

Free-Form Surface Reconstruction from Multiple Images

Chang Shu

Gerhard Roth

Institute for Information Technology
National Research Council Canada
Montreal Road, Building M50
Ottawa, Ontario, Canada K1A 0R6
{chang.shu, gerhard.roth}@nrc-cnrc.gc.ca

Abstract

It is widely appreciated that 3D structures may be computed from multiple 2D images of the same scene given point correspondences between images. Of greater interest, however, is the generation of surfaces that give a compact representation of the geometric model. Assuming we are dealing with smooth surfaces, we show that B-spline is a good choice for this purpose and we describe how to construct it by approximating the 3D data points. The crucial step is the parameterization of the 3D points in a 2D domain. By studying the geometric constraints of multiple views, we show that the original images can be used for parameterization. The implications of the B-spline surfaces for improving the quality of texture mapping is discussed.

1 Introduction

Reconstructing surfaces in 3-space from two-dimensional images is a central problem in computer vision and photogrammetry. Many applications require surface reconstruction. For example, reverse engineering of CAD models from existing objects, virtual and augmented reality, and computer animation.

The theory of 3D structure reconstruction is now well developed for discrete points with known correspondences and camera calibration parameters. However, most applications require a surface representation, particularly in the case where only sparse 3D points are available. Several special cases have been considered. Cross and Zisserman [2] constructs quadric surfaces. Commercial software, for instances, PhotoModeler and ShapeCapture, which are based on photogrammetry techniques [17], construct conic surfaces and spheres. For any non-standard surface types, piecewise linear surfaces are used. This means a large num-

ber of polygons are needed to approximate a smooth surface well.

Cipolla and Giblin [1] and Zhao and Mohr [22] studied surface reconstruction using profiles (apparent contours). The usual set up is to put the object to be constructed on a turntable and use a fixed camera to take a series of images. The advantage of this method is that it can deal with general smooth surfaces with no textures. The drawback is that certain concave part of the surface cannot be recovered.

Sullivan and Ponce [18] fit triangular B-splines to a crude polyhedral model obtained from silhouette based reconstruction methods. The problem of this method is that many surface patches have to be used to approximate a complex surface.

Sinha and Schunck [16] treats the stereo data fitting problem as a large scale optimization problem. Without solving the parameterization problem, the algorithm is inefficient and can fail to converge.

Non-analytical models were also used for representing stereo data. Fua [5] proposed to use particle system to approximate stereo data. Fua and LeClerc [6] and Zhang and Seitz [20] deform a mesh model to fit stereo data. The advantage of these methods is that they work with uncalibrated images. They both solve a nonlinear optimization problem. When the number of points to fit is large, the algorithm is slow and there is no guarantee of convergence.

In this work we investigate the reconstruction of general free-form surfaces. By free-form, we mean surfaces that cannot be represented by a single simple algebraic function. Examples of free-form surfaces abound in the physical world, both natural and man-made objects; for examples human and animal bodies, sculptures, and shapes of automobiles and airplanes.

We consider the situation where users take a small number of images from arbitrary positions. Then a number of feature points in the images are identified and matched across the images by either user assisted or automatic track-

ing methods. Finally, a 3D surface representation is constructed and texture mapped. Figure 1 illustrates this process. Our objective here is to enhance the current model building tools to handle more general surface types.

In section 2 we discuss reconstruction of discrete points using a projective geometry formulation. In section 3 we develop a method for fitting B-spline surfaces on the reconstructed 3D points. In particular, we show that the images and corresponding image points can be used for computing parameters in the fitting process. In section 6 we discuss texture mapping techniques for smooth B-spline surfaces. We demonstrate the improvements in the quality of texture mapping resulting from the B-spline surface fitting. Finally, concluding remarks are given in section 7.

2 3D point reconstruction

Let $\mathbf{X} = [X Y Z 1]^\top$ be the homogeneous coordinate of a point in the 3D world and $\mathbf{x} = [u v 1]^\top$ be the homogeneous coordinate of the corresponding image point. Under the assumption of a pinhole camera model, the relationship between \mathbf{X} and its projection on the image plane is

$$\gamma \mathbf{x} = \mathbf{K}[\mathbf{R} \ \mathbf{t}]\mathbf{X} \quad (1)$$

where γ is a non-zero scale factor, \mathbf{R} is a rotation matrix, and \mathbf{t} is a translation vector. The 3×3 matrix \mathbf{K} contains camera intrinsic parameters

$$\mathbf{K} = \begin{bmatrix} f & s & u_0 \\ 0 & \alpha f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length, α is the aspect ratio, s is the skew factor, and (u_0, v_0) is the position of the principal point on the image plane. Equation 1 is often written in a compact form $\mathbf{x} = \mathbf{P}\mathbf{X}$, where $\mathbf{P} = \mathbf{K}[\mathbf{R} \ \mathbf{t}]$ is a 3×4 perspective projection matrix, also known as camera matrix.

For two views, any two points $\mathbf{x} = \mathbf{P}_1\mathbf{X}$ and $\mathbf{x}' = \mathbf{P}_2\mathbf{X}$ in the two images corresponding to a 3D point \mathbf{X} satisfy the bilinear (epipolar) constraint

$$\mathbf{x}'^\top \mathbf{F} \mathbf{x} = 0$$

where \mathbf{F} is the 3×3 *fundamental matrix*. This matrix can be computed with 8 or more corresponding points [7].

For two images of a planar object, the corresponding points in the two images are related by a linear form, called *homography*:

$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

where \mathbf{H} is a 3×3 matrix. A homography matrix can be estimated using 4 or more corresponding points [8].

From equation 1 we can see that the projection of a 3D point to an image point depends on the camera intrinsic parameters in \mathbf{K} and the position and orientation, \mathbf{R} and \mathbf{t} , of

the camera, called extrinsic parameters. Once these parameters are determined, the camera and its images are called *calibrated*.

A point in a calibrated image back projects to a ray in the three-dimensional space. Two corresponding points in two separate images define two rays that lie in the same plane. Intersecting these two rays results in the reconstruction of a 3D point.

In this work, we carry out the task of point reconstruction in three steps. First, we compute the intrinsic parameters of the camera. Second, we find out the rotation matrix and the translation vector. Third, we compute the ray intersections to obtain 3D points.

In the ideal situation, the user calibrates the camera's intrinsic parameters using a calibration pattern before taking the images of the scene to be reconstructed. Several pattern-based calibration methods have been proposed, for example [9, 19]. We used a recent method suggested by Zhang [21] with a planar checkerboard pattern.

In the case that we do not have the luxury of having the user calibrate the camera beforehand, it is still possible to compute 3D structures from the images, using a technique known as *autocalibration*. In our implementation, we used a method due to Mendonça and Cipolla [12]. This calibration method uses only image point correspondences from which the fundamental matrices are estimated. Comparing to the bundle adjustment methods [17], it takes less time to compute because only a few parameters are involved. In practice, we assume the skew factor to be zero and the principal point is at the center of the image. When more than two views are used, satisfactory results are obtained [15]. We usually get within 10% of the calibration parameter values obtained from using pattern-based calibration methods.

3 B-spline surface fitting

In this section we show that it is possible to represent a smooth surface by reconstructing only a few sparse points on it. This is achieved by fitting a B-spline surface on the reconstructed sample points. This capability is useful to interactive model building systems in which the user has to manually identify feature points and match them in different images. On the other hand, reconstructions resulting from dense stereo may contain too many points. In this case, B-spline fitting provides a compact representation of the surface.

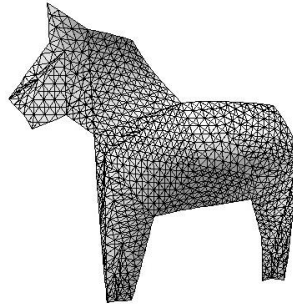
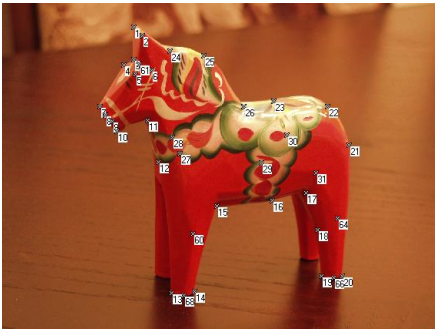
A B-spline surface is the tensor product function given by

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_i^p(u) N_j^q(v) P_{i,j} \quad (2)$$

where $P_{i,j}$ is an array of $(n+1) \times (m+1)$ control points and $N_i^p(u)$ is the i th B-spline basis function of degree p ,



(a)



(b)

Figure 1. Surface reconstruction process

defined recursively as

$$N_i^0(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_i^p(u) = \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u)$$

on a knot vector

$$U = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1}\}.$$

$N_j^q(v)$ is defined analogously [13].

Let Q_1, \dots, Q_N be the reconstructed 3D points. Assume uniform knot sequences, for any fixed degree p and q , we solve the least square problem

$$\min_{P_{i,j}} \sum_{i=1}^N \|Q_i - S(u_i, v_i)\|^2$$

for the control points $P_{i,j}$. Here $(u_1, v_1), \dots, (u_N, v_N)$ are the parameterization of the data points. We discuss parameterization and least square fitting in the next two sections.

4 Parameterization

A parameterization is a one-to-one mapping between a planar domain and the data points in 3-space. For B-spline, the planar domain is a rectangular area $D = [a, b] \times [c, d]$. Therefore, the process of parameterization is to find a function $\Phi : D \rightarrow \{Q_i\}$. For a data point Q_k we seek a pair of parameters $(u_k, v_k) \in D$, that is a point in the parameter domain.

If the data points are organized in a mesh, the inverse map Φ^{-1} should preserve the same mesh topology in the parameter domain. It is also desirable to minimize the shape distortions. Many techniques for mesh parameterization have been proposed, for examples [3, 4, 10]. In addition to surface fitting, parameterization is also essential to many applications such as remeshing, mesh compression, and mesh smoothing.

In our case, the 3D data points are unorganized. Ma and Kruth [11] and Piegl and Tiller [14] suggested the projection of the 3D data points to a base surface such as a Coons patch derived from the boundaries of the surface, then the

parameters of the projected points on the Coons patch are used for fitting the B-spline surface.

Notice that the image of an object is a mapping between the points on the 3D surface and a planar rectangular domain. It is natural to use the images for parameterization. The problem is that we have multiple images and we need one single mapping. Therefore, we need to merge the images into a single one and use that as a parameterization.

Recall from section 2 that the images of a planar object can be related to each other by homographies. If we choose one image as the fixed image, other images that share at least four points with the fixed image can be merged to it, since a homography matrix can be computed from four corresponding points. As shown in Figure 2, the first view and the third view can be merged into the second view by applying the homography matrices H_{12} and H_{32} to them respectively.

In general, the scene under consideration is not planar and the homography is only an approximation to the transformations between images. Therefore, merging images using homography will cause distortions. But since our objective here is parameterization rather than mosaicking images for visualization, distortions are expected.

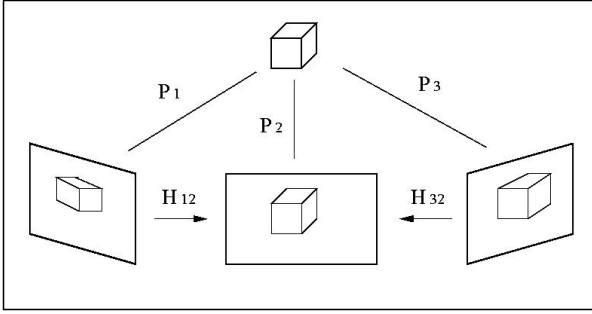


Figure 2. Homographic parameterization

In general, if we have a sequence of overlapping images, I_1, I_2, \dots, I_n , and we choose one I_r ($1 \leq r \leq n$) as the reference view, the transformation applied to the points in image I_k is

$$T_k = \begin{cases} H_{k,k+1}H_{k+1,k+2} \dots H_{r-1,r} & \text{if } 1 \leq k < r \\ H_{k,k-1}H_{k-1,k-2} \dots H_{r-1,r} & \text{if } r < k \leq n \end{cases}$$

where $H_{p,q}$ is the homography matrix between two adjacent views.

Through this kind of piecewise merging, all the visible points in all the images are transformed into a single 2D coordinate system. For each 3D point \mathbf{X} , assume it appears in m images with $\mathbf{u}_1, \dots, \mathbf{u}_m$ as its corresponding homogeneous coordinates. Let $\mathbf{w}_i = T_i\mathbf{u}_i$ and let $\tilde{\mathbf{w}}_i$ be the inhomogeneous coordinate of \mathbf{w}_i . The parameter of \mathbf{X} is

computed as the average of all the transformed points:

$$\mathbf{u}_x = \frac{1}{m} \sum_{i=1}^m \tilde{\mathbf{w}}_i$$

Figure 3 shows a result of a parameterization (Figure 3(b)) from the merge of three views (Figure 3(a)).

5 Least square fitting

Reorder the array of control points by row so that it becomes a linear list of points P_1, P_2, \dots, P_M , where $M = (n+1)(m+1)$. A point P_i ($1 \leq i \leq M$) in the linear list maps to the control point array $P_{s,t}$, where the subscripts s and t are computed as

$$s = \left\lfloor \frac{i}{m+1} \right\rfloor \quad t = i \bmod (m+1).$$

Using the same subscript scheme, we can denote

$$N_i(u, v) = N_s(u)N_t(v).$$

Then we can rewrite equation 2 as

$$S(u, v) = \sum_{i=1}^M N_i(u, v)P_i.$$

For fitting a B-spline surface with given data points Q_1, Q_2, \dots, Q_N , we wish to solve the following equations

$$\sum_{i=1}^M N_i(u_k, v_k)P_i = Q_k \quad 1 \leq k \leq N. \quad (3)$$

Let $\mathbf{P}_x = [P_{1x} \dots P_{Mx}]^T$ and $\mathbf{Q}_x = [Q_{1x} \dots Q_{Nx}]^T$, where P_{ix} and Q_{ix} are the x coordinates of P_i and Q_i respectively.

Write equation 3 in a matrix form

$$\mathbf{N}\mathbf{P}_x = \mathbf{Q}_x \quad (4)$$

where \mathbf{N} is a $N \times M$ matrix of scalars

$$\mathbf{N} = \begin{bmatrix} N_1(u_1, v_1) & \dots & N_M(u_1, v_1) \\ \vdots & \ddots & \vdots \\ N_1(u_N, v_N) & \dots & N_M(u_N, v_N) \end{bmatrix}$$

where $M \geq N$. The least square solution to equation 4 is

$$\mathbf{P}_x = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{Q}_x.$$

The other two coordinates \mathbf{P}_y and \mathbf{P}_z are solved analogously using the same matrix \mathbf{N} and matrix $\mathbf{N}^T\mathbf{N}$.

Figure 4 shows an example of the B-spline surface constructed. Note that the natural B-spline surface goes beyond the data points. The shaded part is the actual surface defined by the data and trimmed away from the original surface using the boundary of the data set.

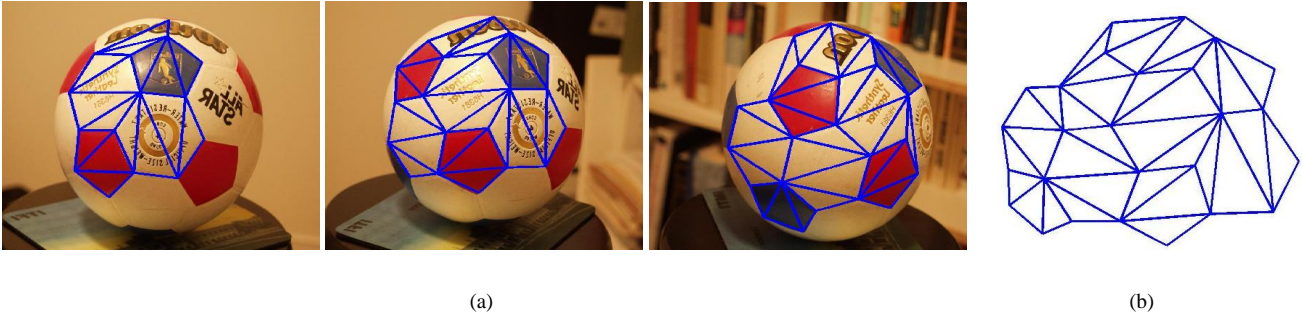


Figure 3. Parameter mesh from merging views

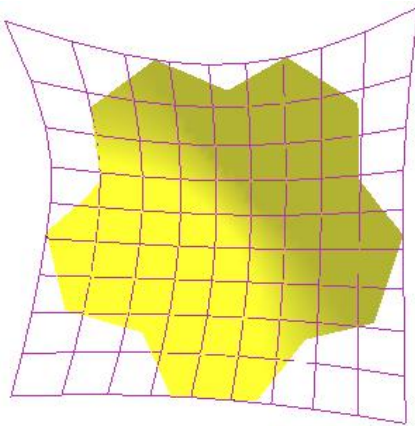


Figure 4. Fitting result of a degree 2 B-spline surface

6 Texture mapping

One of the advantages of reconstructing surfaces from multiple images is that the surfaces can be texture mapped naturally from the images. Coordinates of the image points become the texture coordinates of their corresponding 3D points. A triangle in the 3D mesh maps to a triangle in one of the images. Figure 6(a) shows an example of a texture mapping. As we can see, since there are only a small number of 3D points, the texture mapped model shows obvious artifacts.

As we have a smooth B-spline surface approximating the initially reconstructed 3D points, we can improve the texture mapping by mapping the images to a finer mesh that better approximates the true surface geometry. A hierarchy of meshes with different resolutions can be constructed by refining triangles.

For refining a triangle, we divide the longest of its edges. As shown in Figure 5, if we want to divide the edge AC , we

first compute its mid-point M , and then we project M onto the B-spline surface. The projection finds the point N on the B-spline surface that is closest to M . This can be done using Newton's iteration with initial parameter values being the mid-point coordinates of A and B in the parameter space. Finally, the two triangles $\Delta(A, B, C)$ and $\Delta(A, C, D)$ sharing the edge AC are replaced by four smaller triangles: $\Delta(A, B, N)$, $\Delta(B, C, N)$, $\Delta(N, C, D)$, and $\Delta(A, N, D)$.

Figure 7(c) shows the results of texture mapping on two different levels of refinement. With the initial reconstruction of 37 triangles, we can see obvious artifacts. After B-spline surface fitting and refining to 450 triangles, the texture mapping is much improved but still shows some artifacts. Refining to 7038 triangles, no sign of artifact is shown.

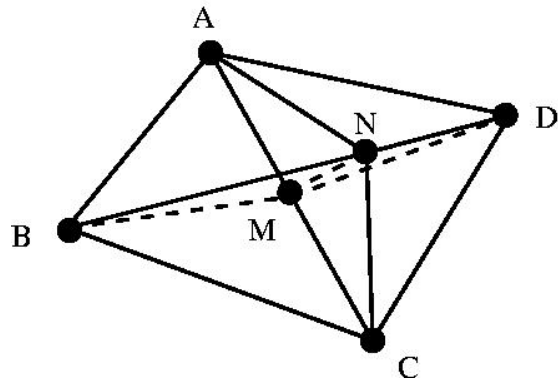


Figure 5. Dividing a triangle edge and projecting the mid edge point onto the B-spline surface

7 Conclusions

We have described a method for reconstructing free-form surfaces from multiple images. Assuming surface smooth-

ness, we fit B-spline surfaces to the reconstructed 3D points. In this way, complex models can be constructed using only sparse feature points. Texture mapping is greatly improved by using a smooth surface representation. The technique of using a homography for parameterizing 3D data points is both simple and efficient. This method is best suited for an interactive system in which users manually select and match a small number of feature points. It is therefore a useful extension to the current photo modeling systems.

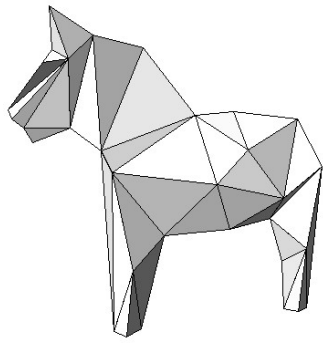
We used projective vision techniques for camera calibration and 3D structure reconstruction. These techniques are simple to implement and in many cases provide fast linear solutions to the reconstruction problem. The linear solutions are sufficient for many applications that do not demand a high accuracy. They can also be used as initial estimations for the more accurate nonlinear iterative methods.

Acknowledgments

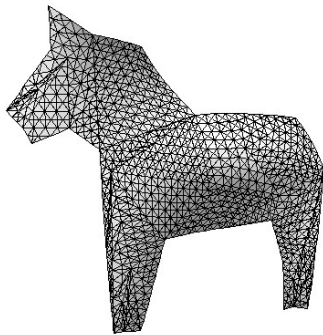
The authors thank the reviewers for their constructive comments and suggestion of references [5], [6], and [20]. We also thank Guy Godin for valuable suggestions toward improvement of this paper.

References

- [1] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2000.
- [2] G. Cross and A. W. Zisserman. Quadric surface reconstruction from dual-space geometry. In *Proceedings of the 6th International Conference on Computer Vision*, pages 25–31, Bombay, India, 1998.
- [3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbury, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Computer Graphics Proceedings, SIG-GRAPH95*, pages 173–182, 1995.
- [4] M. S. Floater. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250, 1997.
- [5] P. Fua. From multiple stereo views to multiple 3-D surfaces. *International Journal of Computer Vision*, 24(1):19–35, 1997.
- [6] P. Fua and Y. LeClerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16:35–56, 1995.
- [7] R. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
- [8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [9] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, Oct. 2000.
- [10] K. Hormann and G. Greiner. MIPS: An efficient global parameterization method. In P.-J. Laurent, P. Sablonnière, and L. Schumaker, editors, *Curve and Surface Design*, pages 153–162. Vanderbilt University Press, Nashville, 2000.
- [11] W. Ma and J. P. Kruth. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, 27(9):663–675, 1995.
- [12] P. R. S. Mendonça and R. Cipolla. A simple technique for self-calibration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR99)*, volume 1, pages 500–505, 1999.
- [13] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, 1997.
- [14] L. A. Piegl and W. Tiller. Parameterization for surface fitting in reverse engineering. *Computer-Aided Design*, 33:593–603, 2001.
- [15] G. Roth and A. Whitehead. Some improvements on two autocalibration algorithms based on the fundamental matrix. In *Proc. International Conference on Pattern Recognition (ICPR2002)*, volume 2, pages 312–315, 2002.
- [16] S. Sinha and B. Schunck. A two-stage algorithm for discontinuity-preserving surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):36–55, 1992.
- [17] C. C. Slama, editor. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.
- [18] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1091–1097, 1998.
- [19] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Aug. 1987.
- [20] L. Zhang and S. Seitz. Image-based multiresolution shape recovery by surface deformation. In *Procs. of SPIE: Videometrics and Optical Methods for 3D Shape Measurement*, pages 51–61, San Jose, CA, 2001.
- [21] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *International Conference on Computer Vision (ICCV99)*, pages 666–673, Corfu, Greece, 1999.
- [22] C. Zhao and R. Mohr. B-spline patches for surface reconstruction in computer vision. In P.-J. Laurent, A. L. Méhauté, and L. Schumaker, editors, *Wavelets, Images and Surface Fitting*, pages 521–528. A K Peters, Wellesley, MA, 1994.



(a)

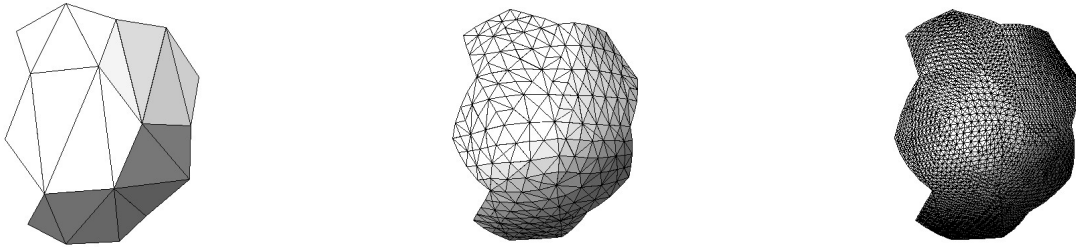


(b)

Figure 6. Wooden horse example. (a) reconstruction and texture mapping (37 triangles); (b) after B-spline fitting and mesh refinement (2346 triangles).



(a)



(b)



(c)

Figure 7. Soccer ball example. (a) input images; (b) initial reconstruction and two levels of refinements with 18, 450, and 7038 triangles; (c) texture mapping on the original reconstruction and the two levels of refinement after B-spline fitting.