Manhattan-World Urban Reconstruction from Point Clouds

Minglei Li[1,2], Peter Wonka[1], and Liangliang Nan[1](✉)

[1] Visual Computing Center, KAUST, Thuwal, Saudi Arabia
mingleili87@gmail.com, pwonka@gmail.com, liangliang.nan@gmail.com
[2] College of Electronic and Information Engineering, NUAA, Nanjing, China

Abstract. Manhattan-world urban scenes are common in the real world. We propose a fully automatic approach for reconstructing such scenes from 3D point samples. Our key idea is to represent the geometry of the buildings in the scene using a set of well-aligned boxes. We first extract plane hypothesis from the points followed by an iterative refinement step. Then, candidate boxes are obtained by partitioning the space of the point cloud into a non-uniform grid. After that, we choose an optimal subset of the candidate boxes to approximate the geometry of the buildings. The contribution of our work is that we transform scene reconstruction into a labeling problem that is solved based on a novel Markov Random Field formulation. Unlike previous methods designed for particular types of input point clouds, our method can obtain faithful reconstructions from a variety of data sources. Experiments demonstrate that our method is superior to state-of-the-art methods.

Keywords: Urban reconstruction · Manhattan-world scenes · Reconstruction · Box fitting

1 Introduction

Obtaining faithful reconstructions of urban scenes is an important problem in computer vision. Many methods have been proposed for reconstructing accurate and dense 3D point clouds from images [4,5,26,31]. Besides these point clouds computed from images, there exist an increasing amount of other types of point clouds, e.g., airborne LiDAR data and laser scans. Although these point clouds can be rendered in an impressive manner, many applications (e.g., navigation, simulation, virtual reality) still require polygonal models as a basis. However, few works have addressed the problem of converting these point clouds into surface models. In fact, reconstructing polygonal models from these point clouds still remains an open problem [17,23].

The main difficulty for urban reconstruction from point clouds is the low quality of the data. For example, the obtained point clouds of urban scenes

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46493-0_4) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part IV, LNCS 9908, pp. 54–69, 2016.
DOI: 10.1007/978-3-319-46493-0_4

typically exhibit significant missing regions, as well as uneven point density. This is because the data acquisition process unavoidably suffers from occlusions. Therefore, incorporation of prior knowledge about the structure of the urban scenes into the reconstruction process becomes necessary. In this work, we aim to tackle the problem of reconstructing Manhattan-world urban scenes from the above mentioned point clouds. Such scenes are common in the real world [4].

Existing methods on urban reconstruction from point clouds are designed to handle particular types of input, i.e., either MVS point clouds [14,29], airborne LiDAR data [12,22,30,33], or laser scans [15,19], and it may not be easy to extend these methods to handle data from other sources. Moreover, most of existing methods require segmentation of a scene into individual buildings [15,28,29], and some of them require to further extract individual facades [19], which often result in long processing times. The semantic segmentation of a scene into meaningful buildings or facades is still an unsolved problem that an automatic approach often generate unsatisfied segmentation [7]. Thus, we seek a fully automatic urban reconstruction solution that does not require this segmentation step.

The key observation in this work is that fitting a set of boxes to point clouds is much more robust than directly fitting a polygonal surface model. Thus, our strategy relies on choosing an optimal subset of boxes from a large number of box hypothesis through optimization. First, the input point cloud is aligned with a global coordinate system, and a large amount of planar segments are detected from the point cloud. Then, these planes are iteratively refined to best fit the input point cloud. The refined planes partition the space of the input data into a grid consisting of a compact set of boxes. We formulate the box selection as an energy minimization problem using a Markov Random Field formulation. After optimization, the chosen subset of boxes serve as lightweight polygonal models that faithfully represent the geometry of the buildings in the scene. Experiments show that our method can handle point clouds from a variety of data sources and obtains faithful reconstruction of the geometry of the scenes. Figure 1 shows an example of our reconstruction.

The main contributions of our work include:

- a framework for the automatic reconstruction of Manhattan-world scenes by directly fitting boxes into point clouds from various sources.
- a Markov Random Field formulation for selecting an optimal subset of candidate boxes to represent the geometry of the buildings in the scene.
- an urban reconstruction approach that does not require segmentation of the scene into individual buildings or facades, and it can handle point clouds from various data sources.

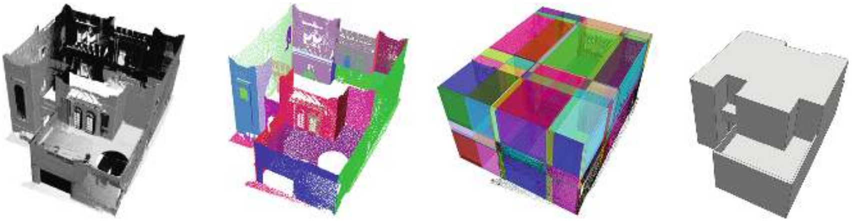
2 Related Work

Many approaches have been proposed for reconstructing physical objects represented by point clouds. A typical data-driven approach (e.g., Poisson reconstruction [11]) can generate very dense surface models from point clouds. These methods have strong requirements that data is complete and free of outliers. However, such requirements are unlikely to be guaranteed during data acquisition. To obtain complete reconstructions, model-driven approaches [4,10,13,19,28] take advantage of prior knowledge (e.g., facades are planar, facade elements are repeating) about the structure of the buildings, which seems to be more promising in reconstructing real world scenes. Some works [2,21,32] focus on recovering topologically correct assembly of hypothesized planes. However, since the initial planes are independent, these approaches may not produce accurate and simple polygonal models. In the following, we mainly review the model-driven approaches that are most related to our method.

Controur-based methods. These methods [12,14,22,30,33] were initially proposed to reconstruct urban scenes from airborne LiDAR data where the roofs of buildings are usually well captured. The typically work flow of these methods is as follows. First, contours or footprints of the buildings are extracted (usually followed by a refinement step) from the roofs of the buildings. Then, the extracted contours or footprints are extruded toward the ground plane, yielding 2.5D reconstructions that approximate the geometry of the buildings in the scene. Since only the data of the roofs of the buildings are considered, significant amount of critical information of the walls of the buildings are intentionally ignored. Thus, some important facade structures may be missing from the results.

Template-based methods. Urban scenes usually exhibit repeating structures, such as windows, doors, etc. A few template-based reconstruction methods have been proposed to reconstruct scenes containing these repeating structures. These methods first segment the point samples of a facade into meaningful segments and then obtain a detailed 3D model by replacing the segments with predefined templates. Starting from an initial coarse model consisting of few boxes constructed with user assistance, Nan et al. [18] perform 2D template matching to choose an appropriate set of detailed 3D templates and determine their locations in the 2D image domain. After that, the detailed 3D templates are positioned by projecting their 2D locations onto the faces of the 3D coarse model using the camera parameters recovered in the previous structure from motion step. Using supervised learning techniques, Lin et al. [15] first classify the input point cloud into different categories, and then decompose and fit the points of each individual building using predefined symmetric and convex blocks. Few works [20,25] also exploit the idea of template matching to reconstruct indoor scenes. Rather than recovering detailed structures of buildings, our goal is to obtain an approximate reconstruction of the buildings in Manhattan-world scenes by fitting boxes directly into point clouds.

Graph-based methods. Many approaches represent the relationships between building elements using graphs and obtain polygonal models by partitioning or optimizing the graph presentation. Garcia et al. [6] propose a surface graph cuts approach for architectural modeling based on a volumetric representation. Hiep et al. [8] reconstruct mesh models of different scales by extracting a visibility consistent mesh from the dense point clouds using a minimum *s*-*t* cut-based



(a) Point cloud (b) Plane primitives (c) Candidate boxes (d) 3D model

Fig. 1. An overview of the proposed approach.

global optimization algorithm, followed by a refinement step using image information. Similarly, Verdie et al. [29] extract a surface model from a dense mesh representation using a min-cut algorithm. Following these methods, we represent the relationship between box hypothesis, and extract an optimal set of boxes to approximate the geometry of the buildings in the scene.

Manhattan-world scene reconstruction. Another large group of papers address the problem of Manhattan-world scene reconstruction. Matei et al. [16] and Venegas et al. [28] first extract regular grammars from LiDAR point clouds. Then, a volume description of the building is established from the classified points. By assuming repetitive structures, Nan et al. [19] interactively create and snap box-like detailed structures for facade reconstruction. To reconstruct indoor scenes complying with the Manhattan world assumption, Furukawa et al. [4] and Ikehata et al. [10] approximate the geometry of indoor scenes by placing axis-aligned planes to fit the MVS point clouds. Since high-level structure information of the scenes are exploited, the reconstruction results from these methods usually outperform those from data-driven approaches that are purely based on geometric fitting, in terms of controllability over both geometric and semantic complexity of the final models. Inspired by these methods, we tackle the problem of Manhattan-world scene reconstruction by fitting a set of boxes directly into the point clouds.

3 Overview

Given a point cloud of a Manhattan-world scene, our method establishes a box approximation of the scene in two major steps: candidate boxes generation and box selection. Figure 1 shows an overview of our method.

Candidate boxes generation. We first extract a large number of planar segments from the input point cloud using a RANSAC algorithm [24]. Considering that the detected planar segments unavoidably contain undesired elements due to noise, outliers, and missing data, we refine these planar segments by iteratively merging pairs of planes that are close to each other. After that, the refined planes partition the space of the input point cloud into a set of axis-aligned boxes with non-uniform sizes (see Sect. 4).

Box selection. In this step, we choose an optimal subset of the candidate boxes to approximate the scene. We formulate the boxes selection as an energy minimization problem, where the objective function is designed to encourage the final model to be confident with respect to the input point samples and meanwhile be simple and compact. Specifically, we design two energy terms, a data fitting term to ensure the fidelity of the final model with respect to the input point cloud, and a smoothness term to encourage geometric consistency of neighboring faces in the final model. The optimal set of boxes are then chosen by minimizing the above energy function using graph cut (see Sect. 5).

4 Candidate Box Generation

4.1 Plane Extraction

Using the Manhattan-world assumption, the major components of a building (i.e., walls and roofs) consist of axis-aligned planes. Thus, we first identify the three dominant directions of the scene, as well as a set of plane hypothesis on which most of the points lie. Then we iteratively refine these planar segments and generate candidate boxes from the refined planar segments.

To determine the three dominant directions of a scene, we identify the top three strong peaks from the histogram of the normal distribution of the point cloud [4]. Then the corresponding normal directions of the three peaks are regarded as the dominant directions. With these dominant directions, we transform the point cloud such that its dominant directions align with the axes of a given coordinate frame. We directly use the normal information if it is given. Otherwise, we estimate the normal at each point using Principal Component Analysis using K-nearest neighbors. Typically, a wide range [16, 30] of K can guarantee good normal estimation.

To extract planar segments from the noisy point clouds, we exploit the RANSAC-based primitive detection method proposed by Schnabel et al. [24]. Considering the noise and outliers in the point clouds, we run the RANSAC algorithm multiple times to generate a large number of initial plane hypothesis. By doing so, appropriate planar segments describing the structure of the scene are more likely to be present in the initial plane hypothesis. We discard planar segments if either their orientations are far away (i.e., more than $20°$) from the three dominant directions, or they have a small number (i.e., 20) of supporting points.

Given the large number of plane hypothesis, we propose an algorithm that iteratively refines these initial planar segments. Specifically, we score each planar segment according to the number of its supporting points. Then, starting from the pair of planar segments with lowest average score, we merge them if the following two conditions are satisfied: (1) the angle between the two planes is less than a threshold θ_t, and (2) the distance from the center of mass of the points associated with one planar segment to that of the other one is less than a threshold d_t. After that, a new planar primitive is suggested by performing a least-squares fitting of the merged points. We repeat this process until no

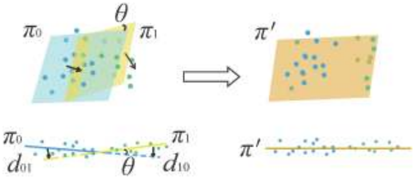
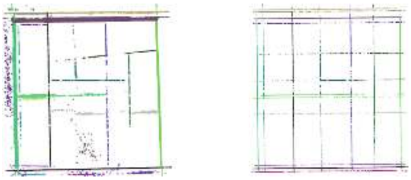


Fig. 2. The merging of two planar segments. Two planes π_0 and π_1 are merged if the angle between them is smaller than a threshold (i.e., $\theta < \theta_t$), and the distance from their mass centers is less than another threshold (i.e., $d_{01} < d_t$ and $d_{10} < d_t$). Then a new plane π' is proposed using a least-squares fitting of the union of the points.



(a) Initial segments (b) Refined segments

Fig. 3. The refinement of the initially extracted planar segments (top-view).

more pairs of planar segments can be merged. As a result, the planar segments are refined such that they are supported by more points and meanwhile the number of planar segments is significantly reduced. Figure 2 shows the merging of two planar segments. Empirically, we set θ_t to 10° and d_t to 0.2 m. A visual comparison of the planar primitives before and after refinement is shown in Fig. 3. As can be seen from (b), the arrangement of the planar segments has been significantly regularized. Meanwhile, the number of planar segments is reduced from 66 to 45.

Missing walls. The above described plane extraction method can detect most of the major planes from the point cloud. However, some critical planes could still be omitted due to the large area of missing data. This is especially true when the data is obtained by airborne equipments. For example, in the point clouds computed from aerial images or obtained by an airborne LiDAR, the walls of the buildings are extremely sparse and incomplete (see Fig. 4(a)). To ensure sufficient information for the reconstruction, we propose to determine the missing walls from their neighboring planar segments. Specifically, we first generate a height map by projecting the points onto the ground plane and rasterizing the height values of the projected points. Then, we smooth the height map by using a bilateral filter [27]. After that, line segments are detected on the height map using the Hough Transform method [3]. Note that since the height map is generated using an orthographic projection, the detected line elements are in fact the intersections of walls and roofs of the building. Thus, these wall planes can be determined by fitting vertical planes to the detected line segments. An illustration of a missing wall detection is shown in Fig. 4. We add the wall planes to the initially detected planar segment set and run the refinement algorithm illustrated in Fig. 2.

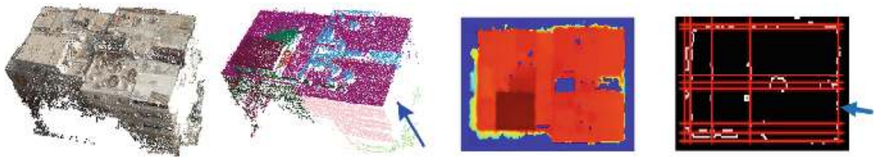

(a) Input (b) Planes (c) Height map (d) Walls (in red)

Fig. 4. Detection of missing walls. The arrow indicates a missing wall. (Colour figure online)

4.2 Candidate Boxes

In this step, we generate box hypothesis from the planar segments extracted in the previous step. According to the orientations, the refined planar segments can be categorized into three groups, i.e. $\mathbf{G}_x$, $\mathbf{G}_y$, and $\mathbf{G}_z$, which are aligned with the three dominant directions, respectively. Intuitively, the supporting planes of these planar segments partition the space of the input point cloud into a set of axis-aligned boxes. Assuming N_x, N_y, and N_z are the numbers of the planes along the three dominant directions (i.e., $|\mathbf{G}_x| = N_x$, $|\mathbf{G}_y| = N_y$, and $|\mathbf{G}_z| = N_z$), the total number of candidate boxes can be computed by

$$N = (N_x - 1) \cdot (N_y - 1) \cdot (N_z - 1). \tag{1}$$

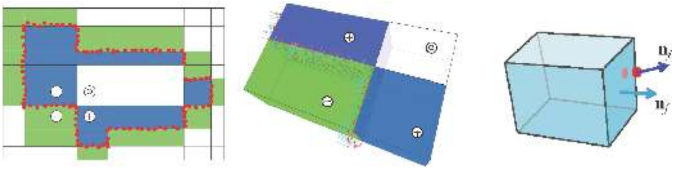
In the next step, we will propose an optimization-based box selection algorithm to choose appropriate candidate boxes to approximate the geometry of the buildings in the scene.

5 Box Selection

Given N candidate boxes $\mathbf{B} = \{b_i\}(1 \leq i \leq N)$ generated in the previous step, our goal is to choose a subset of the these boxes to approximate the 3D geometry of the buildings represented by the point samples $\mathbf{P}$. We formulate the box selection as a labeling problem so as to approximate the geometry of the buildings in the scene by a subset of the candidate boxes that favor high fidelity in data fitting and compactness in the structure of the final model. In the following, we first introduce our objectives. Then, we detail our energy function and a Markov Random Field formulation to minimize the energy function.

5.1 Objectives

To obtain a faithful reconstruction from the sampled points, we consider the following two main factors. First, the faces of the reconstructed model should be as close as possible to the input point cloud. Second, since we are seeking a set of boxes to approximate the 3D geometry of the Manhattan-world scene, the assembly of the chosen boxes should be compact and respect the structural



(a) (b) (c)

Fig. 5. Candidate boxes and supporting points. (a) A 2D illustration of the three types of candidate boxes: positive (blue), negative (green), and blank (white). (b) A zoom-in of the marked region in (a). (c) A supporting point of a face. (Color figure online)

property of the buildings, i.e., the planar facades containing the least number of holes and protrusions. We formulate these two factors as the following two objectives: data fitting and compactness.

Data fitting. We define the data fitting score $S(b_i)$ to measure how well a candidate box b_i is supported by the point cloud. Specifically, the score function $S(b_i)$ is defined as

$$\begin{aligned} S(b_i) &= \textstyle\sum_f^6 \sum_j^M \mathbf{n}_f \cdot \mathbf{n}_j \cdot dist(p_j) \\ dist(p_j) &= \begin{cases} 1/(t + d_j), \; d_j < d_t \\ 0, \qquad\qquad \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{n}_f$ denotes the normal of a face f in box b_i, and $\mathbf{n}_j$ the normal of a supporting point. To measure the fitting quality of f with respect to its supporting points $\{p_j\}(1 \leq j \leq M)$, we only consider points that are projected inside f and have distances smaller than a threshold d_t to this face (see Fig. 5(a)). Here, t is a constant to make sure that very small distances do not receive a huge weight. In our work, we set t to 1.0.

It is obvious that the data fitting score defined by Eq. 2 may have a negative value. We intended to design it in this way so as to distinguish three types of candidate boxes.

- *Positive boxes:* the boxes that have positive data fitting scores, and thus we prefer to choose them to represent the geometry of the building. Examples of this type of boxes are shown in blue in Fig. 5(a).
- *Negative boxes:* the boxes that have negative data fitting scores. The negative boxes are actually outside the volume of the building. This type of boxes are marked in green in Fig. 5(a).
- *Blank boxes:* the boxes supported either by no points, or by very few points, and thus their data fitting scores are close to zero. This type of boxes do not contribute to representing the geometry of the buildings and should be removed. Examples of blank boxes are shown in white in Fig. 5(a).

Boxes with few supporting points have very low data fitting score, thus they may be incorrectly identified as blank boxes resulting in holes in the final model.

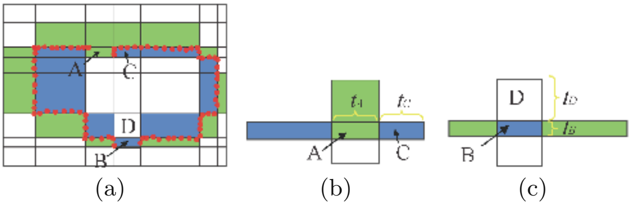


Fig. 6. A 2D illustration of holes and protrusions. Box A is misclassified as *negative* resulting in a hole; box B is misclassified as *positive* resulting in a protrusion. The thickness values t_A and t_C are used to computed the compactness for box pair A-C, and t_B and t_D for box pair B-D. Colors indicate different boxes bypes, i.e., blue for positive boxes, green for negative boxes, and white for blank boxes. (Color figure online)

To tackle this problem, we perform a smoothing procedure on the data fitting score of all blank candidate boxes. Specifically, we re-compute the data fitting score of a blank box as the area and distance weighted average of its neighbors.

$$\begin{aligned} \hat{S}(b_i) &= \textstyle\sum_{j \in Nb_i} w_j \cdot S(b_j) \\ w_j &= \frac{A(f_{ij})/d_{ij}}{\sum_{j \in Nb_i} A(f_{ij})/d_{ij}} \end{aligned} \tag{3}$$

where Nb_i are the direct neighboring boxes contacting b_i by a face; $S(b_j)$ is the data fitting score of box b_j originally computed by Eq. 2; w_j is a weight defined based on the area $A(f_{ij})$ of the contacting face of the two boxes and the distance d_{ij} between their centers.

Compactness. Since we are seeking a set of boxes as the approximate reconstruction of a building, any failure in assigning the label for a candidate box located on the surface of a building will result in a hole or a protrusion. Figure 6 shows two such examples.

To avoid holes and protrusions, we introduce the *compactness* for each pair of adjacent boxes. In our work, we say that a facade is not compact if holes or protrusion exist in this facade. Thus, to favor compactness (i.e., to avoid holes and protrusions), we prefer to assign the same label to two neighboring boxes. If the hole (or the protrusion) is caused by assigning a wrong label to a single box, the decrease in the compactness of the facade can be assessed by some value that is proportional to the thickness of the box. Specifically, we define a pairwise compactness between two neighboring boxes as

$$C_{i,j} = \frac{1}{min(t_i, t_j)} \tag{4}$$

where t_i and t_j are the thickness values of two adjacent boxes b_i and b_j. Intuitively, if one of two adjacent boxes is thin, the decrease in the compactness of assigning different labels to the boxes will be small.

5.2 Optimization

Now we describe how we select appropriate candidate boxes by using a Markov Random Field (MRF) formulation. We first construct an associated graph where the nodes represent all candidate boxes and each edge connects two adjacent boxes. This graph has a 3D grid structure where each node has a maximum number of 6 neighbors (for interior boxes) and a minimum number of 3 neighbors (for boxes at the corners of the 3D grid). We formulate the box selection as probability functions and compute the probabilities of each candidate box belonging to *positive* or *non-positive*. These box types are the labels that will be assigned to the nodes in the graph after the optimization. We employ a graph cut algorithm to partition this highly connected graph structure into optimal and consistent groups of boxes, where the boxes labeled as *positive* will then be assembled as the final approximate reconstruction of the buildings in the scene.

Our objective function consists of a data term and a smoothness term.

- *Data term.* The data term encourages to choose boxes that have higher data fitting scores.

$$D(b_i) = \begin{cases} -S(b_i), \text{ for positive boxes} \\ \ \ S(b_i), \text{ otherwise} \end{cases} \tag{5}$$

- *Smoothness term.* As has been discussed in Sect. 5.1, holes and protrusions should be avoided to ensure that the final reconstruction is compact. Thus, our smoothness term is defined to favor assigning the same label to neighboring boxes.

$$V(b_i, b_j) = \begin{cases} C_{i,j}, \text{ if } min(t_i, t_j) \leq 1 \\ 1, \quad \text{ otherwise} \end{cases} \tag{6}$$

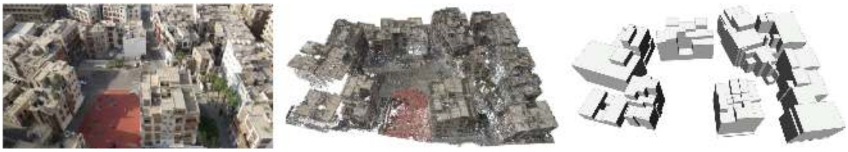
  Note that we set the smoothness weight to be a constant value of 1 if two adjacent boxes are both very thick. This is intended to handle very large boxes (i.e., boxes with a thickness larger than 1 m).

Now, our objective function can be defined as a linear combination of the sum of the above two terms.

$$E(\mathbf{X}) = \sum_{b_i} D(b_i) + \lambda \cdot \sum_{\{b_i, b_j\} \in \mathbb{E}} V(b_i, b_j), \tag{7}$$

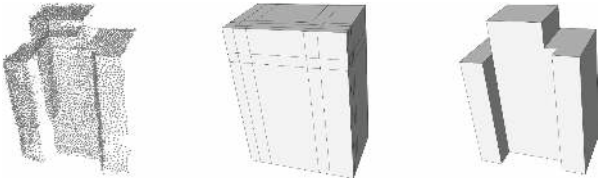
where $\{x_i\} \in \mathbf{X}$ denote the binary label (i.e., *positive* or *non-positive*) assigned to each box; λ is a weight that balances between the data term and the smoothness term. Empirically, the value of λ can be approximately computed as the average of the number of neighboring points within d_t for all data samples, where d_t is the minimum distance between two parallel planes (see Sect. 4).

The above energy function can be efficiently minimized using an existing graph cut method [1]. After the energy being minimized, the assembly of candidate boxes labeled as *positive* approximate the geometry of the buildings represented by the input point cloud.



(a) An aerial image (b) MVS point cloud (c) 3D model

Fig. 7. Reconstruction of a scene consisting of a few complex buildings from MVS point cloud.



(a) Input (b) Candidate boxes (c) 3D model

Fig. 8. Reconstruction of a single building from airborne LiDAR data.

6 Results and Discussion

We have applied our approach on a variety of datasets (including MVS data, airborne LiDAR data, laser scans, and synthetic data) and conducted both qualitative and quantitative evaluations of the proposed method.

In Fig. 7, we show a scene consisting of a few buildings reconstructed from an MVS point cloud taken from Li et al. [14]. This point cloud was computed from aerial images using SfM and MVS [31]. As can be seen from (b), even though large portions of several walls are missing from the input, our method can recover the main structure of each building and obtain a compact 3D polygonal model for the entire scene without segmenting of the scene into individual buildings.

Figure 8 shows the reconstruction result of an individual building from aerial LiDAR data. Similar to the MVS data extracted from aerial images, aerial LiDAR data is even sparser and quite a few walls of the building are missing. As can be seen from this figure, our method successfully reconstructed a polygonal model consisting of a set of boxes approximating the geometry of this building.

We also tested our method on point clouds captured by a laser scanner. Figure 1 shows the reconstruction of a two-floor residential building. The point cloud is obtained using a Leica ScanStation C10 scanner. This data has higher accuracy, but it still contains large missing regions due to occlusions. We can observe that our method generates very faithful reconstruction.

Accuracy. Since the ground truth models are not available, we evaluate the accuracy of our reconstructed models by measuring the average distance of point samples to their nearest faces in the reconstructed models. Figure 9 visualizes our reconstruction error of two examples. For all the examples shown in the paper, our average reconstruction error is less than 8 cm.

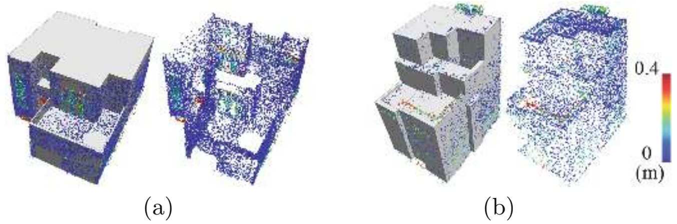


Fig. 9. The accuracy of two reconstructed models shown in Figs. 1 and 12.

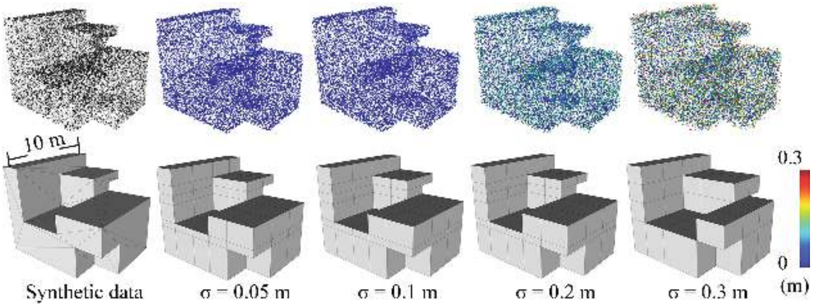
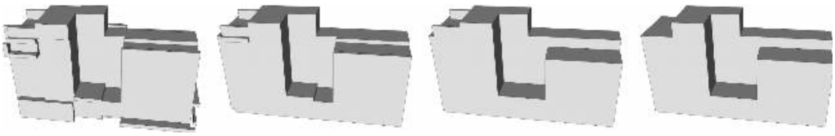


Fig. 10. Reconstruction results from a synthetic data with increasing noise levels.



(a) $\lambda = 0$ (b) $\lambda = 10$ (c) $\lambda = 30$ (d) $\lambda = 50$

Fig. 11. The effect of the parameter λ on the final model. The suggested value of λ is 29.2, which is the average number of neighboring points within 0.2 m.

Robustness. In Fig. 10, we demonstrate the robustness of our approach with respect to different levels of noise using synthetic data. In this example, we can obtain good reconstruction results when the noise level σ is less than 30 cm. However, as the noise goes larger than the minimum distance between two actual planes in the building, the RANSAC failed in extracting appropriate planar segments, yielding an incorrect 3D model.

In the box selection step, the data fitting term and the compactness term work together resulting in faithful reconstruction. To understand how much each term contributes to the final reconstruction, we tested the effect of the weight parameter λ on the final results (Fig. 11). As can be seen from this figure, smaller values of λ (i.e., more data fitting) result in gaps and bumps (a) in the final model due to noise and outliers. Increasing the value of λ favors smooth surfaces (i.e., less holes and protrusions), and thus improves the compactness of the final models. However, a too large value leads to an overly smoothed 3D model (d).

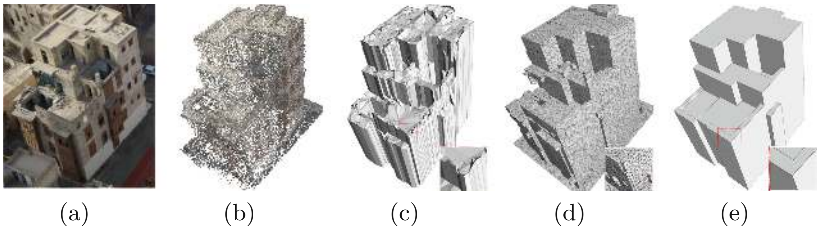
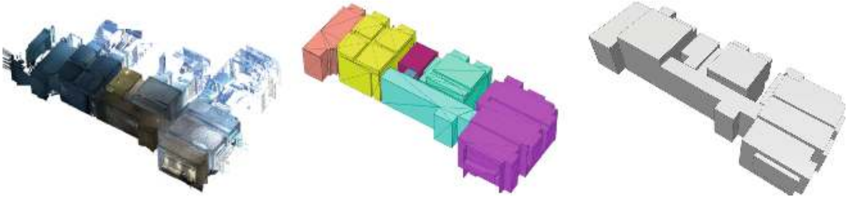


Fig. 12. Comparisons with two state-of-the-art methods. (a) An aerial photograph of the building. (b) MVS point cloud. (c) Reconstruction result using the 2.5D dual contouring method [33]. (d) The result from L_1-based polycube method [9]. (e) Ours.



(a) Input point cloud (b) Result from SIM (c) Our result

Fig. 13. A comparison with the structured indoor modeling method (SIM) on their data [10]. The SIM method requires segmenting the scene into individual rooms (color coded).

Comparisons. We also conducted comparisons with three state-of-the-art methods, namely the 2.5D Dual Contouring [33], L_1-based polycube generation [9], and a structured indoor modeling method [10] (see Figs. 12 and 13). As can be seen from Fig. 12, the result of the 2.5D Dual Contouring method (c) contains large areas of small bumps. This is because this method was initially designed to deal with data that has higher density and accuracy, and it mainly relies on roof information. Thus, it is sensitive to noise and the uneven point distribution in the roofs. The L_1-based polycube method can generate an isotropic dense surface model with more details (d). However, it usually produces undesirable surfaces (i.e., bumps and holes) passing through the outliers and the missing regions. Moreover, this method requires an initial dense 3D model as input (e.g., reconstructed using the Poisson reconstruction method [11]) and a remeshing step as preprocessing. In contrast, our method can generate more compact and visually pleasing reconstruction results (i.e., simple and clean polyhedra) as shown in (e).

In Fig. 13, we show a comparison with the structured indoor modeling approach [10]. Without segmenting the scenes into individual rooms, our method can generate comparable results.

Limitations. Our method is robust to high-levels of noise as shown in Fig. 10. To handle noise and outliers, we need to run the RANSAC algorithm multiple times during the candidate box generation step. Our experiments suggested that 10 iterations of RANSAC is usually enough to ensure that appropriate candidate boxes are proposed, but in extreme cases, it may require more iterations.

7 Conclusions

We presented a method for reconstruction of Manhattan-world scenes from point clouds. Our idea is to approximate the geometry of the buildings in the scene using a set of axis-aligned boxes. Our method is based on a *generate and select* strategy, i.e., we chose an optimal subset of boxes from a large number of candidates to assemble a compact polygonal mesh model. We formulated the box selection as a labeling problem and solved it based on a Markov Random Field formulation. Our reconstruction favors to represent the scene with a compact assembly of boxes and meanwhile respects the fitting to the input point cloud. Experiments demonstrated that our method can obtain good reconstruction for a variety of the data sources. The results of our method are polygonal models with simplified geometric structures, which can be directly used in various applications. Unlike previous methods that were designed to handle specific types of input data, our method does not have any particular requirements on the data source. Further, our method does not require semantically segmenting the input point clouds into individually buildings. Using a simple divide-and-conquer strategy (i.e., partition of the point clouds into small parts), our method can be directly applied for reconstructing large scale urban scenes.

Our method is dedicated to Manhattan-world scene reconstruction. However, it is still possible to reconstruct more general buildings by simply skipping the plane refinement step. As future work, we plan to extend our idea of box selection to polygon selection to handle more general scenes.

Acknowledgements. We thank the reviewers for their valuable comments. We also thank Dr. Neil Smith for providing us the data used in Fig. 7. This work was supported by the Office of Sponsored Research (OSR) under Award No. OCRF-2014-CGR3-62140401, and the Visual Computing Center at KAUST. Minglei Li was partially supported by NSFC (61272327). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro K5200 GPU used for this research.

References

1. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: The 9th IEEE International Conference on Computer Vision, ICCV, vol. 2, pp. 26–33 (2003)
2. Chauve, A.L., Labatut, P., Pons, J.P.: Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1261–1268. IEEE (2010)

3. Fernandes, L.A.F., Oliveira, M.M.: Real-time line detection through an improved hough transform voting scheme. Pattern Recogn. **41**(1), 299–314 (2008)
4. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: CVPR, pp. 1422–1429 (2009)
5. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)
6. Garcia-Dorado, I., Demir, I., Aliaga, D.G.: Automatic urban modeling using volumetric reconstruction with surface graph cuts. Comput. Graph. **37**(7), 896–910 (2013)
7. Golovinskiy, A., Kim, V.G., Funkhouser, T.: Shape-based recognition of 3d point clouds in urban environments. In: ICCV, pp. 2154–2161 (2009)
8. Hiep, V., Keriven, R., Labatut, P., Pons, J.: Towards high resolution large-scale multi-view stereo. In: CVPR, pp. 1430–1437 (2009)
9. Huang, J., Jiang, T., Shi, Z., Tong, Y., Bao, H., Desbrun, M.: L1 based construction of polycube maps from complex shapes. ACM Trans. Graph. **33**(3), 25:1–25:11 (2014)
10. Ikehata, S., Yang, H., Furukawa, Y.: Structured indoor modeling. In: ICCV (2015)
11. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. **32**(3), 29:1–29:13 (2013)
12. Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M.: Structural approach for building reconstruction from a single dsm. IEEE Trans. Pattern Anal. Mach. Intell. **32**(1), 135–147 (2010)
13. Li, M., Nan, L., Liu, S.: Fitting boxes to manhattan scenes using linear integer programming. Int. J. Digital Earth, 1–12 (2016)
14. Li, M., Nan, L., Smith, N., Wonka, P.: Reconstructing building mass models from uav images. Comput. Graph. **54**, 84–93 (2016)
15. Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., Yang, R.: Semantic decomposition and reconstruction of residential scenes from lidar data. SIGGRAPH **32**(4), 66:1–66:10 (2013)
16. Matei, B., Sawhney, H., Samarasekera, S., Kim, J., Kumar, R.: Building segmentation for densely built urban regions using aerial lidar data. In: CVPR, pp. 1–8 (2008)
17. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., van Gool, L., Purgathofer, W.: A survey of urban reconstruction. Comput. Graph. Forum **32**(6), 146–177 (2013)
18. Nan, L., Jiang, C., Ghanem, B., Wonka, P.: Template assembly for detailed urban reconstruction. Comput. Graph. Forum **35**, 217–228 (2015)
19. Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., Chen, B.: Smartboxes for unteractive urban reconstruction. SIGGRAPH **29**(4), 93 (2010)
20. Nan, L., Xie, K., Sharf, A.: A search-classify approach for cluttered indoor scene understanding. ACM Trans. Graph. **31**(6), 1–10 (2012)
21. Oesau, S., Lafarge, F., Alliez, P.: Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS J. Photogram. Remote Sens. **90**, 68–82 (2014)
22. Poullis, C., You, S.: Automatic reconstruction of cities from remote sensor data. In: CVPR, pp. 2775–2782 (2009)
23. Rottensteinera, F., Sohnb, G., Gerkec, M., Wegnerd, J., Breitkopfa, U., Jungb, J.: Results of the isprs benchmark on urban object detection and 3d building reconstruction. ISPRS J. Photogram. Remote Sens. **93**, 256–271 (2014)
24. Schnabel, R., Wahl, R., Klein, R.: Efficient ransac for point-cloud shape detection. Comput. Graph. Forum **26**(2), 214–226 (2007)

25. Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., Guo, B.: An interactive approach to semantic modeling of indoor scenes with an rgbd camera. ACM Trans. Graph. **31**(6), 136:1–136:11 (2012). http://doi.acm.org/10.1145/2366145.2366155
26. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: SIGGRAPH, pp. 835–846 (2006)
27. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV, pp. 839–846 (1998)
28. Vanegas, C.A., Aliaga, D.G., Benes, B.: Building reconstruction using manhattan-world grammars. In: CVPR, pp. 358–365 (2010)
29. Verdie, Y., Lafarge, F., Alliez, P.: Lod generation for urban scenes. ACM Trans. Graph. **34**(3), 15 (2015)
30. Verma, V., Kumar, R., Hsu, S.: 3d building detection and modeling from aerial lidar data. In: CVPR, pp. 2213–2220 (2006)
31. Wu, C.: Visualsfm: A visual structure from motion system, 9 (2011). http://homes.cs.washington.edu/~ccwu/vsfm
32. Zebedin, L., Bauer, J., Karner, K., Bischof, H.: Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5305, pp. 873–886. Springer, Heidelberg (2008)
33. Zhou, Q.-Y., Neumann, U.: 2.5d dual contouring: a robust approach to creating building models from aerial LiDAR point clouds. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 115–128. Springer, Heidelberg (2010)