Vis Comput (2016) 32:601–609
DOI 10.1007/s00371-015-1076-0

ORIGINAL ARTICLE

Automatic segmentation of point clouds from multi-view reconstruction using graph-cut

Rongjiang Pan[1,3] · Gabriel Taubin[2]

Published online: 1 April 2015

Abstract In multi-view reconstruction systems, the recovered point cloud often consists of numerous unwanted background points. We propose a graph-cut based method for automatically segmenting point clouds from multi-view reconstruction. Based on the observation that the object of interest is likely to be central to the intended multi-view images, our method requires no user interaction except two roughly estimated parameters of objects covering in the central area of images. The proposed segmentation process is carried out in two steps: first, we build a weighted graph whose nodes represent points and edges that connect each point to its k-nearest neighbors. The potentials of each point being object and background are estimated according to distances between its projections in images and the corresponding image centers. The pairwise potentials between each point and its neighbors are computed according to their positions, colors and normals. Graph-cut optimization is then used to find the initial binary segmentation of object and background points. Second, to refine the initial segmentation, Gaussian mixture models (GMMs) are created from the color and density features of points in object and background classes, respectively. The potentials of each point being object and background are re-calculated based on the learned GMMs. The graph is updated and the segmentation of point clouds is improved by graph-cut optimization. The second step is iterated until convergence. Our method requires no manual labeling points and employs available information of point clouds from multi-view systems. We test the approach on real-world data generated by multi-view reconstruction systems.

Keywords Graph-cut · Point clouds · Segmentation · Multi-view reconstruction · Fixation constraints

1 Introduction

Three-dimensional models of real-world objects are widely used in archaeology, cultural heritage management, animation, game and other fields. In comparison with laser scanners and structured light scanners, image-based reconstruction of 3D objects is considered the most flexible and low-cost way. Image-based 3D modeling has the advantage over other 3D modeling techniques in terms of equipment availability, affordability, scalability and amount of user requirements. One of the most popular approaches to image-based reconstruction is the combination of structure from motion (SfM) and multi-view stereo (MVS), which produces 3D reconstructions from a set of overlapping images. SFM recovers camera poses and MVS produces dense 3D point clouds. Several commercial [1] and freely available [2–4] software appeared in the past few years. A point cloud with positions, colors and normal vectors is usually recovered from these systems.

However, besides the points of interested object, outliers often exist in the point cloud, as shown in Fig. 1b. Thus, the segmentation of point clouds into object and background is a necessary operation in the 3D reconstruction workflow, which is usually done manually before reconstruc-

✉ Rongjiang Pan
panrj@sdu.edu.cn

Gabriel Taubin
taubin@brown.edu

[1] School of Computer Science and Technology, Shandong University, Jinan, China

[2] School of Engineering, Brown University, Providence, RI, USA

[3] Engineering Research Center of Digital Media Technology, Ministry of Education of PRC, Jinan, China

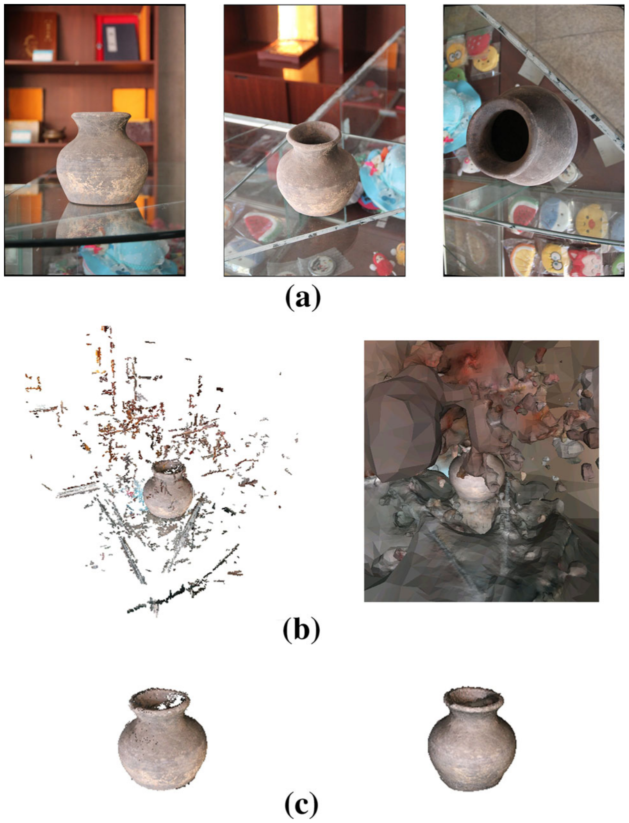


Fig. 1 Segmentation results of a pot model. **a** Three of 39 images used in multi-view reconstruction. **b** The generated 3D point cloud (*left*) and reconstructed mesh (*right*). **c** Segmented 3D point cloud (*left*) and reconstructed mesh (*right*)

tion of a mesh model. Moreover, some applications want an unattended photo-to-3d workflow. For instance, after excavators in an archaeological site upload photos into database, REVEAL system [5] uses the photos to automatically generate accurate 3D models of the artifacts, excavation and site. Figure 1 gives an example of a pot model generated from 39 images using Bundler [3] and PMVS [4]. Without segmentation, the pot mesh from surface reconstruction [6] is connected numerous unwanted outliers. Since the real-world data is usually large and object points are highly entangled with background points, automatically segmenting 3D point clouds is a challenging task without information supplied by the user.

In this paper, we propose a graph-based method for segmenting point clouds generated by a multi-view reconstruction system. Our method is based on the observation that the object of interest is much likely to be central to the intended multi-view images since the object of interest is always being focused upon by the camera and within the central area of the viewport. We begin by a graph-cut optimization to segment the obvious object points leveraging the fixation constraint. Gaussian mixture models (GMMs) are then created and learned from the color and density information of points in object and background classes, respectively. The graph is updated and the segmentation of point clouds is improved by graph-cut. The improvement of GMMs and graph-cut optimization is iterated until convergence. The method does not require the object be fully contained in each view. We show its effectiveness on several real-world data generated from multi-view systems.

The main contributions of this work are:

- We use the fixation constraint to initiate the process of point clouds segmentation. Two intuitive parameters describing the coverage of object in the images are used as a priori assumption about the object. The method requires no manual labeling points and runs automatically without user interaction.
- The color and density features are utilized in the Gaussian mixture models of object and background classes. Since background points are usually much sparsely distributed than object points in the output of multi-view reconstruction system, those background points which have the similar colors to the object points are correctly classified according to the density features.
- We allow for the label changing of all points during iterative graph-cut optimization. This provides label recovery from poor initial hypotheses without hard constraints.

2 Related work

Point cloud segmentation is an ongoing research field in computer vision and computer graphics. Given a noisy point cloud representing a scene, the goal is to separate the points that belong to the object of interest from the background points.

Most previous works focus on point clouds generated by laser or structured light 3D scanners. As the graph-cut based global optimization is widely used in image segmentation [7,8], this framework is also extended to 3D point clouds. A min-cut based method of segmenting objects in large-scale outdoor point cloud scans is proposed in [9]. In the automatic regime, a given or estimated radial scale is used to specify hard constraints for foreground points. Alternatively, the interactive regime allows the user to iteratively add background and foreground constraints. The edges of the graph have weights that decrease with distance between points. The graph-cut framework is also applied to data gathered with active vision robotic heads [10]. Two methods, geometric plane assumption and image saliency, are proposed for seeding the segmentation. Color information is utilized in the point cloud segmentation problem and multiple objects can be identified. In modeling plants from multi-view images, interactive graph-based optimization is implemented to segment the 3D data points and 2D images of individual leaves [11]. They defined a combined distance function for pairs of points using color difference in images and 3D Euclidean distance. Graph-cut based interactive segmentation of 3D point clouds is also discussed in [12], which relied on point labeling input specified by the user and utilized only Euclidean distance between points. It is more challenging to segment point clouds generated from SfM and MVS since there are more background points compared to active scanners which have limited ranges.

Fixation constraint has been used to indicate the object of interest in image, video and multi-view object segmentation (MVOS) [14–16]. Using the fixation point as an object marker, graph-cut is used to find the globally optimal closed contour around the fixation point in the image [14]. In multi-view object segmentation [15], a color model of the object from the image pixels around the camera fixation points is learned. Then, image edges and object color information from calibrated images are combined in a volumetric binary MRF model. The globally optimal segmentation of object is obtained by a graph-cut optimization. In [16], the MVOS problem is solved by a joint graph cuts linking pixels through space and time. Appearance and stereo cues are also combined in an energy minimization framework to solve the MVOS problem in [20]. They propose an automatic approach based on a piecewise planar depth map representation without consistent 3D reconstruction. However, MVOS requires that the object of interest be fully visible in all considered views. In contrast to MVOS whose goal is to segment objects of interest in images and videos, we focus on the segmentation of 3D object points and our method does not require the object be fully contained in each view.

3 Overview

Given a point cloud generated by a multi-view stereo system, each data point has a position **x**, normal vector **n**, RGB color **c** and the images in which it is visible. Our goal is to segment the point cloud into foreground (object) and background. We consider the segmentation problem as a binary labeling problem where each point is labeled as object or background. Our work is based on the graph-cut method described in [7,8]. However, we allow for the label changing of all points during iterative graph-cut optimization and require no user interaction. We create a graph G representing the point cloud. Each point is represented by one node, which is linked to its k-nearest neighbors by edges (the value k is discussed in the results section). In addition, graph Gcontains two terminal nodes s and t, where the source$s$ is the object terminal and the sink t represents the background terminal. An $s - t$ cut of graph G corresponds to a segmentation of the points into object and background. All points remaining connected to s in the $s - t$ cut are considered to be classified as object points and all points remaining connected to t are considered to be classified as background points. The quality of the segmentation is measured by a cost function consisting of two terms, unary potentials and pairwise potentials. Fast and effective min-cut/max-flow algorithms are used to get the global optimum of the cost function [18].

Under the assumptions that the object is located in the central area of all images and its appearance is different from the background, we do not require hard constraints from the user labeling for segmentation. In the proposed method, the fixation constraint is employed to provide an initialization. After obtaining a tentative object and background classification of points by graph-cut optimization, Gaussian mixture models including color and density features of the points are further used to refine the segmentation. The labels of all points can change during iterative graph-cut optimization. This provides label recovery from poor initial hypotheses without hard constraints, especially for the views in which the object is partly contained. An overview of our segmentation algorithm is given in Fig. 2.

Fig. 2 Overview of our algorithm

Initialization

1. Construct a k-nearest neighbors graph on the input points.
2. Compute the point-wise potentials of assigning the point to object and background based on the horizontal and vertical observational parameters describing the coverage of the interested object in the images, and the projected pixel coordinates in the images in which the point is visible.
3. Compute the pairwise potentials between neighboring points according to their distances, normals and colors.
4. Run graph-cut to find a tentative object and background classification of points.

Refinement

1. Create Gaussian mixture models from the color and density features of points in object and background classes, respectively.
2. Update the point-wise potentials according to the likelihoods that the point belongs to the object and background GMMs.
3. Run graph-cut to find an updated object and background classification of points.
4. Repeat Steps 1-3 in Refinement until the classification converges.

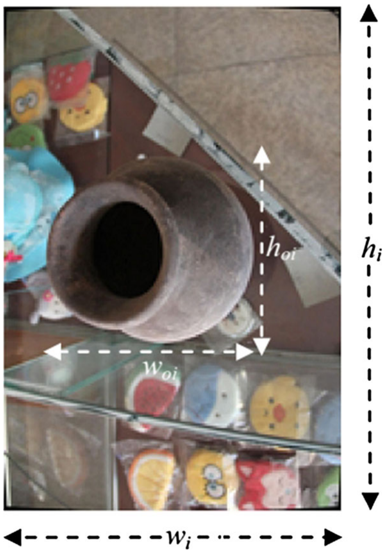


Fig. 3 The observational parameters in our algorithm are satisfied by most images

4 Implementation

4.1 Initial point-wise potentials in initialization

Based on the assumption that the object is located in the central area of all images, the center of the object points can be estimated by finding the approximate point of intersection of the optical axis of the cameras. However, it is difficult to estimate object and background probabilities for a point $p$because the object points are usually not in the center of the cloud point. Instead, our method takes as input two observational parameters r_w and r_h which specify the approximately ratios of the interested object covering along the width w and height h of most images, $r_w \approx \text{average}(w_{oi}/w_i)$ and $r_h \approx \text{average}(h_{oi}/h_i)$, as shown in Fig. 3. We consider a probabilistic model, which deals with uncertainty by taking into account all images where the point is visible. The object and background probabilities for a point p are modeled as a two-dimensional Gaussian function:

$$P_{\text{obj}}(p) = \frac{1}{|I|}\sum_{i\in I}\exp\left(-\frac{\mathrm{d}_{xi}^2}{2\sigma_{xi}^2}-\frac{\mathrm{d}_{yi}^2}{2\sigma_{yi}^2}\right) \tag{1}$$

$$P_{\text{back}}(p) = 1 - P_{\text{obj}}(p) \tag{2}$$

where I denotes the set of images in which point p is visible, d_{xi} and d_{yi} are the horizontal and vertical pixel distances between the projection of point p in image i and the center of the image i, $\sigma_{xi} = 0.5r_w \times w_i$ and $\sigma_{yi} = 0.5r_h \times h_i$ control the fidelity of P_{obj} and P_{back} in image i with width w_i and height h_i.

We express the unary term E_p^{initial} associating a label l_p to point p as follows:

$$E_p^{\text{initial}}(l_p) = \begin{cases} -\log(P_{\text{obj}}(p)), & \text{if } l_p = \text{object} \\ -\log(P_{\text{back}}(p)), & \text{if } l_p = \text{background} \end{cases} \tag{3}$$

Therefore, $E_p^{\text{initial}}(\text{object})$ is low in case p comes from object and $E_p^{\text{initial}}(\text{background})$ is low if p comes from background.

4.2 Pairwise potentials

The pairwise term describes the penalty for placing a segmentation boundary between neighboring points and encourages assigning the same labels to neighboring points that exhibit similar appearance. For a pair of neighboring points p and q, the proposed pairwise potential considering positions, colors and normal vectors of points is expressed by

$$E_{pq}(l_p, l_q) = \begin{cases} \frac{1}{2}\left(\exp\left(\frac{-d(\mathbf{c}_p,\mathbf{c}_q)^2}{2\sigma_c^2}\right)+\exp\left(\frac{-d(\mathbf{x}_p,\mathbf{x}_q)^2}{2\sigma_x^2}\right)\right) & \\ \cdot\max\left(\left\langle \mathbf{n}_p, \mathbf{n}_q\right\rangle, 0\right), & \text{if } l_p \neq l_q \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $d(.,.)$ is the Euclidean distance, σ_c and σ_x indicate the color and position expectations over all neighboring points, $\langle \mathbf{n}_p, \mathbf{n}_q\rangle$ is the dot product of normal vectors. This penalty $E_{pq}(l_p, l_q)$ is high in the regions of similar color and orientation when neighboring points are assigned different labels.

4.3 Point-wise potentials in refinement

After a tentative object and background classification of points using fixation constraint and graph-cut optimization, we create Gaussian mixture models for object and background classes. The RGB color and sampling density of points are used to set up GMMs. The density feature makes the algorithm more robust based on the observation that points in background are usually more sparsely distributed than in object. We define the sampling density $\rho$of a point p as the average distance between p and its k-nearest neighbors N_p:

$$\rho = \frac{1}{|N_p|}\sum_{q\in N_p} d(p, q) \tag{5}$$

where $d(.,.)$ is the Euclidean distance. The sampling density of points is then mapped to the interval [0, 255] to match the RGB color component. Thus, the feature vector for a point is a 4-dimensional vector whose components are RGB value and ρ.

Each GMM, one for the object and one for the foreground, has K Gaussian components ($K = 5$ as suggested in [8]). The parameters for the object and background GMMs are

$\{\pi_{\text{obj}}(k), \mu_{obj}(k), \sum_{\text{obj}}(k)\}$ and $\{\pi_{\text{back}}(k), \mu_{\text{back}}(k), \sum_{\text{back}}(k)\}, k = 1, \ldots, K$, i.e., the mixture weights π, means μ and covariance matrices $\sum$ for the object and background Gaussian distributions. The unary term E_p^{refine} associating a label l_p to point p is now redefined in Eq. (6), where z is the 4-dimensional feature vector of point p, det $\sum$ denotes the determinant of the matrix $\sum$ and $\sum^{-1}$ the inverse of the matrix $\sum$.

E_p(object) is low in case p comes from object and can therefore be severed at low cost. N-edges connect each point to its k-nearest neighbors with weights $E_{pq}(l_p, l_q)$. The weights of N-edges are not changed throughout the execution of our algorithm and can be computed once and reused. In the refinement stage of our algorithm, the graph structure stays the same and enables a substantial speed-up in the iteration.

$$E_p^{\text{refine}}(l_p) = \begin{cases} -\log \sum_{k=1}^{K} \left[\frac{\pi_{\text{obj}}(k)}{(2\pi)^2 \sqrt{\det \sum_{\text{obj}}(k)}} \exp\left(-\frac{1}{2}(z - \mu_{\text{obj}}(k))^T \sum_{\text{obj}}(k)^{-1} (z - \mu_{\text{obj}}(k))\right)\right], & \text{if } l_p = \text{object} \\ -\log \sum_{k=1}^{K} \left[\frac{\pi_{\text{back}}(k)}{(2\pi)^2 \sqrt{\det \sum_{\text{back}}(k)}} \exp\left(-\frac{1}{2}(z - \mu_{\text{back}}(k))^T \sum_{\text{back}}(k)^{-1} (z - \mu_{\text{back}}(k))\right)\right], & \text{if } l_p = \text{background} \end{cases} \tag{6}$$

4.4 Graph-cut optimization

An energy function E is defined so that its minimum should correspond to a good segmentation of the point clouds:

$$E = \sum_p E_p(l_p) + \lambda \sum_{\{p,q\} \in N} E_{pq}(l_p, l_q), \tag{7}$$

where N is the set of all neighboring pairs in the k-nearest neighborhood on the input points, the coefficient $\lambda \geq 0$ controls a relative importance of the two terms. The first term E_p reflects how well the labeling corresponds to the observed data. Equation (3) is used in the initialization stage and Eq. (6) is used in the refinement stage, respectively. The second term $E_{pq}(l_p, l_q)$ measures the consistency between labeling of adjacent points.

This kind of energy function can be represented by Markov random fields (MRFs) and minimized using the standard graph-cut algorithm [7]. In the graph, there are two types of edges between nodes, T-edges and N-edges. T-edges connect each point to the object terminal s and the background terminal t. The weight of the edge connecting point p with s is set to E_p(background) and the weight to t is set to E_p(object).

5 Experimental results

We tested our algorithm on dozens of datasets generated from freely available software. In particular, we employ VisualSFM [2] to estimate camera poses and PMVS/CMVS [4] to recover a point cloud. In our implementation, ANN [17] is used to find the k-nearest neighbors of a point.

The result of segmentation depends on the coefficient λ. For the point cloud given in Fig. 4b, segmentation results for different λ values are shown in Fig. 5. As λ increases, more points are classified into object points and over segmentation is observed. As in image segmentation, the value that gives the best segmentation may differ for different input data. Our experiments show that λ values between 10 and 30 work well for most segmentations.

Then we compare different sizes of the neighborhood for the segmentation. For the point cloud given in Fig. 4b, segmentation results for 8, 12 and 16 nearest neighbors are shown in Fig. 6. We get more accurate segmentation as the size of the neighborhood is increased. In the rest of our experiments, we use 16 nearest neighbors. When only RGB colors are used to set up GMMs, spurious points in the background are incorrectly classified into object, as shown in Fig. 6d.

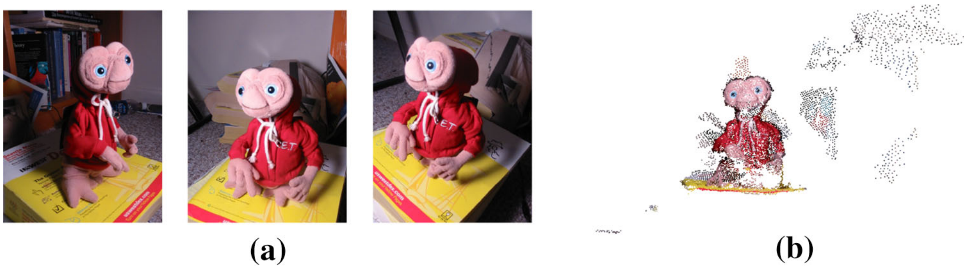


Fig. 4 Three of nine images used in the reconstruction of ET model (**a**) and the generated 3D point clouds (**b**)

Fig. 5 The effect of λ on the segmentation results. **a** $\lambda = 1$, 6022 points. **b** $\lambda = 10$, 9136 points. **c** $\lambda = 30$, 11,396 points. **d** $\lambda = 50$, 11,396 points

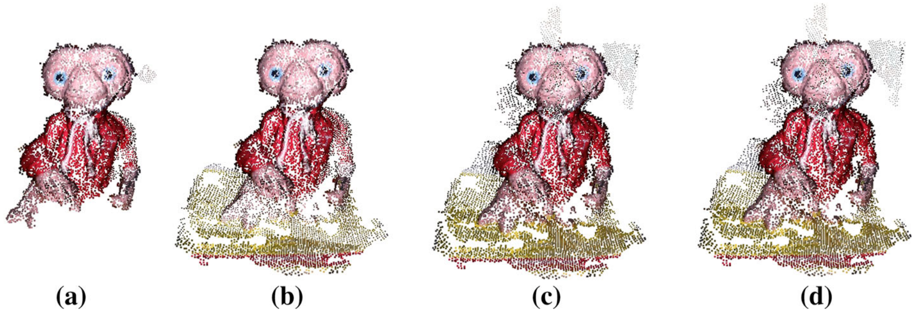
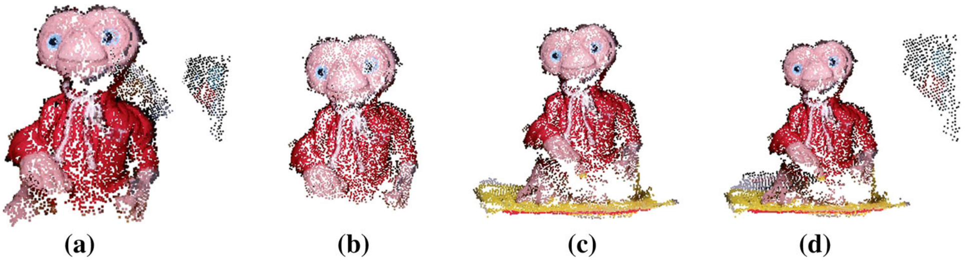


Fig. 6 Segmentation results of ET model with different size *k* of neighborhood. **a** 8-nearest neighbors. **b** 12-nearest neighbors. **c** 16-nearest neighbors. **d** 16-nearest neighbors without density information



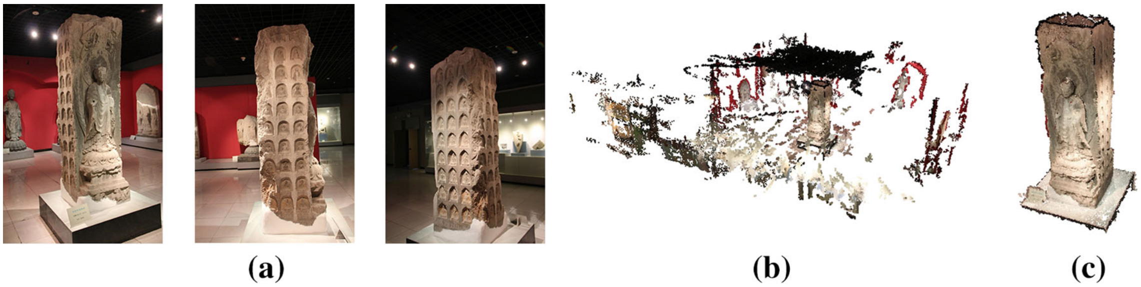


Fig. 7 Three of 80 images used in the reconstruction of Buddha statue (**a**), the generated 3D point clouds (**b**) and segmentation result (**c**)

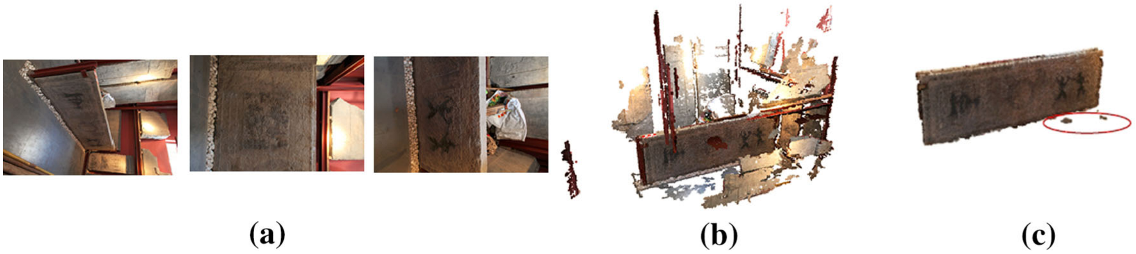


Fig. 8 Three of 79 images used in the reconstruction of a stone relief (**a**), the generated 3D point clouds (**b**) and segmentation result (**c**)

Figures 7, 8 and 9 present segmentation results for more challenging point clouds. The photos are taken in the exhibition halls of a museum. Although there are some similar objects in the background, our method obtains reasonable segmentation results without user interaction.

Moreover, we do not require the object be fully visible in the images used in reconstruction, as shown in Figs. 8 and 9. However, it is still a constraint that the general appearance of the interested object should be different from the backgrounds'. In spite of using point density in the GMMs which can exclude the background points far from the object, those background points that have similar colors and densities with the object points are incorrectly classified, as marked in Figs. 8c and 9c.

We also evaluate our approach on publicly available multi-view dataset [20] (see Figs. 10, 11). The objects of interest are fully visible and in the central area in all considered views. The ground truth segmentation results in the dataset cannot be used as masks in MVS since their resolution are different from the input images. Since the chair in Fig. 11 takes

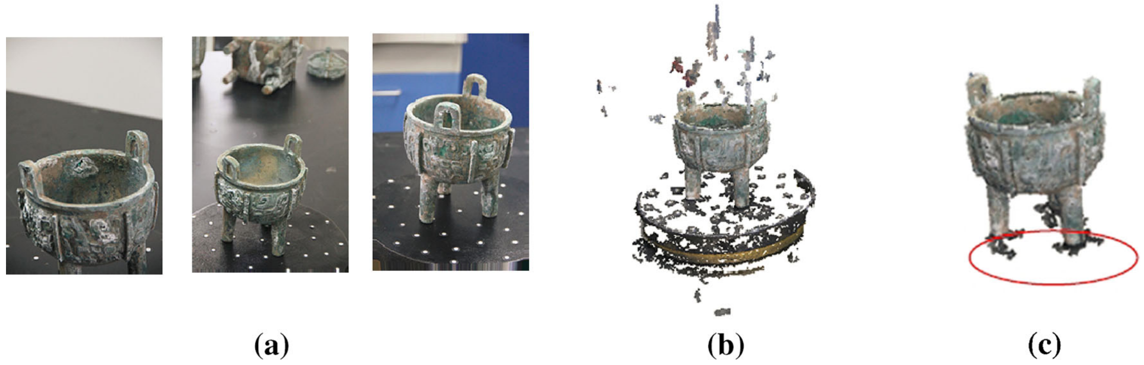


Fig. 9 Three of 90 images used in the reconstruction of a bronze tripod (**a**), the generated 3D point clouds (**b**) and segmentation result (**c**)

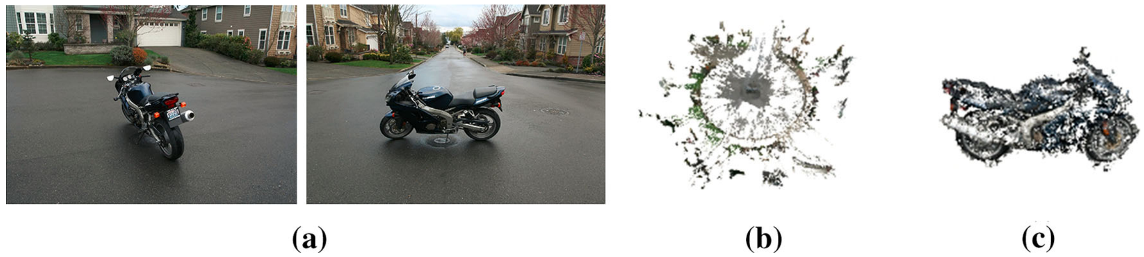


Fig. 10 Two of 35 images used in the reconstruction of a motorbike (**a**), the generated 3D point clouds (**b**) and segmentation result (**c**)

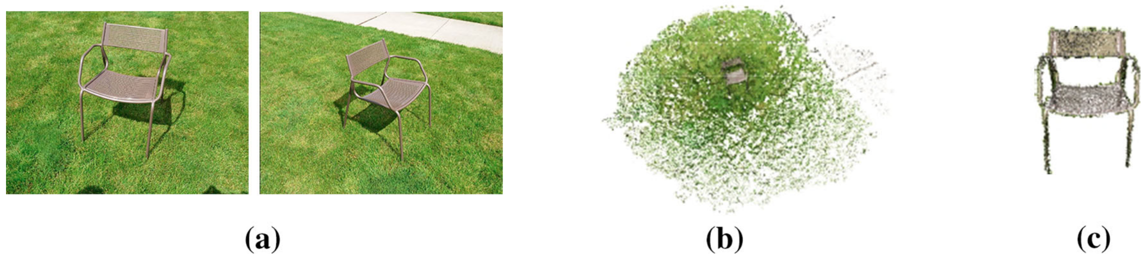


Fig. 11 Two of 41 images used in the reconstruction of a chair (**a**), the generated 3D point clouds (**b**) and segmentation result (**c**)

Table 1 The input parameters and performance of our experiments

Model	r_w	r_h	#points before segmentation	#points after segmentation	Precision (%)	Recall (%)	Runtime (s)
ET	0.6	0.7	12,095	9,139	98.3	100.0	0.42
Bronze tripod	0.5	0.4	78,378	61,119	96.9	100.0	1.85
Chair	0.4	0.5	125,892	13,812	99.2	99.8	2.83
Motorbike	0.3	0.3	139,011	21,467	99.5	98.1	4.45
Buddha statue	0.3	0.8	179, 030	107,484	99.6	100.0	4.87
Stone relief	0.5	0.7	4,433,413	3,443,978	99.3	100.0	262.71
Average					98.8	99.6	

consistent appearance, we set the number of its Gaussian components to two.

Table 1 shows the two parameters r_w and r_h used in the experiments, which are roughly estimated based on the coverage of the interested object along the width and height of the images. In fact, our method is not very sensitive on the parameters and the results remain almost the same when the parameters vary between ±10 % in our experiments. Large values of r_w and r_h lead to more object points.

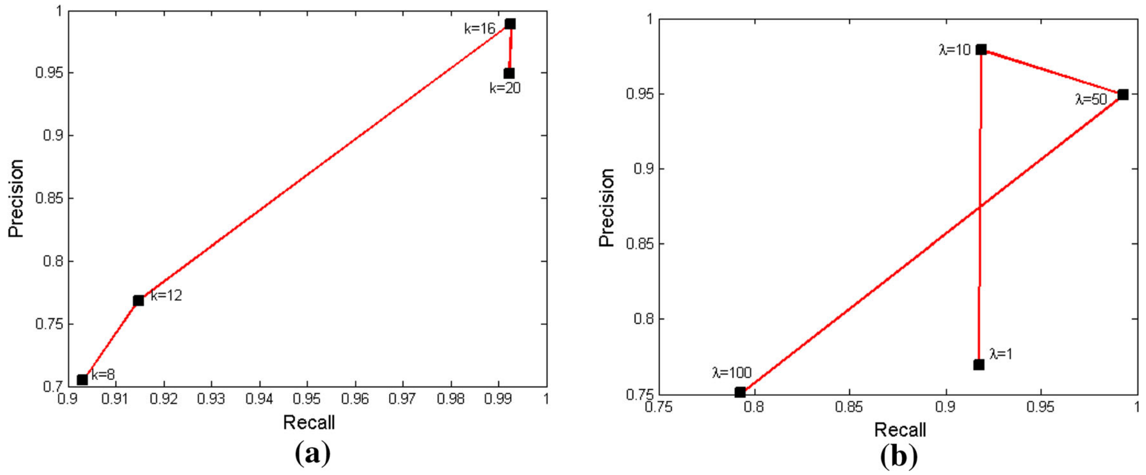


Fig. 12 Precision–recall plots of our algorithm for different values of neighborhood size k ($\lambda = 10$) (**a**) and λ($k = 16$) (**b**)

To evaluate the performance of the algorithm, we make use of the ground truth segmentation obtained by interactive editing with the open-source 3D mesh processing software MeshLab [19]. We use precision and recall measures as in [9]. The precision measure is defined as the ratio of the number of correctly predicted object points to the total number of predicted object points. The recall measure is defined as the ratio of the number of correctly predicted object points to the number of ground truth object points. Precision is high when most of the predicted object points are in the object and low when there is significant over-segmentation. A high recall indicates that most of the object points have been classified into the object. Low recall is the result of under-segmentation. For a perfect segmentation, precision and recall will be 1. Figure 12 shows precision–recall curves of our algorithm for different values of neighborhood size k and λ.

The time reports are obtained on a machine with two 8-core Intel Xeon E5-2560 CPUs, 2.00 GHz and 64 GB of memory and no GPU-based acceleration technique is exploited.

6 Conclusion

We introduce a method for the segmentation of point clouds from multi-view system based on graph-cut. By using the fixation constraint to initialize the segmentation, we do not require hard constraints from the user labeling. Color and density features of points are utilized in the Gaussian mixture models of object and background classes, which improves the robustness of the method. The method has been demonstrated on several challenging real-world datasets.

Acknowledgments This work was supported by State Scholarship Fund of China, Program of Science and Technology Development of Shandong Province (2014GGX101016), the Fundamental Research Funds of Shandong University (2014JC003).

References

1. Agisoft LCC. PhotoScan. http://www.agisoft.ru/products/photoscan
2. Wu, C.: VisualSFM: a visual structure from motion system. http://ccwu.me/vsfm/
3. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring image collections in 3D. ACM Trans. Graph. (Proceedings of SIGGRAPH 2006) **25**(3), 835–846 (2006)
4. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. IEEE Trans. Pattern Anal. Mach. Intell **32**(8), 1362–1376 (2010)
5. Gay, E., Galor, K., Cooper, D.B., Willis, A., Kimia, B.B., Karumuri, S., Taubin, G., Doutre, W., Sanders, D., Liu, S. : REVEAL Intermediate report. In: Proceedings of CVPR Workshop on Applications of Computer Vision in Archaeology (ACVA'10), June 2010
6. Calakli, F., Taubin, G.: SSD-CSR: Smooth signed distance colored surface reconstruction. In: Dill, J., Earnshaw, R., Kasik, D., Vince, J., Wong, P.-C. (eds.) State-of-the-Art Volume on Computer Graphics, Visualization, Visual Analytics, VR and HCI Dedicated to the memory of Jim Thomas (2012)
7. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. ICCV **1**, 105–112 (2001)
8. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309–314 (2004)
9. Golovinskiy, A., Funkhouser, T.: Min-Cut based segmentation of point clouds. IEEE Workshop on Search in 3D and Video (S3DV) at ICCV September 2009 (2009)
10. Johnson-Roberson, M., Bohg, J., Björkman, M., Kragic, D.: Attention-based active 3D point cloud segmentation. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1165–1170, 18–22 Oct. 2010 (2010)
11. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B.: Image-based plant modeling. ACM Trans. Graph. **25**(3), 599–604 (2006)
12. Sedlacek, D., Zara, J.: Graph cut based point-cloud segmentation for polygonal reconstruction, Adv. Visual Comput. pp. 218–227 (2009)

13. Campbell, N.D.F., Vogiatzis, G., Hernandez, C., Cipolla, R.: Automatic 3d object segmentation in multiple views using volumetric graph-cuts. Image Vis. Comput. **28**(1), 14–25 (2010)
14. Mishra, A.K., Aloimonos, Y., Cheong, L.F., Kassim, A.A.: Active visual segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 639–653 (2012)
15. Campbell, N., Vogiatzis, G., Hernandez, C., Cipolla, R.: Automatic object segmentation from calibrated images. Visual Media Production (CVMP) (2011)
16. Djelouah, A., Franco, J.S., Boyer, E., Clerc, F.L., Pérez, P.: Multi-view object segmentation in space and time. ICCV 2013-IEEE International Conference on Computer Vision, Syndey, Australia. pp. 2640–2647, Dec 2013 (2013)
17. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM **45**(6), 891–923 (1998)
18. 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)
19. Cignoni, P., Corsini, M., Ranzuglia, G. : MeshLab: an openSource 3D mesh processing system, ERCIM News, No.73, pp. 45–46, http://meshlab.sourceforge.net, Apr, 2008 (2008)
20. Kowdle, A., Sinha, S.N., Szeliski, R.: Multiple View Object Cosegmentation using Appearance and Stereo Cues, in Proceedings of the 12th European Conference on Computer Vision (ECCV), Springer, Berlin 8 October 2012 (2012)

Rongjiang Pan is a professor in School of Computer Science and Technology, Shandong University, China, and a research scientist in Engineering Research Center of Digital Media Technology, Ministry of Education of PRC. His research interests include computer vision and 3D geometricmodeling. He received aBSc in computer science, aMsc in computer science, and a PhD in computer science from Shandong University, China in 1996, 2001, and 2005, respectively. He was a visiting scholar at the University of West Bohemia in Plzen between 2006 and 2007 and a visiting professor at Brown University between 2014 and 2015. He is a member of the ACM.

Gabriel Taubin is a professor in School of Engineering, Brown University. He earned a Licenciado en Ciencias Matemáticas degree from the University of Buenos Aires, Argentina, and a PhD degree in Electrical Engineering from Brown University. He served as Editor-in-Chief of the IEEE Computer Graphics and Applications Magazine from 2010 to 2013, serves as a member of the Editorial Board of the Geometric Models journal, and has served as associate editor of the IEEE Transactions of Visualization and Computer Graphics. He was named IEEE Fellow for his contributions to the development of three-dimensional geometry compression technology and multimedia standards. His main research interests fall within the following disciplines: Applied Computational Geometry, Computer Graphics, Geometric Modeling, 3D Photography, and Computer Vision. Currently, he is more interested in dynamic shape representation, shape capture, and processing.