3D Foreground Point Segmentation from Background using Centroid-based Min-Cut Method

Cho Cherry Aung
University of Computer Studies, Yangon (UCSY)
Yangon, Myanmar
chocherryaung@ucsy.edu.mm

Thin Lai Lai Thein
University of Computer Studies, Yangon (UCSY)
Yangon, Myanmar
tllthein@ucsy.edu.mm

***Abstract*—Point Cloud generation and surface reconstruction of these point clouds are the two main and common steps for a complete 3D model. This system proposed 3D point cloud segmentation as a joint method that can be used between these two steps to get the best and accurate model. Firstly, the proposed system builds the graph in which each point is used as nodes and these nodes are connected by k-nearest neighbor edges. Based on this graph, then, find the location of the object over the complex and noisy points clouds in which the foreground and background points are closely related. This system is based on the centroid point of the cloud, so it does not need user interaction to predict where is the object location. Based on these predicated object point, the object is segmented by our proposed system, centroid-based Min-Cut Segmentation method. The system experiments on various data set such as large-scale scene and real-world data, in which the points are generated by Structure from Motion (SfM). This proposed strategy provides efficient and gives substantial reductions in time and can be used as the input for the surface reconstruction very well.**

Keywords—Point Cloud Segmentation, Energy Minimization, Min-Cut Segmentation, K-Nearest Neighbor, Graph Algorithms

I. INTRODUCTION

Image segmentation is very popular and mature in computer vision and is still a classic problem for several decades. Nowadays, the new technologies emerged more and more, and our world turns two-dimensional into the three-dimensional world. 3D modeling of an object is still an active research topic and have many problems and they are required in many application areas – archaeology, cultural heritage management, virtual tourism, and museum. Before point cloud can be used in any of these application areas, the data needs to be pre-processed to get accurate shape and smoothness of the objects, and segmentation of point clouds is an important one to get these golds. Hence, many researchers focus on 3D point cloud segmentation that extends and based on image segmentation. There are many ways to get the 3D point clouds from either three-dimensional scanner such as LIDAR (Light Detection and Ranging) and Microsoft Kinect or 3D point clouds that are generated from images which are captured from different viewpoints. For input data, image-based reconstruction is considered the most flexible and low-cost way rather than laser scanners and structured light scanners.

The basic requirement steps for 3D model are Structure from Motion (SfM) that gives both the extrinsic and intrinsic camera parameters and sparse point clouds, and Multiview Stereo (MVS) reconstructs dense 3D information. Surface reconstruction and surface texturing can be used after SfM and MVS to get the texture model with accurate results. Many

(a)

(b)

Fig. 1. (a) (b) Input Point Cloud that is generated by Structure-from-Motion (SfM) (Outliers are described within red circles)

projects cover about SfM which first extracts the feature points and find the feature matching points among the corresponding images. The relative camera poses (rotation and translation) are extracted from the fundamental matrix and the feature matches are triangulated into 3D points. Bundler [9], OpenMVG [6] and VisualSfM [10] are the illustration of SfM and PMVS [11] can be used for multi-view stereo.

In the original point cloud, there are often existed many outliers beside the point cloud of the interested object, as in Fig. 1(a) and (b). These outliers and unnecessary background points may cause devastation to create the mesh model of the interested object. Our system takes as input as point clouds that are generated from Structure-from-Motion (SfM) and intends to remove the outliers and noisy background points which are closely related to the foreground objects.

Some reports have already been presented concerning the research related work of 3D point cloud segmentation. K. Liu and J. Boehm [3] proposed an interactive method based on graph cuts for point cloud segmentation. This paper doesn't provide automatically segment the point cloud and needs user interaction to draw two stokes for indicating the target object and the background respectively. It used MaxFlow-MinCut Algorithm and simple, but which can be potentially applied for general point clouds. A. Golovinskiy, V.G. Kim, and T. Funkhouser [2] describe the location of point clouds and segmentation of this point clouds with a graph-cut algorithm before labeling objects for the urban city. Nearest neighbors' graph is built first and then search the connection points within the radius value, so the results heavily depend on this some radial scale. Their paper able to recognize 65% of the objects and it needs more efficiency. R. Pan and G. Taubin [1] proposed a graph-based method for segmentation point clouds generated by a multi-view reconstruction system. Their paper no need user interaction for points labeling but this approach tests the data generated by PMVS and assumes objects are in the center of the point clouds. The primary difference between this previous related paper and our approach is that we segment the point clouds that are output from Structure-from-Motion (SfM), rather than mesh model that gives as a result by Multi-View Stereo (MVS).

Our system takes as input a direct point clouds procedure from SfM and creates as output a separation foreground object from the background by using centroid-base Min-Cut Segmentation. Min-cut performs binary segmentation: it divides the cloud into two clusters, one with points that do not belong to the object we are interested in (background points) and another with points that are considered part of the object (foreground points) [9]. There are many advantages about segmentation is done directly on the point clouds rather than the mesh model. The foreground objects are extracted from the background areas after surface reconstruction, it will take more times because of the unnecessary outliers and noise point. For example, let total points of an original cloud have 160, 000 points, the object has 60, 000 points and the remaining are unnecessary background points, it may take a long time to construct these all points into the mesh model if the system did not do any segmentation. If the system will do the segmentation, it needs to reconstruct the 3D model with 60, 000 points. Our proposed system will save the processing time and give more accurate results to compare to other methods [1, 2, 3] because it worked directly on point clouds and removed the outliers and reduce unnecessary points in advance before surface reconstruction.

In the next section, we describe the overview of the proposed system for segmentation of point clouds into foreground and background. Then, we discuss how to predict the location of the object automatically by our proposed method are described in section III. In section IV, we outline the segmentation of the point cloud by Min-cut Segmentation. Finally, we describe the experiments and results of the proposed system as in section V. After all, the paper is concluded in the last section.

II. OVERVIEW OF THE PROPOSED SYSTEM

The step by step processing of the proposed system are described as follows:

Step 1: Acquired input point clouds from Structure-from-Motion (SfM).

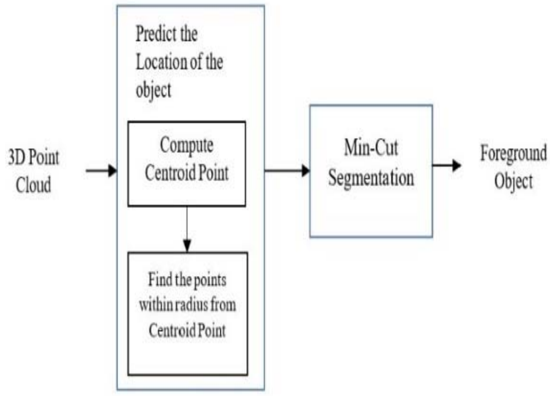

Fig. 2. Overview of the Proposed System

Step 2: Build the graph G = (V, E), whose vertex V represents each point in the cloud and the edges E are connected by K-Nearest Neighbors (KNN).

Step 3: Predict the location of objects using the proposed searching method.

Step 4: Segment the foreground and background by Min-cut Method.

Step 5: Repeat graph-cut optimization until the classification converges.

The points in the cloud that are generated by SfM are used as the input as the first step and overview are presented in Fig. 2. Detail discussion about the remaining steps will be described in the next sections.

III. POINT CLOUD SEGMENTATION BY GRAPH CUTS

Point cloud segmentation is an ongoing research field in computer vision and computer graphics and it has many issues until now. The problem of segmentation is that it needs to consider the foreground has a point that is not only closely related to the background but also how many points has in proximity to the background. Moreover, the point cloud data are usually noisy, sparse, and unorganized.

To resolve these issues, a simple graph-based min-cut method is used in our proposed system to segment the point cloud whether it is foreground or background. As the graph-cut based optimization is widely used in image segmentation [7, 8], this framework is also extended to 3D point clouds. Graph-based methods can segment complex scenes in point cloud data include noise or uneven density with better results compare to other methods. Our system proposed an automatic segmentation method based on graph cuts, it is an optimization process which computes the minimum cut for a weighted graph, to partition point clouds. The next section will firstly introduce the background theory about graphs in detail and build the graph for our proposed system.

A. Graphs

By the background theory, a weighted graph G = (V, E) consists of a set of edges E that connect between the nodes as shown in Fig. 3(a). The special nodes that are called terminals (the source s and the sink t) are added in this graph. Normally, there are two types of edges in the graph: n-links and t-links

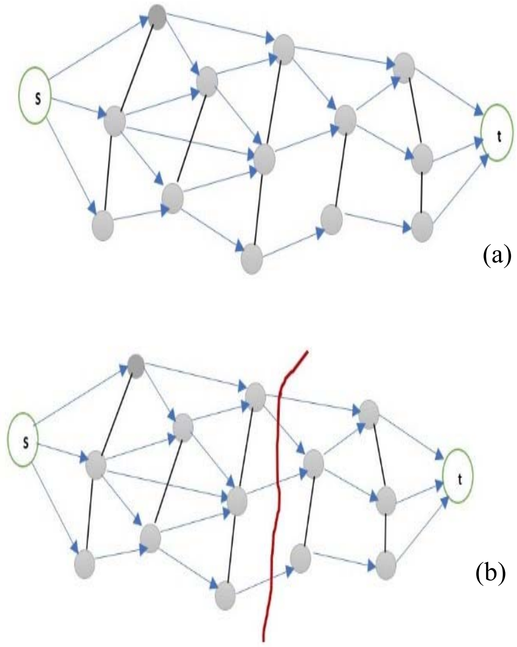

Fig. 3. Example of a Graph (a) A Graph G (b) A Cut on G

which are colored as the black and the blue lines respectively in the figure. N-links connect pairs of neighboring points (p and q) and T-links connect points with terminals (source s and sink t). Then, the graph with two terminals aims to partition the nodes in the graph into two disjoint subsets S and T such that the source s in S and the sink t is in T as illustrated in Fig 3(b).

In this proposed system, we create the graph G = (V, E) will contain n+2 vertices: V = {s, t, v_1, …, v_n} where s and t are the source node and sink node respectively, and v_1, …, v_n represents each 3D point in the cloud and it will encode the binary variable L_p (i.e. means label for point p whether it is object or background and we will discuss in the next section). The edges E are connected between these vertices {v_1, …, v_n}and not on these nodes – source s and sink t in case. All points remaining connected to s are 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 potential function (data energy) and pairwise potential function (smoothness energy). Fast and effective min-cut/ max-flow algorithms are used to get the global optimum of the cost function [4].

B. Prediction of Location of Object on the Point Cloud

In our proposed system, we predict the location of the interested object directly on the point cloud, not onto the mesh model and images. The proposed system assumed that the object is in the central area of all images and most of the object would be located the center over the 3D point clouds. In [1], it takes as input the width and height of the object (w_{oi}, h_{oi}) in images and the width and height of most images (w_i, h_i). Then, calculate the two observational parameters $r_w \approx$ average (w_{oi} / w_i) and $r_h \approx$ average (h_{oi} / h_i) on the input image not on point clouds. But, in our proposed method, the object within the point clouds can be estimated by computing the centroid of a cloud. The centroid is the “center of mass” and it is a point with coordinates that result from computing the mean of the values of all points in the cloud and they are calculated as in (1).

$$P_c = \frac{1}{N} \sum_{i=1}^{N} P_i \qquad (1)$$

Where P_c is the centroid of the 3D points, P_i represents each 3D point in the cloud and N is the total number of scattered 3D points.

After getting the centroid point P_c, the neighbors’ points within a radius (R) are defined by using KdTree [5]. The object and background probabilities for a point p are expressed by as in (2).

$$P_{obj}(p) = \begin{cases} 1 & \text{, if } P_i \text{ is within Radius(R)} \\ 0 & \text{, if } P_i \text{is outside Radius(R)} \end{cases} \qquad (2)$$

where $P_{obj}(p)$ is the probability of point *p* is the foreground object. The black dot represents 3D points and the red points within these radius values are assumed for the object as shown in Fig. 4.

C. Unary Potential Function (Data Energy)

Once the seed points have been predicted, it is necessary to compute for a point p with respect to object or background. The data term cost $D_p(L_p)$ for point p taking label L_p is expressed as follows:

$$D_p(L_p) = \begin{cases} P_{obj}(p) & \text{if } L_p\text{=object} \\ P_{bg}(p) & \text{if otherwise} \end{cases} \qquad (3)$$

D. Pairwise Potential Function (Smoothness Energy)

The smooth term cost sometimes called pairwise term $S_{p,q}$ (L_p, L_q) describes the penalty for placing a segmentation boundary between neighboring points and making sure assigning the same labels that have the similar appearance to their neighboring points. The smooth cost for a pair of neighboring point p and q is calculated by the formula as follows:

$$S_{p,q}(L_p, L_q) = e^{-\left(\frac{d(p,\ q)}{\sigma}\right)^2} \qquad (4)$$

where $d\ (p, q)$ is the Euclidean distance between point p and q, and σ indicates the position expectations over all neighboring points.

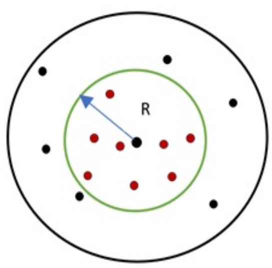

Fig. 4. Prediction the Location of the Object. The Red Points within the radius (R) are assumed the Object

IV. REPRESENTING ENERGY FUNCTION WITH GRAPH-CUT

In the final stage, each point cloud is assigned as the object by minimizing the energy function with smoothness constraint.

$$E = E_{data} + E_{smooth} \tag{5}$$

Where E is the energy function for good segmentation of the point clouds, E_{data} is for data energy and E_{smooth} is for smoothness energy.

$$E_{data} = \sum_{p \in P} D_p(L_p) \tag{6}$$

E_{data} is the sum of a set of labeling correspondences and equation (3) can be used for this data term cost $D_p(L_p)$.

$$E_{smooth} = \sum_{(p,\, q) \in N} S_{p,\, q}\left(L_p\, ,\, L_q\right) \tag{7}$$

E_{smooth} measures the consistency between the labeling of adjacent points and N is the set of all neighboring pairs and it can be calculated by using (4). In this stage, the weight of the edge connecting point p with s is set to the background and the weight t is set to object.

V. EXPERIMENTS AND RESULTS

A few experiments have been done with Core i5 2.50 GHz processor and 8.00 RAM for our proposed system and they are written in C++ and using PCL libraries [12, 14] and tested on the various data. We chose three for the illustration of this proposed system such as pictures in Fig. 6(a) have been used as primary images, the first one is based on Der Hass (79 images) data provided by Multiview Environment official website [13], the second is Hanau model with complex background which contains 114 images and third sculpture contains 18 images. These input images are transformed into 3D points in advance by using open Multiview Geometry (openMVG) [6] and the point clouds are stored in the form of polygon data format (.ply) as in Fig. 6(b) and the background points are colored in red and object points are expressed in white. The Der Hass data, Hanau and sculpture datasets have 36056, 68042 and 177828 points respectively.

Sometimes, the background scene such as walls, human, trees, cars and so on may highly cause trouble to model the 3D objects that are occurred especially in cultural heritage application. Now, our proposed system generated the extraction of the foreground object from complex background scenes with efficiency, timeless and gives the result more accurately. Automatic segmentation results are shown in Fig. 6(c).

In particular, the two data such as Der Hass and Hanau, in which the number of points is 70K, are reconstructed in less than 1 minutes and the 200K Sculpture data is computed within 2 minutes. The time required for our proposed segmentation system is expressed in the table (1) and the comparison results about the processing time before and after segmentation are described in Fig. 5. To prove more evident that the performance of our proposed system, we used two-point clouds that is generated by SfM. One is point-cloud segmented by our proposed method and other is the original

TABLE I. SEGMENTATION TIME (SECONDS)

3D Point Cloud Dataset	Number of Images	Points Before Segmentation	Points After Segmentation	Time (s)
Der Hass Model	79	36056	20396	14.6
Hanau Model	114	68042	21685	57. 913
Sculpture Model	18	177828	173510	116. 07

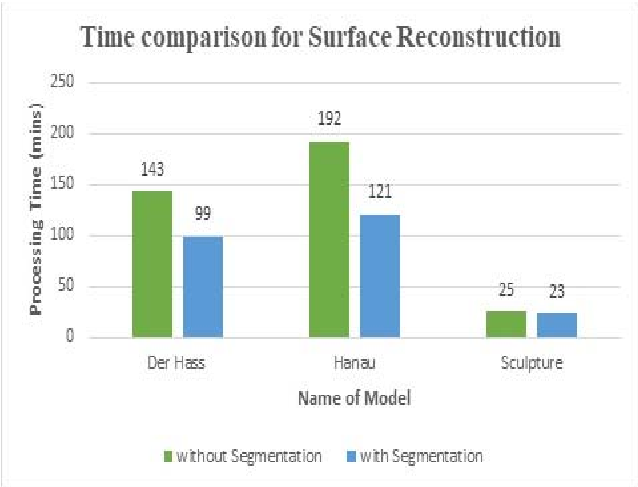

Fig. 5. Surface Reconstruction Time without Segmentation and with Segmentation

point cloud without segmentation, these two kinds of point clouds can be used for surface reconstruction in order to compare the time performance. Surface reconstruction is done by using our proposed segmentation is faster and saves time consuming rather than without segmentation as can see in Fig. 5. The below image Fig. 7 shows accuracy results for Der Hass Model (row1), Hanau Model (row2) and Sculpture Model(row3) by our proposed centroid-based Min-cut method. Each row has an object with the original point cloud, followed by segmentation with [15] and then our proposed centroid-based Min-cut Segmentation. Our centroid-based Min-cut Segmentation is more robust to clutter points of the cloud and gives more accurate by comparison with segmentation results from [15]. These experiments show that our proposed system is used point clouds instead of the mesh model, that not only give the accurate 3D mesh model but also yield efficient time because of the noise and outliers over sparse clouds are already removed by our proposed centroid-based Min-Cut Segmentation.

VI. CONCLUSION

One major gold of our system is to find the location of the objects and perform the segmentation of objects from the background over the point clouds before surface reconstruction. We believe that the proposed system will give the accurate results in this case, where is it the objects, and the segment with a high degree of accuracy within a minimum time cost. This proposed system will reduce the

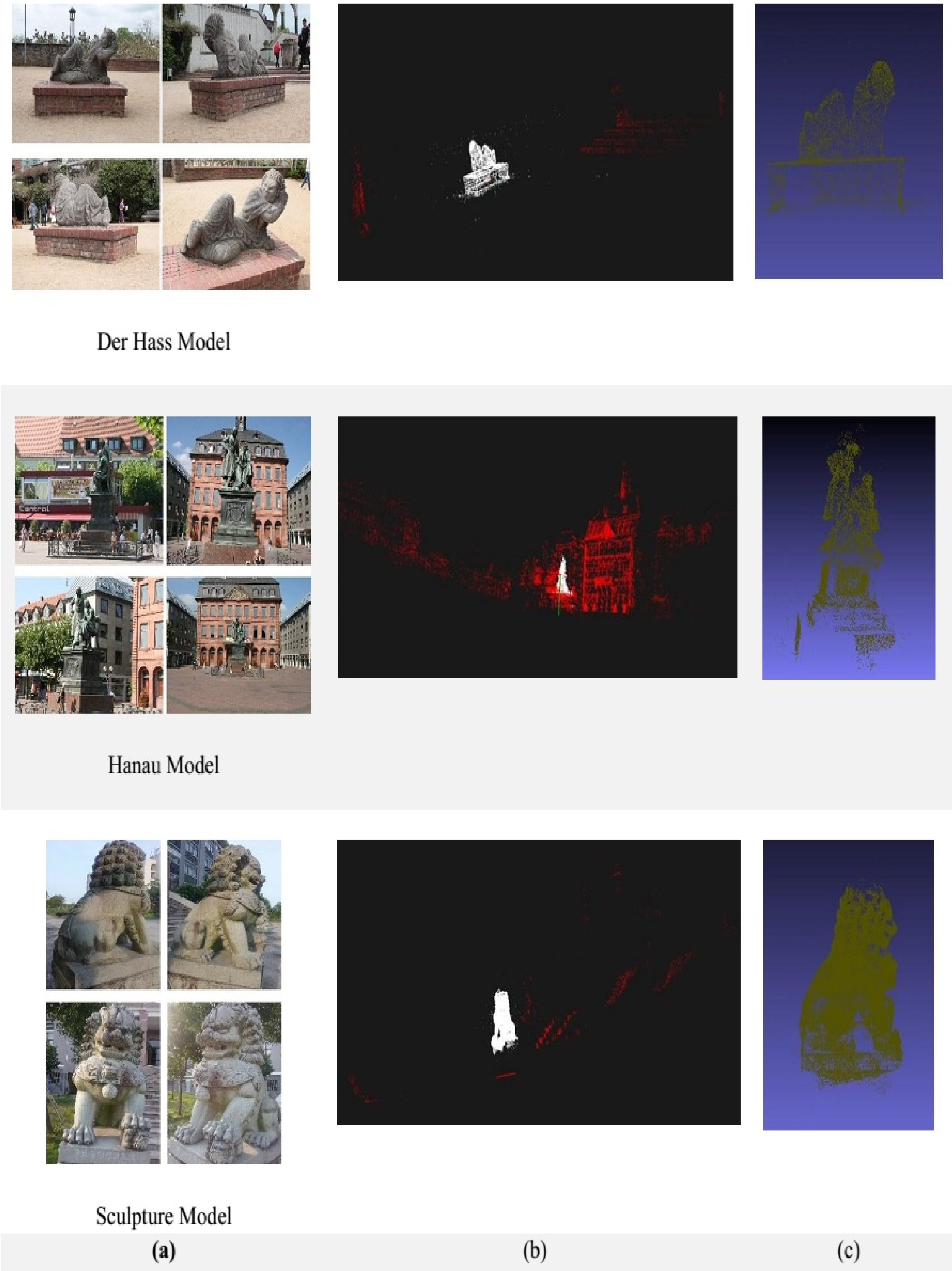

Fig. 6. 3D Point Cloud of Various Dataset (a) Images (b) Input Point Clouds (Before Segmentation, the Backgrounds are expressed in Red) (c) After Segmentation by Proposed Centroid-based Min-Cut Segmentation

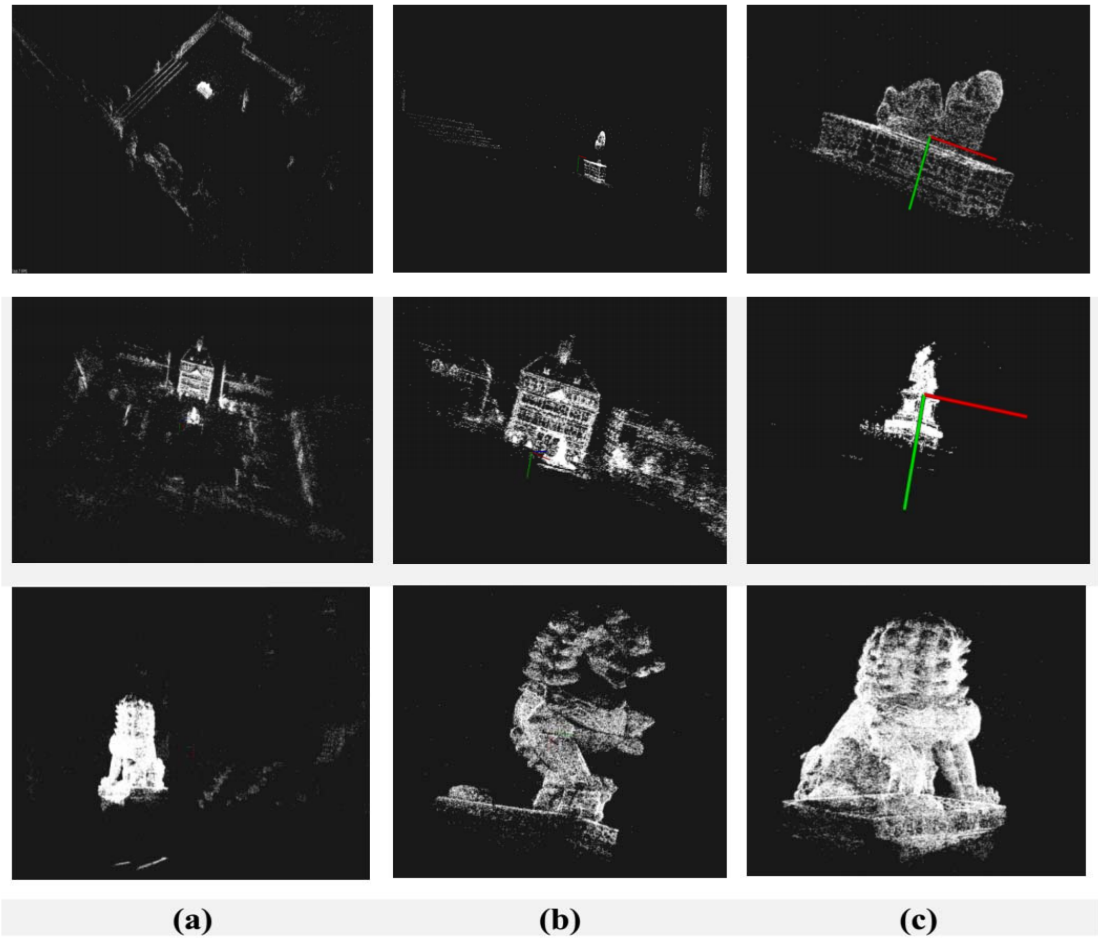

Fig. 7. Comparison Results for Accuracy between [15] and Our Proposed Centroid-based Min-Cut Segmentation Method

wrong object points are considered to reconstruct the surface or mesh model and unqualified results. Moreover, it not only saves running times by energy minimization function but also performs automatically and no need user interaction. Moreover, it provides accurate outputs. In the future work, we should take to complete the level of details with a high degree of automation.

REFERENCES

[1] R. Pan, G. Taubin, Automatic Segmentation of Point Clouds from Multi-view Reconstruction using Graph-cut, Journal of Visual Computer, Vol. 32, Iss. 5, pp. 601-609, 2016

[2] A. Golovinskiy, V.G. Kim, and T. Funkhouser, Shape-based Recognition of 3D Point Clouds in Urban Environments, ICCV, September, 2009.

[3] K. Liu, J. Boehm, A New Framework for Interactive Segmentation of Point Clouds, ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XL-5, Iss. 1, pp. 357-362, 2014.

[4] Y. Boykow, V. Kolmogorov, An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, In IEEE Transactions on PAMI, Vol. 26, No. 9, pp. 1124-1137, Sept, 2004

[5] M. Cornor, P. Kumar, Fast Construction of k-nearest Neighbor Graphs for Point Clouds, IEEE Transactions on Visualization and Computer Graphics, Vol. 16, Iss. 4, pp. 599-608, 2010.

[6] P. Moulon, P. Monasse, and R. Marlet, Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion, ICCV, pp. 3248-3255, 2013.

[7] Y. Boykow, M. P. Jolly, Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images, In Proc. IEEE on ICCV 1,pp. 105-112, July, 2001.

[8] C. Rother, V. Kologorov, A. Blake, GrabCut – Interactive Foreground Extraction using Integrated Graph Cuts, ACM Trans. Graph. 23(3), pp. 309-314, 2004.

[9] N. Snavely, S. M. Seitz, R. Szeliski, Photo Tourism: Exploring Photo Collections in 3D, Transactions on Graphics 25 (3), pp. 835-846, 2006.

[10] C. Wu, Towards Linear-Time Incremental Structure from Motion, in: International Conference on 3D Vision (3DV), 2013, pp. 127-134.

[11] Y. Furukawa, J. Ponce, Accurate, Dense, and Robust Multi-View Stereopsis, Transactions on Pattern Analysis and Machine Intelligence (PAMI) 32 (8), pp. 1362-1376, 2010.

[12] PCL/OpenNIHomepage,http://robotica.unileon.es/index.php/PCL/OpenNI_tutorial_3:_Cloud_processing_(advanced).

[13] MVE Homepage,https://www.gcc.tu-darmstadt.de/home/proj/mve/.

[14] PCL Homepage, http://www.pointclouds.org/.

[15] PCL Hompage, http://pointclouds.org/documentation/tutorials/min_cut_segmentation.php#min-cut-segmentation