Large Scale Urban Scene Modeling from MVS Meshes

Lingjie Zhu[1,2], Shuhan Shen[1,2](✉), Xiang Gao[1,2], and Zhanyi Hu[1,2]

[1] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{lingjie.zhu,shshen,xiang.gao,huzy}@nlpr.ia.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. In this paper we present an efficient modeling framework for large scale urban scenes. Taking surface meshes derived from multi-view-stereo systems as input, our algorithm outputs simplified models with semantics at different levels of detail (LODs). Our key observation is that urban building is usually composed of planar roof tops connected with vertical walls. There are two major steps in our framework: segmentation and building modeling. The scene is first segmented into four classes with a Markov random field combining height and image features. In the following modeling step, various 2D line segments sketching the roof boundaries are detected and slice the plane into faces. Through assigning each face with a roof plane, the final model is constructed by extruding the faces to the corresponding planes. By combining geometric and appearance cues together, the proposed method is robust and fast compared to the state-of-the-art algorithms.

Keywords: Urban reconstruction · Building modeling
Markov random field · Segment based modeling

1 Introduction

Modeling urban environment has been the core part for many applications including navigation, simulation and virtual reality. Although detailed models can be created with modern interactive softwares, it is inevitably tedious and not applicable to large city scale. Actually, automatic generation of urban models from physical measurements remains an open problem [23]. Typically, there are two types of data sources used in the reconstruction: aerial LiDAR (light detection and ranging) and aerial imagery.

Airborne LiDAR point cloud was once the first choice for city-scale modeling [17,27,35]. It is pure geometric data and usually in the form of 2.5D, *i.e.*, the LiDAR sensor captures roof structures well but fails to collect sufficient points

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01252-6_38) contains supplementary material, which is available to authorized users.

on facade. By contrast, meshes derived from oblique images using structure from motion (SfM) and multi-view stereo (MVS) workflows contain walls with details and have rich texture information [2,8,9,11,12]. Although LiDAR point cloud is more accurate, images are much cheaper and more approachable. With advanced automated SfM and MVS workflows like Pix4D [26] and Acute3D [1], people can obtain faithful meshes with realistic textures at large scale. However, these triangulated meshes are particularly dense and noisy because they do not convey any high level semantic or structural information (*e.g.*, road, building, tree, roof and wall). Therefore reconstructing them into more compact models with abstracted semantics has gained increasing attention [14,18,19,25,31,36].

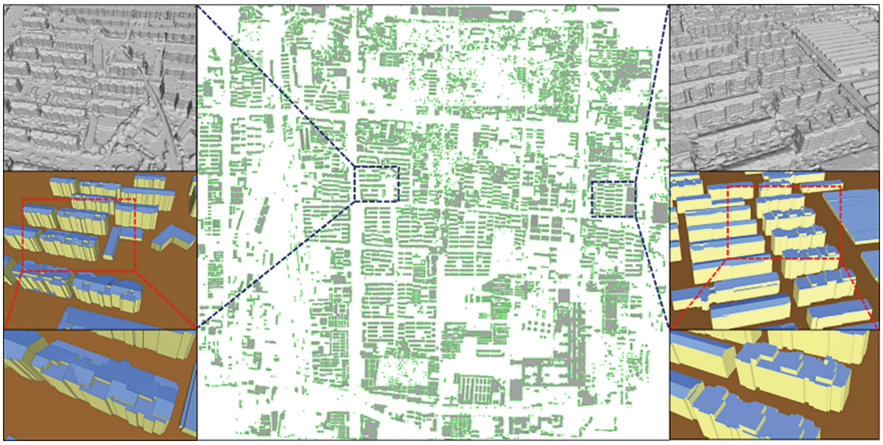
Fig. 1. Modeling of a large urban area. Our input is textured surface meshes generated from 3720 oblique aerial images. It has 92 M triangle faces covering an area of 12.2 km^2. The output model (in the middle) has 4343 buildings with 0.25M faces, enhanced with regularity and semantics. Two close-ups are shown on left and right sides

In the context of urban modeling, practitioners usually wish to present the data with semantics and levels of detail (LODs). Although classic simplification or approximation algorithms [7,13] can generate models of different complexity via controllable parameters. Herein LOD is not only in the sense of data storage or rendering, but also a simplified semantic abstraction of the scene. One aspect of our system is to generate models with LODs conforming to the CityGML [6] standard, which is a widely accepted open data model for representing and exchanging virtual 3D city models. Figure 2 shows the basic semantics and LOD abstractions defined by CityGML [6].

The proposed pipeline takes urban MVS meshes as input and outputs simplified building models with meaningful LODs adhering to the CityGML [6] standard. We utilize the 2.5D characteristic of building structures and cast the modeling as a shape labeling problem. Specifically, we first segment the scene on the orthograph into 4 classes: ground, grass, tree and building. Then various roof boundary segments are detected for each building. The segments slice the plane into pieces of faces. Built on the faces of the segment arrangement, each face is assigned with a roof plane via a Markov random field (MRF) formulation. Extruding the face to their designated plane gives the final model.

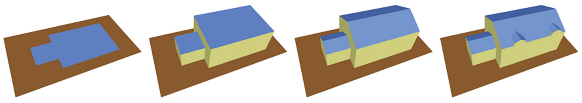

Fig. 2. Modeling semantics and LODs defined by CityGML [6]. From left to right are LODs from 0 to 3 with increasing details: LOD1 is LOD0 rising to the averaged height, LOD2 models the roof top shapes and LOD3 is decorated with superstructures. Semantics are color coded: ground in brown, facade in light yellow and roof top in blue (Color figure online)

The main contributions of our work include:

– A novel line segment based 2D shape labeling method for LOD modeling taking both appearance and geometry cues into consideration.
– A prior embedded shape detection approach, which enhances the regularity of the model with orthogonal facades and symmetric roofs.
– An efficient pipeline to generate LOD models of large urban scene from noisy MVS meshes, and validated on our real-world dataset.

2 Related Work

Two major steps of the proposed pipeline are covered in our review of previous work: urban scene segmentation and modeling.

Segmentation. Although surface mesh segmentation and image segmentation have been around for a long time in computer graphics and computer vision communities respectively. Little literature is devoted to the segmentation of meshes reconstructed from images. Verdie *et al.* [31] compute different geometric features on the surface and classify it by constructing a MRF labeling on the superfacet graph. Similarly, Rouhani *et al.* [28] add other photometric features and using a random forest to compute the MRF pair-wise potentials. Liu *et al.* [21] partition the urban surfaces into structural elements by iteratively clustering faces into bigger primitives with a high-order conditional random field.

Modeling. Speaking of urban modeling, building is of the most interest. Two categories of building modeling have been proposed recently.

Candidate selection is a common modeling strategy, which usually follows the *generation and selection* pattern. Both [19,31] slice the bounding space into candidate 3D cells with planes and transform the modeling into a binary inside/outside labeling problem. Comparing to [19,31] is restricted to Manhattan scene. Apart from the cell selection method, Nan *et al.* [24] formulate the modeling as a cell face selection problem. By putting constraints on the faces sharing an edge, the model is guaranteed to be 2-manifold by solving a linear programing problem.

Contour based modeling is another category. In [18], the authors simplify the boundaries of the pixel-wise labeled height map and generate the model by lifting the boundary polygons to 3D spaces. Given street-level imagery, GIS footprint and MVS meshes, Kelly *et al.* [15] detect the profiles of the footprint contour and generate detailed models with procedural extrusion [16]. Although the generated model is of high-quality, the input data is not always available. Zhu *et al.* [36] extend the variational shape approximation (VSA) [7] algorithm for the urban scene and model the facade contour with regularity.

LOD generation is another concern in city modeling. General mesh simplification [13] or shape approximation [7] methods can generate models of different complexity. However they are essentially geometric error driven, which are not aware of the higher-level structure or regularity presented in the scene. Build upon the detected structure primitive graph, Salinas *et al.* [29] try to preserve the structure when conducting the mesh decimation. None of them could generate LODs with respect to the semantic abstraction.

3 Overview

Taking noisy MVS meshes of large urban scene as input, our method outputs manifold models of low complexity and strong regularity in the form of semantic LODs. The proposed framework consists of two main phases: semantic segmentation and building modeling. Figure 3 shows the overview of the pipeline.

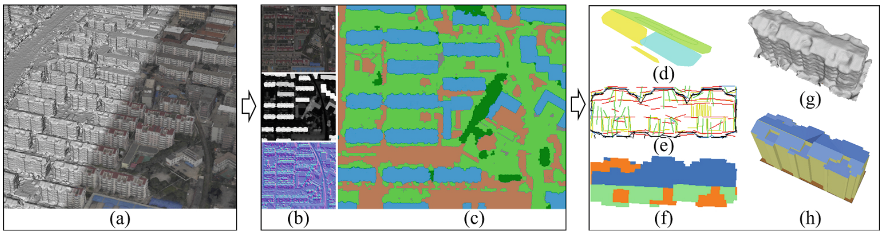

Fig. 3. Large urban scene modeling pipeline of two major steps: semantic segmentation (b)(c) and building modeling (d)–(h). (a) The input raw MVS meshes in terrestrial coordinate, rendered with (right) and without (left) texture; (c) semantic segmentation on the orthographs of (b); (d) plane detection on the roof top; (e) various line segments outlining the roof patch boundaries; (f) line segments arrangement shape labeling; (h) an example of the final model with semantics and LODs of building (g)

For city scale reconstruction, aerial oblique images acquired at high altitude are often used. Although the images are of high resolution and quality, the output MVS surface meshes still suffer from occlusion, shade, weak and repetitive texture. Compared with LiDAR point clouds, MVS meshes are more noisy as we can see in Fig. 3(a)(g).

Memory issues also arise when dealing with city scale data. Rendering such dense and large scene alone can be challenging. To curb the computation burden, we cut the input mesh into several memory manageable blocks and each block can be processed in parallel. In the segmentation part, we fuse both geometry and appearance information on the orthographs, Fig. 3(b)(c). Then we model each building with a segment based modeling method as demonstrated in Fig. 3(d)–(f).

4 Segmentation

Our segmentation step relies on a MRF built on the orthographs to distinguish four classes of urban objects: *ground, grass, tree* and *building*. Through combining a specialized supervised tree classifier and geometric attributes, we can achieve decent result that meets our need with a simple formulation.

Raw MVS surfaces usually contain many geometric and topological defects such as self-intersections, non-manifold edges and floating parts. Instead of operating on the 3D meshes directly like [28,31], we sample the textured mesh into a 2D orthograh representation. Given a grid sampling step, a vertical ray is cast from above at each grid center. Recording the texture color, height and normal of the first intersected point gives us an orthophoto, a depth map and a normal map respectively, as shown in Fig. 4. The obvious advantages of the orthogonal grid sampling are: (1) data are evenly distributed with efficient access, (2) both geometric and texture information are in the same image form, (3) many off-the-shelf image processing algorithms are available.

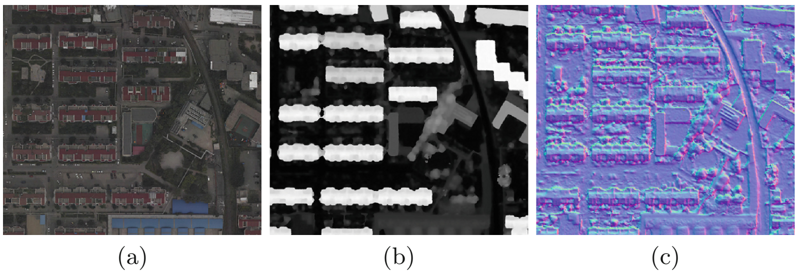

Fig. 4. Orthogonal grid sampling of a block, step size is set to $0.2m$. (a) Orthophoto. (b) Height map (linearly mapped into range [0, 255]). (c) Normal map colored by treating x, y, z components as RGB channels respectively, the blueish color shows that normals are mostly facing upwards (Color figure online)

Recent CNN-based segmentation methods [22,32] have impressive performances. Despite that we choose a traditional boosting and MRF combined method over them because: (1) the large amount of training data is hard to acquire, (2) they are unable to integrate orthophoto features with geometric data (*e.g.*, height or normal) effectively [22], (3) the segmentation here is more treated as a building isolation step rather than a pixel-level precision labeling task, and it serves the subsequent modeling step well in our experiments, Sect. 6.

4.1 Tree Probability

Although buildings may have various textures and colors, trees generally have distinguishable color and similar pattern (Fig. 4). Based on the specialized orthophoto tree detection algorithm [34], we give each pixel a value measuring the tree probability. To be more specific, we compute visual features x_i at each pixel p_i of (1) color: CIE L*a*b* color and the illumination-invariant color [5], (2) texture: Gaussian derivative filter-bank at 3 scales (with $\sigma = 1, \sqrt{2}, 2$) and 6 uniform sampled directions in $[0, \pi)$, (3) entropy of L channel: with window sizes of 5, 9 and 17. With these features, we use the boosting algorithm [10] to train a strong classifier $F(x_i)$ from T basic decision stump $f_t(x_i)$:

$$F(x_i) = \sum_{t=1}^{T} \alpha_t f_t(x_i), \tag{1}$$

where T is set to 200 across our experiments and the weights α_t is learned in the training process. Then a tree probability $t(p_i)$ is given by the sigmoid function:

$$t(p_i) = \frac{1}{1 + e^{-F(x_i)}}. \tag{2}$$

We trained our model on a small area of about $300\,\text{m} \times 300\,\text{m}$, Fig. 5(a) shows the predicted tree probability of the same block in Fig. 4.

4.2 MRF Labeling

With the tree probability and the observation that only trees and buildings rise above the ground of few meters, our label set is $l^p = \{ground, grass, tree, building\}$. Since ground is almost flat in a block, we measure the height with a normalized value defined as:

$$h_n(p_i) = L(max(h(p_i) - h_{ground}, 0)), \tag{3}$$

where $L(x) = 1/(1 + e^{-2(x - h_{trunc})})$ is a logistic function with "S"-shaped curve and its midpoint is modulated by a truncation value h_{trunc}. h_{ground} is the height of block ground. Figure 5(b) shows the normalized height map.

With the tree probability $t(p_i)$ and the normalized height $h_n(p_i)$, each pixel p_i is defined with a likelihood data term:

$$D^p(l_i^p) = \begin{cases} h_n(p_i) \cdot t(p_i) & \text{if } l_i^p = ground \\ h_n(p_i) \cdot (1 - t(p_i)) & \text{if } l_i^p = grass \\ (1 - h_n(p_i)) \cdot (1 - t(p_i)) & \text{if } l_i^p = tree \\ (1 - h_n(p_i)) \cdot t(p_i) & \text{if } l_i^p = building \end{cases}. \tag{4}$$

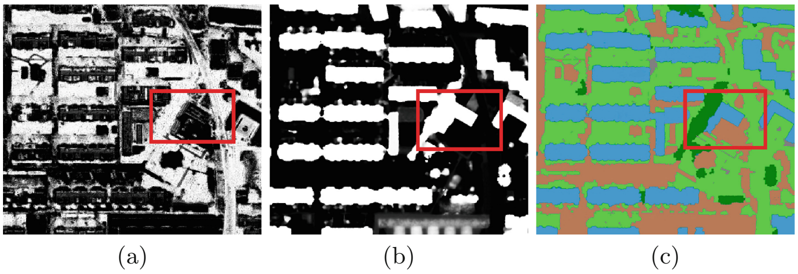

Fig. 5. (a) Tree probability learned from the boosting algorithm. (b) Normalized height map. (c) Result of segmentation: *ground*, *grass*, *tree* and *building* are colored with brown, light green, dark green and blue respectively. Notice the adjacent tree and building are correctly separated in red rectangle (Color figure online)

The pairwise smooth relation is measured by their clamped height difference:

$$V^p(l_i^p, l_j^p) = (1 - [h(p_i) - h(p_j)]_0^1) \cdot \mathbf{1}_{\{l_i^p \neq l_j^p\}}, \tag{5}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the characteristic function. Then the objective function on the image grid graph $\mathcal{G}^p = \{\mathcal{V}^p, \mathcal{E}^p\}$ can be written as:

$$E^p(l^p) = \sum_{i \in \mathcal{V}^p} D^p(l_i^p) + \mu \sum_{\{i,j\} \in \mathcal{E}^p} V^p(l_i^p, l_j^p), \tag{6}$$

where l is the label configuration of the orthophoto. μ is the balance and set to 1 in all experiments shown in this paper. The proposed energy function can be efficiently minimized by graph-cuts [3,4].

To start the segmentation, we use the lowest point in the block as an initial estimation of h_{ground}. After running MRF once, all ground pixels are averaged to give better estimation of h_{ground}. With updated h_{ground}, the normalized heights h_n are recomputed and the MRF formulation is solved again. This iterative technique will give more accurate ground height estimation and thus better segmentation of the scene.

In our experiment, h_{ground} typically converges after 3 iterations and h_{trunc} is set to $3m$ as the height of an one story building. After the segmentation, we apply a morphology open operation to break weakly connected buildings followed by a close operation to eliminate small holes. An example of the labeling result is shown in Fig. 5(c).

5 Modeling

With the labeled block, each connected *building* region is isolated from the scene. The objective of this section is to reconstruct buildings into LOD models with enhanced regularity.

Since images for urban reconstruction are usually obtained from aerial vehicles, facades suffer more from occlusion and shade than the roofs. Therefore, quite a few works [18,27,35] take advantage of the 2.5D nature of the buildings and mainly use the roof information for modeling. Here 2.5D means buildings can be viewed as piece-wise planar roofs connected with vertical walls. Both [27,35] deal with LiDAR point clouds and extract roof contours by analysing the point geometric features, making them vulnerable to noise and outliers. Li *et al.* [18] uses a pixel-wise labeling to extract the contours of the roof patches. However none of them tries to model the regularity (*e.g.*, symmetry, orthogonality and parallelism) or is capable of generating models with LODs.

In this section, we propose a segment based modeling approach. We first construct a roof boundary segment arrangement on the ground plane, then assign each arrangement face a roof plane label. Extruding each face to the assigned plane gives the model. Comparing to previous work, we consolidate different data sources together, producing models with LODs and enhanced regularity.

5.1 Facade Directions

One common regularity of urban building is that its facade has two orthogonal directions. We detect the local orthogonal directions of the facade with the RANSAC algorithm.

Specifically, the input data $\{n_f\}$ is a set of 2D unit normals constructed from the horizontal projections of face normals that are within an angle threshold ang_{thre} to the ground plane. Each time a hypothesis normal n_o is generated. Together with its orthogonal counterpart n'_o, the inlier set is defined as:

$$\{n | min(cos^{-1}(n \cdot n_o), cos^{-1}(n \cdot n'_o)) < ang_{thre}, n \in \{n_f\}\}. \quad (7)$$

Then, the maximum consensus set is considered as the facade orthogonal directions. Figure 6(a)(b) show the detection of the facade directions on a small house provided by Pix4D [26] open data set.

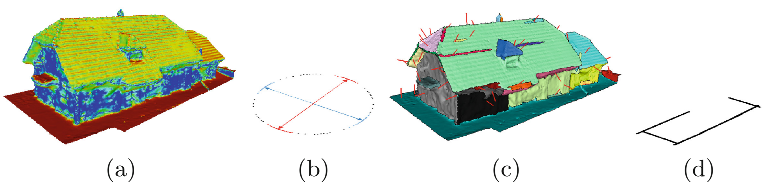

Fig. 6. Facade directions and segments detection on the Pix4D [26] house. (a) Z component of face normals with the jet color map. (b) The near horizontal unit normal circle $\{n_f\}$, two orthogonal (red and blue arrows) directions are detected. (c) VSA [7] segmentation on the mesh with color coded proxies and unit normal of short red line. (d) The facade segments detection by projecting near vertical proxies on the ground. Notice the boundary is not closed due to occlusion (Color figure online)

5.2 Line Segments

Generally roof tops are piece-wise planar and their boundaries projected on the ground are polygons. This inspires us to detect line segments that outline the roof patches. With the orthogonal sampling in Sect. 4, each building has an orthophoto, a normal map and a height map. We utilize these complementary data sources to detect various edge segments.

Image Segments. The most straight forward boundary cue is line segments from orthophoto. Due to the noise and mismatch, MVS mesh generation algorithms generally perform some smoothing operations, making the reconstructed surface corner rounded (Fig. 3(g)). Although it is difficult to locate the edges from geometric data, it can be easily spotted in the image. Therefore, we use the line segment detection (LSD) [33] algorithm to detect line segments from the orthophoto, as illustrated in Fig. 7(a).

Facade Segments. Apart from the visual cues, we also detect segments from geometry data. The facade contouring segments is detected with the plane approximation method VSA [7]. Specifically, a minimum area (set to $10\,\mathrm{m}^2$) is used to estimate the number of proxies used to approximate the building shape. And we use the random seeding to give a rough and fast segmentation of the mesh. By projecting vertical proxies (controlled by ang_{thre}) onto the ground, we have facade line segments bounding the roofs, as shown in Fig. 7(c)(d).

Height Map Segments. When the building facade is touching a tree, there is no mesh at the facade thus VSA [7] is unable to detect any proxy here. To solve this problem, we treat height map as an intensity image and use the same LSD [33] algorithm to detect height discontinuity segments as shown in Fig. 7(c).

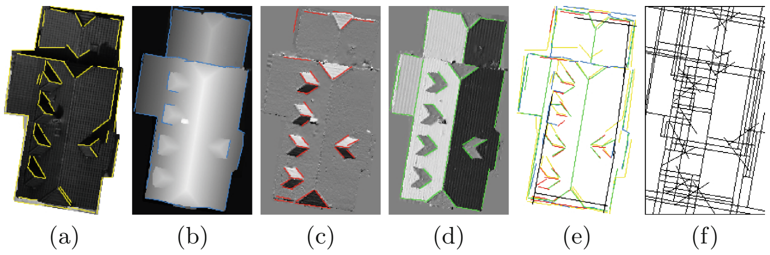

Fig. 7. Segments detected from different sources of the house in Fig. 6. (a) Image segments. (b) Height map segments. (c)(d) *Directional normal variation* maps and the detected ridge segments. (e) The stacked segments from (a)–(d) and facade segments in Fig. 6(d). (f) The regularized line segments slices the plane into faces

Normal Map Segments. While height map segments is evident where sufficient height disparity exists, it is unable to detect the ridge lines where two roof patches meet. In contrast, ridges are quite noticeable in the normal map as shown in Fig. 4(c). Unfortunately, there is no clear definition of the gradient of a 3-channel image, so we can not apply the LSD [33] on it directly.

One important structural characteristic of the building is that roof normals are in the planes constructed by the z axis and facade normals [15,16], as depicted in Fig. 8(a). We construct two *directional normal variation* maps by computing the dot products of the normal map and two orthogonal facade directions in Sect. 5.1 respectively, highlighting the normal changes along each direction. Then we can apply the LSD [33] on the two generated intensity images and spot the ridge lines easily, as illustrated in Fig. 7(c)(d).

Stacking all these line segments together, we can outline the complete contours of major roof patches, as shown in Fig. 5(e). To enhance the regularity of the contours, we employ the two step regularization technique described in [36]: (1) segments parallel to the facade directions are reoriented to the facade directions first, (2) then coplanar segments are repositioned and merged into one longer segment. Figure 7(f) shows the regularized segments slice the plane into smaller polygon faces. The collinear threshold is set to $0.5m$ in our experiment.

5.3 Shape Detection

An often underestimated obstacle in modeling from MVS data is the detection of shape primitives that are complete and with regularity. Unlike previous works [17,20] using the *first-detect-then-regularize* strategy, we extend the RANSAC [30] algorithm to model regularities into the shape detection process.

Roof Direction Detection. Before detecting roof planes, we detect prominent roof directions on the normal map first. All normals are collected in $\{n_r\}$ as the input for the RANSAC. When a hypothesis direction n_s is proposed, it is snapped to the nearest roof direction plane (based on the characteristic discussed in normal map segments). Similar to Sect. 5.1, the inlier set is defined as

$$\{n | min(cos^{-1}(n \cdot n_s), cos^{-1}(n \cdot n'_s)) < ang_{thre}, n \in \{n_r\}\}, \tag{8}$$

where n'_s is the z-symmetry counter part of n_s. Figure 8(a)(b) shows the roof principal direction detection.

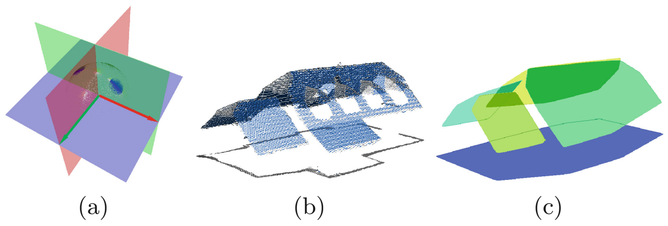

Fig. 8. Roof shape detection of the building in Fig. 6. (a) The normal sphere $\{n_r\}$ and roof direction planes. (b) Detected symmetric roof direction inliers (blue points). (c) 5 detected shapes: 2 pairs of z-symmetric/parallel planes and a horizontal plane (Color figure online)

Roof Plane Detection. Based on the RANSAC shape detection [30], we detect planes along the roof principal directions on the height map. In the same spirit to roof direction detection, each hypothesis plane is snapped to the nearest roof principal directions. Figure 8(c) shows the detected roof planes along the roof directions. With the direction constraints stemmed from facades, the detected planes $\{P_i\}$ is inherently encoded with parallelism and z-symmetry.

5.4 LOD Modeling

With the regularized segment arrangement in Sect. 5.2, we have a dual graph $\mathcal{G}^f = \{\mathcal{V}^f, \mathcal{E}^f\}$, where each face f_i is a vertex and adjacent faces is connected with an edge. Another MRF is build on $\mathcal{G}^f$ with the detected roof planes as the label set $l^f = \{P_i\}$.

MRF Formulation. In the height map in Sect. 4, each pixel p_i is treated as a 3D point p'_i. To measure how well a plane is fitted to the points bounded by face f_i, each face f_i is defined with a data term:

$$D^f(l_i^f) = \sum_{p_k \in f_i} \|p'_k, l_i^f\|. \tag{9}$$

The pairwise smooth relation between two adjacent faces f_i, f_j is weight by their shared edge length $len_{i,j}$:

$$V^f(l_i^f, l_j^f) = len_{i,j} \cdot (1 - \frac{3\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}) \cdot \mathbf{1}_{\{l_i^f \neq l_j^f\}}, \tag{10}$$

where λ_0 denotes the minimum eigenvalues of the covariance matrix of points $\{p'_i | p_i \in f_i \vee f_j\}$ measuring the planarity of two faces [31]. $\mathbf{1}_{\{\cdot\}}$ is the characteristic function. With the data and smooth terms balanced by β, the objective function of a labeling configuration l^f is:

$$E^f(l^f) = \sum_{i \in \mathcal{V}^f} D^f(l_i^f) + \beta \sum_{\{i,j\} \in \mathcal{E}^f} V^f(l_i^f, l_j^f). \tag{11}$$

We use the same graph-cut [3,4] algorithm to solve the above problem, and β is set to 10 in our experiments.

LOD Generation. Given the 2D faces labeled with roof planes, models of LODs can be extracted by projecting the face vertices to the corresponding plane, as illustrated in Fig. 9 (same house in Fig. 6):

– LOD0: the outer boundary of non-ground face is the building's footprint,
– LOD1: each face is extruded to the plane's averaged height,
– LOD2: each face is extruded to the corresponding roof planes.

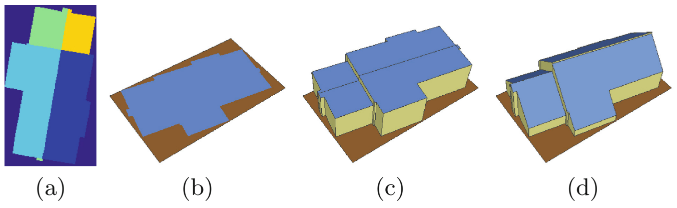

Fig. 9. LOD generation. (a) Segment arrangement color coded with the plane labels in Fig. 8(c). (b) LOD0 model of the outer most boundary. (c) LOD1 model extruded to the averaged heights. (d) LOD2 model extruded to the corresponding 3D roof planes

6 Results and Discussion

Our method is implemented in C++ with the CGAL and the max-flow library [3, 4]. In this section, we test our method on both open data set and our own large urban scene data. Experiments show that our modeling workflow can generate regular and accurate building models with different LODs. Both qualitative and quantitative assessments are conducted.

Modeling Quality. Three aspects are considered in evaluating the modeling quality: reconstruction error, model complexity and regularity. In Fig. 10, we compare the modeling results on the public Pix4D [26] house model in Fig. 6. Two recent *candidate selection* methods [24,31] and two classic shape simplification algorithms [7,13] are compared with our method.

Generally, geometric error driven methods like VSA [7] and QEM [13] can generate models of LODs with controllable parameters. Although both methods are able to reduce the complexity of the model dramatically while keeping the error low, they are more sensitive to noise and outliers and unable to convey the global regularity of the model.

More recent approaches [19,24,31] employ the *slice and selection* strategy. While [19,31] choose a cell based selection and assembly method, Nan *et al.* [24] take the edge selection approach. As illustrated in Fig. 10, both methods can produce models of high regularity but the accuracy is low. One common difficulty of the 3D slicing methods is that they suffer from noise and incomplete primitives. For example, facades can be severely occluded because the images are captured from above. Without the complete structures of the model, both [24,31] failed in Fig. 10 (in rectangles). In the proposed method, we consolidate various complementary information from appearance to geometry into one modeling framework. We are able to infer the incomplete structures from the scene.

The proposed workflow can reach the balance of model complexity, accuracy and regularity. In Table 1, our LOD2 model has slightly higher error than [7,13] but much lower complexity and more regular surfaces.

Scalability and Performances. By orthogonal grid sampling, we simplify the 3D modeling into a 2D shape labeling problem, making the modeling relatively fast. In fact, the modeling speed is more affected by the sampling step size. In

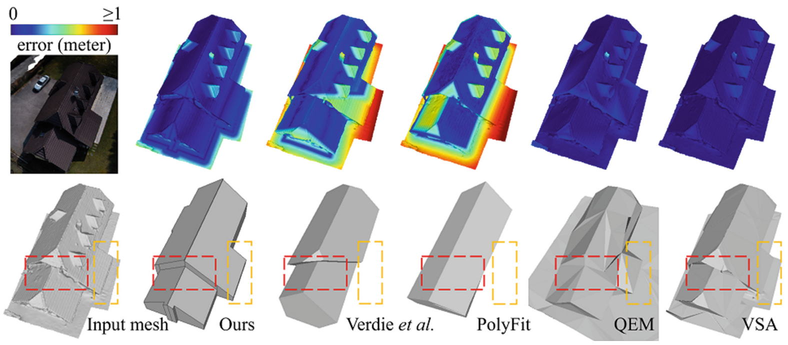

Fig. 10. Modeling quality comparison of accuracy and regularity to our LOD2 model. At each column we have the reconstructed model below and the corresponding modeling error to the input mesh on the top. The accuracy is measured by the Hausdorff distance from the output models to the input mesh, color scaled from blue to red. Both Verdie *et al.* [31] and Polyfit [24] suffer from incomplete primitives of the protruded parts (in red and orange rectangle). QEM [13] and VSA [7] have higher quality but less regularity (Color figure online)

Table 1. Comparison of modeling accuracy and complexity. The accuracy is measured by the RMS of the Hausdorff distance to the input mesh. The generated model complexity is accessed by the number of triangle faces. Our LOD2 model can reach the balance of accuracy and complexity

Method	Our LOD1	Our LOD2	Verdie *et al.* [31]	PolyFit [24]	QEM [13]	VSA [7]
RMS	2.25	0.21	1.16	1.27	0.14	0.11
#Face	24	24	153	52	149	830

Table 2. Running time on different data sets. Data complexity measured by the number of triangles. We run all modeling in a sequential implementation, except for the large city district in Fig. 1

Input	Segmentation	Modeling
Pix4D house (418.6 k faces, Fig. 10)	5 s	8 s
Apartment (92 k faces, second row in Fig. 11)	4 s	3 s
Block (588 k faces, Fig. 3)	25 s	42 s
City district (92 M faces, Fig. 1)	22 min	49.5 min

Table 2, we list the timing of different sizes of data set. All timings are measured on a PC with a 4 cores Intel Xeon CPU clocked at 3.7 GHz. VSA [7] and QEM [13] are either iterative method or sequential, neither of them scales well with data. The computational speed of PolyFit [24] relies on the linear programming solver. When there are too many intersections in the model, the solver may take really long time.

As Table 2 shows our pipeline scales well with data and can deal with input meshes with up to a hundred million triangles. Figure 1 shows a large urban area reconstructed from 356 high-resolution oblique aerial images by Pix4D [26]. The flight height is 750 m and the averaged definition is 0.1 m. The input mesh has 94M triangles covering an area of 12.2 km^2. We cut the data into 16×16 blocks and run them in parallel. The orthogonal sampling step is set to 0.2 m. Buildings that are cut at the block borders are later merged to avoid unnatural seams. On the left and right sides of Fig. 1 are two close-ups. Although obvious defects are visible in the input building's mesh model, we can still recover the sharp structures of roof tops and small protrusions on facades with the parallelism, coplanarity and z-symmetry regularity. Figure 11 shows some other modeling results. More results are available in the supplementary materials.

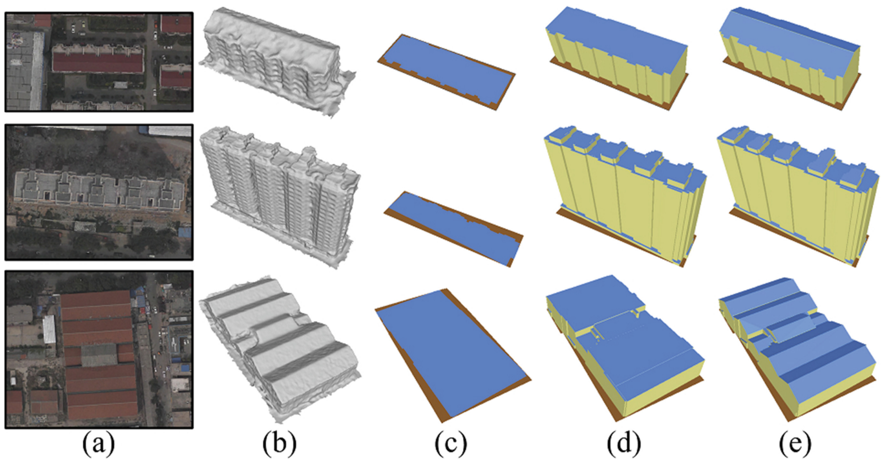

Fig. 11. LOD modeling on typical urban buildings. From top to bottom rows are ordinary residential building, high rising apartment and low connecting houses. From left to right at each row are: (a) orthograph, (b) isolated input building mesh, (c)–(e) LOD0, LOD1, LOD2 models respectively

7 Conclusions

In this paper we present a complete framework of LOD modeling from MVS meshes of large urban scene. We first segment the scene with a simple yet effective MRF formulation by fusing the visual and geometry features. The subsequent modeling method take advantage of the structure characteristics of the urban building and transform the 3D modeling into a 2D shape labeling problem. With line segments detected from complementary data sources, we are able to model the scene with strong regularity and low complexity. The result of our method is regular polygon models with LODs conforming to the CityGML standard [6], which can be used for presentation or further processing.

Acknowledgment. This work was supported by the Natural Science Foundation of China under Grants 61632003, 61333015 and 61421004. We also thank China TOPRS Technology Co. Ltd. for the city dataset in the paper.

References

1. Acute3D: Acute3D. https://www.acute3d.com/
2. Agarwal, S., et al.: Building Rome in a day. Commun. ACM **54**(10), 105–112 (2011)
3. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1124–1137 (2004)
4. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
5. Chong, H.Y., Gortler, S.J., Zickler, T.: A perception-based color space for illumination-invariant image processing. In: ACM SIGGRAPH, pp. 61:1–61:7 (2008)
6. CityGML: CityGML. http://www.opengeospatial.org/standards/citygml
7. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. In: ACM SIGGRAPH, pp. 905–914 (2004)
8. Cui, H., Gao, X., Shen, S., Hu, Z.: HSfM: hybrid structure-from-motion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2393–2402 (2017)
9. Cui, H., Shen, S., Gao, W., Hu, Z.: Efficient large-scale structure from motion by fusing auxiliary imaging information. IEEE Trans. Image Process. **22**(11), 3561–3573 (2015)
10. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
11. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1434–1441 (2010)
12. Furukawa, Y., Ponce, J.: Accurate camera calibration from multi-view stereo and bundle adjustment. Int. J. Comput. Vis. **84**(3), 257–268 (2009)
13. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: ACM SIGGRAPH, pp. 209–216 (1997)
14. Hofer, M., Maurer, M., Bischof, H.: Efficient 3D scene abstraction using line segments. Comput. Vis. Image Underst. **157**(4), 167–178 (2017)
15. Kelly, T., Femiani, J., Wonka, P., Mitra, N.J.: BigSUR: large-scale structured urban reconstruction. ACM Trans. Graph. **36**(6), 204:1–204:16 (2017)
16. Kelly, T., Wonka, P.: Interactive architectural modeling with procedural extrusions. ACM Trans. Graph. **30**(2), 14:1–14:15 (2011)
17. Lafarge, F., Mallet, C.: Building large urban environments from unstructured point data. In: IEEE International Conference on Computer Vision (ICCV), pp. 1068–1075 (2011)
18. Li, M., Nan, L., Smith, N., Wonka, P.: Reconstructing building mass models from UAV images. Comput. Graph. **54**, 84–93 (2016)
19. Li, M., Wonka, P., Nan, L.: Manhattan-world urban reconstruction from point clouds. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 54–69. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_4

20. Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., Mitra, N.J.: GlobFit: consistently fitting primitives by discovering global relations. In: ACM SIGGRAPH, pp. 52:1–52:12 (2011)
21. Liu, J., Wang, J., Fang, T., Tai, C.L., Quan, L.: Higher-order CRF structural segmentation of 3D reconstructed surfaces. In: IEEE International Conference on Computer Vision (ICCV), pp. 2093–2101 (2015)
22. Liu, Y., Piramanayagam, S., Monteiro, S.T., Saber, E.: Dense semantic labeling of very-high-resolution aerial imagery and LiDAR with fully-convolutional neural networks and higher-order CRFs. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1561–1570 (2017)
23. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Gool, L.V., Purgathofer, W.: A survey of urban reconstruction. Comput. Graph. Forum **32**(6), 146–177 (2013)
24. Nan, L., Wonka, P.: PolyFit: polygonal surface reconstruction from point clouds. In: IEEE International Conference on Computer Vision (ICCV), pp. 2372–2380 (2017)
25. Nguatem, W., Mayer, H.: Modeling urban scenes from pointclouds. In: IEEE International Conference on Computer Vision (ICCV), pp. 3857–3866 (2017)
26. Pix4D: Pix4D. https://pix4d.com/
27. Poullis, C.: A framework for automatic modeling from point cloud data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2563–2575 (2013)
28. Rouhani, M., Lafarge, F., Alliez, P.: Semantic segmentation of 3D textured meshes for urban scene analysis. ISPRS J. Photogramm. Remote. Sens. **123**, 124–139 (2017)
29. Salinas, D., Lafarge, F., Alliez, P.: Structure-aware mesh decimation. Comput. Graph. Forum **34**(6), 211–227 (2015)
30. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for point-cloud shape detection. Comput. Graph. Forum **26**(2), 214–226 (2007)
31. Verdie, Y., Lafarge, F., Alliez, P.: LOD generation for urban scenes. ACM Trans. Graph. **34**(3), 30:1–30:14 (2015)
32. Volpi, M., Tuia, D.: Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. IEEE Trans. Geosci. Remote. Sens. **55**(2), 881–893 (2017)
33. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: A fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. **32**(4), 722–732 (2010)
34. Yang, L., Wu, X., Praun, E., Ma, X.: Tree detection from aerial imagery. In: ACM GIS, pp. 131–137 (2009)
35. 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). https://doi.org/10.1007/978-3-642-15558-1_9
36. Zhu, L., Shen, S., Hu, L., Hu, Z.: Variational building modeling from urban MVS meshes. In: IEEE International Conference on 3D Vision (3DV), pp. 318–326 (2017)