Image-based Object Modeling by Fitting Salient Lines and Geometric Primitives

Meng-Hong Cho
Inst. of Multimedia Engineering,
National Chiao Tung University
Hsinchu City, Taiwan
combo1966@gmail.com

I-Chen Lin
Inst. of Multimedia Engineering and
Dept. of Computer Science,
National Chiao Tung University
Hsinchu City, Taiwan
ichenlin@cs.nctu.edu.tw

ABSTRACT

With modern vision techniques and depth sensing devices, it becomes possible for common users to acquire the shape of an object from a set of color or depth images from different views. However, the estimated 3D volume or point clouds, disturbed by noise and errors, cannot directly be applied for graphics usage. This paper presents a two-stage method for reconstructing 3D graphics models from point clouds and photographs. Unlike related work that immediately fitted primitives for the point clouds, we propose finding the primary planes through salient lines in images in advance, and extracting auxiliary planes according to the symmetric properties. Then, a RANSAC method is used to fit primitives for the residual points. Intuitive editing tools are also provided for rapid model refinement. The experiments demonstrate that the proposed automatic stages can generate more accurate results. Besides, the user intervention time is less than that by a well known modeling tool.

Keywords

Image-based modeling, primitive fitting, line features.

1 INTRODUCTION

Reconstructing 3D models is an essential and important topic in the computer graphics and vision fields. The reconstructed models can be used for various applications, from object analysis, interior and urban planning to visual special effects, and so forth. With the rapid advances in depth estimation techniques, users can now purchase consumer-level depth cameras, such as Microsoft Kinect, ASUS Xtion Pro. Software tools, such as ARC 3D, 123D Catch, and my3Dscanner, can help users obtain their three-dimensional models by taking a set of photographs. These tools mainly employ structure-from-motion technologies to compute camera poses (extrinsic parameters) and 3D point clouds of a target object. However, these point clouds usually contain holes and defects resulted from noise, view occlusion, and inaccurate point correspondences. They require additional processing or considerable manual adjustment before applied to graphics usage, e.g. model editing, graphics rendering, and 3D printing.

Primitive fitting is a popularly used approach for modeling from point clouds. It decomposes point clouds into multiple parts and finds the most likely primitive fit for each part. The modern primitive fitting performs satisfactorily for accurate laser-scanned point clouds. Nevertheless, when we applied primitive fitting to point clouds generated by photogrammetric methods, the larger noise makes the fitting process of higher degrees of ambiguity, resulting in unstable and noise-sensitive results. From another aspect, Debevec et al. [Deb96a] involved more user intervention and proposed a pioneering architectural modeling method. They required users to compose a target scene by rough geometric models, and to draw corresponding lines from images. Their system then adjusted the geometric models by aligning the lines of primitives with lines assigned by users.

Our work is inspired by the aforementioned approaches, but we would like to lessen user intervention and lower the chances of ambiguity in fitting. We consider that the lines and planes are valuable cues in modeling of man-made objects, and propose a novel two-stage modeling method. Given a set of photographs and corresponding 3D point clouds, the proposed system first extracts salient lines from photographs, and then finds planes which are associated with these lines and conform to our objective rules. While taking these planes as foundation, the ambiguity situations are alleviated. Partial primitive fittings

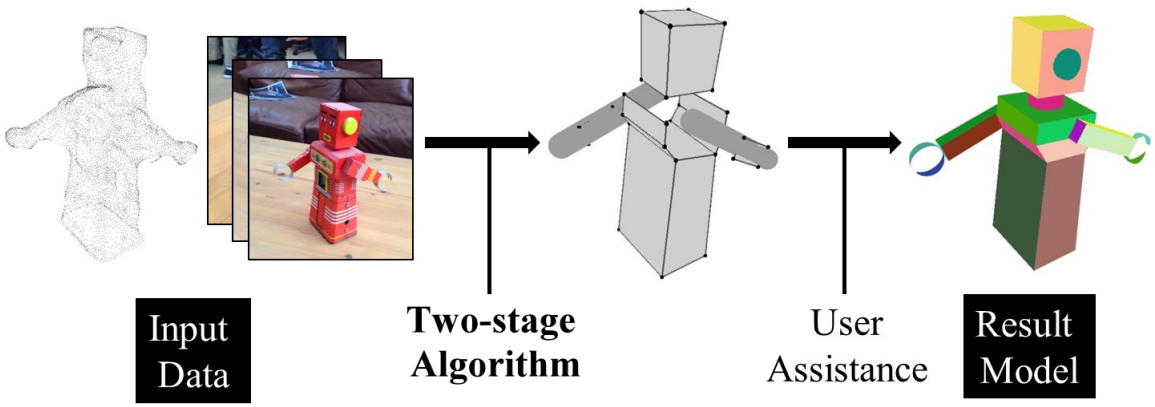

Figure 1: Overview of the proposed method.

is then applied to estimate the residual regions. As shown in Fig.1, the proposed system automatically generated models by fitting salient lines, planes and partial primitives. It allows user assistance to refine the reconstructed model. The user study demonstrates that the proposed method requires less user intervention time than that by a known modeling tool.

2 RELATED WORK

Reconstructing three-dimensional geometric models has attracted considerable interests. This section reviews research on model reconstruction from images and point cloud.

Cipolla and Robertson [Cip99a], Werner and Zisserman [Wer02a], and Schindler and Bauer[Sch03a] used multiple images as input, and generated a coarse model consisting of orthogonal planes. The properties of vanishing points are applied to obtain planes and camera parameters. Snavely et al. [Sna06a] proposed a Photo tourism system. It computed the view points of photographs, and found the correspondences between reconstructed 3D points and images. Izadi et al. [Iza11a] proposed a GPU pipeline to track and fuse a sequence of the point clouds into a volume structure. The estimated 3D scenes are of less noise and can be used for augmented reality (AR) applications. Nguyen et al. [Ngu13a] focused on high-resolution texture mapping for image-based modeling. They projected best-fitting images onto surface segments and merged the segments by using a graph-cut method.

Li et al. [Li11a] introduced a method to reconstruct a man-made object from scanned point sets. They focused on discovering global relations among parts of model to correct primitives, and applied several iterations of RANSAC fitting [Sch07a] and constrained optimization. Lafarge and Alliez [laf13a] adopted a two-step method for point-cloud-based modeling. They first detected a set of planar primitives from the input point set, and then they used min-cut formulation to reconstruct the surface of model. By contrast, our method extracted image salient lines and associated planes and combined a RANSAC fitting framework. Nan et al. [Nan10a] proposed an interactive modeling system for buildings with repetitive structural elements. Arikan et al. [Ari13a] introduced an intelligent snapping algorithm to best fit the input data and maintain the planarity of the polygons.

Several related articles compared different image-based methods or tools. Azevedo et al. [Aze09a] [Aze10a] analyzed the accuracy of volume-based modeling and shape-from-motion from multiple-view images. Bernardes et al. [Ber14a] discussed using image-based modeling tools as a replacement of laser scanning in the archaeological process.

3 PREPROCESSING AND OVERVIEW

The inputs of the proposed system are photographs and point clouds of a target object. The point clouds can be estimated by different active or passive vision techniques. In this paper, we acquired point clouds and photographs from the Autodesk 123D Catch gallery [Aut14a]. In the preprocessing stage, we currently utilized the Grabcut method [Rot04a] to extract foreground objects from the input photographs. The user intervention in preprocessing can be further lessened by using recent segmentation methods, e.g. [Lin15a]. The line segment detector (LSD) method [Von10a] is used to obtain line segment features from the foreground regions of photographs and masked images. Based on the epipolar geometry and constraint, our system finds the potential correspondences of line segments between different views, and it converts the two-dimensional line segment pairs into three-dimensional line segments. Besides, 3D line segments that are far from the point clouds are removed from the valid set. Fig. 2 depicts the result of this step.

The proposed system consists of two automatic stages, *selection of planes associated with salient lines* and *residual point cloud fitting with primitives*. After two

automatic processing stages, a friendly interface is provided for rapid user-assistance refinement. Fig. 3 shows the flowchart of the automatic stages. Section 4, 5 and 6 present the L-plane selection, primitive fitting and refinement stages, respectively.

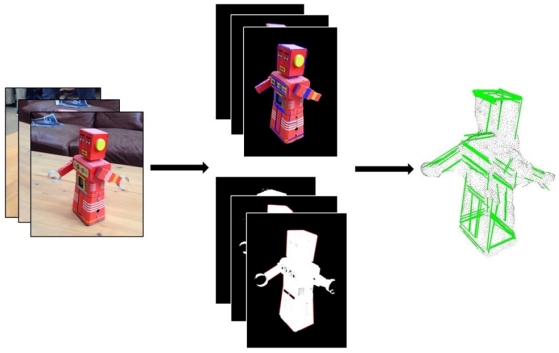

Figure 2: Result of salient line feature extraction.

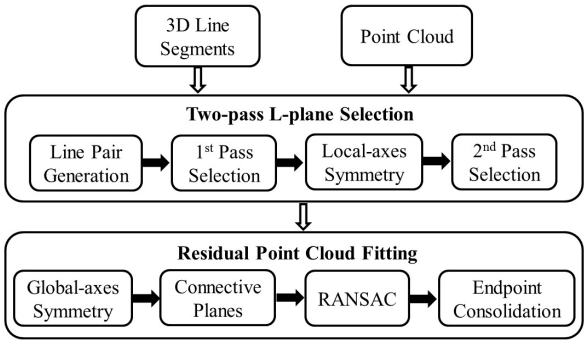

Figure 3: The flowchart of the automatic modeling stages.

4 FUNDAMENTAL FRAME MODELING WITH L-PLANES

4.1 Finding valid L-planes

We use the term *L-plane* to concisely represent a quadrangular plane derived from two or more three-dimensional salient lines. As shown in Fig. 4, either parallel or perpendicular line segments can form an L-plane candidate. The next step is to form a valid L-plane set from the naive candidates. We found the overlap ratio between an L-plane candidate and points of point clouds is useful for this task. For a point, we project it onto an L-plane candidate. If the projection (point-to-plane) distance is within a predefined range, we inspect whether the point is inside the L-plane region, as shown in Fig. 5. For an L-plane, the surface area that is covered by the input point clouds is named $Area_{points}$. Since there are always noise and measurement errors, in practice, we divided an L-plane area into regular cells, and evaluated the sum of area of cells that have close and valid point projections.

For each L-plane, we compute the filled ratio between the surface area of included points $Area_{points}$ and the rectangular area of L-plane $Area_{rect}$:

$$FilledRatio = \frac{Area_{points}}{Area_{rect}} \quad (1)$$

We exclude candidates of which *FilledRatio* values are less than a threshold and form the valid L-plane set.

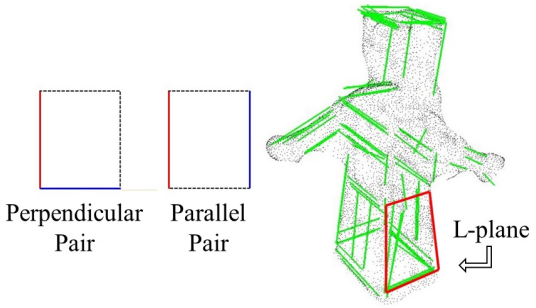

Figure 4: L-planes from pairs of salient lines.

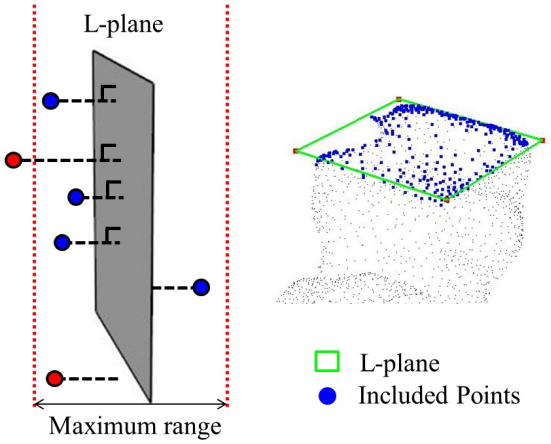

Figure 5: Check the coverage ratio between an L-plane candidate and point clouds.

4.2 First-pass selection with an objective function

The valid L-plane set is derived from salient lines in photographs, which can be influenced by texture on the object, discretized error, and other noise. An L-plane extracted in the above subsection is possibly overlapped with others, and therefore, we have to further extract a more compact L-plane set for the fundamental frame of a target model.

Before we introduce our objective function, we first measure the quality of a valid L-plane by its distance to point clouds. Assume an $L\text{-}plane_i$ includes N_i valid projection points. The projection distance between the included points V_j to the $L\text{-}plane_i$ is named $Dist(V_j, L\text{-}plane_i)$ (as the dashed lines in Fig. 5). The quality of

an $L\text{-}plane_i$ is defined as the average of valid point-to-plane distances.

$$Quality_i = \frac{1}{N_i}\sum_j Dist(V_j,\ L\text{-}plane_i) \qquad (2)$$

We propose an objective function to select a set of L-planes, S, which covers the maximum surface area of model accurately (of high quality) and with less overlap. The objective function E_{sel} is defined as:

$$E_{sel} = Area_{ent} - \sum_{s\in S}(Area_s - \omega \cdot Quality_s), \qquad (3)$$

where $Area_{ent}$ is the surface area of the entire point cloud set, $Area_s$ represents the covered surface area of $L\text{-}plane_s$, and ω is a weight to adjust the importance of quality. $Quality_s$ in Eq.3 can be regarded a penalty of fitting error for plane s. The optimum is substantially a complex combinatorial problem. In this paper, we adopt a heuristic algorithm to efficiently approximate this problem. At each iteration, the proposed system finds a single L-plane s that can minimize the E_{sel} value, and it then removes the $Area_s$ from $Area_{ent}$. The selection procedure continues until there is no adequate L-plane.

4.3 Second-pass selection by local-axes symmetry

Since symmetry characteristics occur in a large portion of man-made objects, we intend to find more valuable L-planes (which may be missed during salient line extraction) according to symmetry of the primary L-planes selected above.

Given a pair of primary L-planes which are parallel or perpendicular to each other, we evaluate their local symmetric center and axis. This pair of L-planes are then flipped and shifted to check whether there are rotational or mirror symmetry situations. The filled ratio in Eq. 1 is again used to verify the new L-plane candidate. For error tolerance, when an L-plane is flipped or shifted, the proposed system also moves it backward and forward within a small offset, and takes the position with the highest filled ratio as the new L-plane. Fig. 6 shows the symmetric L-plane candidates derived from the primary L-planes.

With a set of symmetric L-plane candidates, We also apply the objective function in Eq.3 to extract a set of auxiliary L-planes. The entire area $Area_{ent}$ for the second pass selection is now the final value of the first pass selection. The symmetric L-planes and results of the two-pass selection are shown in Fig. 7.

5 RESIDUAL POINT CLOUD FITTING

By the two-pass selection, we have estimated the fundamental frame of the point cloud. However, there

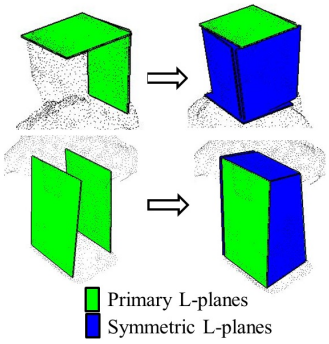

Figure 6: Symmetric L-plane candidates.

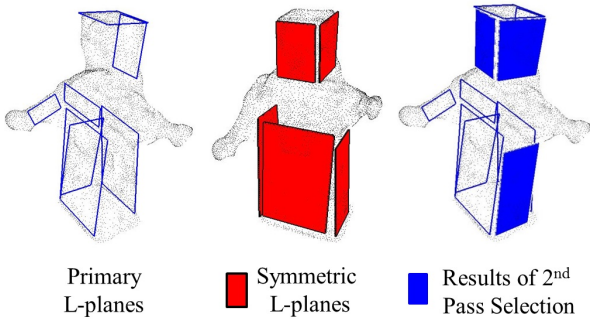

Figure 7: Symmetric L-planes and results of two-pass selection.

are still points uncovered by the selected L-planes. To fit these residual points, this stage further utilizes two fitting procedures, which are L-planes by global-axes symmetry, and RANSAC primitive fitting, and adjusts polygon endpoints.

5.1 Extension with global-axes symmetry

In 4.3, we have exploited the local symmetry for L-plane finding. After getting the fundamental frame of point clouds, we would like to find its global axes, and use them to acquire more symmetric L-planes. One common thought for retrieving the global axes is using principal component analysis (PCA). However, as the robot case in Fig. 7, the second principal axis is roughly along the arm direction. By contrast, users usually regard the normals of building façades or pedestals as the axes of an object. Therefore, we apply box fitting on selected L-planes. A box is formed by at least three L-planes. The plane normals, lengths and distances of adjacent edges are used to determine whether these L-planes are adequate to form a box. We then estimate the box center and the average normals of every two opposite planes to form possible global axes of symmetry. Fig. 8 shows an example of box fitting of selected L-planes. Extended L-planes extracted by global-axes symmetry are shown in Fig. 9.

5.2 Plane connection

Since the L-planes obtained so far may not totally cover the whole model, connective planes are generated to fill

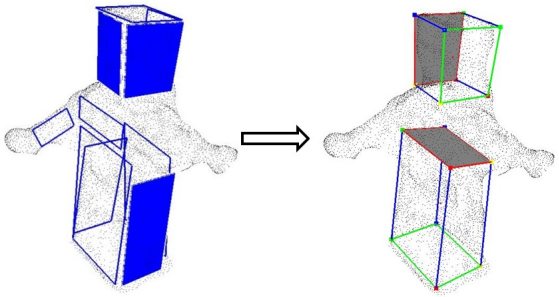

Figure 8: Box fitting of selected L-planes.

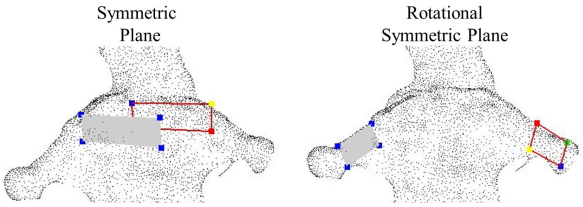

Figure 9: Examples of mirror and rotational symmetric planes by global axes.

small gaps between between two L-planes. The connection process is activated when two selected L-planes have similar plane normals, their distance is smaller than a distance T_{dis}, and their adjacent edges have similar lengths. Examples of connective planes are shown in Fig. 10.

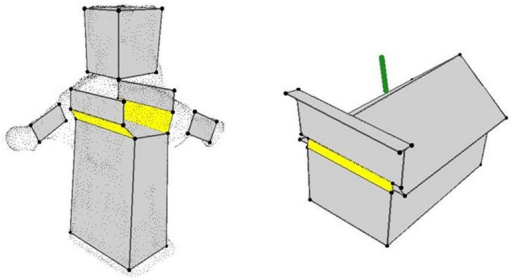

Figure 10: Examples of connective planes.

5.3 Primitive fitting by RANSAC

In this step, the state-of-the-art RANSAC-based method [Sch07a] is applied for fitting the residual point cloud. This local RANSAC-based approach can detect basic shapes, including plane, cylinder, sphere, cone and tori from unorganized point clouds. Nevertheless, our experiment found that the later three primitives were relatively unstable in fitting, and therefore, we adopted using only the plane and cylinder primitives.

Since our photogrammetry-based input point clouds are sometimes seriously disturbed by noise, the results of RANSAC detection cannot fit the target boundary accurately. Additional processes have to be applied for extracting the vertices of primitives from a subset point cloud estimated by the RANSAC method. First, we exploit the Delaunay triangulation to rearrange the mesh of subset points. Second, we apply geometric α-shape for finding the boundary points. Then, boundary points of included angles smaller than a threshold are set to be vertices. Fig. 11 shows an example of primitive boundary estimation.

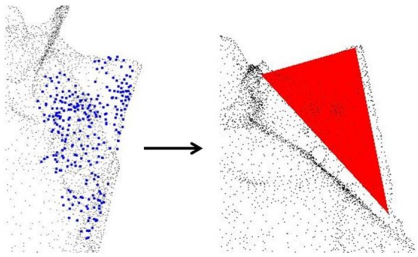

Figure 11: Primitive boundary estimation from a point cloud subset.

5.4 Endpoint consolidation

The last step in automatic modeling is to merge close endpoints of L-planes and primitive planes together. As mentioned in 5.1, we have found boxes of L-planes as candidates of global axes. In this step, the proposed system also first merges vertices around the box vertices, and then merges other points. Fig. 12 shows an example.

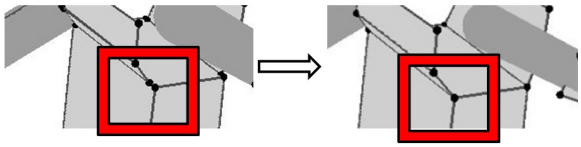

Figure 12: Examples of endpoint consolidation.

6 USER-ASSISTANCE REFINEMENT

Even though fitting the L-plane frame first improves the latter results of primitive fitting, users may still want to adjust the automatically estimated model. Hence, the proposed system also provides users with a friendly interface for rapid model editing. As shown in Fig. 13, users can turn polygons to the transparent modes and align the vertices with input photographs. As shown in Fig. 14, users can simply draw a two-dimensional stroke on the screen to select two edges and our system will predict possible operations, e.g. creating a new plane, creating a plane and its symmetry, in a row of suggestive operations for users to select. Users can also select a polygon as the working plane and insert a new primitive, such as a cylinder or sphere, aligned with the plane. The parameters of primitives, e.g. sizes, lengths, types, can also be adjusted in the control panel. Please refer to the supplementary video in which the user interface, editing processes and results are demonstrated.

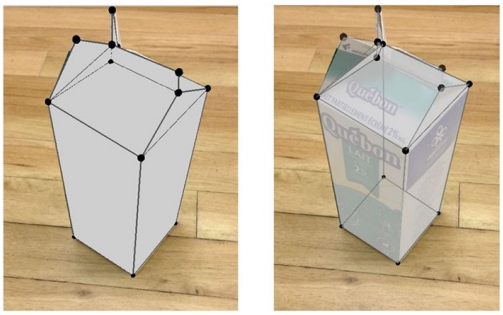

Figure 13: Overlapping the model and photograph on interface.

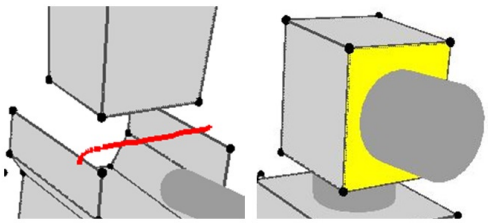

Figure 14: Generating a new plane by a stroke and a new cylinder on the selected working plane.

7 EXPERIMENTS

7.1 Result comparison

To demonstrate the effectiveness of the proposed method, we apply a state-of-the-art RANSAC method by Schnabel et al. [Sch07a] and the proposed method to several data sets from the 123D Catch gallery. As shown in Fig. 15, if we directly apply the RANSAC method, the results can be distorted due to the noise-disturbed point clouds. When the boundary of components in the target is not clear in point clouds, they cannot be accurately extracted by a RANSAC method. For instance, the signboard and frontal face of the house are considered a single plane in front of the house by direct RANSAC, and the top of the drink carton is fitted by quadrangle and triangles. By contrast, the proposed method estimated the L-plane frame in advance and the results are closer to users' expectation.

7.2 Texture mapping

In order to obtain the texture map for each polygon, the proposed system chooses the input photos in the most frontal view and transfers their texture. We also apply the perspective correction for the photos of large view angles. Examples about models with texture mapping are shown in Fig. 16.

7.3 Computation time

The content complexity of input photographs substantially influences the computation time. When a target object in photos has complicate patterns, such as

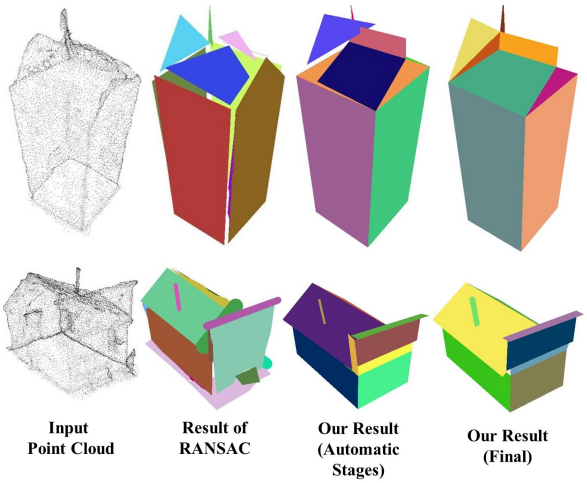

Figure 15: Comparison between the results of advanced RANSAC[Sch07a] and the proposed method.

Figure 16: Texture mapping on the result models.

wooden texture or nets, the proposed system retrieves more salient lines from images and constructs more 3D line segments for L-plane estimation, and therefore prolongs the computation time. The computation time of each result model is listed in Table 1. Fig. 17 demonstrates the results generated by the proposed automatic stages and final results with user refinement. The experiment was performed on a desktop computer with an Intel i7 3.4GHz CPU and 16GB RAM. Currently, a large portion of the system is developed in a single thread. OpenMP [Omp14a] library is applied for searching corresponding line segments in parallel; OpenGL [Ogl14a] and GLUI [Glu14a] libraries are applied for interface development.

7.4 User study

To evaluate how easily users can use our interface to refine automatically estimated models, we conducted user study with five volunteers. All participants received ten-minute demonstration and practice about our interface. Each user had to edit four models estimated by the proposed automatic stages. Table 2 lists the editing time of each user. Moreover, to compare the proposed modeling procedure with popular modeling methods, we also asked users to reconstruct these four

Model	Image num.	Point num.	Auto stage	User refine.
House	24	13297	65.81 sec	$\approx$ 4 min
Robot	15	6155	27.31 sec	$\approx$ 5 min
Mailbox	22	6045	33.44 sec	$\approx$ 4 min
Bigben	37	14526	281.1 sec	$\approx$ 12 min
Drink	23	8310	48.64 sec	$\approx$ 3 min
Lomo	21	13522	45.191 sec	$\approx$ 8 min

Table 1: Information of each result model shown in Fig. 17. The columns from left to right: model name, input image number, input point number, computation time in the automatic stage, and user refinement time

Ours	Mailbox	Bigben	Drink	House
User 1	3 min	6 min	2 min	6 min
User 2	3 min	6 min	2 min	2 min
User 3	3 min	8 min	3 min	2 min
User 4	2 min	6 min	1 min	3 min
User 5	3 min	7 min	3 min	2 min
Average	2.8 min	6.6 min	2.2 min	3 min

Table 2: Editing time of each user by using our refinement interface.

models by using Google SketchUp Make. Similarly, users received a ten-minute video demo and practice about SketchUp interface, and they were required to build the same four models. Table 3 lists the editing time by using SketchUp Make.

These participants had never used our interface and the SketchUp Make software. Since our automatic stages provided approximate models, users just needed fewer simple operations to complete the models. According to users' comments on the SketchUp Make, they can rapidly create a simple building model, but it is relatively not intuitive for modeling several specific parts, such as the roof region of the drink (carton) model.

	Mailbox	Bigben	Drink	House
User 1	9 min	9 min	8 min	9 min
User 2	16 min	9 min	13 min	9 min
User 3	7 min	15 min	8 min	13 min
User 4	8 min	9 min	10 min	7 min
User 5	8 min	11 min	13 min	10 min
Average	9.6 min	10.6 min	10.4 min	9.6 min

Table 3: Editing time of each user by using SketchUp Make.

8 CONCLUSION AND FUTURE WORK

This paper presents a novel framework to reconstruct 3D graphics models from a set of photos and point clouds. Our system first extracts salient line segments from images, and then reconstructs the fundamental 3D frame of a target based on the line segments and plane symmetry properties. Modeling based on this frame can lessen the ambiguity situations and substantially improve the accuracy of the following primitive fitting. The proposed framework can automatically estimate primitive models and it provides friendly refinement interface. User evaluation demonstrates that it can also lessen user editing time compared to a known tool.

There are several possible future directions. An interesting extension is to construct the fundamental frame according to more salient features, such as curves. It can further reduce the ambiguity in residual fitting. Besides, applying shape-from-shading techniques on surfaces [Lin10a] or using advanced fitting methods [Li11a] for residual points can further refine the estimated models.

ACKNOWLEDGMENTS

Authors would like to thank the volunteers who participated in user evaluation. This project was partially supported by Ministry of Science and Technology, Taiwan under grant no. MOST 103-2221-E-009-143 and MOST 104-2218-E-009-008.

9 REFERENCES

[Ari13a] Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., and Maierhofer, S. O-snap: Optimization-based snapping for modeling architecture. ACM Trans. Graph., vol.32, no.1, article 6, 2013.

[Aut14a] Autodesk 123D Catch Gallery, http://www.123dapp.com/Gallery/catch.

[Aze09a] Azevedo, T.C.S., Tavares, J.M.R.S., Vaz, M.A.P. 3D object reconstruction from uncalibrated images using an off-the-shelf camera. Advances in Computational Vision and Medical Image Processing Computational Methods in Applied Sciences. vol. 13, pp 117-136, 2009.

[Aze10a] Azevedo, T.C.S., Tavares, J.M.R.S., Vaz, M.A.P. Three-dimensional reconstruction and characterization of human external shapes from two-dimensional images using volumetric methods. Computer Methods in Biomechanics and Biomedical Engineering, vol.13, no.3, pp.359-369, 2010.

[Ber14a] Bernardes, P., Magalhaes, F., Ribeiro, J., Madeira, J., Martins, M. Image-based 3D modelling in archaeology: application and evaluation, Proc. Intl. Conf. Computer Graphics, Visualization and Computer Vision (WSCG), pp. 159-166, 2014.

[Cip99a] Cipolla, R., and Robertson, D. 3D models of architectural scenes from uncalibrated images and vanishing points. Proc. Intl. Conf. Image Analysis and Processing, 1999. , pp. 824-829, 1999.

[Deb96a] Debevec, P.E., Taylor, C.J., and Malik, J. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. Proc. SIGGRAPH'96, pp.11-20, 1996.

[Glu14a] GLUI User Interface Library, http://glui.sourceforge.net/

[Iza11a] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. Proc. ACM symp. User interface software and technology, pp.559-568, 2011.

[laf13a] Lafarge, F., and Alliez, P. Surface reconstruction through point set structuring. Comput. Graph. Forum, vol.32, no.2, pp.225-234, 2013.

[Li11a] Li, Y., Wu, X., Chrysanthou. Y., Sharf, A., Cohen-Or. D., and Mitra, N.J. Globfit: consistently fitting primitives by discovering global relations. ACM Trans. Graph., vol.30, no.4, article 52, 2011.

[Lin10a] Lin, I.-C., Chang, W.-H., Lo, Y.-S., Peng, J.-Y, Lin, C.-Y. Image-based detail reconstruction of non-Lambertian surfaces. Computer Animation and Virtual Worlds, vol.21, no.1, pp.55-68, 2010.

[Lin15a] Lin, I.-C., Lan, Y.-C., Cheng, P.-W. SI-Cut: Structural inconsistency analysis for image foreground extraction. To appear in IEEE Trans. Visualization and Computer Graphics, 2015.

[Nan10a] Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., and Chen, B. Smartboxes for interactive urban reconstruction. ACM Trans. Graphics, vol. 29, no.4, article 93, 2010.

[Ngu13a] Nguyen, H.M., Wunsche, B., Delmas, P., Lutteroth, C., van der Mark, W., Zhang, E. High-definition texture reconstruction for 3D image-based modeling, Proc. Intl. Conf. Computer Graphics, Visualization and Computer Vision (WSCG), pp.39-48, 2013.

[Ogl14a] OpenGL (Open Graphics Library), https://www.opengl.org.

[Omp14a] OpenMP (Open Multi-Processing Library), http://openmp.org/wp/

[Rot04a] Rother, C., Kolmogorov, V., and Blake, A. GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Trans. Graphics, vol.23, no.3, pp. 309-314, 2004.

[Sch03a] Schindler, K., and Bauer, J. A model-based method for building reconstruction. Proc. Intl. Conf. Computer Vision workshop on Higher-Level Knowledge in 3D Modeling and Motion, pp.74-82, 2003.

[Sch07a] Schnabel, R., Wahl, R., and Klein, R. Efficient ransac for point-cloud shape detection. Comput. Graph. Forum, vol.26, no.2, pp.214-226, 2007.

[Sna06a] Snavely, N., Seitz, S.M., and Szeliski, R. Photo tourism: Exploring photo collections in 3d. ACM Trans. Graph., vol.25, no.3, pp.835-846, 2006.

[Von10a] von Gioi, R.G., Jakubowicz, J., Morel, J.M., and Randall, G. LSD: A fast line segment detector with a false detection control. IEEE Trans. Pattern Analysis and Machine Intelligence, vol.32, no.4, pp.722-732, 2010.

[Wer02a] Werner, T., and Zisserman, A. New techniques for automated architectural reconstruction from photographs. Proc. European Conf. Computer Vision-Part II, pp. 541-555, 2002.

Figure 17: The leftmost column: results of automatic stages; the other columns: final results in three different views and corresponding input photos.