A Low-Cost Model Acquisition System for Computer Graphics Applications

Minh Tran, Amitava Datta, and Nick Lowe

School of Computer Science & Software Engineering
University of Western Australia
Perth, WA 6009
Australia
{tranm03,datta,nickl}@csse.uwa.edu.au

Abstract. Most 3D objects in computer graphics are represented as polygonal mesh models. Though techniques like image-based rendering are gaining popularity, a vast majority of applications in computer graphics and animation use such polygonal meshes for representing and rendering 3D objects. High quality mesh models are usually generated through 3D laser scanning techniques. However, even the inexpensive laser scanners cost tens of thousands of dollars and it is difficult for researchers in computer graphics to buy such systems just for model acquisition. In this paper, we describe a simple model acquisition system built from web cams or digital cameras. This low-cost system gives researchers an opportunity to capture and experiment with reasonably good quality 3D models. Our system uses standard techniques from computer vision and computational geometry to build 3D models.

1 Introduction

Polygonal mesh models are widely used in computer graphics for representing and rendering complex 3D objects. The surface of a 3D object is usually represented as a triangulated mesh in such models. While most users and researchers in computer graphics routinely use such models, quite often it is difficult for them to acquire their own models according to their specific requirements. The most popular model acquisition system is a 3D laser scanner which is usually an expensive device. Even an inexpensive laser scanner may cost tens of thousands of dollars. Hence, it is difficult for researchers in computer graphics to buy such systems for model acquisition. Most researchers depend on the models available from a few research labs such as the Stanford 3D scanning repository [4] and Georgia Tech large model archive [6] where high quality models have been acquired through laser scanning. However, sometime this is too restrictive since the researchers do not have the freedom to experiment with specific models according to their requirements. In many cases, it is necessary to experiment with models with specific topological features for designing efficient data structures and techniques in computer graphics.

In this paper, we discuss a simple model acquisition system built from web cams and digital cameras that can be used as a low-cost alternative for model acquisition. This low-cost system gives researchers an opportunity to capture and experiment with reasonably

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2659, pp. 1054–1063, 2003.

good quality 3D models. Our system employs standard techniques from computer vision and computational geometry to build 3D models.

A major research area within computer vision is stereo reconstruction from images taken from monocular (multiple images with a single camera) or polynocular views (single images with multiple cameras) [3]. Three-dimensional (3D) reconstruction from multiple images attempts to simulate human perception of our 3D world from disparate two-dimensional (2D) images. A realistic representation of objects from camera images can provide an inexpensive means of object or scene modeling compared to specialized hardware such as laser scanners.

Automatic object modeling has well-known applications including the construction of 3D polygonal models for computer graphics applications, which is the focus of this paper. Once a camera captures an object, by identifying and exploiting specific scene information in the image, we can retrieve the depth, i.e., the third dimension. Next, we produce a cloud of 3D points by locating the depth values for various points of interest. An appropriate visualization technique is then employed to establish connectivity between the unstructured point cloud representing the object surface. The quality of object surface representation is dependent on the accuracy of depth retrieval. It is possible to reconstruct an object from a single image, but having multiple images of a scene improves the accuracy at which depth of object points can be determined. Further, matching identical points between images relies on the exploitation of image information including camera parameters, intensity, edge pixels, lines and regions. Analyzing and identifying areas or features, whose characteristics are preserved among disparate views, can establish point correspondences between successive images. Camera parameters give information regarding the transformation or projection of the object from the 3D world onto a 2D image. Its knowledge improves the accuracy and efficiency of point matching and can be determined before or during the matching process. In our system, the task of 3D reconstruction can be divided into roughly four stages: *camera calibration, correspondence, recovery of depth,* and *visual representation.*

The rest of the paper is organized as follows. We describe our methodology in Section 2. The results obtained from our system are discussed in Section 3 along with an example. Finally we conclude with some remarks and possible future work in Section 4.

2 Our Methodology

In this section, we discuss the techniques we have used for implementing the model acquisition system. Most of these techniques are taken from the existing computer vision and stereo vision literature with modifications to suit our needs. For standard techniques in stereo vision, see the book by Hartley and Zisserman [3]. The stereo reconstruction problem has been discussed in several papers [1,5,2,7]. In particular, Beardsley *et al.* [1] and Mandal *et al.* [5] discuss the problem of extracting a 3D model from multiple stereo images. They use multiple cameras and focus mainly on reconstructing outdoor scenes. In this paper, we concentrate on extracting 3D models for indoor objects. Hence, our techniques are considerably simpler than those in [1,5] since the images are limited to local objects captured in front of a plain background.

When a camera captures an object, a projective transformation from 3D space to 2D takes place. A typical model of the camera is displayed in Figure 1(a). In Figure 1(a), an object point in 3D space, (X, Y, Z), is mapped onto the image plane according to the line joining (X, Y, Z) and the camera's center of projection, P. The image point $\mathbf{p}$, coincides with the position where this ray intersects the image plane, (x, y) in Figure 1(a).

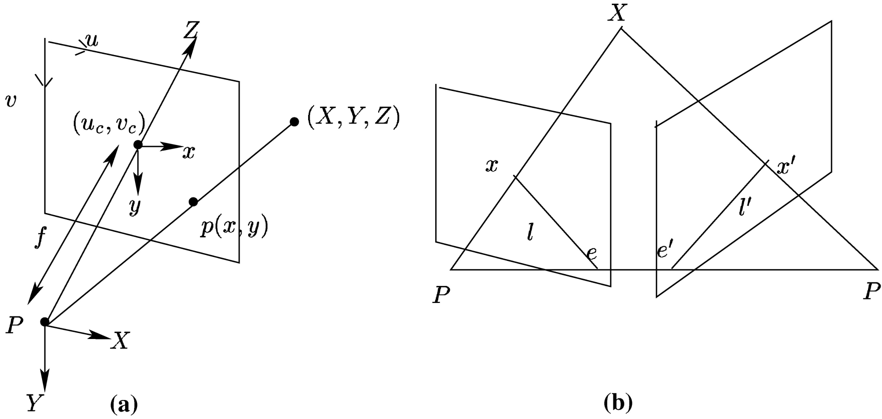

Fig. 1. (a) This is a typical camera model where a point of an object in 3D space, (X, Y, Z), is projected onto a viewing plane, $\mathbf{p}$, by a viewing ray that converges at the center of projection, P. **(b)** Two views of an object possess a unique geometric relationship expressed by the epipolar constraint. The plane formed by the two center of projections and the 3D object point of interest is called the epipolar plane. Where this plane intersects with the viewing planes are the images' respective epipolar lines.

The coordinates of the point $p(x, y)$ can be found through the simple geometric calculation. $x = \frac{Xf}{Z}$ and $y = \frac{Yf}{Z}$, where f is the focal length of the camera. The image coordinates u, v of p can be described in terms of x and y such that $u = u_c + \frac{x}{p_w}$ and $v = v_c + \frac{y}{p_h}$, where p_w is pixel width, p_h is pixel height, and (u_c, v_c) represents the principle point formed by the perpendicular projection of the camera center, P, onto the image plane. The mapping of the object from 3D space to 2D can be expressed in homogeneous coordinates as follows. $s[x, y, 1]^T = K[X, Y, Z, 1]^T$, where the intrinsic camera calibration matrix K can be represented as

$$K = \begin{bmatrix} \frac{f}{p_w} & 0 & 0\,0 \\ 0 & \frac{f}{p_h} & 0\,0 \\ u_c & v_c & 1\,0 \end{bmatrix} \quad (1)$$

and s is an arbitrary scalar value. The images acquired in our experiments are assumed to have negligible lens distortion therefore no rectification techniques were applied. The focal point P is located at a distance perpendicular to the image center i.e., principle point, (u_c, v_c). This distance is known as the focal length f. The principle point, focal length and P are collectively referred to as the intrinsic parameters of the camera along with effective pixel size and any radial distortion coefficient of the lens. Intrinsic parameters are characterized by their invariance to the camera's geographic position and orientation.

On the other hand, extrinsic parameters are dependent on the camera's orientation relative to the world reference frame and consist of a rotation matrix and a translation vector. Object points such as (X, Y, Z) in Figure 1(a) are described with respect to some world reference frame. These points must be transformed to coincide with the camera axes made possible by a rotation followed by a translation. The transformation of a 3D point to a 2D coordinate can now be represented as $\mathbf{x} = K\ [R|\mathbf{t}]\ \mathbf{X}$ where K represents the intrinsic camera parameters, R is a $4{\times}3$ rotation matrix, and $\mathbf{t}$ a $4{\times}1$ translation vector. The relationship between the projections of the object from the world coordinates into image coordinates can be found in the camera parameters, which can be solved via various camera calibration techniques. Cameras can be actively calibrated using calibration targets or passively from image correspondences.

2.1 Camera Calibration

A typical calibration method uses a calibration target. Depth is inferred by finding image points and using the known camera parameters to solve for (X, Y, Z). Camera calibration requires the known 3D positions of certain known pixels on the calibration target, and their respective image coordinates to solve linearly for C, the $3{\times}3$ camera calibration matrix composed of $K[R|\mathbf{t}]$. In this paper, the cameras are calibrated actively using a calibration target. This enables a metric reconstruction and also the recovery of epipolar geometry.

The retrieval of camera parameters unlocks the epipolar geometry between two images, which can be exploited during the matching process. The relationship between two images and the 3D object is illustrated in Figure 1(b). Image epipoles, $\mathbf{e}$ and $\mathbf{e}'$, are located at the intersection of the baseline joining the cameras' optical centers with the image plane. An epipolar plane is formed by the cameras' center of projection and the 3D point of interest, $\mathbf{X}$ in Figure 1(b). The epipolar line is the intersection of this plane with the image plane. From Figure 1(b), it is clear that a point in one image corresponding to a point in 3D space has its matching point contained in the epipolar line in the second image. Therefore, the epipolar constraint not only reduces the match search from the image area to a line, but also improves the robustness of the matching process. However, in 3D reconstruction, the value $\mathbf{X}$ is unknown and it is what we are trying to determine. We identify the epipolar line through the fundamental matrix described by Hartley and Zisserman [3] as the algebraic representation of epipolar geometry.

2.2 The Fundamental Matrix

The fundamental matrix encapsulates the relationship between image points in a stereo pair. This relationship is represented in the equation $\mathbf{x}'^T F\mathbf{x} = 0$ where $\mathbf{x}'$ is the correspondence of $\mathbf{x}$. In accordance with projective geometry, the dot product between a point $\mathbf{x}'$ located on a line $\mathbf{l}'$ and $\mathbf{l}'$ is 0. Thus, $\mathbf{x}'.\mathbf{l}' = \mathbf{x}'^T\mathbf{l}' = \mathbf{x}'^T F\mathbf{x} = 0$ resulting in $\mathbf{l}' = F\mathbf{x}$ where $\mathbf{l}'$ is the epipolar line. Conversely, $\mathbf{l} = \mathrm{F}^T\mathbf{x}'$. Therefore, the epipolar line in the second image, $\mathbf{l}'$, corresponding to a point in the first image, $\mathbf{x}$, can be identified through the fundamental matrix, F and vice versa.

The camera projects a point in 3D space into 2D image coordinates. The fundamental matrix can be described in terms of camera matrices. In projective geometry, the cross product of two points returns a line containing the two points. Therefore, $\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_x \mathbf{x}$ where $\mathbf{e}'$ is the epipole and $[\mathbf{e}']_x$ is a 3×3 skew-symmetric matrix of $\mathbf{e}'$. Let $\mathbf{e}' = (e_1, e_2, e_3)$, then $[\mathbf{e}']_x$ is defined as:

$$[\mathbf{e}']_x = \begin{bmatrix} 0 & -e3 & e2 \\ e3 & 0 & -e1 \\ -e2 & e1 & 0 \end{bmatrix} . \tag{2}$$

The cross product of the two vectors $\mathbf{e}'$ and $\mathbf{x}'$, can be expressed as $\mathbf{e}' \times \mathbf{x}'$ = $[\mathbf{e}']_x \mathbf{x}'$ = $(\mathbf{e}'^T [\mathbf{x}']_x)^T$. If H represents the homography, mapping $\mathbf{x}$ with $\mathbf{x}'$ such that $\mathbf{x}' = H\mathbf{x}$, we obtain the following relationship. $\mathbf{l}' = [\mathbf{e}']_x H\mathbf{x} = F\mathbf{x}$, and thus $F = [\mathbf{e}']_x H$. The fundamental matrix can be obtained by identifying point correspondences ($\mathbf{x}' F \mathbf{x} = 0$) or from camera parameters ($F = [\mathbf{e}'] C' C^+$). Since F will be used in the matching process, that is to find $\mathbf{x}$ and $\mathbf{x}'$, F is derived from camera matrices found from image calibration.

2.3 Point Matching

Exploitation of epipolar geometry, through the estimation of the fundamental matrix (ideally) enables robust establishment of point correspondences whilst improving the efficiency of each search. A stereo pair of pixels enables the recovery of depth through triangulation. Two popular approaches to stereo matching are the intensity-based and feature-based methods. Typically, feature-based methods are robust to significant change of viewpoints and depth disparity, but are ineffective in matching areas of smooth changing intensity values. On the other hand, intensity-based methods work well in textured areas, but large depth discontinuity along with change of viewpoint can lead to mismatches. Since depth values of objects typically do not vary drastically and in the hope that dense matches will improve structure reconstruction, we use a cross-correlation intensity-based matching method to locate point correspondences.

Cross correlating matches between image pairs improves the accuracy of matches identified. Pixels are matched according to the similarity or correlation between neighborhood intensity values. To improve accuracy of the matching process, once a match is found in the second image, a corresponding match is searched for in the first image (hence the name cross-correlation). If the match identified in the first image is different from the initial pixel, the match is rejected. This was a common occurrence when the algorithm attempted to match featureless areas in our experiments. The user determines the neighborhood size or window size, centered at the point of interest. Region W_1 in image 1 is matched with a region W_2 in image 2 according to their correlation coefficient given by the following equation:

$$c(W_1, W_2) = \frac{2cov(W_1, W_2)}{var(W_1) + var(W_2)} . \tag{3}$$

This equation is referred to by Sara [8] as the modified normalized cross-correlation algorithm, which has an advantage of tending to zero when similarly textured areas have different contrasts. The correlation coefficient $c(W_1, W_2)$ ranges between 0 and 1. The higher the coefficient the higher the correlation between the two regions W_1 and W_2.

To preserve the structure of the object surface, edge pixels are matched first and then point correspondences between image pairs are searched for at regular intervals. The search is conducted along the pixel's corresponding epipolar line to improve the accuracy and efficiency of each matching attempt. Accepting matches above a certain threshold and imposing the similarity constraint where a pixel and its match will have similar intensity values further improves the accuracy of the matching process. All image processing was conducted with grey image values. Therefore, all of the images were converted to grey scale before processing. Since the search was conducted along the epipolar line, if a match was found, it would be restricted to lie within that line. Therefore, the two rays back-projecting from the respective camera centers lie on the epipolar plane (Figure 1(b)) and will intersect at a point in 3D space.

2.4 Depth Inference

Since a correspondence pair back-project to the same point in 3D space, its depth can be approximated through geometric triangulation. Linear inference of depth is employed since metric reconstruction is assumed. The relationship between the common (X, Y, Z) point in 3D space and the point correspondences (x_1, y_1) and (x_2, y_2) is expressed homogeneously as:

$$s_i[x_i, y_i, 1]^T = C_i[X, Y, Z, 1]^T \; for \; i = 1, 2 \tag{4}$$

where C_1, C_2 and s_1, s_2 are the camera calibration matrices and scalar factor of image 1 and image 2 respectively. The camera calibration matrix C is a 3×4 matrix. The (X, Y, Z) values can be obtained by substituting C_1 and C_2 (the two calibration matrices) in Equation 4 and solving. The resulting structure is a cloud of points in the $3D$ space representing the surface of the object.

We use Delauney triangulation for generating a triangular mesh from this cloud of points. A $3D$ Delauney triangulation satisfies the following two conditions :

- three points are connected to form a triangle such that no neighboring points can be contained within or on the circle circumscribed by the vertex points,
- the outer edges of the union of all triangles form a convex hull.

3 Results

We now discuss the results obtained by our low-cost acquisition system. Image noise is inevitably introduced through the digitization of continuous scene information during the image acquisition process. Matching images taken from a 1600×1200 digital camera produced denser matches compared to several 640×480 web cameras, given the same matching parameters (i.e. window size, edge detection threshold, number of images matched and correlation co-efficient threshold).

3.1 Point Matching

Finding point correspondences between images is by far the most significant bottleneck in the reconstruction process. Attempting to match four 640×480 pictures using epipolar and similarity constraints, edge detection threshold of 0.1, window size of 10, and cross-correlation coefficient of 0.8, matching every fifth pixel can take hours to complete. For the web cameras, the object was taken with a uniform or featureless background. Since an intensity-based matching method was employed, featureless areas were not matched. However, the background was not completely featureless and non-object matches were found. Therefore, we found it necessary to find a 'mask' corresponding to the object for eliminating the background.

A window-based approach possesses some deficiencies when attempting to match depth discontinuities and textureless areas. Mismatches occur where the change in view has changed the neighborhood of the corresponding pixel in other images. A typical instance of this is where the pixel cannot be seen in other images because it is blocked by another part of the scene. Particularly, if the pixel corresponds with depth discontinuities in the image, the view disparity can change the neighborhood intensity values or occlude the pixel of interest. It is clear that area-based matching would fail if many pixels along the epipolar line have neighborhoods of the same intensities. Textureless and patterned areas cause this problem of ambiguity but are combated by the cross-correlation approach.

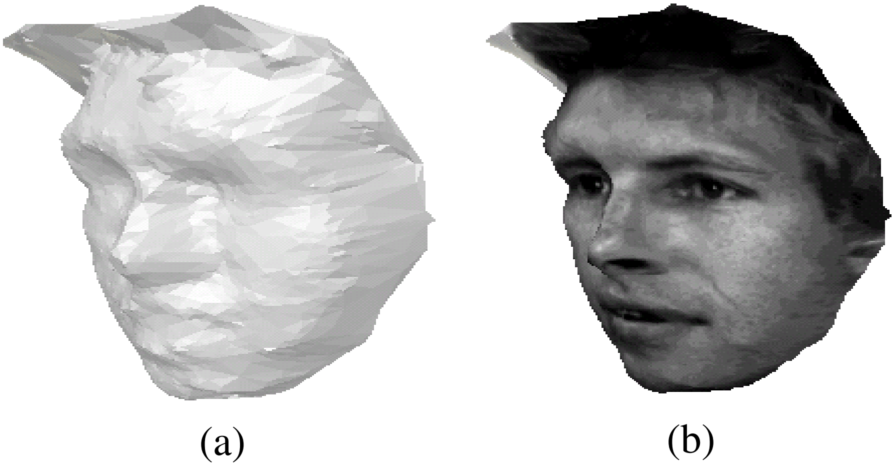

Fig. 2. (a) The recovered structure of the face from Figure 3. The face is shown with lighting and shading. **(b)** Surface details can be enhanced by texture mapping the image onto the retrieved surface. We have used the original image in Figure 3(a) as the texture.

Mismatches can be prevented during the matching phase of the reconstruction process. This can be achieved by setting the correlation coefficient to a high value in conjunction with epipolar constraints. In our experiments, there was a high correlation between matching accuracy and accurate matches. Having a high correlation coefficient threshold prevents the occurrence of false positives but may also reject accurate matches. This is deemed appropriate, since the depth retrieval step is sensitive to errors.

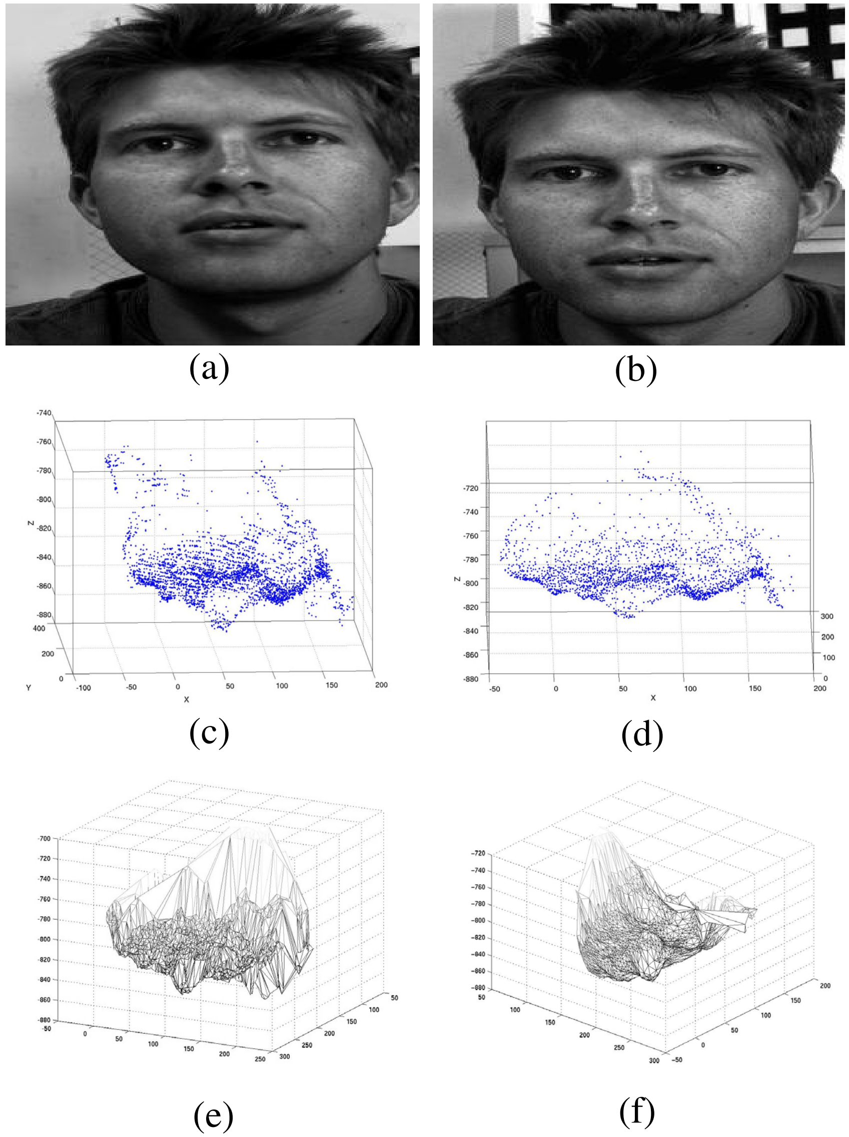

Fig. 3. Results from our model acquisition system. **(a),(b)** The stereo image pairs used in one of our reconstructions. **(c)** The initial point cloud obtained after matching. After applying a median filter over the face data points, the outliers located in front of the face are adjusted to the facial surface. **(d)** The point cloud after applying median and averaging filters. The resulting point cloud represents a smooth surface with little or no outliers. **(e)** The recovered structure after triangulating point cloud in (c). **(e)** The recovered structure after triangulating the point cloud in (d). All images are reproduced with permission from Frédéric Devernay.

3.2 Depth Inference

Even though the matches obtained are ‘roughly’ accurate, mismatches still occur and the estimated 3D point array contains outliers. Typically the 3D points reflect the general shape of object, but this is much maligned by a litter of outliers. Since object points congregate in a localized cluster, outliers can have large distances between themselves and their nearest neighbor and are sparsely populated relative to object point density. Therefore points that are far away from the majority of the points (assumed to be object points) can be identified and removed from the data set. In our implementation, points with the mean distance of its 11 or so nearest neighbors that deviate from the average distance by more than the standard deviation are removed.

Outliers occur due to erroneous depth estimates from inaccurate matches, or matches identified that do not correspond to the local object being reconstructed (non-object matches). Outliers resulting from matches often produce spurious results when triangulating 3D points. This can result in recovered object points being squashed into a sheet of 3D points rather than a polygonal mesh. After the removal of spurious outliers, 3D data points that lie near but not on the object surface still exist. Applying a median filter can adjust these points back to the object surface without losing the underlying object structure. The results of median-filtering the facial data points, using a neighborhood of the 20 or so nearest points, is displayed in Figure 3(c). Further, applying both a median and an average filter improves the quality of the point cloud considerably as shown in Figure 3(d).

3.3 Data Visualization

Data visualisation aims to establish connectivity between an unstructured set of 3D points. Implementing 2D Delauney triangulation is simpler than the 3D approach especially if the data points are aligned with the axis. Otherwise, the data set would need to be rotated by an appropriate angle to align the depth axis with one of the axes in the world reference frame. If the object points are not rotated so that the depth values are aligned with a world reference frame axis, 2D triangulation can result in a ‘spiky’ object structure or a skewed surface representation.

Triangulating the points in 2D can have a jagged effect once returned to 3D. This occurs when neighboring points in 2D are triangulated have large discrepancies in depth values. The resulting topology, looking along the depth axis, clearly represents the object structure. But once turned on its side is populated with a series of peaks and troughs. This is illustrated in Figures 3(e) and 3(f). The resulting 3D point set can be further smoothed by an averaging filter applied to all of the points with the filter the size of its connected neighbors.

4 Conclusion

We have implemented a system for low-cost model construction using inexpensive web cams and digital cameras. We have experimented with many objects extensively and one of our reconstructions is shown in Figure 3. The triangulated mesh in Figure 3 has several thousand triangles which is quite good for computer graphics applications. The resulting models can be used for computer graphics applications as shown in Figure 2. The reconstructed model with shading, lighting and texture mapping is shown in Figure 2.

Currently, our system uses only two cameras. As a result, it is not possible to acquire a complete 3D model. We plan to extend the system so that we can place the object within a circular array of web cams or digital cameras. We plan to build the model piecewise from pairs of cameras and then reconstruct a complete model from these partial models.

Acknowledgments. The research of the second and third authors is supported by Western Australian Interactive Virtual Environments Centre (IVEC) and Australian Partnership in Advanced Computing (APAC).

References

1. P.A. Beardsley, P.H.S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *ECCV (2)*, pages 683–695, 1996.
2. G. Farneback. The stereo problem. Technical report, Computer Vision Laboratory, Linköping University, February 2001.
3. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
4. Stanford Graphics Laboratory. The stanford 3d scanning repository. `http://graphics.stanford.edu/data/3Dscanrep/`.
5. C. Mandal, H. Zhao, B. C. Vemuri, and J.K. Aggarwal. 3D shape reconstruction from multiple views. Available: `http://citeseer.nj.nec.com/149335.html`.
6. Georgia Institute of Technology. Georgia tech large model archive. `http://www.cc.gatech.edu/projects/large_models/`.
7. Marc Pollefeys. 3D modeling from images. `http://www.esat.kuleuven.ac.be/~pollefey/SMILE2/tutorial.html`, June 2000.
8. R. Sara. Accurate natural surface reconstruction from polynocular stereo. In F. Solina A. Leonardis and R. Bajcsy, editors, *Proceedings NATO Advanced Research Workshop Confluence of Computer Vision an Computer Graphics*, number 84 in 3, pages 69–86. Kluwer Academic Publishers, 2000.