ACQUIRING 3D MODELS FROM IMAGES FOR MULTIMEDIA SYSTEMS

WEE KHENG LEOW, ZHIYONG HUANG, LUPING ZHOU, INDRIYATI ATMOSUKARTO, YONG ZHANG

School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543
leowwk, huangzy, zhoulupi, indriyat, zhangyon@comp.nus.edu.sg

Multimedia systems integrate and process multiple types of contents such as text, image, graphics, video, and sound. In addition to text, image, and video, 3D graphics is becoming an important Internet medium for information exchange. To use 3D information, 3D graphics models must be acquired, and 3D model acquisition remains a very challenging task. This paper presents a system for automated acquisition of 3D models. After highlighting the main components of the system, this paper focuses on a new algorithm for constructive polygonization of 3D points. The acquisition results of real objects are illustrated using VRML.

1 Introduction

The rapid progress and integration of computing, telecommunications, and broadcasting technologies have brought about the emergence of multimedia systems. These systems integrate and process information in multiple media types, including text, image, graphics, video, and sound. While text, images, and video are currently the predominant types of information exchanged over the Internet, 3D graphics is becoming another important Internet medium for information exchange. The increasing importance of 3D graphics is realized by the development of efficient graphics hardware and, particularly, the standardization of 3D information format called Virtual Reality Modeling Language (VRML) and the development of VRML browsers such as the CosmoPlayer plug-in for web browsers.

To use 3D information in multimedia systems, 3D graphical models must be acquired. In the past, geometric modeling methods have been used successfully to create well structured models such as mechanical and architectural parts. It is, however, difficult to use these methods for acquiring complex natural scenes and objects. With the maturity of computer vision technology, there is now a trend towards reverse engineering the acquisition process by recovering the geometric and topological information from measurements of real scenes and objects.

We have implemented a framework for automated acquisition of 3D models of small natural objects for use in multimedia systems. The framework uses

Figure 1. The acquisition hardware platform consisting of a PC, a PC-controlled turntable, and a CCD camera.

a CCD camera to capture different views of an object placed on and rotated by a PC-controlled turntable (Fig. 1). It then applies a modified Kanade-Lucas-Tomasi tracker to math corresponding feature points in consecutive images. Next, 3D coordinates of matched 2D feature points are computed using an accurate and robust Ray Intersecting algorithm.[13] Finally, a novel polygonization algorithm is used to construct a triangle mesh of the recovered 3D points and a smoothing algorithm is applied to smooth the surfaces of the constructed mesh.

The first part of the system, including the hardware configuration and the Ray Intersecting algorithm for computing 3D points, has been presented in a previous paper.[13] This paper focuses on the second part: the reconstruction of the object's surfaces from the recovered 3D points. In this paper, a novel *frontier advancing* polygonization algorithm for constructing a triangle mesh from 3D points will be described (Section 3). After constructing the mesh, a distance-weighted Laplacian algorithm is applied to smooth the surfaces of the mesh to produce a more pleasing 3D model (Section 4). Finally, 3D acquisition results will be illustrated using VRML (Section 5).

2 Related Work

2.1 Model Acquisition

Virtual Reality (VR) techniques have an important application in multimedia systems. A recent example is the use of 3D VR theater in a multimedia presentation of "Mysteries of Egypt" at the Canadian Museum of Civilization.[14] The system presented 3D VR tours of the tombs of Nefertari and Tutankhamun, and computers are used to render scenes in real time using a series of viewpoints as the viewer walks through the scenes. In another example of multimedia systems,[10] a 3D agent is animated to describe the layout of the 3D objects in a virtual environment based on a speech text.

The success of such applications depends heavily on the 3D contents of the multimedia systems. Easy acquisition of 3D models would facilitates the development of such systems. Recent works on this topic focused on the acquisition and use of 3D human faces.[4,7,9,11] Among them, Kshirsagar et al.[11] integrated MPEG-4 compatible synthetic face and body into a real-time system which allows the user to interactively control the animation of the virtual human face.

In addition to modeling human faces, the acquisition of general object models is also important for multimedia systems. There are two major approaches to the acquisition of general 3D models: structured light and multiple image views. The first approach projects a laser beam onto an object and uses a camera to capture the surface contours for reconstructing the object's surfaces. Such laser scanning products[16,19] can produce accurate results but are quite expensive. The small, hand-held version[16] requires a lot of work to manually scan all the surfaces of an object. The second approach uses one or more cameras to capture multiple views of an object and uses the multiple views to recover 3D coordinates of the feature points on the object's surface.[15,18] Our method takes the second approach. In particular, it uses affordable equipments such as a turntable and a camera and it automatically acquires a 360° model of an object.

2.2 Polygonization of 3D Points

The abstract problem of polygonization is defined by Hoppe et al. as follows:[8] given a set of points in 3D space without any information about the structure or organization, construct a polygon mesh, possibly with boundary, from the 3D points. Unlike in the ideal case, the 3D points recovered from an image sequence are constrained by the features present in the images. These 3D points cannot satisfy the dense sampling requirement of the crust algorithm[1] and the uniform sampling requirement of the α-shape.[3] The zero-set algorithm[8] does not have specific requirements about the 3D points. However, it is a volume-based algorithm that produces an approximating mesh[1] whereas our algorithm produces an interpolating mesh.

Another major research problem is that the reconstruction of surface discontinuities such as edges and corners have not been solved by existing polygonization algorithms.[1] Our polygonization algorithm attempts to solve this problem by looking for possible surface discontinuities. It then constructs the mesh starting from relatively smooth and flat surfaces and working progressively towards surface discontinuities.

Most existing polygonization algorithms are based on Delaunay triangulation or, its dual, the Voronoi diagram. The Voronoi diagram is a versatile geometric structure that naturally produces small polygons in areas with densely distributed sample points and large polygons in coarsely sampled areas. Therefore, it is very suitable for polygonizing feature points recovered from images.

Several algorithms for Delaunay triangulation are well known. Green and Sibson devised an incremental algorithm that computes the Voronoi diagram [5] of Delaunay triangulation, of a set of points. Lawson developed an algorithm by flipping diagonals of triangles [12] and Guibas, Knuth, and Sharir presented an optimal implementation of Lawson's method based on randomized algorithm.[6] These algorithms, and most other randomized incremental algorithms in computational geometry, all work according to the principle of *structure maintaining*:[2] To add the next triangle, the algorithm first determines which part of the current structure has to be changed to resolve conflicts with the new triangle. Then, it updates the structure locally, removing the conflicting triangles and adding the new triangle.

The principle of structure maintaining is inefficient and inconvenient for our application. To handle surface discontinuities, our algorithm adopts the strategy of constructing the mesh starting at relatively smooth and flat surfaces and working progressively towards edges and corners. This strategy can be implemented more elegantly and efficiently by adopting a mesh construction process that only adds triangles and never removes triangles.

3 Frontier Advancing Polygonization

The frontier advancing polygonization algorithm consists of two main steps:

1. Identifying reliable points:
   Reliable points are 3D points that lie on relatively flat surfaces. 3D points that lie near surface discontinuities are called *ambiguous points*. The closer a point is to an edge or a corner, the larger is its ambiguity.

2. Advancing Mesh Frontier:
   A mesh is first constructed around a reliable point. Then, 3D points near the frontier of the mesh are added, and the process continues in increasing order of ambiguity.

These main steps are described in detail in the following subsections.

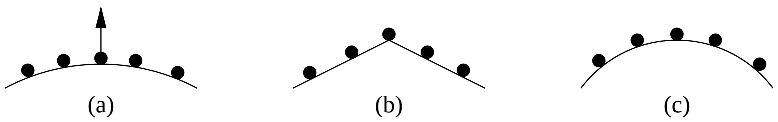


Figure 2. Identifying reliable points. (a) 3D points (dots) distributed over a relatively flat surface (line, viewed from the side) have a small third eigenvalue, whereas those lying on a sharp edge (b) or a surface with a large curvature (c) have large third eigenvalues. The arrow indicates the direction of the third eigenvector.

3.1 Identifying Reliable Points

Reliable points that lie on relatively flat surfaces can be identified using Principal Component Analysis (PCA). Given a set of 3D points, PCA computes the three eigenvectors $\mathbf{e}_i$ and eigenvalues λ_i, $i = 1, 2, 3$, in decreasing eigenvalue. If the set of points lie on a relatively flat surface, the third eigenvalue λ_3 would be very small (Fig. 2a), approaching the value 0 when the surface approaches a 3D plane. Conversely, for a set of points distributed near a surface discontinuity (Fig. 2b) or a surface with a large curvature (Fig. 2c), λ_3 would be large. Therefore, the third eigenvalue λ_3 can serve as a measure of the ambiguity of a 3D point, and reliable points can be identified as follows:

A1: Identifying Reliable Points:
For a point p,
  Find the neighbors of p within a sphere of radius r centered at p.
  Perform PCA on this set of 3D points.
  If $\lambda_3/\lambda_1 < \Gamma_a$,
    p is a reliable point.
  Else, p is an ambiguous point.

The radius r is varied adaptively to capture about 20 points in the sphere so that λ_3 can be estimated reliably. The threshold Γ_a can be a fixed parameter because it is independent of the sampling density of the 3D points. In the current implementation, it is fixed at 0.05.

3.2 Frontier Advancing Polygonization

After identifying reliable points, an initial mesh is first constructed around a randomly chosen reliable point, say, c. Essentially, the polygonization algorithm has to determine which of c's neighbors should be considered as *mesh points*, i.e., points to be connected to c to form the mesh triangles. It can be

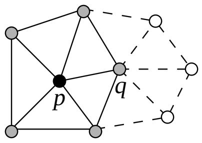


Figure 3. Mesh construction. The initial mesh (solid lines) is constructed around a reliable point p (black dot). The mesh points (gray dots) of p form the frontier of the initial mesh. Subsequently, the frontier is advanced by extending the mesh (dashed lines) and completing the mesh around the frontier point q. The process continues at the frontier points in increasing order of ambiguity of the frontier points.

proved that the algorithm constructs a Delaunay triangulation of 3D points *lying on a plane.*

First, the algorithm constructs a mesh around a sample point as follows:

A2: Constructing A Mesh Around A Point
Given a point c and its neighbors,
  Set p_0 as the nearest neighbor of c.
  Find the point p_1 whose bisector $L(c, p_1)$ intersects $L(c, p_0)$ at the location v_1 nearest to the mid-point between c and p_0.
  Repeat for $i \geq 1$,
    Find the point p_{i+1} whose bisector $L(c, p_{i+1})$ intersects $L(c, p_i)$ at the location v_{i+1} nearest to v_i.
  Until $p_n = p_0$.
  Connect the vertices $p_i, i = 1, \ldots, n$, to c to form a mesh.

Once the initial mesh is constructed, the polygonization algorithm advances the mesh frontier by extending the mesh around the frontier points (Fig. 3). The same algorithm A2 is used to construct the mesh around a frontier point except that neighbors that are already included in the mesh are retained and never removed. The algorithm just adds triangles to complete the mesh around a frontier point. This mesh extension process continues in increasing order of ambiguity of the frontier points.

The frontier advancing polygonization algorithm can now be summarized as follows:

A3: Advancing Mesh Frontier
While there are points not connected to the mesh,
  Construct an initial mesh around the least ambiguous point.

For each frontier point c in increasing order of ambiguity,
   Complete the mesh around c using algorithm A2.
   Include new frontier points in the list of frontier points, and
   exclude c from the list of frontier points.

Algorithm A3 constructs the mesh starting from reliable points and advancing the frontier towards ambiguous points. If 3D points along the edges and at the corners are sampled, the algorithm will form an edge at the meeting place of two advancing frontiers, and a corner at the place where more than two frontiers meet. This polygonization method thus reduces the error in constructing surface discontinuities.

4 Mesh Smoothing

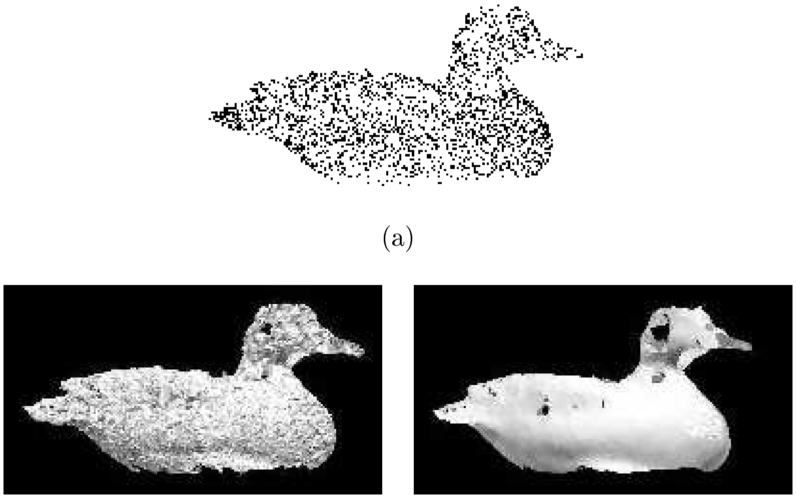
The mesh constructed from the recovered 3D points tends to have rough surfaces. A distance-weighted Laplacian algorithm [17] is used to smooth the mesh surfaces. Specifically, the 3D coordinate $\mathbf{x}$ of a point is iteratively updated by a distance-weighted average of the 3D coordinates $\mathbf{x}_i$ of its neighbors in the mesh:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{x}(t) + \lambda U(\mathbf{x}(t)) \\ U(\mathbf{x}(t)) &= \frac{1}{\sum_i w_i} \sum_i w_i \, \mathbf{x}_i(t) - \mathbf{x}(t) \end{aligned} \tag{1}$$

where t is the iterative time step, λ is a positive constant that determines the rate of update, and w_i is inversely proportional to the distance d_i between $\mathbf{x}$ and $\mathbf{x}_i$. The parameter λ determines the rate of convergence of the smoothing process. Empirical studies show that iterating the process 50 times with a λ value of 0.1 produces consistently good results.

5 Test Results

Tests were performed to assess the performance of the frontier advancing polygonization and smoothing algorithms. Figure 4 illustrates the mesh constructed from 3D data points recovered by our 3D model acquisition system. The general shape of the duck's body, head, neck, and wing are properly reconstructed. Some holes are observed on the model due to the lack of feature points on the original object's surface. The smoothing algorithm can even out the rough surfaces, turning them into smooth surfaces without disrupting important surface discontinuities.



(a) (b) (c)

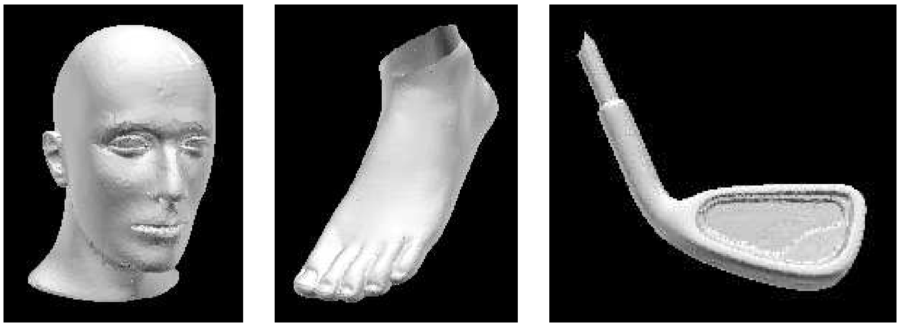
Figure 4. Reconstruction result. (a) 3D points recovered by the model acquisition system. (b) Mesh constructed by the frontier advancing polygonization algorithm. (c) Result of smoothing the mesh by the smoothing algorithm.

The polygonization algorithm was also tested on standard 3D data points available in the Microsoft Research web site research.microsoft.com/~hhoppe. Figure 5 shows that the algorithm can construct complete mesh with fine details even though the 3D points were sampled at different sampling rates at different parts of the models. The reconstructed groves on the golf club show that the algorithm can construct surface discontinuities.

Figure 6 illustrates an example of incorporating the reconstructed model into a VRML application. The background and the water in the scene are composed on 2D images. Three instances of the duck model with different sizes were created in VRML. The surfaces of the ducks were painted with appropriate colors.

6 Conclusion

This paper presented a system for the automated acquisition of 3D models of objects. In particular, it focused on a novel frontier advancing polygonization algorithm for constructing meshes the 3D points recovered by the system. In



(a) (b) (c)

Figure 5. Mesh construction results of standard 3D points. Details such as th e the eyes, ear, nose, and lips of the head, the toes of the foot, and the groves of the golf club are accurately constructed.

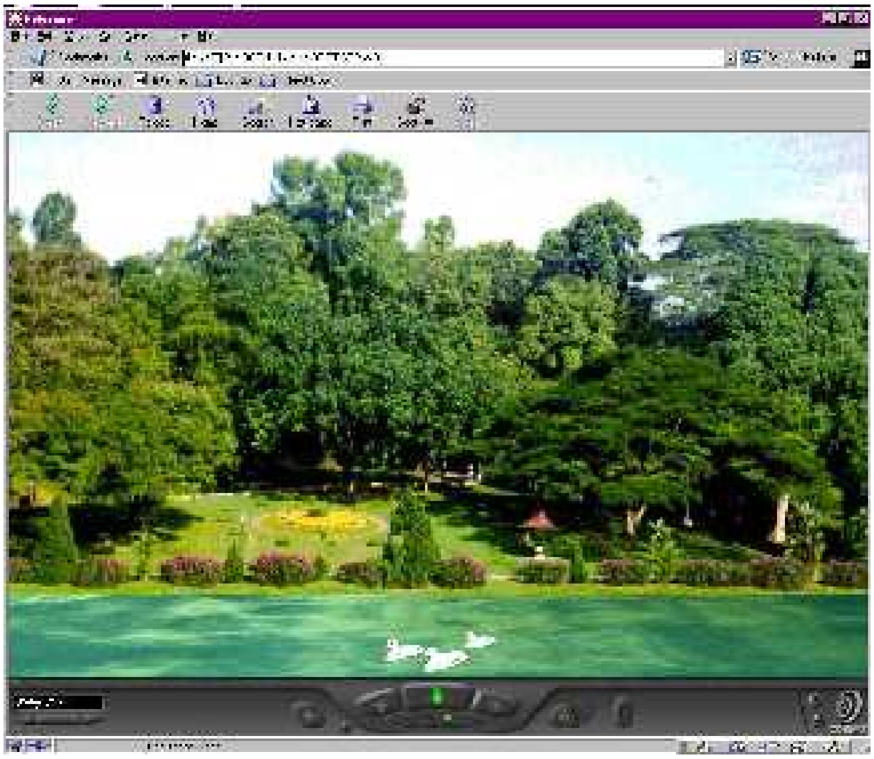
contrast to existing polygonization algorithms, the new method constructs a mesh starting at relatively smooth and flat surfaces and advances the mesh frontier towards surface discontinuities. Moreover, it only adds triangles to the mesh and never removes triangles. Test results show that the method can construct meshes of 3D points well. A VRML example that includes a reconstructed model is also illustrated.

Acknowledgments

This research is supported by NUS Academic Research Fund RP3982704.

References

1. N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH '98*, pp. 415-421, 1998.
2. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications.* Springer-Verlag, pp. 181-207, 1997.
3. H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. on Information Theory*, 29, pp. 551-559, 1983.
4. M. Gotla and Z. Huang. A Minimalist Approach to facial Reconstruction. In *Proc. MMM '99*, pp. 377-387, 1999.



(a)

Figure 6. (a) A VRML example of a garden scene with three instances of the recovered duck model. (b) A closer view of the ducks.

5. P. J. Green and R. R. Sibson. Computing Dirichlet tessellations in the plane. *Comp. J.*, pp. 168-173, 1978.
6. L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 2, pp. 381-413, 1992.
7. H. F. Heimbrecht and G. Trogemann. Modeling Facial Expressions by Interactive Evolution. In *Proc. MMM '99*, pp. 351-366, 1999.
8. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH '92*, pp. 71-78, 1992.
9. X Jiang, Q. Qiang, and J. Zhu. 3-D Human Face Reconstruction and Animation for Chinese Speech Training in Oral Deaf Education. In *Proc. MMM '99*, pp. 367-376, 1999.
10. K. Kakizaki. Generating the Animation of a 3D Agent from Explanatory Text. *Proc. ACM Multimedia '98*, pp. 139-144, 1998.
11. S. Kshirsagar, M. Escher, G. Sannier, and N. Magnenat-Thalmann. Multimodel Animation System Based on the MPEG-4 Standard. In *Proc. MMM '99*, pp. 215-232, 1999.
12. C. L. Lawson. Transforming triangulation. *Discrete Math.*, 3, pp. 365-372, 1972.
13. W. K. Leow, Z. Huang, Y. Zhang, and R. Setiono. Rapid 3D model acquisition from images of small objects. In *Proc. Geometric Modeling and Processing 2000*, pp. 33-44, 2000.
14. G. MacDonald and F. Corcoran. CMC's 3D Virtual Reality Theatre VR Tours of Two Tombs During the Mysteries of Egypt Exhibition. In *Proc. MMM '99*, pp. 167-177, 1999.
15. Y. Matsumoto, H. Terasaki, K. Sugimoto, and T. Arakawa. A portable three-dimensional digitizer. In *Proc. of Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, pages 197–204, 1997.
16. ModelMaker. 3D Scanners Ltd, UK. http://www.3dscanners.com.
17. Y. Ohtake, A. G. Belyaev, I. A. Bogaevski. Polyhedral Surface Smoothing with Simultaneous Mesh Regularization. In *Proc. GMP 2000*, pp. 229-237, 2000.
18. ReCaSs. Fraunhofer Institute for Computer Graphics (IGD), Germany. http://www.igd.fhg.de/projects/index.html.
19. REPLICA. 3D Scanners Ltd, UK. http://www.3dscanners.com.