2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Automatic Building Extraction from Oblique Aerial Images

Xiaofeng Sun[1, 3], Shuhan Shen[1, 2], and Zhanyi Hu[1, 2]
[1] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Key Laboratory for Aerial Remote Sensing Technology of NASG, Beijing 100039, China
email: { xiaofeng.sun, shshen, huzy }@nlpr.ia.ac.cn

***Abstract*—In this paper we propose an automatic urban building extraction method for oblique aerial images. Five steps are included in this method: point cloud generation, grid partition, feature extraction, building detection and building reconstruction. Taking advantages of recent progress in large-scale Structure from Motion (SfM) and Multiple View Stereo (MVS), dense point cloud is generated first. Then, we project the point cloud into a regularly spaced grid in XY plan, and convert the building extraction problem into an image segmentation problem. By combining the strength of the geometric attribute and spectral attribute, three complementary features are extracted and a MRF based graph model along with an energy function is created. Points belonging to buildings are recognized by minimizing this function, and prismatic 3D building models are reconstructed accordingly.**

Keywords—building detection; building reconstruction; point cloud; oblique aerial images; MRF

I. INTRODUCTION

Building extraction in urban scene is an exciting research topic which essential for a variety of applications such as map updating, urban planning, virtual tourism, city modeling and automatic driving. Generally speaking, it can be subdivided into two interdependent tasks, detection and reconstruction, and both tasks are very difficult by their own right, especially in dense environments [1, 2]. Despite the high volume of previous work in the field, there are many unsolved problems, especially when it comes to the development of fully automatic methods [3, 4].

Based on the types of data sources employed, the existing approaches can be roughly categorized into single-image-based extraction, multiple-images-based extraction, point-cloud-based extraction and data-fusion-based extraction. Single-image-based extraction only makes use of the available optical [5] or Synthetic Aperture Radar (SAR) [6] image as the sole data source. For the reason that the information contained in this data source is limited, they are usually used in some simple or sparse urban scene for 2D building footprints extraction. To extract buildings from dense urban areas, Fradkin et al. [1] first reconstruct the scene surface, then extract building facades from the 3D data accumulation in object space based on multiple-overlap aerial images. Xiao et al. [7] directly detect building facades in multiple-overlap oblique images using edge and height information, then verify and refine the building hypotheses by 3D points generated by dense image matching. For the point-cloud-based extraction, although a number of novel methods were proposed in recent years [8-10], the gap between the state of the art and the desired goal of automatic modeling from point cloud data still remains wide [9]. To utilize the complementary characteristics of multi-source data, numerous methods haven been proposed to perform the data fusion in the community of photogrammetry and remote sensing [11] and a comparative analysis of different data-fusion-based automatic building extraction methods was done by Khoshelham et al. [12].

In this paper we propose a novel oblique aerial images based automatic building extraction method. Large-scale Structure from Motion (SfM) and Multiple View Stereo (MVS) are first used to generate dense point cloud, then buildings are extracted from these 3D points automatically. Compared to LIDAR (light detection and ranging) point cloud, MVS point cloud has some adverse properties for building extraction, i.e. lower accuracy, higher but irregular density distribution and large area points missing in texture-poor regions [13, 14], which draw a higher demand to our extraction method. However, considering it's low-cost in data acquisition relative to LIDAR and high-consistency relative to multi-source data by data-fusion. We think it is suitable for building extraction.

The rest of this paper is organized as follows. Details of the proposed building extraction approach are presented in Section 2. Experimental results and discussion are reported in Section 3, followed by some concluding remarks and future work in Section 4.

II. ALGORITHM DESCRIPTION

The proposed method consists of five steps: point cloud generation, grid partition, feature extraction, building detection and building reconstruction. The framework of this method is illustrated in Fig. 1. First we use SfM+MVS algorithm to generate dense point cloud based on the oblique aerial images. Then, the generated 3D point cloud is partitioned by a regular spaced grid and three types of features are extracted. Finally, according to these features, buildings are distinguished from other objects in the scene, and 3D building models are reconstructed in prismatic style.

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

A. Point Cloud Generation and Grid Partition

After a comparative analysis to the state-of-the-art SfM and MVS approaches, a global SfM method that fuses auxiliary imaging information [15] and a PatchMatch Stereo based MVS method [16] are used in our pipeline due to the robustness and computational efficiency requirements for large-scale aerial images modeling. The output of this algorithm are 3D points with color and normal information.

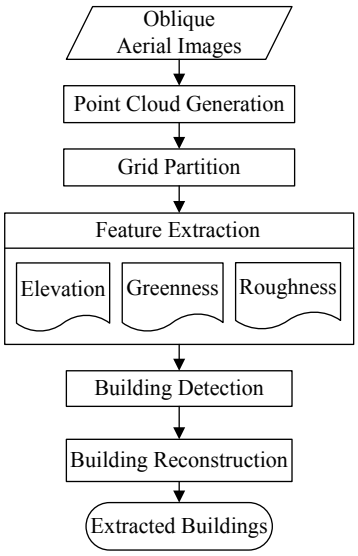


Fig. 1. Flowchart for building extraction.

Then, we convert the building extraction task into a 2D image segmentation problem. Specifically, the 3D point cloud is projected to the 2D XY plane first, then the plane is partitioned by a regularly spaced grid, as shown in Fig. 2. After that, the grid is considered as an image, and each grid cell is considered as an "image pixel"; the pixel value in the image is calculated from all the 3D points located in the corresponding grid cell or interpolated by its neighbors for empty cell. It should be noted that all our following steps in this paper are focus on this virtual image.

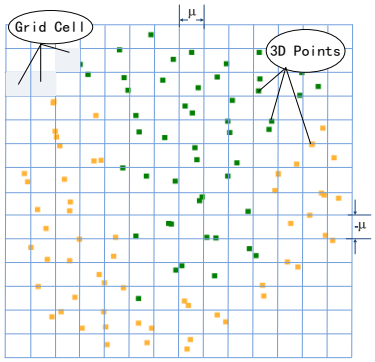


Fig. 2. Grid partition. 3D points are partitioned by a regularly spaced grid in XY plane.

Cell size μ is an important parameter in this step which directly related to the final extraction accuracy. Theoretically, the smaller of this value is, the higher of extraction accuracy is. However, considering the fact that too small value will lead to a dramatically increasing in computation cost and empty cell number, which are also harmful to the extraction. In practice, we set this value according to the average point number in each grid cell, and we found 2~3 is a good choice for the average point number in all our cases.

B. Feature Extraction

Three features are computed for each cell c_i of the grid generated in last subsection:

1) The elevation feature f_e measures the height of the cell c_i relative to the ground (buildings, trees or poles in urban scene usually have a high response to this feature) :

$$f_e(c_i) = z_{c_i} - z_{g_i}, \tag{1}$$

where z_{g_i} denotes the ground height of the cell c_i and we computed it through the algorithm proposed in [17] and z_{c_i} is defined as a function of the 3D points located in the cell, for a certain point $p(x, y, z)$:

$$z_{c_i} = \max_{p \in c_i}(z_p). \tag{2}$$

2) The greenness feature f_g indicates the average ratio of the green spectrum relative to the red and blue spectrum around the cell c_i (vegetation usually have a high response to this feature, buildings and trees can be separated by this):

$$f_g(c_i) = \frac{1}{N_{c_i}} \sum_{P \in \Omega_{c_i}} \frac{G_p}{R_p + G_p + B_p}, \tag{3}$$

where Ω_{c_i} is the neighborhood of c_i, N_{c_i} is the number of points belonging to Ω_{c_i}, and (R_p, G_p, B_p) is the spectral intensities of the point p.

3) The roughness feature f_r denotes the roughness of the cell c_i and it is expressed by the normal vectors of 3D points (trees or other clutters in urban scene usually have a high response to this feature; ground, buildings or other man-made objects usually have a low response for their local planarity):

$$f_r(c_i) = \frac{1}{N^*_{c_i}} \sum_{\{p,q\} \in \Omega_{c_i}} \arccos \frac{\mathbf{n}_p \cdot \mathbf{n}_q}{|\mathbf{n}_p| \cdot |\mathbf{n}_q|}, \tag{4}$$

where $\mathbf{n}_p$ and $\mathbf{n}_q$ are the normal vectors of points p and q respectively, $N^*_{c_i}$ is the number of normal pairs.

It should be noted that, similar to Lafarge et al. [10], all the three type features are normalized to $[0,1]$ before they are inputted to the next build detection step.

C. Build Detection

From the features computed per cell, we can model the building detection problem as a binary classification one. More specifically, for integrating the contextual and spatial consistency prior to the classification, a Markov Random Field (MRF) based combinatorial optimization model is constructed. The quality of a label configuration l is measured by energy E:

$$E(l) = \sum_{i \in G} D_i(l_i) + \lambda \sum_{\{i,j\} \in N} V_{i,j}(l_i, l_j), \quad (5)$$

where D_i and $V_{i,j}$ denote the data term and smoothness term respectively, balanced by parameter λ. G denotes the whole grid which containing all the cells, and N denotes all the adjacent cell pairs.

Based on the prior knowledge of the urban scene and the features described above, we can define a building as a tall object with a low greenness and a low roughness. Furthermore, we found that one of the most distinct attribute with respect to the MVS point cloud investigated in this paper is there are lots of points existing in building facades which significantly different with the aerial LIDAR point cloud used by the majority of other existing approaches. According to this observation, the building façade can be recognized by the accumulations (the number, N_i, of points falling to each cell), of the projected 3D points. Then, the data term can be defined as:

$$D_i(l_i) = \begin{cases} \min(1 - f_e, f_g, f_r) & l_i = 1, N_i < \rho \\ 1 - \min(1 - f_e, f_g, f_r) & l_i = 0, N_i < \rho \\ 0 & l_i = 1, N_i \geq \rho \\ \eta & l_i = 0, N_i \geq \rho \end{cases}, \quad (6)$$

where ρ is a thread for distinguish the building façade from others. In other word, if a cell contains more than ρ points, it is more likely to be a building cell with the label $l_i = 1$, and we give a penalty η to label **0** and 0 to label **1**; otherwise we penalize the label configuration according to the features we extracted above.

The pairwise smoothness term $V_{i,j}$ between two adjacent cells c_i and c_j is expressed by a piecewise smooth and discontinuity preserving model:

$$V_{i,j} = K \cdot (1 - e^{-\cos\langle \mathbf{f}_i, \mathbf{f}_j \rangle}), \quad (7)$$

where K is some constant, $\mathbf{f}_i$ and $\mathbf{f}_j$ are feature vectors for cells c_i and c_j respectively.

Finally, a graph cut algorithm [18] is used to solve this energy minimization problem and cells with the label $l_i = 1$ are considered as the detected buildings.

D. Building Reconstruction

After the building detection process, we can get a binary image with buildings as foreground. To remove some small objects and smooth the classification result, we perform a morphological open and a morphological close operation to this binary image successively. Then, the connected component analysis is used to mark cells to different building entities. Note that we assign each building entity, consisting of connected cells classified as building and 3D points inside them, a unique ID.

For each building entity, we first extract its boundary by a contour finding algorithm proposed by Suzuki et al. [19], then the boundary is simplified by the Douglas-Peucker algorithm [20]. At the same time, we gather the heights information of building roof and building footprint from the 'z' coordinate of 3D points belonging to this building entity. Specifically, the maximum and minimum are assigned to building roof and building footprint respectively. According to these information, the building entity can be reconstructed on a very generic level, comparable to the "block" model, defined as LOD1 (Level of Detail 1) in City-GML [21]. Finally, all building entities are processed through the same workflow and prismatic 3D building models are reconstructed automatically.

III. EXPERIMENTAL RESULTS

A. Testing Datasets

We test our method on two oblique aerial image datasets, and we named them "area 1" and "area 2" for short in this section. They are all captured by a penta-view oblique digital photogrammetric equipment (TOPDC-5, Fig. 3). The nadir camera head is equipped with a 47mm lens and all the other 4 camera heads are equipped with 80 mm lenses mounted with tilt angles of 45°. The image resolution of nadir view and oblique view are 9334x6000 (56 megapixel) and 7312x5474 (40 megapixel) respectively. In each image dataset, there are 50 images with the ground sample distance (GSD) around 10 cm.

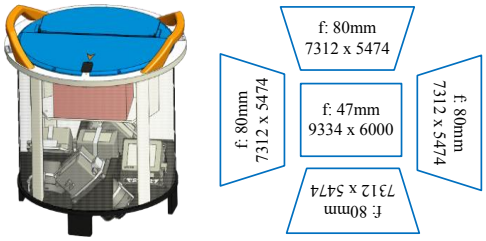
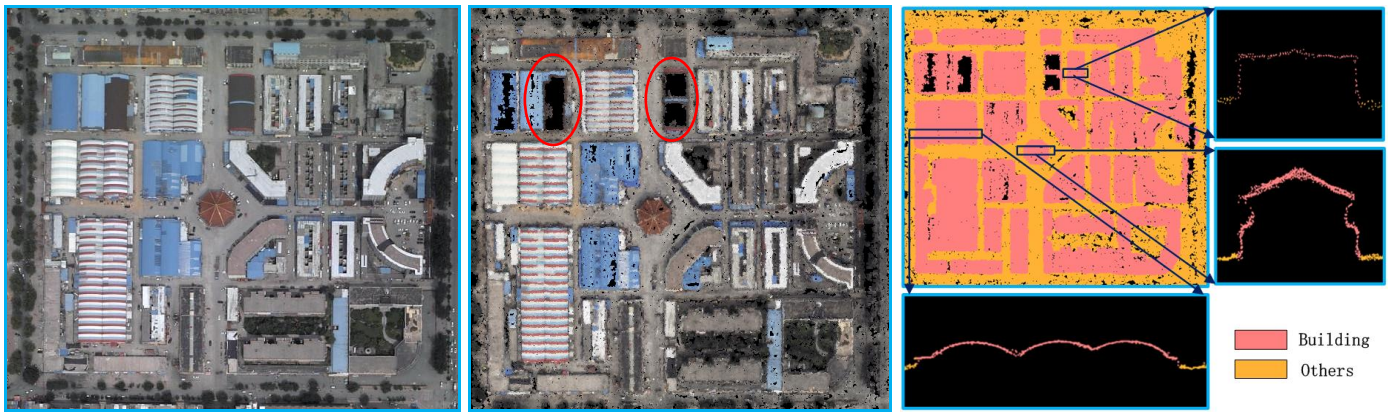


Fig. 3. TOPDC-5 penta-view oblique digital photogrammetric equipment.

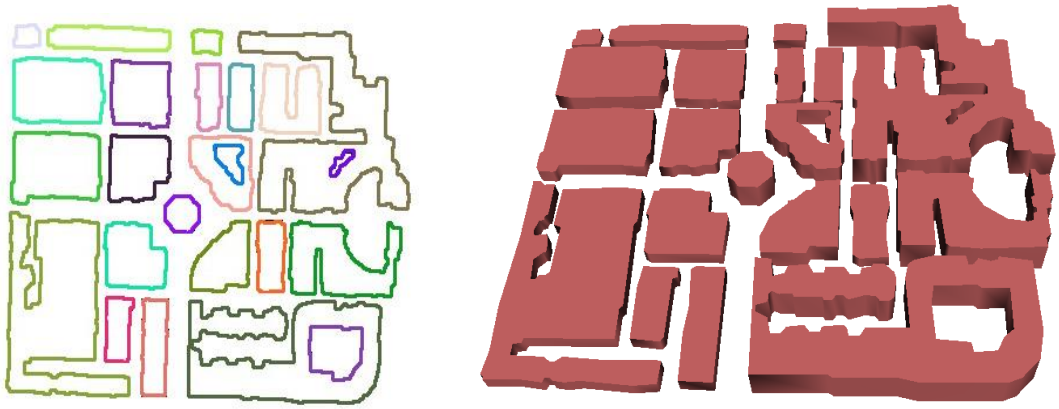
B. Experiment Result

The generated dense point cloud in area 1 is shown in Fig.4 (b). Comparing it with the original aerial image in Fig.4 (a), we can see although there exist some obvious holes (inside the red ellipses) due to the existence of texture-poor regions, we did successfully obtain a high-quality, dense and gross-error-free point cloud. In Fig.4 (c), the 3D points belonging to buildings are separated from others. The top view for the whole area and some locally enlarged side views are both shown.



(a) (b) (c)

Fig. 4. (a) One of the original oblique aerial images in area 1. (b) Generated dense point cloud by SfM+MVS. (c) Detected 3D building points (pink) in area 1, the top view for the whole area and some locally enlarged side views are both shown.

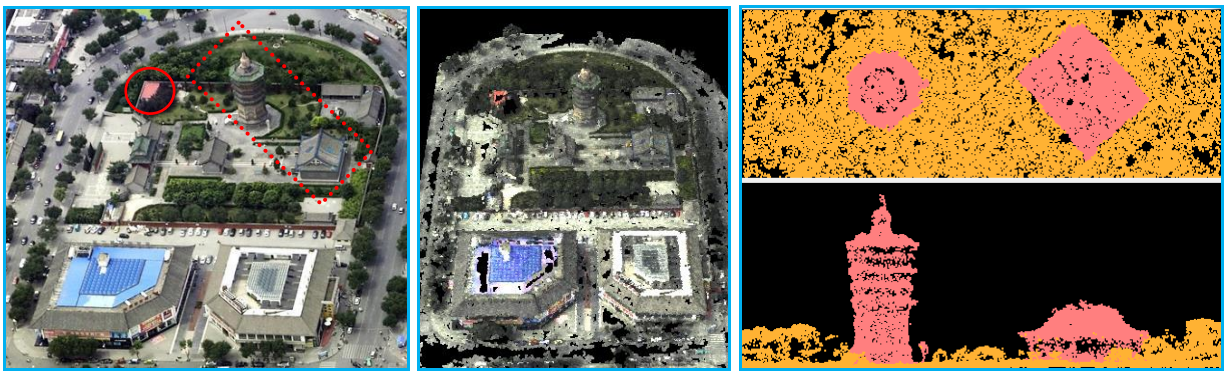


(a) (b)

Fig. 5. (a) Boundaries for detected buildings in area 1, different building entities are rendered in different colors. (b) Extracted 3D building models in area 1, all buildings are expressed in prismatic style.

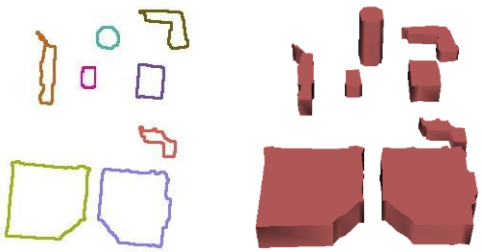
The boundaries for detected buildings in area 1 are shown in Fig.5 (a), different building entity with different color. According to these boundaries, along with the extracted 3D building points in Fig.4 (c), 3D prismatic building models are reconstructed one by one, as shown in Fig.5 (b).

Based on the same procedure and the same parameters, we perform our method on area 2 (a small scenic spot). In Fig.6, one of the typical oblique aerial image, the generated point cloud with the same view, and the locally enlarged top view and side view for detected 3D building points (pink) inside the red rectangle in Fig.6(a) are all shown. In addition, the boundaries for detected buildings and the reconstructed 3D prismatic building models of this area are illustrated in Fig.7(a) and Fig.7(b) respectively.



(a) (b) (c)

Fig. 6. (a) One of the original oblique aerial images in area 2. (b) Generated dense point cloud by SfM+MVS. (c) Top view and side view for detected 3D building points (pink) inside the red rectangle in (a).



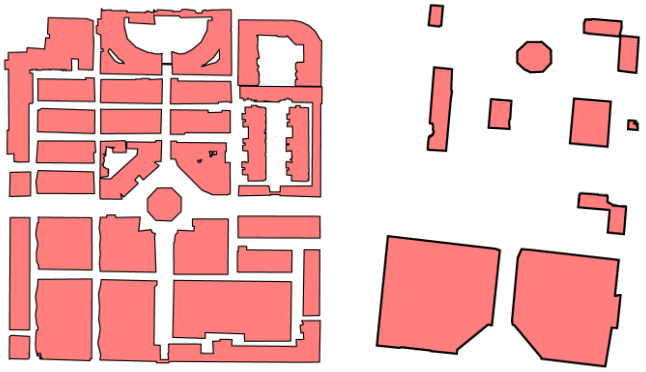
(a) (b)

Fig. 7. (a) Boundaries for detected buildings in area 2, different building entities are rendered in different colors. (b) Extracted 3D building models.

C. *Accuracy Assessment*

The experiment results illustrated in Fig.4-Fig.7 indicate that our proposed automatic building extraction scheme is reasonable and desirable. In addition to this qualitative analysis resort to human inspection, a quantitative analysis also performed by us.

Since there are not existing ground truth data in our test areas. For assessing the extraction accuracy, we first manually label the buildings in our test areas in the original oblique aerial images, then the extracted 3D building points are projected back to the labeled images to verify whether the extraction is correct. Considering the actual requirement for the evaluation and the labor of manually labeling many high resolution images, five images (three for area 1 and two for area 2) were labeled finally. For convenience, we named them "ground truth images" and two of them are shown in Fig.8.



(a) (b)

Fig. 8. (a) One of the manually labeled ground truth images in area 1. (b) One of the manually labeled ground truth images in area 2.

In the evaluation process, for each labeled ground truth image, all the visible 3D points are projected back to it by the aid of the camera projection matrix calculated by the SfM algorithm and the visible information outputted by the MVS algorithm. Then each point's label is compared to the ground truth image and one of the following result is assigned to this point:

- TP (True Positive): the point is correctly recognized as building.
- TN (True Negative): the point is correctly recognized as other objects (not building).
- FN (False Negative): the point is incorrectly recognized as other objects.
- FP (False Positive): the point is incorrectly recognized as building.

Deriving from the above quantities, three metrics are adopted to assess the extraction accuracy:

- RR (Recall Rate): the probability that a building point is recognized, defined as TP/(TP+FN).
- R (Reliability): the probability that a point recognized as building is actually a building, defined as TP/(TP+FP).
- OA (Overall Accuracy): the overall accuracy of the extraction, defined as (TP+ TN)/(TP+FP+TN+FN).

By marking the incorrectly recognized points in the ground truth images, four error maps for building extraction are rendered, FP in green and FN in blue, in Fig.9.

(a) (b)

(c) (d)

Fig. 9. Error maps for building extraction. False positive samples are rendered in green; false negative samples are rendered in blue. (a) and (c) are both for area 1 (different views); (b) and (d) are both for area 2.

From the experiment results above, we could see that most of buildings in our test areas are extracted successfully, including the buildings with matching holes in Fig.4(b), except for some extremely small ones surrounded by clutters, e.g. the one inside the red ellipse in Fig.6(a). This demonstrates the effectiveness of the proposed method. Error maps in Fig.9 also reveal that the majority of extraction errors are located around the building borders. This is because: one the one hand, the MVS algorithm cannot reconstruct sufficient 3D points near building borders; on the other hand, building extraction is an inherently uncertain problem in some extent, and the border of

a building is sometimes hard to be defined accurately in a cluttered urban scene even by human.

The three metrics (RR, R and OA) of our results measured on five manually labeled ground truth images are shown in TABLE I. We could see that all the metrics in different ground truth images and different areas are close and consistent. The overall accuracy in area 1 is 95.0%-95. 6%, and 96.2%-97.2 for area 2, which is comparable with the data-fusion-based method reported in [12]. However, because of the lacking of accurate ground truth data for high resolution aerial image based building extraction and source code of other methods, we cannot find a suitable public dataset and perform a full comparison with other state-of-the-art methods currently.

TABLE I. ACCURACY ASSESSMENT RESULTS

Ground truth image ID		**Metrics (%)**		
		Recall Rate (RR)	***Reliability (R)***	***Overall Accuracy (OA)***
Area 1	1	97.4	95.4	95.0
	2	98.5	95.2	95.6
	3	98.8	93.0	95.2
Area 2	1	98.5	89.7	97.2
	2	97.2	91.3	96.2

IV. CONCLUSIONS

In this paper we propose an automatic urban building extraction method for oblique aerial images. Five steps are included in this method: point cloud generation, grid partition, feature extraction, building detection and building reconstruction. Taking advantages of recent progress in SfM and MVS, dense point cloud is generated first. Then we project the point cloud into a regularly spaced grid in XY plan, the building extraction problem is converted to an image segmentation one. By combining the strength of the geometric attribute and spectral (color) attribute, three features are extracted and a MRF based graph model along with an engine function is created. Points belonging to buildings are recognized by minimizing the function, and prismatic 3D building models are reconstructed accordingly.

Although we attained a high extraction accuracy in this work, there did exist some extraction errors, especially near building borders and some small buildings surrounded by clutters. In addition, the reconstructed 3D building models are relative simple, and we would consider more complex and accurate building models in the future. Another issue is the testing of our method on a larger area, and comparing it with the other state-of-the-art methods systematically if possible.

ACKNOWLEDGMENT

We thank the company of TopRS in China for providing us with the valuable oblique aerial images. This work was supported in part by the National High Technology R&D Program of China (863 Program) under Grant 2015AA124102, and in part by the Natural Science Foundation of China under Grants 61333015, 61473292 and 61273280.

REFERENCES

[1] M. Fradkin, H. Maitre, and M. Roux, "Building detection from multiple aerial images in dense urban areas," Computer Vision and Image Understanding, vol. 82, pp. 181-207, Jun 2001.

[2] H. Mayer, "Automatic object extraction from aerial imagery - A survey focusing on buildings," Computer Vision and Image Understanding, vol. 74, pp. 138-149, May 1999.

[3] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer, "A Survey of Urban Reconstruction," Computer Graphics Forum, vol. 32, pp. 146-177, Sep 2013.

[4] F. Lafarge, "Some new research directions to explore in urban reconstruction," 2015 Joint Urban Remote Sensing Event, 2015.

[5] Y. Zhang, "Optimisation of building detection in satellite images by combining multispectral classification and texture filtering," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54, pp. 50-60, Feb 1999.

[6] A. Ferro, D. Brunner, and L. Bruzzone, "Automatic Detection and Reconstruction of Building Radar Footprints From Single VHR SAR Images," IEEE Transactions on Geoscience and Remote Sensing, vol. 51, pp. 935-952, Feb 2013.

[7] J. Xiao, M. Gerke, and G. Vosselman, "Building extraction from oblique airborne imagery based on robust façade detection," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 68, pp. 56-68, 2012.

[8] Y. Verdie, F. Lafarge, and P. Alliez, "LOD Generation for Urban Scenes," Acm Transactions on Graphics, vol. 34, Apr 2015.

[9] C. Poullis, "A framework for automatic modeling from point cloud data," IEEE Transactions on Pattern Anal Mach Intell, vol. 35, pp. 2563-75, Nov 2013.

[10] F. Lafarge and C. Mallet, "Creating Large-Scale City Models from 3D-Point Clouds: A Robust Approach with Hybrid Representation," International Journal of Computer Vision, vol. 99, pp. 69-85, 2012.

[11] J. Zhang and X. Lin, "Advances in fusion of optical imagery and LiDAR point cloud applied to photogrammetry and remote sensing," International Journal of Image and Data Fusion, pp. 1-31, 2016.

[12] K. Khoshelham, C. Nardinocchi, E. Frontoni, A. Mancini, and P. Zingaretti, "Performance evaluation of automated approaches to building detection in multi-source aerial data," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 65, pp. 123-133, 2010.

[13] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[14] J. Y. Rau, J. P. Jhan, and Y. C. Hsu, "Analysis of Oblique Aerial Images for Land Cover and Point Cloud Classification in an Urban Environment," IEEE Transactions on Geoscience and Remote Sensing, vol. 53, pp. 1304-1319, Mar 2015.

[15] H. N. Cui, S. H. Shen, W. Gao, and Z. Y. Hu, "Efficient Large-Scale Structure From Motion by Fusing Auxiliary Imaging Information," IEEE Transactions on Image Processing, vol. 24, pp. 3561-73, Nov. 2015.

[16] S. H. Shen, "Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes," IEEE transactions on image processing, vol. 22, pp. 1901-1914, 2013.

[17] K. Q. Zhang, S. C. Chen, D. Whitman, M. L. Shyu, J. H. Yan, and C. C. Zhang, "A progressive morphological filter for removing nonground measurements from airborne LIDAR data," IEEE Transactions on Geoscience and Remote Sensing, vol. 41, pp. 872-882, Apr 2003.

[18] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 1222-1239, Nov 2001.

[19] S. Suzuki and K. Abe, "Topological structural-analysis of digitized binary images by border following," Computer Vision Graphics and Image Processing, vol. 30, pp. 32-46, 1985.

[20] D. H. Douglas and 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, vol. 10, pp. 112-122, 1973.

[21] G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele, "OGC City Geography Markup Language (CityGML) En-coding Standard," 2012.