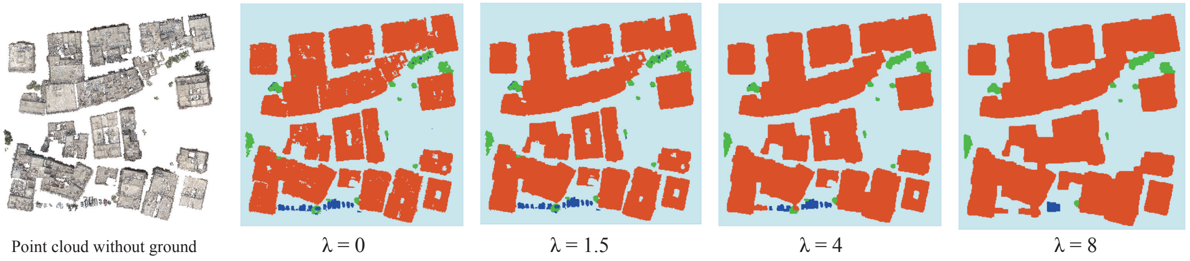


Figure 8: The effect of varying parameter λ (in Equation 4) on the segmentation results. Red, green, and blue colors represent *building*, *tree*, and *others* respectively.

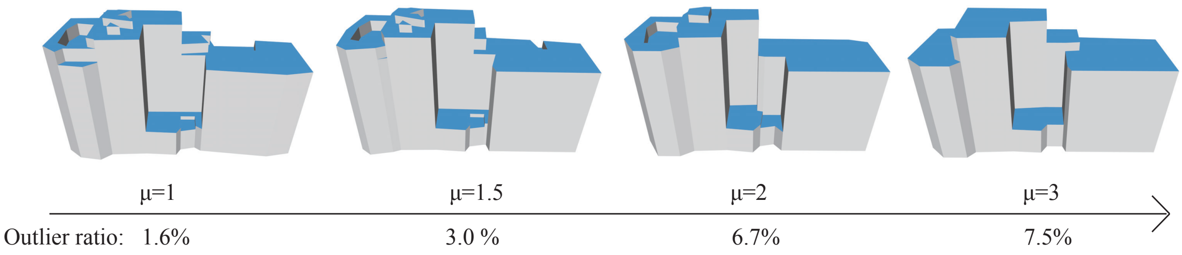


Figure 9: The effect of varying parameter μ (in Equation 7) on the final reconstruction results. Here outlier ratio is defined as the percentage of points whose distances to the 3D model are larger than $0.8m$. Blue color represents the building roofs.

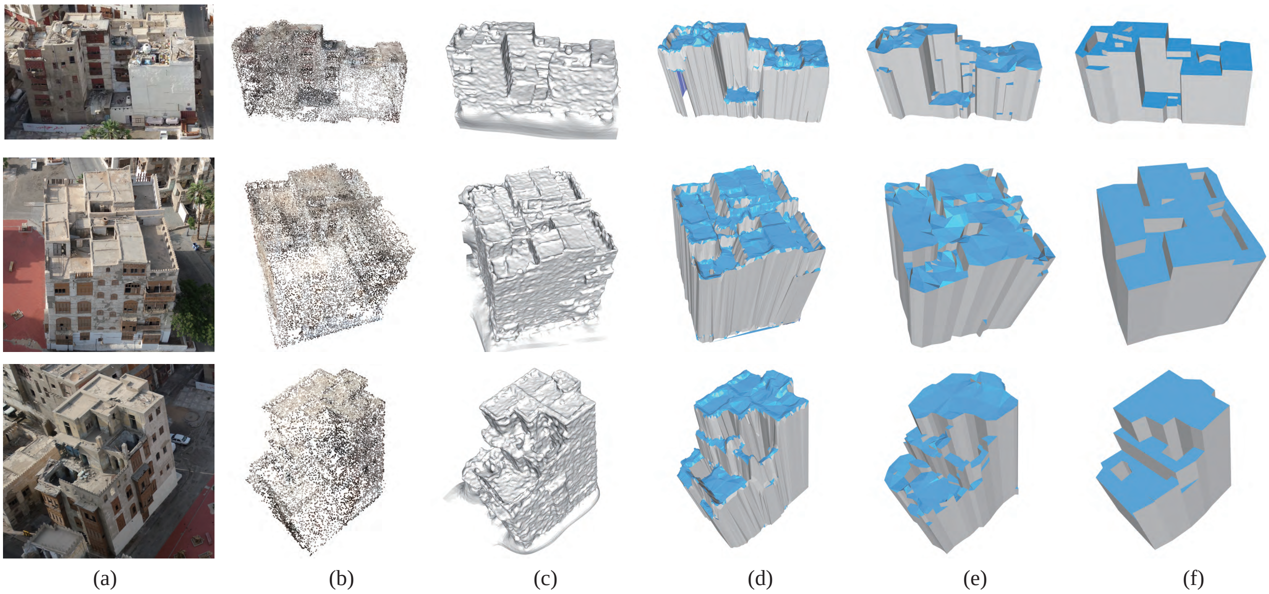


Figure 10: Comparison of the reconstruction results of our approach with other methods on three individual buildings (in different rows). (a) Photo of the building; (b) Point cloud; (c) Surface model reconstructed by Screened Poisson Reconstruction algorithm [36]; (d) DEM simplification result [37]; (e) Result of 2.5D Dual Contouring method [18]; (f) Our result.

ignore more smoothness constraints, which results in more gaps and holes in the segmentation results due to noise and missing data. On the contrary, increasing the value of λ will encourage close segments to be merged. However, as our experiments demonstrate, the value of λ in the range $[1.4, 4.3]$ can guarantee similar satisfactory segmentation results.

In Figure 9, we demonstrate the robustness of our roof extraction algorithm on the final reconstruction result in terms of varying parameter μ. Similar to the effect of varying λ in the object level segmentation step, μ controls how much smoothness constraints are preferred in the energy function. Intuitively, increasing the value of μ encourages larger planar roofs in the final reconstruction. Our experiments reveal that the value of μ in a range of $[1.5, 3.0]$ usually generates similar compact 3D models.

Comparison. We also conduct comparisons with three methods: Surface simplification from Digital Elevation Model (DEM) [37], Screened Poisson Reconstruction [36], and 2.5D Dual Contouring [18].

Figure 10 shows the reconstruction results of three individual buildings. The results of the DEM simplification method are competitive in terms of fitting quality to the point clouds. However, it can not produce straight roof boundaries. The Screened Poisson Reconstruction method [36] can generate an isotropic dense mesh surface from the point clouds. This method, however, can not handle local incompleteness (i.e., holes in the point clouds) caused by occlusions. Besides, since the result is represented as a single surface approximating the entire scene, it is rather difficult to differentiate individual buildings in the reconstruction. The results from the 2.5D Dual Contouring [18] method contain large areas of small bumps. This is because the 2.5D Dual Contouring algorithm is initially designed to deal with airborne LiDAR point clouds that mainly consist of points of building roofs with uniform density and higher accuracy. Thus, it is sensitive to our noisy point clouds computed from images using SfM and MVS. Compared with these approaches, our method can generate a simplified polygonal model that is visually pleasing and satisfactory for various applications or can be used as input for further processing.

In Table 1, we show a quantitative comparison with the aforementioned methods on the buildings shown in Figure 10. As can be seen from this table, the Screened Poisson Reconstruction method wins in terms of precision, but the final surfaces are more fluctuating. Our method has similar accuracy as the 2.5D Dual Contouring method, but it has a more compelling performance and our results have the simplest geometric structure. Our approach is seeking a tradeoff between accuracy and automatic reconstruction.

Furthermore, we also run our method on LiDAR point cloud data provided by [18]. As shown in Figure 11, our method also can deal with LiDAR data and can obtain a similar compact reconstruction results as the primitive-based method proposed in [13].

Accuracy and scalability. To intuitively evaluate the accuracy of the reconstructed models, we show the overlay of the point clouds onto the polygonal models in Figure 12, where color coding indicates the error magnitude. Our method has an average fitting error less than 0.2 m for the scene.

Besides the individual buildings, the experiments also demonstrate that our reconstruction framework has satisfactory performance on large scenes (see Figures 6 and 7). We record the running times for these scenes, which can be seen in Table 2. Both the object level segmentation and roof extraction for the

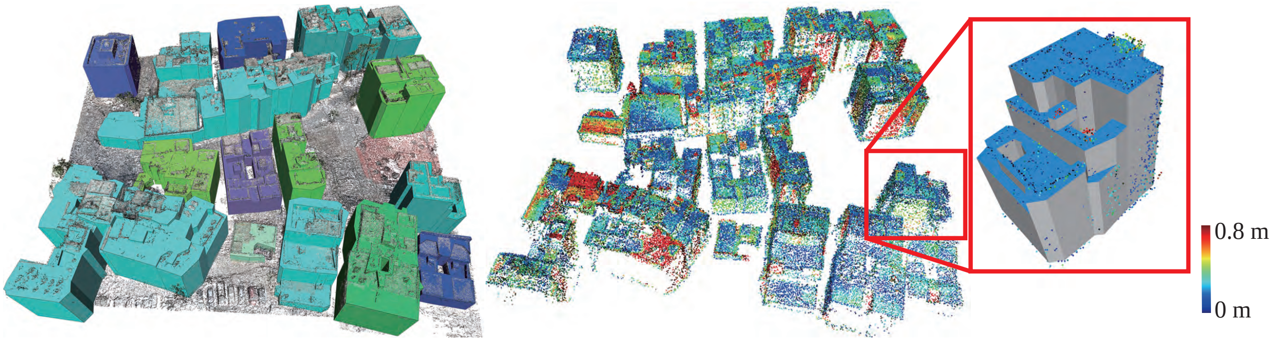


Figure 12: Point cloud overlaid on the reconstructed models. Color indicates the distances from points to their nearest faces in the model.

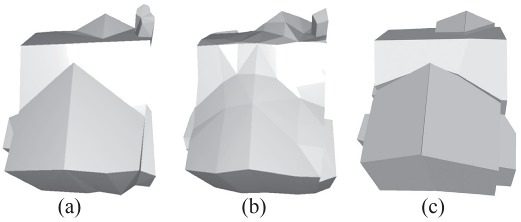


Figure 11: A comparison of our method with the primitive-based method proposed in [13] on a LiDAR point cloud. (a) The model obtained by [13]; (b) 2.5D Dual Contouring result [18]; (c) Our result.

Table 1: Statistical comparison of running times (in seconds), mesh sizes (face number), and mean errors (in meters, defined as the average distance of the points to the model) of our method with 2.5D Dual Contouring [18] (2.5D for short) and Screened Poisson reconstruction [36] (SPR) methods on the buildings shown in Figure 10.

		2.5D [18]	SPR [36]	Ours
Figure 10	Time	0.31	7.66	0.40
(top)	# Faces	2,250	56,344	675
20.6 k points	Error	0.076	0.068	0.096
Figure 10	Time	0.28	8.42	0.38
(middle)	# Faces	3,599	110,287	387
29.6 k points	Error	0.086	0.044	0.053
Figure 10	Time	0.17	1.97	0.19
(bottom)	# Faces	1,382	66,968	207
13.9 k points	Error	0.093	0.103	0.106

two scenes take only a few seconds. Thus, our method is quite suitable for processing large scale urban environments.

Limitations. During the reconstruction, we mainly rely on the roof information of the buildings. We assume that there is only one flat ground in each scene and the roofs are parallel to the ground plane. Since the models are obtained by extruding prisms from the ground plane to the roofs, the reconstructed buildings always lie in the same ground plane, and they are actually 2.5D reconstructions. Given point clouds with vertical facades, our current formulation simply ignores these vertical facade information and only uses the information given by the roof points.

Another limitation is that the piecewise planar roof structure assumption becomes too restrictive when dealing with atypical architectures, e.g., buildings with curved roofs or facades. Currently our method can not handel these types of buildings.

Table 2: Running times (in seconds) of the two core steps (object level segmentation and roof extraction) for the two large scenes shown in Figure 6 and Figure 7.

	Segmentation	Roof extraction
Figure 6	2.36	3.36
Figure 7	15.01	6.92

7. Conclusions and Future Work

This paper presented an automatic framework for reconstructing large scale urban scenes from UAV images. We introduce an effective segmentation algorithm which segments the data based on statistical analysis of their geometric properties using a low resolution grid structure. Roofs are extracted and their contours are simplified and refined using a similar grid structure of higher resolution. By using the proposed MRF formulations on the statistical information, our method is able to handle a higher level of noise and outliers. Experiments on various scenes show the reconstructed polygonal models are more compact and regular compared with state-of-the-art methods.

Currently, we only use the roof information for the reconstruction. Although the walls are sparse, they do provide extra constraints on the geometry of the buildings. As a future work, we would like to exploit the wall information to regularize the roof extraction algorithm. Another interesting problem could be approximating the trees in the scenes using template models from a database.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the KAUST Visual Computing Center.

References

[1] N. Haala, M. Kada, An update on automatic 3d building reconstruction, ISPRS Journal of Photogrammetry and Remote Sensing 65 (2010) 570–580.

[2] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, W. Purgathofer, A survey of urban reconstruction, in: Computer Graphics Forum (STAR Proceedings of Eurographics), 2013.

[3] F. Rottensteinera, G. Sohnb, M. Gerkec, J. Wegnerd, U. Breitkopfa, J. Jungb, Results of the isprs benchmark on urban object detection and 3d building reconstruction, ISPRS Journal of Photogrammetry and Remote Sensing 93.

[4] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, C. Silva, State of the art in surface reconstruction from point clouds, in: S. Lefebvre, M. Spagnuolo (Eds.), Eurographics 2014 - State of the Art Reports, 2014.

[5] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, R. Szeliski, Building rome in a day, Communications of the ACM 54 (10) (2011) 105–112.

[6] Y. Furukawa, J. Ponce, Accurate, dense, and robust multiview stereopsis, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (8) (2010) 1362–1376.

[7] C. Wu, S. Agarwal, B. Curless, S. M. Seitz, Multicore bundle adjustment, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 3057–3064. doi:10.1109/CVPR.2011.5995552.
URL `http://dx.doi.org/10.1109/CVPR.2011.5995552`

[8] P. Müller, G. Zeng, P. Wonka, L. Van Gool, Image-based procedural modeling of facades, in: ACM SIGGRAPH 2007 Papers, SIGGRAPH '07, 2007.

[9] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, M. Pollefeys, Interactive 3d architectural modeling from unordered photo collections, ACM Transactions on Graphics (TOG) 27 (5) (2008) 159:1–159:10.

[10] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, L. Quan, Image-based façade modeling, in: ACM Transactions on Graphics (TOG), Vol. 27, ACM, New York, NY, USA, 2008, pp. 161:1–161:10. doi:10.1145/1409060.1409114.
URL `http://doi.acm.org/10.1145/1409060.1409114`

[11] I. Garcia-Dorado, I. Demir, D. G. Aliaga, Automatic urban modeling using volumetric reconstruction with surface graph cuts, Computers & Graphics 37 (7) (2013) 896–910.

[12] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, R. Yang, Semantic decomposition and reconstruction of residential scenes from lidar data, ACM Transactions on Graphics (TOG) 32 (4) (2013) 66:1–66:10.

[13] F. Lafarge, C. Mallet, Building large urban environments from unstructured point data, in: Computer Vision (ICCV), 2011 IEEE International Conference on, Barcelona, Spain, 2011, pp. 1068–1075. doi:10.1109/ICCV.2011.6126353.

[14] F. Lafarge, P. Alliez, Surface reconstruction through point set structuring, Computer Graphic Forum 32 (2) (2013) 225–234.

[15] Q.-Y. Zhou, U. Neumann, Fast and extensible building modeling from airborne lidar data, in: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, ACM, 2008, p. 7.

[16] C. Poullis, S. You, Automatic reconstruction of cities from remote sensor data, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2775–2782. doi:10.1109/CVPR.2009.5206562.

[17] F. Lafarge, X. Descombes, J. Zerubia, M. Pierrot-Deseilligny, Structural approach for building reconstruction from a single dsm, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 135–147. doi:10.1109/TPAMI.2008.281.

[18] Q.-Y. Zhou, U. Neumann, 2.5d dual contouring: A robust approach to creating building models from aerial lidar point clouds, in: Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III, ECCV'10, 2010, pp. 115–128.

[19] Q.-Y. Zhou, U. Neumann, 2.5d building modeling with topology control, in: CVPR, IEEE Computer Society, 2011, pp. 2489–2496.

[20] Q.-Y. Zhou, U. Neumann, 2.5d building modeling by discovering global regularities, in: CVPR, IEEE Computer Society, 2012, pp. 326–333.

[21] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, B. Chen, Smartboxes for interactive urban reconstruction, ACM Transactions on Graphics (TOG) 29 (4) (2010) 93:1–93:10. doi:10.1145/1778765.1778830.
URL `http://doi.acm.org/10.1145/1778765.1778830`

[22] C. A. Vanegas, D. G. Aliaga, B. Benes, Automatic extraction of manhattan-world building masses from 3d laser range scans, IEEE Transactions on Visualization & Computer Graphics 18 (10) (2012) 1627–1637.

[23] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, H. Towles, Detailed real-time urban 3d reconstruction from video, Int. J. Comput. Vision 78 (2-3) (2008) 143–167.

[24] M. Arikan, M. Schwärzler, S. Flöry, M. Wimmer, S. Maierhofer, O-snap: Optimization-based snapping for modeling architecture, ACM Transactions on Graphics (TOG) 32 (1) (2013) 6:1–6:15.

[25] L. Nan, C. Jiang, B. Ghanem, P. Wonka, Template assembly for detailed urban reconstruction, Eurographics 2015, Computer Graphics Forum 34 (2).

[26] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. N. Sinha, B. Talton, L. W. 0002, Q. Yang, H. Stewnius, R. Yang, G. Welch, H. Towles, D. Nistr, M. Pollefeys, Towards urban 3d reconstruction from video, in: 3DPVT, IEEE Computer Society, 2006, pp. 1–8.

[27] V. Hiep, R. Keriven, J. P. P. Labatut, Towards high resolution large-scale multi-view stereo, Miami, US, 2009, pp. 1430–1437.

[28] Y. Verdie, F. Lafarge, P. Alliez, Lod generation for urban scenes, ACM Transactions on Graphics (TOG) 34 (3) (2015) 15.

[29] C. Wu, Visualsfm: A visual structure from motion system, URL: http://homes. cs. washington. edu/˜ ccwu/vsfm 9.

[30] Y. Boykov, V. Kolmogorov, Computing geodesics and minimal surfaces via graph cuts, in: Proceedings of the Ninth IEEE International Conference on Computer Vision, Vol. 2 of ICCV '03, 2003, p. 26.

[31] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1124–1137.

[32] M. Carlberg, P. Gao, G. Chen, A. Zakhor, Classifying urban landscape in aerial lidar using 3d shape analysis, in: Image Processing (ICIP), 16th IEEE International Conference on, IEEE, 2009, pp. 1701–1704.

[33] P. Labatut, J.-P. Pons, R. Keriven, Robust and efficient surface reconstruction from range data, Computer Graphics Forum 28 (8) (2009) 2275C2290. doi:10.1111/j.1467-8659.2009.01530.x.

[34] R. Schnabel, R. Wahl, R. Klein, Efficient ransac for point-cloud shape detection, Computer Graphics Forum 26 (2) (2007) 214–226.

[35] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 10 (2) (1973) 112–122.

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

[37] G. Priestnall, J. Jaafar, A. Duncan, Extracting urban features from lidar digital surface models, Computers, Environment and Urban Systems 24 (2000) 65–78.