Interactive 3D Modeling Using Only One Image

Sujin Liu
National University of Singapore
liusujin@comp.nus.edu.sg

Zhiyong Huang *
National University of Singapore
huangzy@comp.nus.edu.sg

ABSTRACT

For virtual reality systems, modeling of 3D objects and scenes is important and challenging. In this paper, we present an image-based interactive 3D modeling framework consisting of three major modules: photogrammetric modeling, human interaction, and texture mapping. These three modules are not sequentially used and they are mixed in the whole modeling process. The major idea is to explore the use of images in interactive modeling systems to achieve the automation. In particular, the use of only one image is addressed. On one side, unlike the common fully interactive modeling framework, the users are not required to specify some low level details interactively which can be derived automatically from the image. On the other side, it still requires human interactions to do some high level tasks that the algorithms are difficult to perform automatically. We have implemented the framework and experimental results are good.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.3.7 [**Computer Graphics**]: Three - Dimensional Graphics and Realism

Keywords

Modeling of 3-D shape, 2-D image, human computer interaction, texture mapping.

1. INTRODUCTION

For virtual reality systems, modeling of 3D objects and scenes is important and challenging [22, 27]. The ease of creating 3D models is crucial for the success of any virtual reality (VR) systems. Although much progress has been made on geometric modeling systems, they are mainly used in computer aided design (CAD) and yet to be used to create geometric models for VR systems, usually with irregular shapes such as animals and plants.

Inspired by recent interactive work such as the Teddy and SKETCH, where new human computer interactive techniques were introduced for rapid designing of 3D freeform objects [15] and modeling of CSG-like models consisting of simple primitives [28], we propose a hybrid approach of 3D modeling for virtual reality systems. The major idea is to introduce the use of images [6, 24] into interactive frameworks. It can be considered as an extension of the Teddy and SKETCH. It is not fully automatic. The use of images will release human from modeling of the low level details while some high level tasks that the algorithms are yet to solve automatically still requires human interaction. In a brief summary, the major features of our method are:

1. It requires human interaction. By applying human interaction, we can avoid the use of some complicated procedures of computer vision such as camera calibration [20] and procedures of computational geometry such as polygonization of unorganized points [1].

2. It has automatic process. A user can start from a good initial 3D model. It is automatically derived from the image of the object.

3. It is not a framework of the stereo vision. Only one image is used. Thus, it is not necessary to solve the correspondence between two images. As one image can not determine a unique shape in 3D, the modeling result is always an approximation of the real object. However, for virtual reality systems, it meets the visual requirement.

4. It is not model-based and does not require a generic 3D model. So the shapes to be modeled are not constrained by the initial generic 3D models as in [8].

5. It is general though we have not implemented modeling of the shapes with holes.

The remaining paper is organized as follows. Section 2 briefly reviews some related work. Section 3 describes our framework in detail with the focus on the use of images in the interactive modeling framework. Section 4 presents our implementation and shows some modeling results. Section 5 concludes this paper.

*Department of Computer Science, School of Computing, NUS, Singapore 117543.

2. RELATED WORK

Geometric modeling can be roughly divided into forward and reverse approaches. In the forward approach, the real model of the object is not available or not directly used in the modeling process. The shape of the object is usually modeled by an iterative bottom up manner. Oppositely, in the reverse approach, images or the real measurements of the object are directly used. The purpose of the reverse approach is to achieve automation.

The most popular forward methods are based on CSG/B-rep framework, i.e., the framework of constructive solid geometry (CSG) and boundary representation (B-rep) [10]. They use geometric primitives to hierarchically build up through successive shape operations and transformations. Recent work includes SKETCH which introduces new interfaces for rapid modeling of CSG-like models consisting of simple primitives [28]. This approach is most suitable for CAD systems where shapes of objects are relatively regular and require precise representation. Other work includes Teddy [15] where new interfaces and interactive techniques are introduced for rapid designing 3D freeform objects. In particular, Teddy allows a first-time user to model a moderately complex object within minutes. A more recent work is presented in [27] where a virtual environment is created by directly input to computer of a hand-drawn perspective sketch. Another type of forward modeling methods are implicit surface based [10]. The user specifies the skeleton of the intended model and the system constructs smooth, natural-looking surfaces around it [23].

The most popular reverse methods are 3D digitizing (range image) based. Using such devices as the Cyberware scanner [7], a dense mesh of 3D points can be derived from a set of laser-ranged images of a real model. Then triangulation techniques of computational geometry are applied to recover the topology and geometry of the object [1]. An early work is on human body modeling [25], while recent work includes accuracy incrementing reconstruction by multi-round digitizing [6]. Currently, 3D digitizing is widely used in the entertainment industries such as video games and film production. A different type of reverse modeling methods are digital image (intensity image) based [24], with some well known research topics such as camera calibration [20], shape from motion [18, 26], and stereo vision [9]. In computer graphics, it is widely used in facial modeling and animation [12, 17, 19]. The approach takes the advantage that digital images are much easier to get than the range images.

The forward approach emphasizes on human interaction, while the reverse approach on automation. Combining both approaches is expected to achieve better performance. This idea was explored in the Facade system [8] where a hybrid approach, with automation achieved from the use of images, is proposed to construct large buildings. The results are very impressive. The best use of human interaction and automation together is the central concern of the hybrid approach. Our work differs from the Facade system in that ours is not model-based and does not require generic 3D models at the beginning. So the shapes to be modeled are not constrained by these 3D models.

An interesting walk-through framework TIP (Tour Into the Picture) also used one image of a scene [14]. Different from our work, in TIP, the background of a scene model consists of at most five rectangles, whereas hierarchical polygons are used as a model for each foreground object. Another work was on face recognition [2]. Using one image, the method exploited prior knowledge of faces to generate their views under different rotations and used these views for the recognition.

3. OUR WORK

The overview of the framework is shown in Figure 1. We describe the first module, photogrammetric modeling, in more detail. As human interaction and texture mapping are similar to other work, we only briefly describe them.

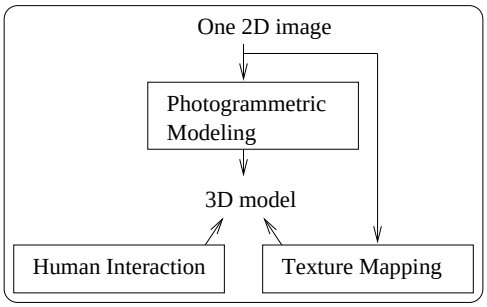

Figure 1: Overview of the framework.

3.1 Photogrammetric Modeling

In this subsection we present the photogrammetric modeling. It is the heart of the framework. The purpose is to achieve the automation for the modeling. The input is one 2D image of the object to be modeled (Figure 2, left). First, the shape boundary of the object is extracted (Figure 2, right). Next, the skeleton of image is constructed (Figure 3). Based on the shape boundary and skeleton, the 2D triangle mesh of the object is constructed by a 2D constrained Delaunay triangulation algorithm (Figure 3). Next, the 3D mesh is created (Figure 5, Left). Finally, the complete 3D mesh is generated after adding the back facing mesh (Figure 5, Right).

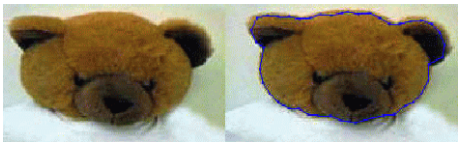

Figure 2: Contouring: shape boundary extraction.

The first step, contour extraction, derives the shape boundary of the object in its image. It is done by color clustering [16]. The algorithm classifies the pixels into different clusters by comparing result of the color threshold of each cluster. For our case, the number of clusters is two, foreground and background. The initial clusters are derived from sample pixels picked interactively, with one pixel representing each cluster. Clusters are iteratively expanded by selectively adding the neighboring pixels automatically. Its color threshold is also automatically updated to the value of the new center. It works well for images with its foreground distinguishable from background by colors and not

necessary the pure color background as some other methods. The algorithm traces the shape boundary and a list of 2D points are derived automatically which forms a closed shape boundary (Figure 2).

The second step is to derive the skeleton of the 2D image from its shape boundary [11]. The purpose of this step is for the triangulation in the third step. We have developed an algorithm based on the feature tracking and minimal spanning tree. First, feature points of the image are derived using the well known KLT tracker [26]. These feature points represent the discontinuity of the shape and will take the role as the joints of the shape skeleton. Second, the resulting feature points are connected together to form the skeleton. An implementation of KLT tracker is available for us to use on the web [3]. Briefly, good features are located by examining the minimum eigenvalue of each 2 by 2 gradient matrix, and features are tracked using a Newton-Raphson method of minimizing the difference between the two windows. Multiresolution tracking allows for even large displacements between images. We have adapted the implementation in our system. After deriving the 2D feature points, we connect them to form the skeleton of the shape. For doing it, we have used the standard minimal spanning tree algorithm [5]. The Euclidean distance between two feature points is used as the weight of the edge between them. The intuition is that the 2D shape skeleton can be constructed by linking the neighboring joints (2D feature points). One example of the resulting skeleton (white line) is shown in Figure 3.

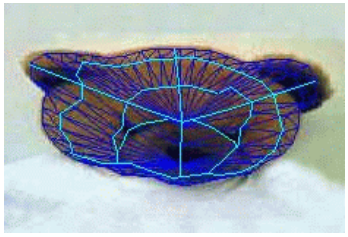

Figure 3: Skeleton (white line) and 2D mesh (dark line).

The third step is to derive a 2D mesh. The resulting triangles should consist of the shape boundary and skeleton derived from the previous two steps. The constrained Delaunay triangulation, a well researched topic in computational geometry [4], is implemented that uses the shape boundary and skeleton as its input. A free available software Qhull is used [21]. A constrained Delaunay triangulation of a set of line segments (the shape boundary and skeleton) is the triangulation of the endpoints where the distance between them is the length of the shortest path which does not cross a line segment. To approximate the shape better, we can add more feature points interactively and connect them to the shape boundary and skeleton for the use in the triangulation. One example of a 2D mesh result (dark line) is shown in Figure 3.

The last step is to derive a 3D mesh which is the major step requires the human interaction. It is done by lifting vertices of 2D mesh with different height values. Because of the use of one image, we can not derive the very accurate depth values using the computer vision techniques such as the stereo method. However, not using them, we also can avoid the complexity and inaccuracy of the algorithms.

In the Teddy system [15], as no images are used, a simple method is applied: for each vertex on the 2D mesh, the displacement depends on its position in the model, i.e., vertices in the central part of the shape are lifted more than those of the boundary. The initial value can be set proportional to the average length of its incident edges in 2D. For decreasing the efforts of human interaction, we take a different approach based on the use of image intensity (shape from shading) [13] to estimate the displacement for each vertex of the 2D triangle. The method of shape from shading is based on the measurement result of the different intensities present in the images to obtain the depth information. However, in order to derive the accurate values, it requires the special specification of the lighting but it is not true for the images we used. Thus, the depth values derived directly from shading are not accurate. So the vertex clusters based on the inaccurate depth values are not accurate. They must be finely adjusted using the human interaction. As only a limited number of vertices need to be adjusted, e.g., about 20 for the bear example, it is not a heavy task for the user. Finally, we displace different clusters by the user specified height values (as the recovered depth values are not accurate).

Now, we have a similar problem as the Teddy system: after the lifting, the resolution of 3D mesh decreases. We need to add more vertices by interpolation of the existing vertices similar to the Teddy system (Figure 4). Finally, a back facing mesh can be generated with either a plane mesh (for example, bear in Figure 5, Left) or a symmetric mesh of the front facing mesh (for example, fish in Figure 10). The interactive editing is applied to derive the final shape (Figure 5, Right).

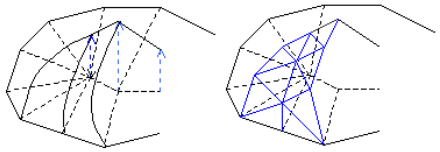

Figure 4: Interpolation to have more vertices in 3D mesh.

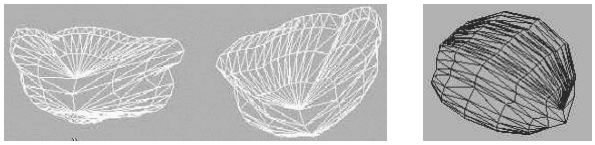

Figure 5: Left: front facing 3D Mesh. Right: back facing mesh.

3.2 Human Interaction

We implement a user interface that supports the 2D and 3D visualization and manipulation. Manipulations can be directly applied on the 3D object (front and back facing mesh) such as picking, grouping, adding, deleting, displacing, etc.

Detailed description can be found in [15]. A snapshot of the user interface of our system is shown in Figure 6.

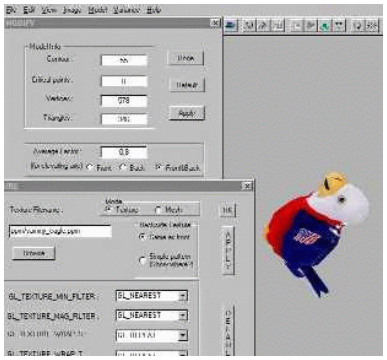

Figure 6: User interface: a snapshot.

3.3 Texture Mapping

Finally, the image is reused for texture mapping. For the front facing mesh, it is straight forward because we have kept the vertex correspondence between 3D and 2D mesh from the photogrammetric modeling (Figure 7, Left). It is a problem for the back facing mesh. We have to use human interaction. For modeling of the bear, we solve it by interactively selecting a small area of the image such as part of hairy face of the bear. Then, we use this small image to remove features such as eyes of the front image. The resulting image is used for texture mapping of the back facing mesh (Figure 7, Right).

Figure 7: Left: front face result of texture mapping. Right: back face result of texture mapping.

4. RESULTS

We implemented our framework on PC with an Intel Pentium III 450MHz processor using Microsoft Visual C++ and OpenGL. The algorithm of each step can run in real time for all the examples we tested. More modeling results are shown for a fish (Figure 8, 9 and 10), seagull(Figure 11, 12 and 13), bobby (Figure 14, 15 and 16), and mouse (Figure 17, 18 and 19).

Figure 8: Fish: input image.

5. CONCLUSIONS

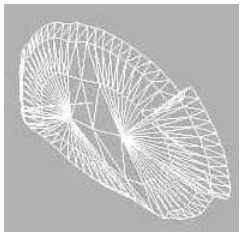

Figure 9: Fish: reconstructed 3D Mesh.

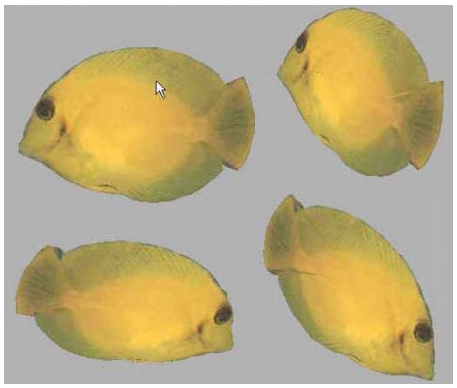

Figure 10: Fish: after texture mapping.

We have presented a hybrid 3D modeling framework combining human interaction (forward approach) and automation achieved from the use of one image (reverse approach). Photogrammetric modeling is applied to get a good approximation from the use of one image. Complementary to the photogrammetric modeling, the use of interactive techniques can solve some high level tasks easily. We have done some experiments. The results are good in visual quality and suitable to use in a virtual reality system.

6. ACKNOWLEDGMENTS

We appreciate the anonymous reviewers for the constructive critics. We thank Dr. Leow Wee Kheng and Zhang Yong of the Center for Heuristics in Information Mining and Extraction, NUS for discussions and the use of color clustering software package.

This work was supported partly by Academic Research Grant (RP3982704) and research student scholarship of National University of Singapore.

Figure 11: Seagull: input image.

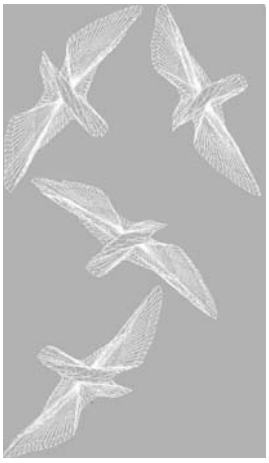

Figure 12: Seagull: reconstructed 3D Mesh.

Figure 13: Seagull: after texture mapping.

Figure 14: Bobby: input image.

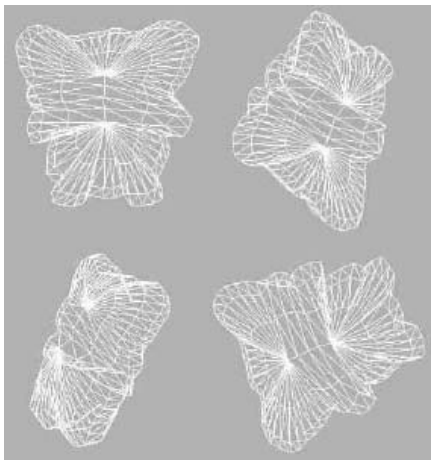

Figure 15: Bobby: reconstructed 3D Mesh.

7. REFERENCES

[1] N. Amenta, M. Bern, and M. Kamvysselis. A New Voronoi-Based Surface Reconstruction Algorithm. *Proc. ACM SIGGRAPH'98*, 415-421 (1998).

[2] D. Beymer and T. Poggio. Face Recognition from One Example View. *Proc. IEEE ICCV'95*, 500-507 (1995).

[3] S. Birchfield. KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker. *http:// vision.stanford.edu/ birch/ klt/*, Stanford University, 2000.

[4] J. D. Boissonnat and M. Yvinec. Algorithmic Geometry. *Cambridge University Press*, 266-281 (1998).

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. *The MIT Press*, 498-513 (1990).

[6] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. *Proc. ACM SIGGRAPH'96*, 303-312 (1996).

[7] Cyberware Laboratory, 4020/RGB 3D Scanner with Color Digitizer. *Cyberware Laboratory, Inc, Monterey, California*, (1990).

[8] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach. *Proc. ACM SIGGRAPH'96*, 11-20 (1996).

[9] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. *Proc. of ECCV'92*, 563-578 (1992).

[10] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. Computer Graphics, Principles and Practice. *Addison-Wesley Publishing Company*, 534-539, 557-558, 1047-1048 (1996).

[11] R. C. Gonzalez and R. E. Woods. Digital Image Processing. *Addison-Wesley Publishing Company*, 491-503 (1992).

[12] M. Gotla and Z. Huang. A Minimalist Approach to Facial Reconstruction. *In the proceedings of MMM'99* 377-388 (1999).

[13] B. K. P. Horn. Image Intensity Understanding. *Artificial Intelligence*, 8:201-231 (1977).

[14] Y. Horry, K. Anjyo, and K. Arai. Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image. *Proc. SIGGRAPH'97*, 225-232 (1997).

Figure 16: Bobby: after texture mapping.

Figure 17: Mouse: input image.

[15] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. *Proc. ACM SIGGRAPH'99*, 409-416 (1999).

[16] R. Jain, R. Kasturi, and B. G. Schunck. Machine Vision. *McGraw-Hill, Inc.*, 180-191 (1995).

[17] W. S. Lee, P. Kalra, and N. M. Thalmann. Model Based Face Reconstruction For Animation. *In the proceedings of MMM'97*, 323-338 (1997).

[18] W. K. Leow, Z. Huang, Y. Zhang, and R. Setiono. Rapid 3D Model Acquisition from Images of Small Objects. *Proc. Geometric Modeling and Processing'2000*, 33-44 (2000).

[19] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing Facial Expressions from Photographs. *Proc. of ACM SIGGRAPH'98*, 75-84 (1998).

[20] M. Pollefeys, R. Koch, and L. van Gool. Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters. *Proc.ICCV'98*, 90-95 (1998).

[21] Qhull. Home pages for Qhull. *http:// www.geom.umn.edu/ software/qhull/*, Geometric Center, The University of Minnesota.

[22] T. Sakaguchi and J. Ohya. Modeling and Animation of Botanical Tree for Interactive Virtual Environment. *Proc. VRST'99*, (1999).

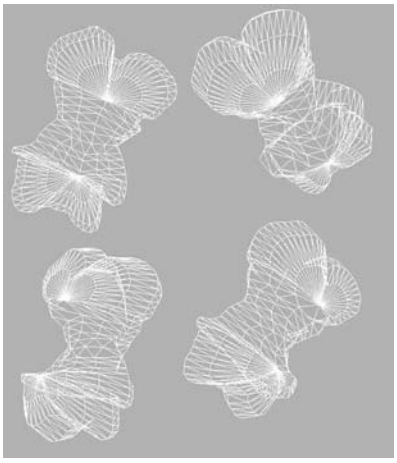

Figure 18: Mouse: reconstructed 3D Mesh.

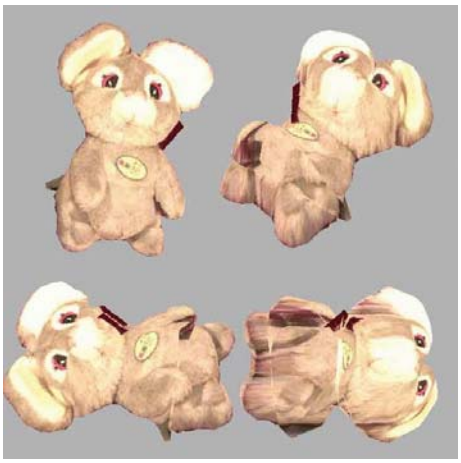

Figure 19: Mouse: after texture mapping.

[23] J. Shen and D. Thalmann. Interactive Shape Design Using Metaballs and Splines. *Implicit Surfaces'95*, 187-196 (1995).

[24] R. Szeliski and S. B. Kang. Recovering 3D Shape and Motion from Image Streams using Non-linear Least Squares. *Journal of Visual Communication and Image Representation*, 5(1), 10-28 (1994).

[25] N. M. Thalmann and D. Thalmann. The Direction of Synthetic Actors in the Film Rendez-vous a Montreal. *IEEE Computer Graphics & Applications*, 7(12), 9-19 (1987).

[26] C. Tomasi and T. Kanade. Shape and Motion from Image Streams under orthography: A Factorization Method. *IJCV*, 9, 137-154 (1992).

[27] A. Turner, D. Chapman, and A. Penn. Sketching a Virtual Environment: Modeling Using Line-Drawing Interpretation. *Proc. VRST'99*, (1999).

[28] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: An Interface for Sketching 3D Scenes. *Proc.*

ACM SIGGRAPH'96, 163-170 (1996).