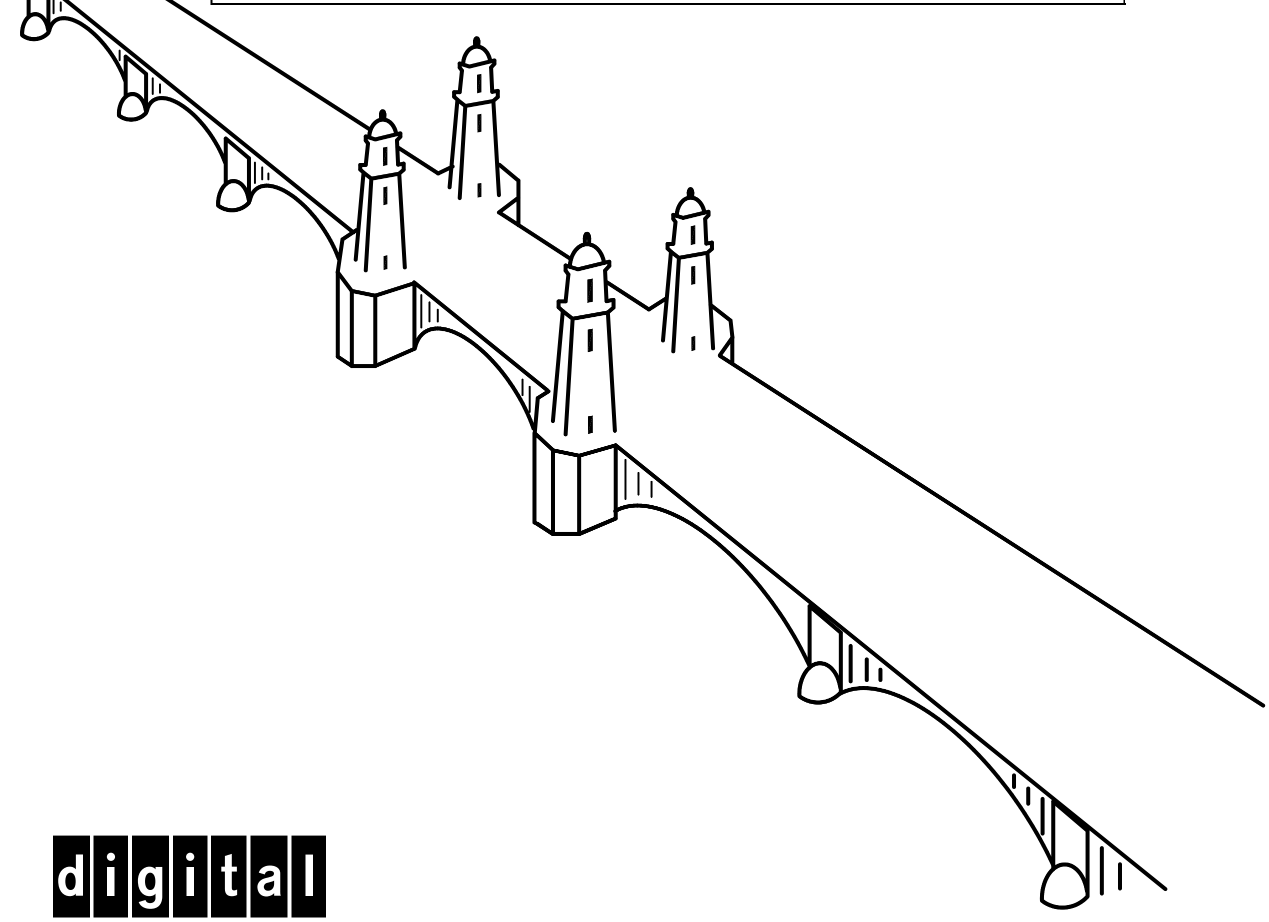
Robust Shape Recovery from Occluding Contours Using a Linear Smoother

Richard Szeliski and Richard Weiss

Digital Equipment Corporation
Cambridge Research Lab

CRL 93/7 December, 1993



CAMBRIDGE RESEARCH LABORATORY
Technical Report Series

Cambridge Research Laboratory
One Kendall Square
Cambridge, Massachusetts 02139

Robust Shape Recovery from Occluding Contours Using a Linear Smoother

Richard Szeliski and Richard Weiss[1]

Digital Equipment Corporation
Cambridge Research Lab

CRL 93/7 December, 1993

Abstract

Recovering the shape of an object from two views fails at occluding contours of smooth objects because the extremal contours are view dependent. For three or more views, shape recovery is possible, and several algorithms have recently been developed for this purpose. We present a new approach to the multiframe stereo problem which does not depend on differential measurements in the image, which may be noise sensitive. Instead, we use a linear smoother to optimally combine all of the measurements available at the contours (and other edges) in all of the images. This allows us to extract a robust and dense estimate of surface shape, and to integrate shape information from both surface markings and occluding contours.

Keywords: Computer vision, image sequence analysis, motion analysis and multiframe stereo, shape and object representation, occluding contours (profiles).

[1]Computer and Information Science Department, University of Massachusetts at Amherst, Amherst, MA 01003

Contents

List of Figures

List of Tables

1 Introduction

Most visually-guided systems require representations of surfaces in the environment in order to integrate sensing, planning, and action. The task considered in this paper is the recovery of the 3D structure (shape) of objects with piecewise-smooth surfaces from a sequence of profiles taken with known camera motion. The *profile* (also known as the *extremal boundary* or *occluding contour*) is defined as the image of the *critical set* of the projection map from the surface to the image plane. Since profiles are general curves in the plane without distinguished points, there is no *a priori* pointwise correspondence between these curves in different views. However, given the camera motion, there is a correspondence based on the *epipolar constraint*. For two images, i.e., classical *binocular stereo*, this epipolar constraint is a set of straight lines which are the intersection of the *epipolar planes* with the image plane. The epipolar plane through a point is determined by the view direction at that point and the instantaneous camera translation direction.

In the case of contours that are not view dependent, e.g., creases (tangent discontinuities) and surface markings, many techniques have been developed for recovering the 3D contour locations from two or more images under known camera motion [Marr and Poggio, 1979; Mayhew and Frisby, 1980; Arnold, 1983; Bolles *et al.*, 1987; Baker and Bolles, 1989; Matthies *et al.*, 1989]. Techniques have also been developed for simultaneously estimating contour locations and camera positions [Tsai and Huang, 1984; Faugeras *et al.*, 1987; Horn, 1990]. However, for smooth curved surfaces, the critical set which generates the profile is different for each view. Thus, the triangulation applied in two-frame stereo will not be correct along the occluding contour for smooth surfaces. For the same reason, it is often not possible to determine the camera motion from the images unless some assumptions are made either about the surface or the motion [Arborgast and Mohr, 1992; Giblin *et al.*, 1992]. On the other hand, the fact that the critical sets sweep out an area means that the connectivity of the surface points can be determined, i.e., one obtains a surface patch rather than a set of points.

The problem of reconstructing a smooth surface from its profiles has been explored for known planar motion by Giblin and Weiss [Giblin and Weiss, 1987] and subsequently for more general known motion by Vaillant and Faugeras [Vaillant, 1990; Vaillant and Faugeras, 1992] and Cipolla and Blake [Blake and Cipolla, 1990; Cipolla and Blake, 1990; Cipolla and Blake, 1992]. These approaches are either based on a differential formulation and analysis, or they use curve fitting but still only use three frames. First order temporal derivatives are usually computed as differences from pairs of frames, and second order derivatives from triples. Unfortunately, determining differential quantities reliably from real images in this way is difficult. Even fitting curves to data from three images can be unsatisfactory. This has led Cipolla and Blake to use relative measurements in order to cancel some of the error due to inadvertent camera rotation. Their approach approximated image contours with B-snakes which require initialization for each contour that is tracked. In addition, B-snakes implicitly smooth the contours in the image. Since the recovery of 3D points is a linear problem (as we will show in this paper), the smoothing can be done in 3D on the surface, where more context can be used in the detection of discontinuities so that detailed structure can be preserved.

It is natural to consider surface reconstruction as an optimal estimation problem. To overcome the limitations of previous algorithms, the approach we develop in this paper applies standard techniques from estimation theory (Kalman filtering and smoothing) to make optimal use of each measurement without computing differential quantities. First, we derive a *linear* set of equations between the unknown shape (surface point positions and radii of curvature) and the measurements. We then develop a robust linear smoother ([Gelb, 1974; Bierman, 1977]) to compute statistically optimal current and past estimates from the set of contours. Smoothing allows us to combine measurements on both sides of each surface point.

Our technique produces a complete surface description, i.e., a network of linked 3D surface points, which provides us with a much richer description than just a set of 3D curves. Some parts of the surface may never appear on the profile. In some cases this is due to occlusion either by the same surface or another one. In other cases, it is due to limitations of the camera trajectory [Giblin and Weiss, 1994]. Since the method presented here also works for arbitrary surface markings and creases, a larger part of the surface can be reconstructed than from occluding contours of the smooth pieces alone. Our approach also addresses the difficult problem of contours that merge and split in the image, which must be resolved if an accurate and complete 3D surface model is to be constructed.

The method we develop has applications in many areas of computer vision, computer aided design, and visual communications. The most traditional application of visually based shape recovery is in the reconstruction of a mobile robot's environment, which allows it to perform obstacle avoidance and planning tasks [Curwen *et al.*, 1992]. Visually based shape recovery can also be used to develop strategies for robotics grasping and manipulation tasks, or as an off-line technique to automatically "learn" object descriptions for object or pose recognition tasks.

In less traditional applications, our system could be used to perform *reverse engineering*, i.e., to automatically acquire 3D computer aided design (CAD) description of real-world objects or prototypes, or even to construct a "3D fax" for transmitting 3D object descriptions and images between desktops.

Our paper is structured as follows. We begin in Section 2 with a description of our edge detection, contour linking, and edge tracking algorithms. In Section 3, we discuss the estimation of the epipolar plane for a sequence of three or more views. Section 4 presents the linear measurement equations which relate the edge positions in each image to the parameters of the circular arc being fitted at each surface point. Section 5 then reviews robust least squares techniques for recovering the shape parameters and discusses their statistical interpretation. Section 6 shows how to extend least squares to a time-evolving system using the Kalman filter, and develops the requisite forward mapping (surface point evolution) equations. Section 7 extends the Kalman filter to the linear smoother, which optimally refines and updates previous surface point estimates from new measurements. Section 9 presents a series of experiments performed both on noisy synthetic contour sequences and on real video images. We close with a discussion of the performance of our new technique and a discussion of future work.

2 Contour detection and tracking

The problem of edge detection has been extensively studied in computer vision [Marr and Hildreth, 1980; Canny, 1986]. The choice of edge detector is not crucial in our application, since we are interested mostly in detecting strong edges such as occluding contours and visible surface markings.[1] For our system, we have chosen the *steerable filters* developed by Freeman and Adelson [Freeman and Adelson, 1991], since they provide good angular resolution at moderate computation cost, and since they can find both step and peak edges. We have used both the G_1 and (G_2, H_2) sets of filters, with the default parameters suggested by Freeman and Adelson. An example of our edge detector operating on the input image in Figure 1a is shown in Figure 1b.

Once discrete edgels have been detected, we use local search to link the edgels into contours. We find the two neighbors of each edgel based on proximity and continuity of orientation. Note that in contrast to some of the previous work in reconstruction from occluding contours [Cipolla

[1] Unlike many edge detection applications, however, our system provides us with a quantitative way to measure the performance of an edge detector, since we can in many cases measure the accuracy of our final 3D reconstruction.

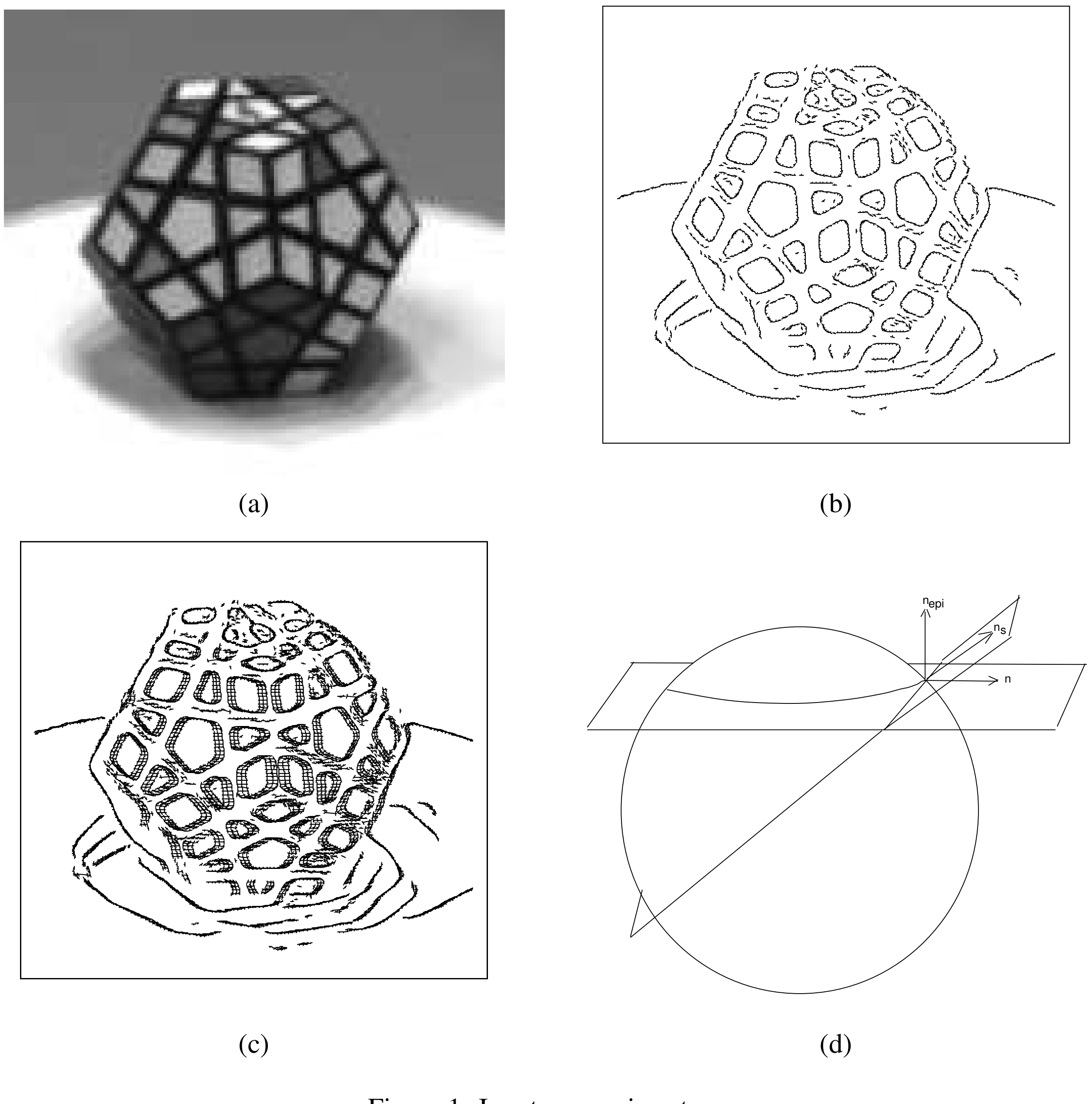


Figure 1: Input processing steps

(a) sample input image (dodecahedral puzzle), (b) estimated edgels and orientations (maxima in $|G_1|^2$), (c) tracked edgels, (d) correspondence of points on the occluding contours using the epipolar constraint.

and Blake, 1990; Cipolla and Blake, 1992; Blake *et al.*, 1993], we do not fit a smooth parametric curve to the contour since we wish to directly use all of the edgels in the shape reconstruction, without losing detail.[2] The curve fitting problem is essentially one of detecting outliers. Since the 3D reconstruction provides more context, smoothing in 3D should be preferred.

We then use the known epipolar line constraints (Section 3) to find the best matching edgel in the next frame. Our technique compares all candidate edgels within the epipolar line search range (defined by the expected minimum and maximum depths), and selects the one which matches most closely in orientation, contrast, and intensity (see Figure 1c). Once an initial estimate for the 3D location of an edgel has been computed, the search range can be dramatically reduced (see Section 5.3).

Since contours are maintained as a list of discrete points, it is necessary to resample the edge points in order to enforce the epipolar constraint on each track. We occasionally start new tracks if there is a sufficiently large (2 pixel wide) gap between successive samples on the contour. While we do not operate directly on the spatiotemporal volume, our tracking and contour linking processes form a virtual surface similar to the *weaving wall* technique of Harlyn Baker [Baker, 1989]. Unlike Baker's technique, however, we do not assume a regular and dense sampling in time.

3 Reconstructing surface patches

The surface being reconstructed from a moving camera can be parametrized in a natural way by two families of curves [Giblin and Weiss, 1987; Cipolla and Blake, 1990]: one family consists of the critical sets on the surface; the other is tangent to the family of rays from the camera focal points. The latter curves are called *epipolar curves* and together with the critical sets form the *epipolar parametrization.* This parametrization can always be used except when the profile is singular or when the normal to the surface is perpendicular to the direction of camera translation [Giblin and Weiss, 1994]. For a pair of stereo images, each viewing direction together with the translation vector from one camera center to the other determines a plane called an *epipolar plane.* The same construction holds in the case of motion: in the limit, as the time between samples goes to zero, the plane determined by a view direction and the camera translation velocity will also be called an epipolar plane. For a more detailed discussion of epipolar curves see [Giblin and Weiss, 1994].

[2] However, we do perform a small amount of curvature-dependent smoothing along the curves to reduce noise. This can be viewed as part of the edge extraction stage.

The problem is that any smooth surface reconstruction algorithm which is more than a first order approximation requires at least three images and, that in general, the three corresponding tangent rays will not be coplanar. However, there are many cases when this will be a good approximation. One such case is when the camera trajectory is almost linear. If the camera trajectory is linear, then the epipolar planes form a pencil of planes containing that line. Under orthographic projection, if the camera motion is planar, then all of the epipolar curves will be planar as well.

Cipolla and Blake [Cipolla and Blake, 1990; Cipolla and Blake, 1992] and Vaillant and Faugeras [Vaillant, 1990; Vaillant and Faugeras, 1992] noticed that to compute the curvature of a planar curve from three tangent rays, one can determine a circle which is tangent to these rays. See Figure 2. The assumption that one needs to make is that the surface remains on the same side of the tangent rays. This is true for intervals of the curve which do not have a singularity or zero curvature.

Given three or more edgels tracked with our technique, we would like to compute the location of the surface and its curvature by fitting a circular arc to the lines defined by the view directions at those edgels. In general, a non-singular space curve will have a unique circle which is closest to the curve at any given point. This is called the *osculating circle*, and the plane of this circle is called the *osculating plane*. It is easy to see that if the epipolar curve is non-singular, then the epipolar plane is an estimate of its osculating plane [Cipolla and Blake, 1992], and the lines defined by the view directions are close to this plane and can be projected onto it. The accuracy of the computation of the radius of curvature depends on the conditioning of this projection. Since in the limit, the epipolar plane is the osculating plane for the epipolar curve, the epipolar curves should be the most robust to reconstruct by projecting onto this plane.

The relationship between the curvature of a curve such as the epipolar curve and the curvature of the surface is determined by the angle between the normal to the curve and the normal to the surface. The curvature of the curve scaled by the cosine of this angle is the *normal curvature*. The curvature of a surface can be thought of as a function which assigns to every tangent direction v a value which is the curvature k_v of the normal slice in that direction. If v is the tangent to the epipolar curve, k_{epi} is the curvature of the epipolar curve, and ϕ is the angle between the epipolar plane and the plane containing v and the surface normal, then the relationship among them is given by the equation

$$k_v = k_{epi} \cos \phi \tag{1}$$

This gives Meusnier's Theorem which says that the normal curvature is the same for all curves on the surface with a given tangent direction. Since the normal to the surface can be determined from

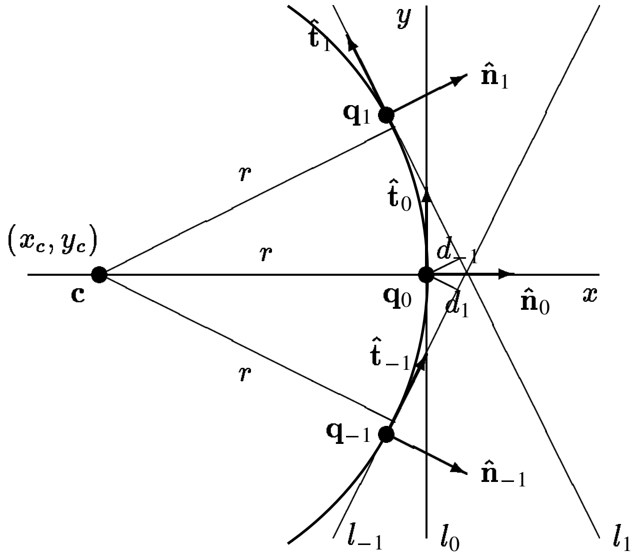


Figure 2: Local coordinate axes and circle center point calculation

In this figure, the points $\mathbf{p}_i$ and $\mathbf{q}_i$ are coincident. In general, the $\mathbf{q}_i$ will lie somewhere along the tangent rays, and the $\mathbf{p}_i$ will be the points of tangency to the osculating circle.

---

the image, the normal curvature can be obtained from the epipolar curve.

4 Measurement equations

Once we have selected the epipolar plane as the reconstruction plane for fitting the circular arc, we must compute the set of lines in this plane which should be tangent to the circle. This can be done either by projecting the 3D lines corresponding to the linked edgels directly onto the plane, or by intersecting the tangent planes (defined by the edgels and their orientations) with the reconstruction plane.

We represent the 3D line corresponding to an edgel in frame i by a 3D point $\mathbf{q}_i$ and a direction $\hat{\mathbf{t}}_i$. The point $\mathbf{q}_i$ is chosen to be the intersection of the viewing ray with a reference plane $z = z_0$. The direction is given by $\hat{\mathbf{t}}_i = \mathcal{N}(\mathbf{q}_i - \mathbf{c}_i)$, where $\mathbf{c}_i$ is the camera center and $\mathcal{N}()$ normalizes a vector. We choose one of these lines as the *reference frame* $(\hat{\mathbf{n}}_0, \hat{\mathbf{t}}_0)$ centered at $\mathbf{q}_0$ (where $\hat{\mathbf{n}}_0 = \hat{\mathbf{t}}_0 \times \hat{\mathbf{n}}_{\text{epi}}$), e.g., by selecting the middle of n frames for a batch fit, or the last frame for a Kalman filter. This line lies in the reconstruction plane defined by $\hat{\mathbf{n}}_{\text{epi}}$.

If we parameterize the osculating circle by its center $\mathbf{c} = (x_c, y_c)$ and radius r (Figure 2), we find that the tangency condition between line i and the circle can be written as

$$c_i x_c + s_i y_c + r = d_i \tag{2}$$

where $c_i = \hat{\mathbf{t}}_i \cdot \hat{\mathbf{t}}_0$, $s_i = -\hat{\mathbf{t}}_i \cdot \hat{\mathbf{n}}_0$, and $d_i = (\mathbf{q}_i - \mathbf{q}_0) \cdot \hat{\mathbf{n}}_i$. *Thus, we have a linear estimation problem in the quantities (x_c, y_c, r) given the known measurements (c_i, s_i, d_i).* This linearity is central to the further developments in the paper, including the least squares fitting, Kalman filter, and linear smoother, which we develop in the next three sections.

5 Least squares fitting

While in theory the equation of the osculating circle can be recovered given the projection of three non-parallel tangent lines onto the epipolar plane, a much more reliable estimate can be obtained by using more views. Given the set of equations (2), how can we recover the best estimate for (x_c, y_c, r)? Regression theory [Albert, 1972; Bierman, 1977] tells us that the minimum least squared error estimate of the system of equations $\mathbf{A}\mathbf{x} = \mathbf{d}$ can be found by minimizing

$$e = |\mathbf{A}\mathbf{x} - \mathbf{d}|^2 = \sum_i (\mathbf{a}_i \cdot \mathbf{x} - d_i)^2. \tag{3}$$

This minimum can be found by solving the set of *normal equations*[3]

$$(\mathbf{A}^T\mathbf{A})\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{d} \tag{4}$$

or

$$(\sum_i \mathbf{a}_i \mathbf{a}_i^T)\hat{\mathbf{x}} = \sum_i \mathbf{a}_i d_i.$$

A statistical justification for using least squares will be presented shortly (Section 5.1).

In our circle fitting case, $\mathbf{a}_i = (c_i, s_i, 1)$, $\mathbf{x} = (x_c, y_c, r)$, and the normal equations are

$$\begin{bmatrix} \sum_i c_i^2 & \sum_i c_i s_i & \sum_i c_i \\ \sum_i s_i c_i & \sum_i s_i^2 & \sum_i s_i \\ \sum_i c_i & \sum_i s_i & \sum_i 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ r \end{bmatrix} = \begin{bmatrix} \sum_i c_i d_i \\ \sum_i s_i d_i \\ \sum_i d_i \end{bmatrix}. \tag{5}$$

[3] Alternative techniques for solving the least squares problem include *singular value decomposition* [Press *et al.*, 1986] and Householder transforms [Bierman, 1977].

If we solve the above set of equations directly, the estimates for x_c and r will be very highly correlated and both will be highly unreliable (assuming the range of viewpoints is not very large). This can be seen both by examining Figure 2, where we see that the location of **c** is highly sensitive to the exact values of the d_i, or by computing the covariance matrix $\mathbf{P} = (\mathbf{A}^T\mathbf{A})^{-1}$ (Section 5.1).

We cannot do much to improve the estimate of r short of using more frames or a larger camera displacement, but we can greatly increase the reliability of our shape estimate by directly solving for the *surface point* (x_s, y_s), where $x_s = x_c + r$ and $y_s = y_c$.[4] The new set of equations is thus

$$c_i x_s + s_i y_s + (1 - c_i) r = d_i. \tag{6}$$

While there is still some correlation between x_s and r, the estimate for x_s is much more reliable (Section 5.1). Once we have estimated (x_s, y_s, r), we can convert this estimate back to a 3D surface point,

$$\mathbf{p}_0 = \mathbf{q}_0 + x_s \hat{\mathbf{n}}_0 + y_s \hat{\mathbf{t}}_0, \tag{7}$$

a 3D center point

$$\mathbf{c} = \mathbf{q}_0 + (x_s - r)\hat{\mathbf{n}}_0 + y_s \hat{\mathbf{t}}_0 = \mathbf{p}_0 - r\hat{\mathbf{n}}_0, \tag{8}$$

or a surface point in the i^{th} frame

$$\mathbf{p}_i = \mathbf{c} + r\hat{\mathbf{n}}_i = \mathbf{p}_0 + r(\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_0), \tag{9}$$

where

$$\hat{\mathbf{n}}_i = \hat{\mathbf{t}}_i \times \hat{\mathbf{n}}_{\text{epi}}$$

is the osculating circle normal direction perpendicular to line l_i (Figure 2).

5.1 Statistical interpretation

The least squares estimate is also the *minimum variance* and *maximum likelihood* estimate (optimal statistical estimate) under the assumption that each measurement is contaminated with additive Gaussian noise [Bierman, 1977]. If each measurement has a different variance σ_i^2, we must weight each term in the squared error measure (3) by $w_i = \sigma_i^{-2}$, or, equivalently, multiply each equation $\mathbf{a}_i \cdot \mathbf{x} = d_i$ by σ_i^{-1}.

[4] While the point (x_s, y_s) will not in general lie on the line $(\mathbf{q}_0, \hat{\mathbf{t}}_0)$, the tangent to the circle at (x_s, y_s) will be parallel to $\hat{\mathbf{t}}_0$.

In our application, the variance of d_i, σ_i^2, can be determined by analyzing the edge detector output and computing the angle between the edge orientation and the epipolar line

$$\sigma_i^2 = \sigma_e^2/(\hat{\mathbf{l}}_i \cdot \hat{\mathbf{n}}_{\text{epi}})^2 = \sigma_e^2/(1 - (\hat{\mathbf{m}}_i \cdot \hat{\mathbf{n}}_{\text{epi}})^2),$$

where σ_e is the variance of $\mathbf{q}_i$ along the surface normal $\hat{\mathbf{m}}_i$. This statistical model makes sense if the measurements d_i are noisy and the other parameters (c_i, s_i) are noise-free. This is a reasonable assumption in our case, since the camera positions are known but the edgel locations are noisy. The generalization to uncertain camera locations is left to future work.

When using least squares, the covariance matrix of the estimate can be computed from $\mathbf{P} = (\mathbf{A}^T\mathbf{A})^{-1}$. We can perform a simple analysis of the expected covariances for n measurements spaced θ apart. Using Taylor series expansions for $c_i = \cos i\theta$ and $s_i = \sin i\theta$, and assuming that $i \in [-m \ldots m]$, $n = 2m + 1$, we obtain the covariance matrices

$$P_3^c \approx \begin{bmatrix} 6\theta^{-4} & 0 & -6\theta^{-4} \\ 0 & \frac{1}{2}\theta^{-2} & 0 \\ -6\theta^{-4} & 0 & 6\theta^{-4} \end{bmatrix} \quad \text{and} \quad P_3^s \approx \begin{bmatrix} 1 & 0 & -2\theta^{-2} \\ 0 & \frac{1}{2}\theta^{-2} & 0 \\ -2\theta^{-2} & 0 & 6\theta^{-4} \end{bmatrix}$$

where P_3^c is the 3 point covariance for the center-point formulation, and P_3^s is the 3 point covariance for the surface-point formulation. As we can see, variance of the surface point local x estimate is four orders of magnitude smaller than that of the center point. Similar results hold for the overdetermined case ($n > 3$). Extending the analysis to the asymmetrical case, $i \in [0 \ldots 2m]$, we observe that the variance of the x_s and y_s estimates increases.

5.2 Robustifying the estimate

To further improve the quality and reliability of our estimates, we can apply *robust statistics* to reduce the effects of *outliers* which are due to grossly erroneous measurements as well as large changes in the surface curvature [Huber, 1981]. Many robust techniques are based on first computing *residuals*, $r_i = d_i - \mathbf{a}_i \cdot \mathbf{x}$, and then re-weighting the data by a monotonic function

$$(\sigma_i')^{-2} = \sigma_i^{-2} g(|r_i|)$$

or throwing out measurements whose $|r_i| \gg \sigma_i$. Alternatively, least median squares can also be used to compute a robust estimate, but at an increased complexity.

In our application, outliers occur mainly from gross errors in edge detection (e.g., when adjacent edges interfere) and from errors in tracking. Currently, we compute residuals after each batch fit, and keep only those measurements whose residuals fall below a fixed threshold.

5.3 Predicting 2D locations for tracking

Once a 3-D estimate for an edgel location has been computed, this can be used to predict where the edgel would appear in the next frame, and hence to improve the correspondence produced by the tracking stage. When no 3-D information is available, we project the viewing ray passing through a 2-D edgel into the next frame to give us the epipolar search line. We use the intersection of the viewing ray with a minimum and maximum depth plane to determine the endpoints which limit the search range.

When a 3-D position estimate is available, we project the 3-D position and covariance estimate into the new reconstruction plane. The position on the screen of the edgel then gives us the middle of the search range, while a multiple of the standard deviation in the local x direction (which is parallel to the image plane and in the reconstruction plane and hence along the epipolar line) times the epipolar line determines the limits of the search range. More formally, the endpoints of the search line are

$$\mathcal{P}_i(\mathbf{p}_i \pm \alpha\sigma_x\hat{\mathbf{n}}_0)$$

where $\mathcal{P}_i$ projects points in 3-D onto the ith frame, and σ_x^2 is the variance in the local x direction.

Our approach is similar in spirit to the *validation gate* approach used by Blake *et al.* for their Kalman-filter snake tracking [Blake *et al.*, 1993]. Even more sophisticated data association techniques could be used to disambiguate multiple intersecting tracks [Bar-Shalom and Fortmann, 1988].

6 Kalman filter

The Kalman filter is a powerful technique for efficiently computing statistically optimal estimates of time-varying processes from series of noisy measurements [Gelb, 1974; Bierman, 1977; Maybeck, 1979]. In computer vision, the Kalman filter has been applied to diverse problems such as motion recovery [Rives *et al.*, 1986], multiframe stereo [Matthies *et al.*, 1989], and pose recovery [Lowe, 1991]. In this section, we develop a Kalman filter for contour-based shape recovery in two parts:

first, we show how to perform the batch fitting of the previous section incrementally; second, we show how surface point estimates can be predicted from one frame (and reconstruction plane) to another.

The *update* or *data processing* part of a Kalman filter takes a current estimate $\tilde{\mathbf{x}}_i$ with its associated covariance $\tilde{\mathbf{P}}_i$ and produces an updated estimate $\hat{\mathbf{x}}_i$ and covariance $\hat{\mathbf{P}}_i$ by processing a single measurement

$$d_i = \mathbf{a}_i \cdot \mathbf{x}_i. \tag{10}$$

The traditional Kalman filter formulation [Gelb, 1974] first computes a Kalman gain matrix

$$\mathbf{K}_i = \tilde{\mathbf{P}}_i \mathbf{a}_i (\mathbf{a}_i^T \tilde{\mathbf{P}}_i \mathbf{a}_i + \sigma_i^2)^{-1}, \tag{11}$$

where σ_i^2 is the variance associated with measurement i. It then increments the state estimate by adding a weighted residual

$$\hat{\mathbf{x}}_i = \tilde{\mathbf{x}}_i + \mathbf{K}_i (d_i - \mathbf{a}_i \cdot \tilde{\mathbf{x}}_i). \tag{12}$$

and decrements the covariance matrix

$$\hat{\mathbf{P}}_i = \tilde{\mathbf{P}}_i - \mathbf{K}_i \mathbf{a}_i^T \tilde{\mathbf{P}}_i. \tag{13}$$

Applying this Kalman filter to our circular arc fitting task is straightforward, since each of our tangent lines is of the required form (10), $d_i = \mathbf{a}_i \cdot \mathbf{x}_i$. More numerically stable or computationally efficient forms of the Kalman filter have also been developed [Bierman, 1977], but we have not yet implemented them to see if they improve our performance.

The update part of the Kalman filter is derived directly from the measurement equation (2) [Gelb, 1974]. It provides an incremental technique for estimating quantities in a *static* system, e.g., for refining a set of (x_s, y_s, r) estimates as more edgels are observed. For our application, however, we need to produce a series of surface points which can be linked together into a complete surface description. If we were using batch fitting, we would perform a new batch fit centered around each new 2D edgel. Instead, we use the complete Kalman filter, since it has a much lower computational complexity. The Kalman filter provides a way to deal with *dynamic* systems where the state $\mathbf{x}_i$ is evolving over time. We identify each measurement $\mathbf{x}_i$ with the surface point (x_s, y_s, r) whose local coordinate frame is given by $(\hat{\mathbf{n}}_i, \hat{\mathbf{t}}_i, \hat{\mathbf{n}}_{\text{epi}})$ centered at $\mathbf{q}_i$ in frame i.

The second half of the Kalman filter requires a *system model* or *process model* which predicts the evolution of state variables over time [Gelb, 1974]. For our smooth surface model, we assume that r (the third component of $\mathbf{x}$) can vary slowly over time, but that the other two components have no associated process noise, i.e., $\mathbf{s} = (0, 0, s_r)$.

The overall sequence of processing steps is therefore the following. Initially, we perform a batch fit to $n \geq 3$ frames, using the last frame as the reference frame. Next, we convert the local estimate into a global 3D position (7) and save it as part of our final surface model. We use this 3D estimate to construct a reduced search range for edgels during the tracking phase. Then, we project the 3D surface point and its radius onto the next frame, i.e., into the frame defined by the next 2D edgel found by the tracker.[5] Then, we update the state estimate using the local line equation and the Kalman filter updating equations. We repeat the above process (except for the batch fit) so long as a reliable track is maintained (i.e., the residuals are within an acceptable range). If the track disappears or a robust fit is not possible, we terminate the recursive processing and wait until enough new measurements are available to start a new batch fit.

7 Linear smoothing

The Kalman filter is most commonly used in control systems applications, where the current estimate is used to determine an optimal control strategy to achieve a desired system behavior [Gelb, 1974]. In certain applications, however, we may wish to refine old estimates as new information arrives, or, equivalently, to use "future" measurements to compute the best current estimate. Our shape recovery application falls into this latter category, since the accuracy of the estimate depends on the range of viewing angles for the measurements, and this can be increased by taking measurements from both sides of the 3D curve corresponding to a given visible occluding contour. In addition, it should be noted that if the curvature of the epipolar curve is not constant, then for each interval over which it is monotonic, filtering rather than smoothing will introduce a bias.

The generalization of the Kalman filter to update previous estimates is called the *linear smoother* [Gelb, 1974]. The smoothed estimate of $\mathbf{x}_i$ based on all the measurements between 0 and N is denoted by $\hat{\mathbf{x}}_{i|N}$. Three kinds of smoothing are possible [Gelb, 1974]. In *fixed-interval smoothing*, the initial and final times 0 and N are fixed, and the estimate $\hat{\mathbf{x}}_{i|N}$ is sought, where i varies from 0 to N. In this case, each point in the model is estimated from all of the data in a track. In *fixed-point*

[5] For even higher accuracy, we could use the 2D projection of our 3D surface point as the input to our tracker.

smoothing, i is fixed and $\hat{\mathbf{x}}_{i|N}$ is sought as N increases. Each point is updated as new data is obtained, i.e. there is a separate smoother for each point. In *fixed-lag smoothing*, $\hat{\mathbf{x}}_{N-L|N}$ is sought as N increases and L is held fixed. This has the advantage that each point is estimated within a fixed amount of time from when it appears on the profile, and it is only estimated once. Since the lag time is non-zero, information on both sides of the critical set are used.

For surface shape recovery, both fixed-interval and fixed-lag smoothing are of interest. Fixed-interval smoothing is appropriate when shape recovery is performed off-line from a set of predetermined measurements. The results obtained with fixed-interval smoothing should be identical to those obtained with a series of batch fits, but at a much lower computational cost. The fixed-interval smoother requires a small amount of overhead beyond the regular Kalman filter in order to determine the optimal combination between the outputs of a forward and backward Kalman filter [Gelb, 1974; Bierman, 1977].

For our contour-based shape recovery algorithm, we have developed a new fixed-lag smoother, which, while sub-optimal, allows us to predict the position of the contour in successive images and simplifies the tracking problem. Our fixed-lag smoother begins by computing a *centered* batch fit to $n(\geq 3)$ frames. The surface point is then predicted from frame $i-1$ to frame i as with the Kalman filter, and a new measurement from frame $i+L$, $L = \lfloor n/2 \rfloor$ is added to the predicted estimate. The addition of measurements ahead of the current estimate is straightforward using the projection equations for the least-squared (batch) fitting algorithm.

Our modified fixed-lag smoother and the optimal fixed-lag smoother incorporate the same information into the current estimate, but use slightly different relative weightings of the data. Intuitively, the optimal smoother weights the data most heavily towards the middle (inversely proportional to the distance from the current estimate), while our modified smoother weights the data most heavily towards the front (most recent measurement). For systems where the process noise σ_s^2 is much smaller than the measurement noise σ_i^2, the results should be similar. We examine the relative performance of the batch estimator, Kalman filter, and sub-optimal linear smoother in Section 9.

8 Building a complete surface description

The batch fitting, Kalman filter, and linear smoothers all produce a series of surface point estimates, one for each input image. Because our reconstruction takes place in object space, features such as surface marking and sharp ridges are stationary in 3D (and have $r = 0$). For these features, we would prefer to produce a single time-invariant estimate. While the detection of stationary features could be incorporated into the Kalman filter or smoother itself, we currently defer this decision to a post-processing stage, since we expect the estimates of position and radius of curvature to be more reliable after the whole sequence has been processed. The post-processing stage collapses successive estimates which are near enough in 3D (say, less than the spacing between neighboring sample points on the 3D contour). It adjusts the neighbor (contour) and temporal (previous/next) pointers to maintain a consistent description of the surface.

To fit a complete surface to the data while interpolating across small gaps, a variety of techniques could be used. Traditionally, physically-based deformable models [Terzopoulos and Metaxas, 1991; Pentland and Sclaroff, 1991] have been used to fit such sparse and incomplete 3-D data. An alternative, which does not suffer from the restrictions on topology imposed by previous techniques, is to use a self-triangulating system of particles to model and interpolate the surface [Szeliski *et al.*, 1993]. We plan to investigate the intergration of this system with our multiframe stereo algorithm in future work.

9 Experimental results

To determine the performance of our shape reconstruction algorithm, we generated a synthetic motion sequence of a truncated ellipsoid rotating about its z axis (Figure 3). The camera is oblique (rather than perpendicular) to the rotation axis, so that the epipolar curves are not planar, and the reconstruction plane is continuously varying over time. We chose to use a truncated ellipsoid since it is easy to analytically compute its projections (which are ellipses, even under perspective), and since its radius of curvature is continuously varying (unlike, a cylinder).

When we run the edge images through our least-squares fitter or Kalman filter/smoother, we obtain a series of 3D curves. The curves corresponding to the surface markings and ridges (where the ellipsoid is truncated) should be stationary and have 0 radius, while the curves corresponding to the occluding contour should continuously sweep over the surface.

We can observe this behavior using a three-dimensional graphics program we have developed for displaying the reconstructed geometry. This program allows us to view a series of reconstructed curves either sequentially (as an animation) or concurrently (overlayed in different colors), and to vary the 3D viewing parameters either interactively or as a function of the original camera position

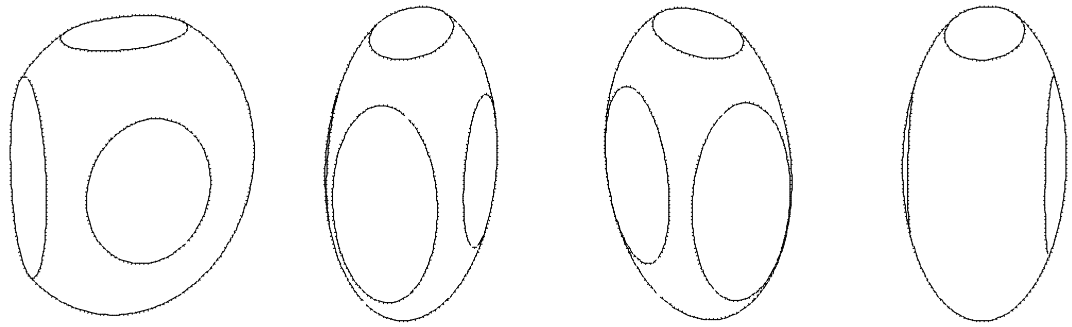


Figure 3: Four images from synthetic truncated ellipsoid sequence.
The top and left hand side are truncated (cut off), while the front and back sides are inscribed with an ellipse (surface marking).

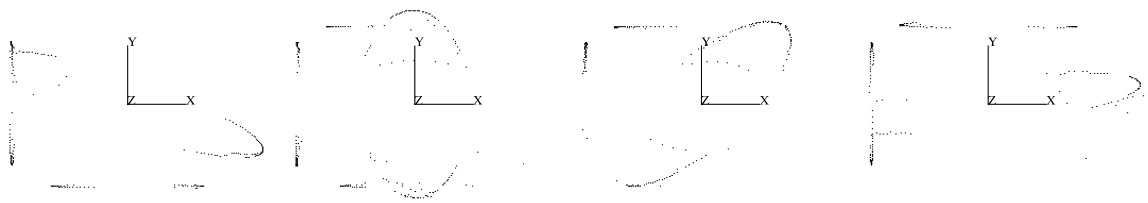


Figure 4: Top view of reconstructed 3D curves.
The surface markings and ridges are stationary, while the occluding contours (ellipse) sweeps around the object.

for each frame. Figure 5 shows all of the 3D curves overlayed in a single image. As we can see, the 3D surface is reconstructed quite well. The left hand pair of images shows an oblique and top view of a noise-free data set, using the linear smoother with $n = 7$ window size. The right-hand pair shows a portion of the reconstructed surface, showing both the profile and epipolar curves.

To obtain a quantitative measure of the reconstruction algorithm performance, we can compute the root median square error between the reconstructed 3D coordinates and the true 3D coordinates (which are known to the synthetic sequence generating program). Table 1 shows the reconstruction error and percentage of surface points reconstructed as a function of algorithm choice and various parameter settings. The table compares the performance of a regular 3-point fit with a 7-point moving window (batch) fit, and a linear fixed-lag smoother with $n = 7$. Results are given for the noise-free and $\sigma_i = 0.1$ pixels case. The different columns show how by being more selective about

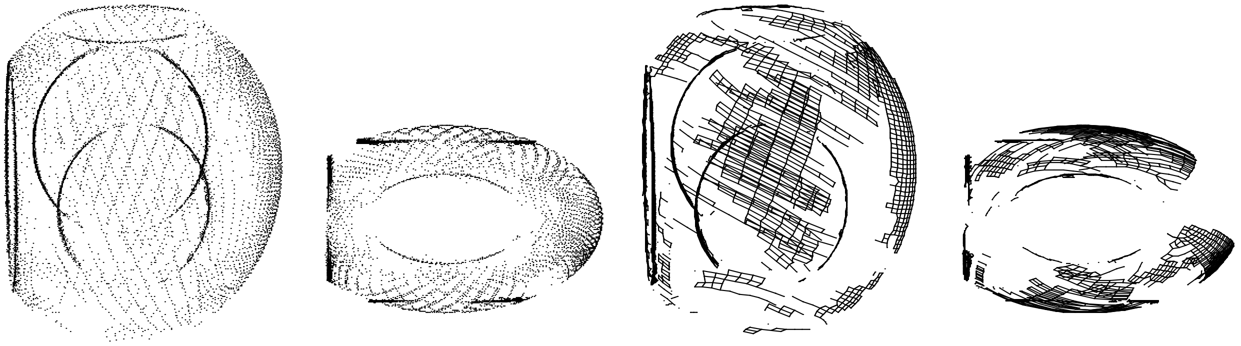


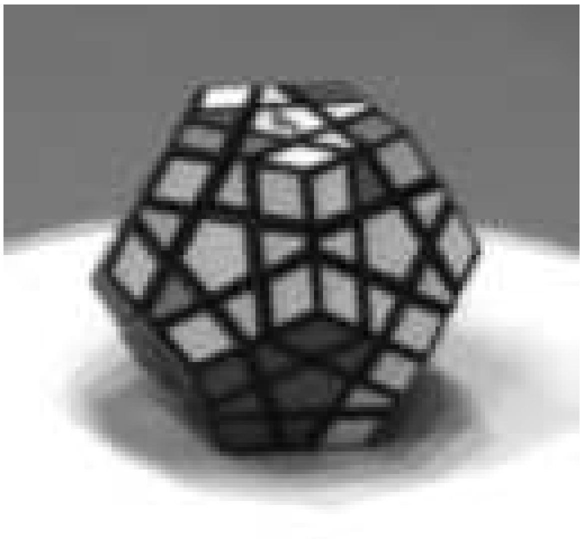
Figure 5: Oblique and top view of reconstructed 3D surface (all 3D curves superimposed). The left pair shows only the reconstructed profile curves, while the right pair shows the profiles linked by the epipolar curves (only a portion of the complete meshed surface is shown for clarity). A total of 72 images spaced 5° apart were used.

which 3D estimates are considered valid (either by requiring more frames to have been successfully fit, or lowering the threshold on maximum covariance), a more reliable estimate can be obtained at the expense of fewer recovered points. For noisefree data, the 3 point algorithm is better because it is less sensitive to curvature variation. However, for noisy data, the 7 point algorithms are better, with batch fitting performing slightly better than linear smoothing.

We have also applied our algorithm to the four real image sequences shown in Figure 6. These sequences were obtained by placing an object on a rotating mechanized turntable whose edge has a Gray code strip used for reading back the rotation angle [Szeliski, 1991; Szeliski, 1993]. The camera motion parameters for these sequences were obtained by first calibrating the camera intrinsic parameters and extrinsic parameters to the turntable top center, and then using the computed turntable rotation. Figure 7 shows the edges extracted from each of these images.

Figure 8 shows two views of each set of reconstructed 3D curves. We can see that the overall shape of the objects has been reconstructed quite well. We show only the profile curves, since the epipolar curves would make the line drawing too dense for viewing at this resolution. Figure 9 shows both the profile curves and the epipolar curves for selected portions of the soda can and coffee objects.

As a final example, Figure 10 shows some partial results (10 reconstructed profile curves) from an image sequence of a coffee mug. This example demonstrates that our method can handle objects



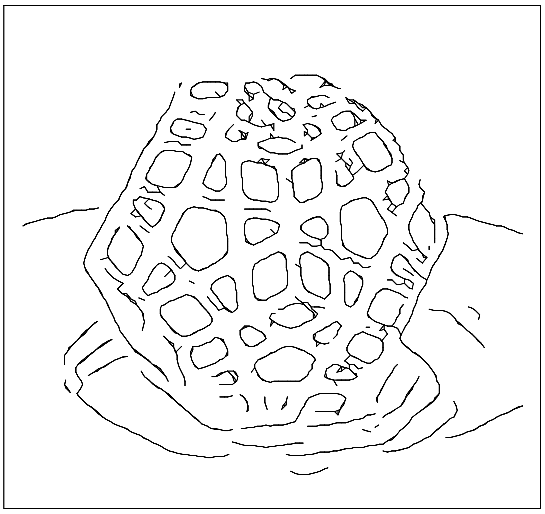
a

b

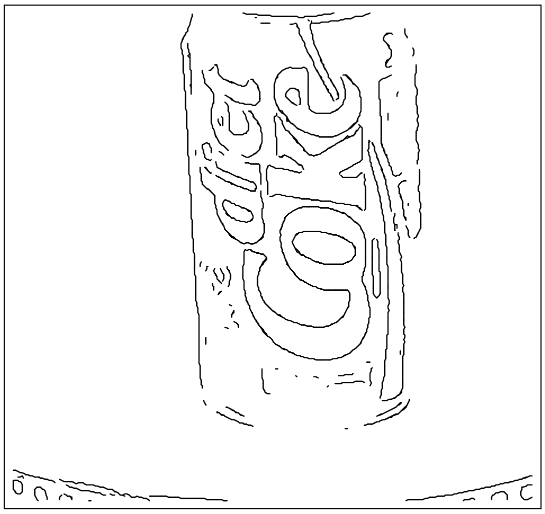
c

d

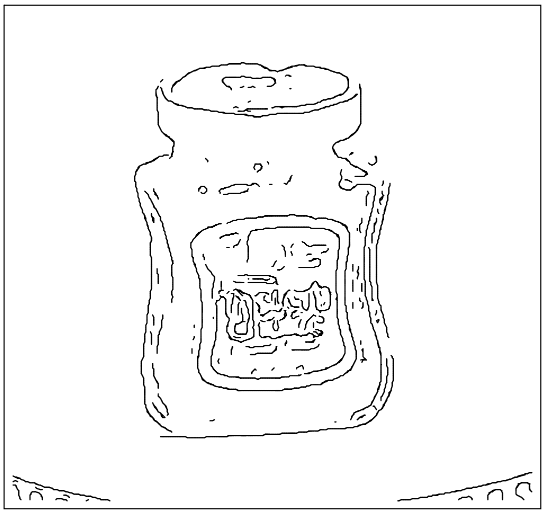
Figure 6: Sample real image sequences used for experiments
(a) dodecahedron (b) soda can (c) coffee (d) tea.



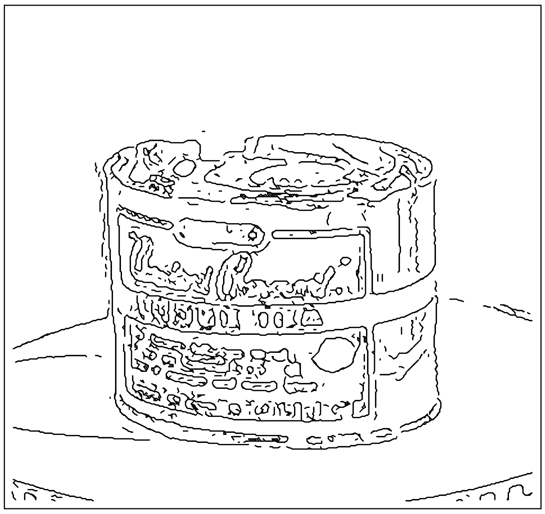
a



b



c



d

Figure 7: Extracted edges
(a) dodecahedron (b) soda can (c) coffee (d) tea.

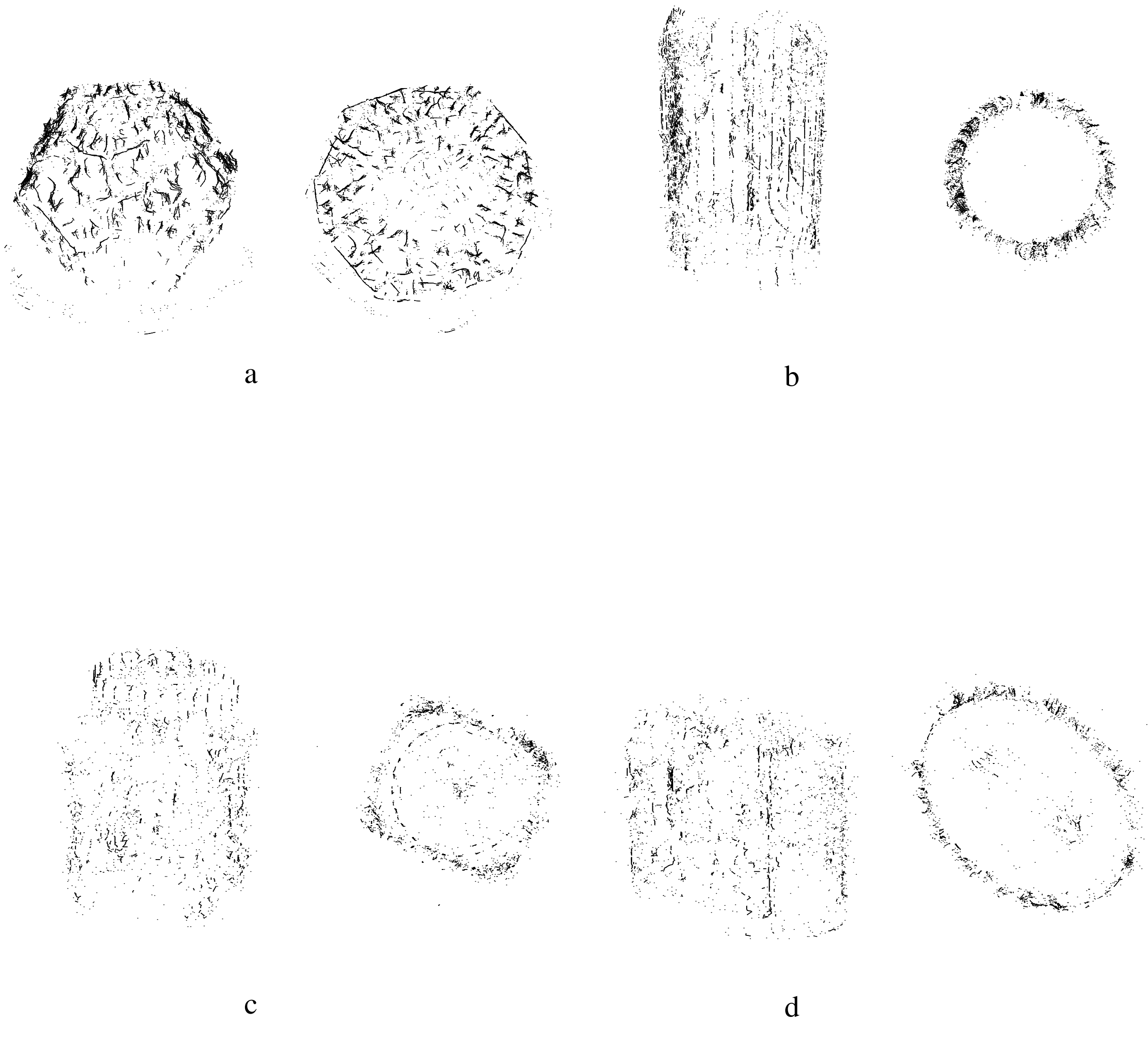


Figure 8: 3D reconstructed points

(a) dodecahedron, (b) soda can, (c) coffee, and (d) tea.

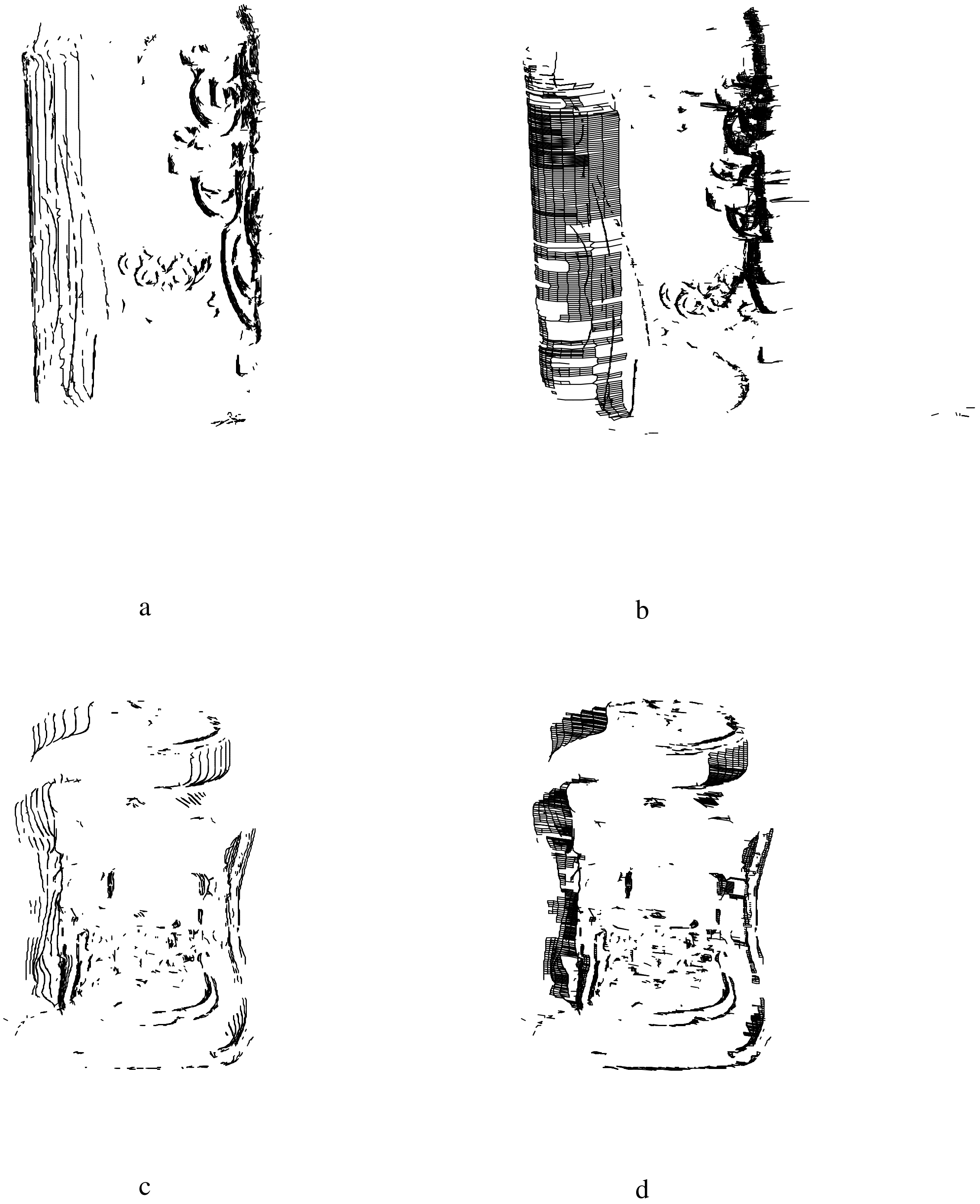
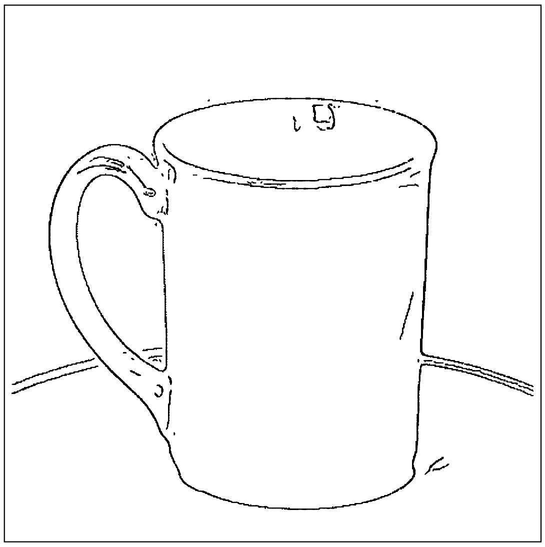


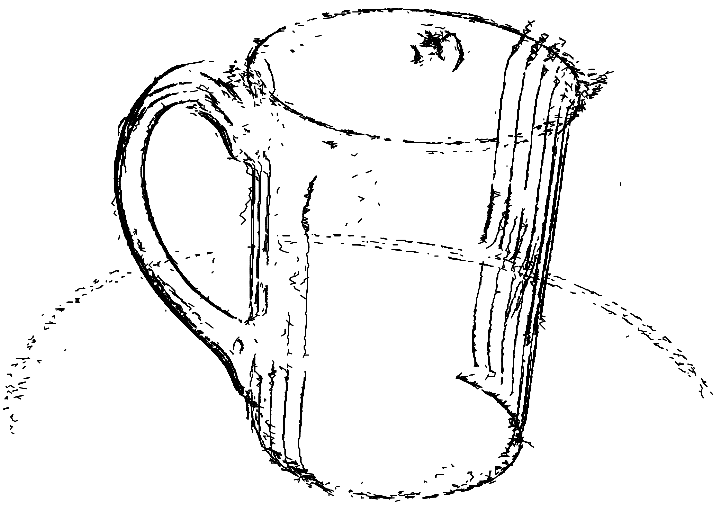
Figure 9: Profile and epipolar curves

(a–b) soda can (c–d) coffee.

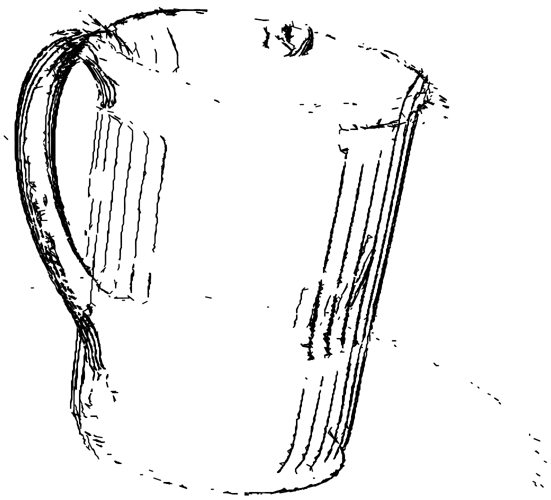
a



b



c



d

Figure 10: Another example: coffee cup.
Note that objects with interior holes (non-trivial topology) can be easily handled by this method.

algorithm	n	σ_i	$n_f \geq 3$	$n_f \geq 7$	$n_f \geq 7 \wedge \sigma_x^2 < 0.5$
smoother	7	0.0	.0074 (77%)	.0046 (45%)	.0044 (38%)
smoother	7	0.1	.0114 (74%)	.0054 (41%)	.0051 (36%)
batch	7	0.0	.0042 (79%)	.0036 (56%)	.0035 (43%)
batch	7	0.1	.0074 (77%)	.0054 (53%)	.0051 (42%)
batch	3	0.0	.0008 (77%)		
batch	3	0.1	.0159 (75%)		

Table 1: Root median square error and percentage of edges reconstructed for different algorithms, window sizes (n), input image noise σ_i, and criteria for valid estimates (n_f: minimum number of frames in fit, σ_x^2: covariance in local x estimate).
These errors are for an ellipse whose major axes are $(0.67, 0.4, 0.8)$ and for a 128×120 image.

with interior holes, since we are not limited to only following the external silhouettes of the objects. In future work, we plan to study the events which occur when multiple silhouette curves obscure each other in the image sequence (which corresponds to points of bitangency in 3D).

10 Discussion and Conclusion

This paper extends previous work on both the reconstruction of smooth surfaces from profiles (edge-based multiframe stereo) and on the epipolar analysis on spatiotemporal surfaces. The ultimate goal of our work is the construction a complete detailed geometric and topological model of a surface from a sequence of views together with an estimate of uncertainty. Towards this end, our observations are connected by tracking edges over time as well as linking neighboring edges into contours. The information represented at each point includes the position, surface normal, and curvatures (currently only in the viewing direction). In addition, error estimates are also computed for these quantities. Since the sensed data does not provide a complete picture of the surface, e.g., there can be self-occlusion or parts may be missed due to coarse sampling or limitations on the camera trajectory, it is necessary to build partial models. In the context of active sensing and real-time reactive systems, the reconstruction needs to be incremental as well.

Because our equations for the reconstruction algorithm are linear with respect to the measurements, it is possible to apply statistical linear smoothing techniques, as we have demonstrated. This satisfies the requirement for incremental modeling, and provides the error estimates which are needed for integration with other sensory data, both visual and tactile. The application of statistical methods has the advantage of providing a sound theoretical basis for sensor integration and for the reconstruction process in general [Szeliski, 1989; Clark and Yuille, 1990].

In future work, we intend to develop a more complete and detailed surface model by combining our technique with regularization-based curve and surface models. We also plan to investigate the integration of our edge-based multiframe reconstruction technique with other visual and tactile techniques for shape recovery.

References

[Albert, 1972] A. Albert. *Regression and the Moore-Penrose Pseudoinverse*. Academic Press, New York, 1972.

[Arborgast and Mohr, 1992] E. Arborgast and R. Mohr. An egomotion algorithm based on the tracking of arbitrary curves. In *Second European Conference on Computer Vision (ECCV'92)*, pages 467–475, Springer-Verlag, Santa Margherita Liguere, Italy, May 1992.

[Arnold, 1983] R. D. Arnold. *Automated Stereo Perception*. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University, March 1983.

[Baker, 1989] H. H. Baker. Building surfaces of evolution: The weaving wall. *International Journal of Computer Vision*, 3(1):50–71, 1989.

[Baker and Bolles, 1989] H. H. Baker and R. C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3(1):33–49, 1989.

[Bar-Shalom and Fortmann, 1988] Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*. Academic Press, Boston, 1988.

[Bierman, 1977] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York, New York, 1977.

[Blake and Cipolla, 1990] A. Blake and R. Cipolla. Robust estimation of surface curvature from deformation of apparent contours. In *First European Conference on Computer Vision (ECCV'90)*, pages 465–474, Springer-Verlag, Antibes, France, April 23–27 1990.

[Blake *et al.*, 1993] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contour. *International Journal of Computer Vision*, 11(2):127–145, October 1993.

[Bolles *et al.*, 1987] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[Cipolla and Blake, 1990] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Third International Conference on Computer Vision (ICCV'90)*, pages 616–623, IEEE Computer Society Press, Osaka, Japan, December 1990.

[Cipolla and Blake, 1992] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, November 1992.

[Clark and Yuille, 1990] J. J. Clark and A. L. Yuille. *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers, Boston, Massachusetts, 1990.

[Curwen *et al.*, 1992] R. Curwen, A. Blake, and A. Zisserman. Real-time visual tracking for surveillance and path planning. In *Second European Conference on Computer Vision (ECCV'92)*, pages 879–883, Springer-Verlag, Santa Margherita Liguere, Italy, May 1992.

[Faugeras *et al.*, 1987] O. D. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. In *First International Conference on Computer Vision (ICCV'87)*, pages 25–34, IEEE Computer Society Press, London, England, June 1987.

[Freeman and Adelson, 1991] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

[Gelb, 1974] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, Massachusetts, 1974.

[Giblin and Weiss, 1987] P. Giblin and R. Weiss. Reconstruction of surfaces from profiles. In *First International Conference on Computer Vision (ICCV'87)*, pages 136–144, IEEE Computer

Society Press, London, England, June 1987.

[Giblin and Weiss, 1994] P. J. Giblin and R. S. Weiss. Epipolar fields on surfaces. In *Third European Conference on Computer Vision (ECCV'94)*, Springer-Verlag, Stockholm, Sweden, (to appear) 1994.

[Giblin *et al.*, 1992] P. J. Giblin, J. E. Rycroft, and F. E. Pollick. Moving surfaces. In *Mathematics of Surfaces V, Inst of Math and its Applications Conference*, Cambridge University Press, Sept 1992.

[Horn, 1990] B. K. P. Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, January 1990.

[Huber, 1981] P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.

[Lowe, 1991] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.

[Marr and Hildreth, 1980] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London*, B 207:187–217, 1980.

[Marr and Poggio, 1979] D. C. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London*, B 204:301–328, 1979.

[Matthies *et al.*, 1989] L. H. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[Maybeck, 1979] P. S. Maybeck. *Stochastic Models, Estimation, and Control*. Volume 1, Academic Press, New York, New York, 1979.

[Mayhew and Frisby, 1980] J. E. W. Mayhew and J. P. Frisby. The computation of binocular edges. *Perception*, 9:69–87, 1980.

[Pentland and Sclaroff, 1991] A. Pentland and S. Sclaroff. Closed-form solutions for physically-based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):715–729, July 1991.

[Press *et al.*, 1986] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, 1986.

[Rives *et al.*, 1986] P. Rives, E. Breuil, and B. Espiau. Recursive estimation of 3D features using optical flow and camera motion. In *Conference on Intelligent Autonomous Systems*, pages 522–532, Elsevier Science Publishers, December 1986. Also appeared in 1987 IEEE International Conference on Robotics and Automation.

[Szeliski, 1989] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Publishers, Boston, Massachusetts, 1989.

[Szeliski, 1991] R. Szeliski. Shape from rotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 625–630, IEEE Computer Society Press, Maui, Hawaii, June 1991.

[Szeliski, 1993] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.

[Szeliski *et al.*, 1993] R. Szeliski, D. Tonnesen, and D. Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 82–87, New York, New York, June 1993.

[Terzopoulos and Metaxas, 1991] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, July 1991.

[Tsai and Huang, 1984] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27, January 1984.

[Vaillant, 1990] R. Vaillant. Using occluding contours for 3D object modeling. In *First European Conference on Computer Vision (ECCV'90)*, pages 454–464, Springer-Verlag, Antibes, France, April 23–27 1990.

[Vaillant and Faugeras, 1992] R. Vaillant and O. D.. Faugeras. Using extremal boundaries for 3-D object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):157–173, February 1992.