International Journal of Computer Vision, 9:2, 83–112 (1992)
© 1992 Kluwer Academic Publishers, Manufactured in The Netherlands.

Surface Shape from the Deformation of Apparent Contours

ROBERTO CIPOLLA
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, England

ANDREW BLAKE
Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, England

Received

Abstract
The spatiotemporal analysis of deforming silhouettes (*apparent contours*) is here extended using the mathematics of perspective projections and tools from differential geometry. Analysis of the image motion of a silhouette or apparent contour enables computation of local surface curvature along the corresponding contour generator on the surface, assuming viewer motion is known. To perform the analysis, a spatiotemporal parameterization of image-curve motion is needed, but is underconstrained (a manifestation of the well-known aperture problem). It is shown that an *epipolar* parameterization is most naturally matched to the recovery of surface curvature.

One immediate facility afforded by the analysis is that surface patches can be reconstructed in the vicinity of contour generators. Once surface curvature is known, it is possible to discriminate extremal contours from other fixed curves in space. Furthermore, the known robustness of *parallax* as a cue to depth extends to the case of surface curvature. Its derivative—*rate of parallax*—is shown theoretically to be a curvature cue that is robust to uncertainties in the known viewer motion. This robustness has been confirmed in experiments.

Finally, the power of the new analysis for robotics applications is demonstrated. Illustrations are given of an Adept robot, equipped with a CCD camera, circumnavigating curved obstacles. When further equipped with a suction gripper the robot manipulator can pick up an object by its curved surface, under visual guidance.

1 Introduction

For a smooth arbitrarily curved surface—especially in man-made environments where surface texture may be sparse—the dominant image feature is the *apparent contour* or silhouette. The apparent contour is the projection of the locus of points on the object—the *contour generator* or *extremal boundary*—which separates the visible from the occluded parts of a smooth, opaque, curved surface.

The apparent contour and its deformation under viewer motion are potentially rich sources of geometric information for navigation, object manipulation, motion planning, and object recognition. Barrow and Tenenbaum [4] pointed out that surface orientation along the apparent contour can be computed directly from image data. Koenderink [29] related the curvature of an apparent contour to the intrinsic curvature of the surface (Gaussian curvature); the sign of Gaussian curvature is equal to the sign of the curvature of the image contour. Convexities, concavities, and inflections of an apparent contour indicate, respectively, convex, hyperbolic, and parabolic surface points. Giblin and Weiss [23] extended this by adding viewer motions to obtain quantitative estimates of surface curvature. A surface (excluding concavities in opaque objects) can be reconstructed from the envelope of all its tangent planes, which in turn are computed directly from the family of apparent contours/silhouettes of the surface, obtained under motion of the viewer. By assuming that the viewer follows a *great circle* of viewer directions around the object, they restricted the problem of analyzing the envelope of tangent planes to the less general problem of computing the envelope of a family of lines in a plane. Their algorithm was tested on noise-free, synthetic data (on the assumption that apparent contours had been distinguished from other image contours) demonstrating the reconstruction of a planar curve under orthographic projection.

In the first part of this article this will be extended to the general case or arbitrary nonplanar, curvilinear viewer motion under perspective projection. The geometry of apparent contours and their deformation under viewer-motion are related to the differential geometry of the observed object's surface. In particular it is shown how to recover the position, orientation and 3D shape of visible surfaces in the vicinity of their contour generators from the deformation of apparent contours and *known* viewer motion. The theory for small, local viewer motions is developed to detect extremal boundaries and distinguish them from *occluding edges* (discontinuities in depth or orientation), surface markings or shadow boundaries.[1]

A consequence of the theory concerns the robustness of relative measurements of surface curvature based on the relative image motion of nearby points in the image—*parallax*-based measurements. Intuitively it is relatively difficult to judge, moving around a smooth, featureless object, whether its silhouette is extremal or not—that is, whether curvature along the ray is bounded or not. This judgement is much easier to make for objects which have at least a few surface features. Under small viewer motions, features are "sucked" over the extremal boundary, at a rate that depends on surface curvature. Our theoretical findings exactly reflect the intuition that the "sucking" effect is a reliable indicator of relative curvature, regardless of the exact details of the viewer's motion. Relative measurements of curvature across two adjacent points are shown to be entirely immune to uncertainties in the viewer's rotational velocity. The dependency on the viewer's translational velocity is also greatly reduced.

The second part of this article describes the implementation of these theories and results of experiments performed with a CCD camera mounted on the wrist joint of a 5-axis Adept 1 SCARA robot arm. A computationally efficient method for extracting and tracking image contours based on B-spline snakes is presented. Measurement of image velocities and accelerations at image contours is used to recover surface position and 3D shape of surfaces in the vicinity of their apparent contours. Error and sensitivity analysis substantiate the claims that parallax methods are orders of magnitude less sensitive to the details of the viewer's motion than single-point measurements. The techniques are also used to detect apparent contours and discriminate them from other fixed image features. We describe the realtime implementaions of these algorithms for use in tasks involving the active exploration of visible surface geometry. The visually derived shape information is successfully used in modeling, navigation, and the manipulation of piecewise smooth curved objects.

2 Theoretical Framework

In this section the theoretical framework for the subsequent analysis of apparent contours and their deformation under viewer motion is presented. We begin with the properties of apparent contours and their contour generators and then relate these first to the descriptions of local 3D shape developed from the differential geometry of surfaces, and then to the analysis of visual motion of apparent contours.

2.1 The Apparent Contour and Its Contour Generator

Consider a smooth object. For each vantage point, all the rays through the vantage point that are tangent to the surface can be constructed. They touch the object along a smooth curve on its surface which we call the *contour generator* [38] or alternatively the *extremal boundary* [3], the *rim* [29], the *fold* [6] or the *critical set* of the visual mapping [12, 23] (figure 1).

For generic situations (situations that do not change qualitatively under arbitrarily small excursions of the vantage point) the contour generator is part of a smooth space curve (not a planar curve) whose direction is not in general perpendicular to the ray direction. The contour generator is dependent on the local surface geometry and on the vantage point in a simple way which will be elucidated below. Moreover, each vantage point will, in general, generate a different contour generator. Movement of the viewer causes the contour generator to "slip" over the visible surface.

The image of the contour generator—here called the apparent contour but elsewhere also known as the occluding contour, profile, outline, silhouette, or limb—will usually be smooth (figure 1). However, as a consequence of the contour generator being a space curve, there may exist a finite number of rays that are tangent not only to the surface but also to the contour generator. At these points, the apparent contour of a transparent object will cusp. For opaque surfaces, however, only one branch of the cusp is visible and the contour ends abruptly [34, 29].

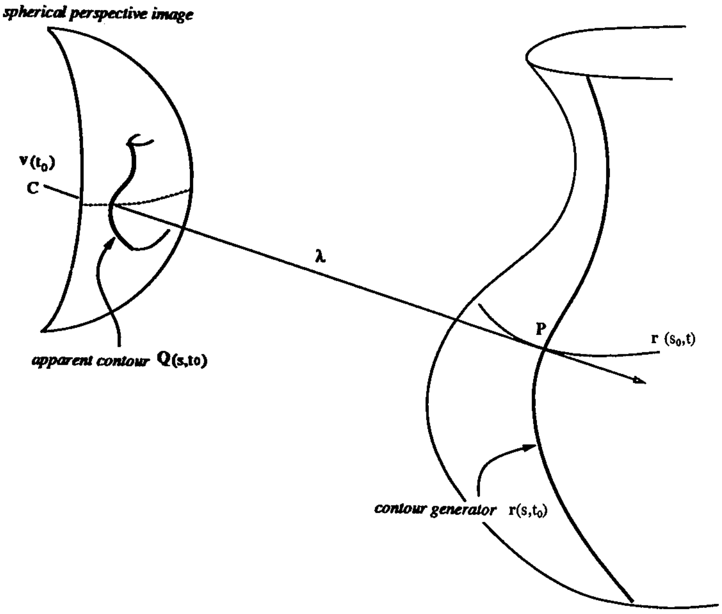

Fig. 1. Surface and viewing geometry. P lies on a smooth surface which is parameterized locally by $\mathbf{r}(s, t)$. For a given vantage point, $\mathbf{v}(t_0)$, the family of rays emanating from the viewer's optical center (C) that touch the surface defines an s-parameter curve $\mathbf{r}(s, t_0)$—the contour generator from vantage point t_0. The spherical perspective projection of this contour generator—the apparent contour, $\mathbf{Q}(s, t_0)$—determines the direction of rays which graze the surface. The distance along each ray, CP, is λ.

2.2 Surface Geometry

In the following, descriptions of local 3D shape are developed directly from the differential geometry of surfaces [18, 20, 31].

Consider a point P on the contour generator of a smooth, curved surface in R^3 and parameterized locally by a vector-valued function $\mathbf{r}(s, t)$. The parametric representation can be considered as covering the surface with two families of curves [35]: $\mathbf{r}(s, t_0)$ and $\mathbf{r}(s_0, t)$ where s_0 or t_0 are fixed for a given curve in the family. For the analysis of apparent contours and their deformation with viewer motion it is necessary to choose the one-parameter family of views to be indexed by a time parameter t, which will also parameterize viewer position for a moving observer. The s and t parameters are defined so that the s-parameter curve, $\mathbf{r}(s, t_0)$, is the contour generator from a particular view t_0 (figure 1). A t-parameter curve $\mathbf{r}(s_0, t)$ can be thought of as the 3D locus of points grazed by a light ray from the viewer, under viewer motion. Such a locus is not uniquely defined. Given a starting point $s = s_0$, $t = t_0$, the correspondence, as the viewer moves, between "successive" (in an infinitesimal sense) contour generators is not unique. Hence there is considerable freedom to choose a spatiotemporal parameterization of the surface, $\mathbf{r}(s, t)$.

The local surface geometry at P is determined by the tangent plane (surface normal) and a description of how the tangent plane turns as we move in arbitrary directions over the surface (figure 2). This can be specified in terms of the basis $\{\mathbf{r}_s, \mathbf{r}_t\}$ for the tangent plane (where for convenience $\mathbf{r}_s$ and $\mathbf{r}_t$ denote $\partial\mathbf{r}/\partial s$ and $\partial\mathbf{r}/\partial t$, the tangents to the s- and t-parameter curves respectively),[2] the surface normal (a unit vector, $\mathbf{n}$) defined so that

$$\mathbf{r}_s \cdot \mathbf{n} = 0 \qquad (1)$$

$$\mathbf{r}_t \cdot \mathbf{n} = 0 \qquad (2)$$

and the derivatives of these quantities with respect to movement over the surface. These are conveniently packaged in the first and second fundamental forms as follows. For a tangent-plane vector at P, $\mathbf{w}$, the first fundamental form, $I(\mathbf{w}, \mathbf{w})$, is used to express the length of any infinitesimal element in the tangent plane ([18], p. 92):

$$I(\mathbf{w}, \mathbf{w}) = \mathbf{w} \cdot \mathbf{w} \qquad (3)$$

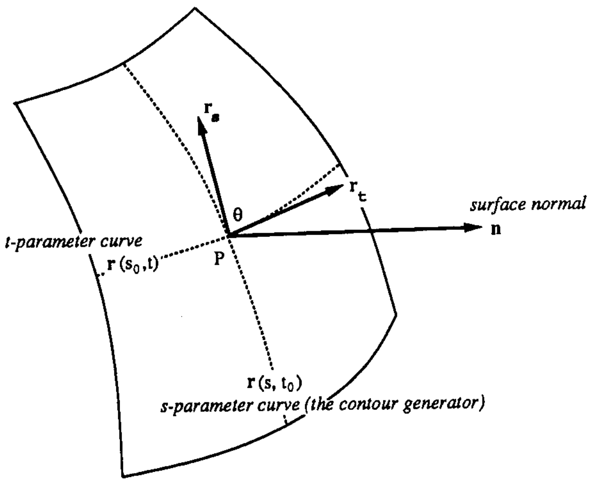

Fig. 2. The tangent plane. Local surface geometry can be specified in terms of the basis $\{\mathbf{r}_s, \mathbf{r}_t\}$ for the tangent plane (where $\mathbf{r}_s$ and $\mathbf{r}_t$ denote the tangents to the s and t-parameter curves respectively and are not in general orthogonal) and the surface normal $\mathbf{n}$ (a unit vector). In differential surface geometry the derivative of these quantities with respect to movement over the surface is used to describe surface shape.

It can be represented by a matrix of coefficients, $\mathbf{G}$, with respect to the basis $\{\mathbf{r}_s, \mathbf{r}_t\}$ where

$$\mathbf{G} = \begin{bmatrix} \mathbf{r}_s \cdot \mathbf{r}_s & \mathbf{r}_s \cdot \mathbf{r}_t \\ \mathbf{r}_t \cdot \mathbf{r}_s & \mathbf{r}_t \cdot \mathbf{r}_t \end{bmatrix} \tag{4}$$

The second fundamental form $II(\mathbf{w}, \mathbf{w})$, quantifies the "bending away" of the surface from the tangent plane. It is defined by ([18], p. 141):

$$II(\mathbf{w}, \mathbf{w}) = -\mathbf{w} \cdot \mathbf{L}(\mathbf{w}) \tag{5}$$

where $\mathbf{L}(\mathbf{w})$ is the derivative of the surface normal, $\mathbf{n}$, in the direction $\mathbf{w}$. $\mathbf{L}$ is in fact a linear transformation on the tangent plane. It is also called the *shape operator* [43] or the *Weingarten map* [42]. In particular for the basis vectors $\{\mathbf{r}_s, \mathbf{r}_t\}$:

$$\mathbf{L}(\mathbf{r}_s) = \mathbf{n}_s \tag{6}$$

$$\mathbf{L}(\mathbf{r}_t) = \mathbf{n}_t \tag{7}$$

and the coefficients of the second fundamental are given by matrix $\mathbf{D}$, where

$$\mathbf{D} = \begin{bmatrix} \mathbf{r}_{ss} \cdot \mathbf{n} & \mathbf{r}_{st} \cdot \mathbf{n} \\ \mathbf{r}_{ts} \cdot \mathbf{n} & \mathbf{r}_{tt} \cdot \mathbf{n} \end{bmatrix} \tag{8}$$

The geometry of the surface is completely determined locally up to a rigid motion in R^3 by these two quadratic forms. It is, however, sometimes more convenient to characterize the surface by *normal* curvatures in specific directions in the tangent plane.[3] The *normal* curvature in the direction $\mathbf{w}$, κ^n, is defined by [20]:

$$\kappa^n = \frac{II(\mathbf{w}, \mathbf{w})}{I(\mathbf{w}, \mathbf{w})} \tag{9}$$

The maximum and minimum normal curvatures are called the *principal* curvatures. The corresponding directions are called the *principal* directions.[4]

It will now be shown how to make these quadratic forms explicit from image measurable quantities. This requires relating the differential geometry of the surface to the analysis of visual motion.

2.3 Imaging Model

A monocular observer can determine the orientation of any ray projected onto its imaging surface. The observer cannot, however, determine the distance along the ray of the object feature that generated it. A general model for the imaging device is therefore to consider it as determining the direction of an incoming ray, which we can choose to represent as a unit vector. This is equivalent to considering the imaging device as a spherical pin-hole camera of unit radius (figure 1). The use of spherical projection (rather than planar), which has previously proven to be a powerful tool in structure-from-motion [32, 39], makes it feasible to extend the theory of Giblin and Weiss [23] to allow for perspective. Its simplicity arises from the fact that there are no special points on the image surface, whereas the origin of the perspective plane is special and the consequent loss of symmetry tends to complicate mathematical arguments.

For perspective projection the direction of a ray to a world point, P, with position vector $\mathbf{r}(s, t)$, is a unit vector on the image sphere $\mathbf{Q}(s, t)$ defined at time t by

$$\mathbf{r}(s, t) = \mathbf{v}(t) + \lambda(s, t)\mathbf{Q}(s, t) \tag{10}$$

where $\lambda(s, t)$ is the distance along the ray to the viewed point P and $\mathbf{v}(t)$ is the viewer's position (figure 1).

For a given vantage position t_0, the apparent contour determines a continuous family of rays $\mathbf{Q}(s, t_0)$ emanating from the camera's optical center, which touch the surface so that

$$\mathbf{Q} \cdot \mathbf{n} = 0 \tag{11}$$

where $\mathbf{n}$ is the surface normal. Equation (11) defines both the contour generator and the apparent contour. The moving monocular observer at position $\mathbf{v}(t)$ sees a family of apparent contours swept over the image sphere. These determine a two-parameter family of rays in R^3, $\mathbf{Q}(s, t)$. As before with $\mathbf{r}(s, t)$, the parameterization is underdetermined but that will be fixed later.

2.4 Viewer and Reference Coordinate Systems

Note that $\mathbf{Q}$ is the direction of the light ray in the fixed reference/world frame for R^3. It is determined by a spherical image position vector $\tilde{\mathbf{Q}}$ (the direction of the ray in the camera/viewer coordinate system) and the orientation of the camera coordinate system relative to the reference frame. For a moving observer the viewer coordinate system is continuously moving with respect to the reference frame. The relationship between $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$ can be conveniently expressed in terms of a rotation operation $R(t)$ [27]:

$$\mathbf{Q} = R(t)\tilde{\mathbf{Q}} \tag{12}$$

The frames are defined so that instantaneously, at time $t = 0$, they coincide,

$$\mathbf{Q}(s, 0) = \tilde{\mathbf{Q}}(s, 0) \tag{13}$$

and have relative translational and rotational velocities of $\mathbf{U}(t)$ and $\boldsymbol{\Omega}(t)$ respectively:

$$\mathbf{U} = \mathbf{v}_t \tag{14}$$

$$\boldsymbol{\Omega} \wedge \tilde{\mathbf{Q}} = R_t\tilde{\mathbf{Q}} \tag{15}$$

The relationship between temporal derivatives of measurements made in the camera coordinate system and those made in the reference frame is then given instantaneously at time $t = 0$ by (differentiating (12)):

$$\mathbf{Q}_t = \tilde{\mathbf{Q}}_t + \boldsymbol{\Omega} \wedge \tilde{\mathbf{Q}} \tag{16}$$

where (as before) the subscripts denote differentiation with respect to time and $\wedge$ denotes a vector product.

3 The Static Analysis of Apparent Contours

3.1 Geometric Properties of the Contour Generator and Its Projection

We now establish why the contour generator is a rich source of information about surface geometry. The physical constraints of *tangency* (all rays at a contour generator are in the surface's tangent plane) and *conjugacy* (the special relationship between the direction of the contour generator and the ray direction) provide powerful constraints on the local geometry of the surface being viewed and allow the recovery of surface orientation and the sign of Gaussian curvature directly from a single image of the contour generator.

3.1.1 Tangency. Both the tangent to the contour generator, $\mathbf{r}_s$, (obtained by differentiating (10)),

$$\mathbf{r}_s = \lambda_s\mathbf{Q} + \lambda\mathbf{Q}_s \tag{17}$$

and the ray, $\mathbf{Q}$, must (by definition) lie in the tangent plane of the surface. From the tangency conditions

$$\mathbf{r}_s \cdot \mathbf{n} = 0$$

$$\mathbf{Q} \cdot \mathbf{n} = 0$$

and (17), we see that the tangent to the apparent contour also lies in the tangent plane of the surface

$$\mathbf{Q}_s \cdot \mathbf{n} = 0 \tag{18}$$

This allows the recovery of the surface orientation $\mathbf{n}$ (defined up to a sign) directly from a single view $\mathbf{Q}(s, t_0)$ using the direction of the ray and the tangent to the apparent (image) contour

$$\mathbf{n} = \frac{\mathbf{Q} \wedge \mathbf{Q}_s}{|\mathbf{Q} \wedge \mathbf{Q}_s|} \tag{19}$$

This result is also valid for projection onto the plane. It is a trivial generalization to perspective projection of the well-known observation of Barrow and Tenenbaum [3, 4].

3.1.2 Conjugate Direction Relationship of Ray and Contour Generator. The *tangency* conditions constrain the contour generator to the tangent plane of the surface. In which direction does the contour generator run? The direction is determined by the second fundamental form and the direction of the ray. In particular, the ray direction, $\mathbf{Q}$, and the tangent to the contour generator, $\mathbf{r}_s$, are in *conjugate* directions with respect to the second fundamental form [33, 29]. That is, the change in surface normal (orientation of the tangent plane) for an infinitesimal movement in the direction of the contour generator is orthogonal to the direction of the ray. This is intuitively obvious for orthographic projection since the normal will continue to be orthogonal to the line of sight as we move along the contour generator.

This is immediately apparent in the current framework for perspective projection since the second fundamental form has the property that

$$II(\mathbf{Q}, \mathbf{r}_s) = -\mathbf{Q} \cdot \mathbf{L}(\mathbf{r}_s) = -\mathbf{Q} \cdot \mathbf{n}_s \qquad (20)$$

which, by differentiating (11) and substituting (18), is zero:

$$\mathbf{Q} \cdot \mathbf{n}_s = 0 \qquad (21)$$

The ray direction, **Q**, and the contour generator are not in general perpendicular but in *conjugate* directions. Let θ be the angle between the ray direction **Q** and the tangent $\mathbf{r}_s$ to the extremal contour. In general $-\pi/2 \leq \theta \leq \pi/2$. For example, for any point on a sphere, the contour generator will be perpendicular to the ray. However for any point on a cylinder, the *conjugate* direction of any ray is in the *asymptotic* direction, (i.e., parallel to the axis of a cylinder), and the contour generator will then run along this direction. The special case $\theta = 0$ occurs when the ray **Q** lies along an *asymptotic* direction of the surface. The tangent to the contour generator and the ray are parallel—*asymptotic* directions are *self-conjugate*. A *cusp* is generated in the projection of the contour generator, seen as an ending of the apparent contour for an opaque surface [34].

3.2 Static Properties of Apparent Contours

It is now well established that static views of extremal boundaries are rich sources of surface geometry [4, 29, 11, 23]. The main results are summarized below. Simple derivations can be found in [14].

3.2.1 Surface Normal. Computation of orientation on a textured surface patch would usually require (known) viewer motion to obtain depth, followed by spatial differentiation. In the case of a contour generator however, the tangency condition (11), (18) means that surface orientation $\mathbf{n}(s, t_0)$ can be recovered directly from a single view of an apparent contour $\mathbf{Q}(s, t_0)$ from the ray (**Q**) and image tangent directions ($\mathbf{Q}_s$):

$$\mathbf{n}(s, t_0) = \frac{\mathbf{Q} \wedge \mathbf{Q}_s}{|\mathbf{Q} \wedge \mathbf{Q}_s|}$$

The temporal and spatial differentiation that, for the textured patch, would have to be computed with attendant problems of numerical conditioning, is done, for extremal boundaries, by the physical constraint of tangency.

Note that the sign of the orientation can only be determined if it is known on which side of the apparent contour the surface lies. This information may not be reliably available in a single view. It is shown below, however, that the "sidedness" of the contour generator can be unambiguously determined from the deformation of the apparent contour under known viewer motion. In the following, we choose the convention that the surface normal is defined to point away from the solid surface. This arbitrarily fixes the direction of increasing s-parameter of the apparent contours so that $\{\mathbf{Q}, Q_s, \mathbf{n}\}$ form a right-handed orthogonal frame.

3.2.2 Sign of Normal Curvature Along the Contour Generator. The curvature of the apparent (image) contour has the same sign as the normal curvature along the contour generator.

The curvature of the apparent contour, κ^p, computed as the *geodesic* curvature of the curve, $\mathbf{Q}(s, t_0)$, on the image sphere:[5,6]

$$\kappa^p = \frac{\mathbf{Q}_{ss} \cdot \mathbf{n}}{|\mathbf{Q}_s|^2} \qquad (22)$$

is simply related to the *normal* curvature of the contour generator, κ^s, by [14]:

$$\kappa^p = \frac{\lambda \kappa^s}{\sin^2 \theta} \qquad (23)$$

where (as before) θ is the angle between the ray and the contour generator. Since surface depth λ must be positive, the sign of κ^s must, in fact, be the same as the sign of κ^p.

In the case of viewing a parabolic point, $\kappa^s = 0$, and an inflection is generated in the apparent contour. A similar result was derived for orthographic projection by Brady et al. [11]. In the special case of a cusp, $\theta = 0$ and the apparent contour has infinite curvature while the contour generator has zero normal curvature. This can be considered in the limit.

3.2.3 Sign of Gaussian Curvature. For opaque surfaces the sign of the Gaussian curvature, K, can be inferred from a single view of an extremal boundary from the sign of the curvature of the apparent contour. This follows from a simple relationship between the Gaussian curvature, K; the curvature κ^t of the normal section at P containing the ray direction; the curvature κ^p of the apparent contour (perspective projection) and the depth λ [29, 31] (derived in section 4.3.4):

$$K = \frac{\kappa^p \kappa^t}{\lambda} \qquad (24)$$

The sign of κ^t is always the same at a contour generator. For P to be visible, the *normal section* must be convex at a contour generator—a concave surface point can never appear on a contour generator of an opaque object.[7] Distance to the contour generator, λ, is always positive. Hence the sign of κ^p determines the sign of Gaussian curvature. Convexities, concavities, and inflections of an apparent contour indicate, respectively, convex, hyperbolic, and parabolic surface points.

As before when we considered the surface normal, the ability to determine the sign of the Gaussian curvature relies on being able to determine on which side of the apparent contour the surface lies. This information is not readily available from image contour data. It is however available if it is possible to detect a contour-ending since the local surface is then hyperbolic (since the surface is being viewed along an *asymptotic* direction) and the apparent contour must be concave at its endpoint [34]. Detecting *cusps* by photometric analysis is a nontrivial exercise.

4 The Dynamic Analysis of Apparent Contours

4.1 Spatiotemporal Parametrization

The previous section showed that static views of apparent contours provide useful qualitative constraints on local surface shape. The viewer must however have discriminated apparent contours from the images of other surface curves (such as surface markings or discontinuities in surface orientation) and have determined on which side of the image contour the surface lies.

When the viewer executes a *known* motion, then surface depth can, of course, be computed from image velocities [10, 26]. This is correct for static space curves but it will be shown that it also holds for extremal contour generators even though they are not fixed in space. Furthermore, if image accelerations are also computed, then full surface curvature (local 3D shape) can be computed along a contour generator. Giblin and Weiss demonstrated this for orthographic projection and planar motion [23]. We now generalize these results to arbitrary nonplanar, curvilinear viewer motion and perspective projection. This requires the choice of a suitable spatiotemporal parameterization for the image, $\tilde{\mathbf{Q}}(s, t)$, and surface, $\mathbf{r}(s, t)$.

As the viewer moves the family of apparent contours, $\tilde{\mathbf{Q}}(s, t)$, is swept out on the image sphere (figure 3). However the spatiotemporal parameterization of the family is not unique. The mapping between contour generators, and hence between apparent contours, at successive instants is underdetermined. This is essentially the "aperture problem" for contours, considered either on the spherical perspective image $\tilde{\mathbf{Q}}(s, t)$, or on the Gauss sphere $\mathbf{n}(s, t)$, or between space curves

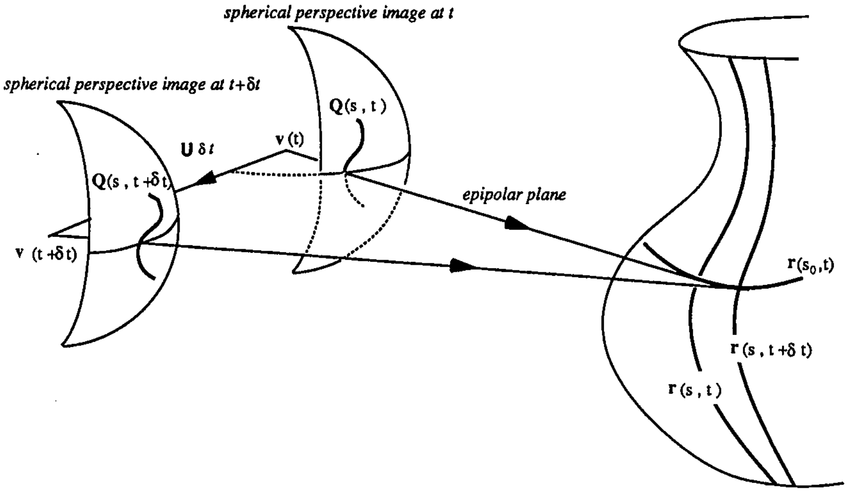
spherical perspective image at t
spherical perspective image at t+δt
Q(s,t)
U δt
v(t)
Q(s,t+δt)
v (t+δt)
epipolar plane
r(s₀,t)
r(s,t+δt)
r(s,t)

Fig. 3. Epipolar parameterization. A moving observer at position $\mathbf{v}(t)$ sees a family of contour generators $\mathbf{r}(s, t)$ indexed by the time parameter t. Their spherical perspective projections are represented by a two-parameter family of apparent contours $\mathbf{Q}(s, t)$. For the epipolar parameterization t-parameter curves ($\mathbf{r}(s_0, t)$ and $\mathbf{Q}(s_0, t)$) are defined by choosing the correspondence between successive contours to be in an epipolar plane that is determined by the translational velocity, $\mathbf{U}$ and the direction of the ray, $\mathbf{Q}$.

on the surface $\mathbf{r}(s, t)$. The choice is arbitrary since the image contours are projections of different 3D space curves.

4.2 Epipolar Parameterization

A natural choice of parameterization (for both the spatiotemporal image and the surface), is the *epipolar parameterization* defined by

$$\mathbf{r}_t \wedge \mathbf{Q} = 0 \tag{25}$$

The tangent to the t-parameter curve is chosen to be in the direction of the ray, $\mathbf{Q}$. The physical interpretation is that the grazing/contact point is chosen to "slip" along the ray. The tangent-plane basis vectors, $\mathbf{r}_s$ and $\mathbf{r}_t$, are therefore in *conjugate* directions. The advantage of the parameterization is clear later, when it leads to a simplified treatment of surface curvature and a unified treatment of the projection of rigid space curves and apparent contours.

A natural *correspondence* between points on successive snapshots of an apparent contour can now be set up. These are the lines of constant s on the image sphere. Differentiating (10) with respect to time and enforcing (25) leads to a "matching" condition

$$\mathbf{Q}_t = \frac{(\mathbf{U} \wedge \mathbf{Q}) \wedge \mathbf{Q}}{\lambda} \tag{26}$$

The corresponding ray in the next viewpoint (in an infinitesimal sense), $\mathbf{Q}(s_0, t + \delta t)$, is chosen so that it lies in the plane defined by $(\mathbf{U} \wedge \mathbf{Q})$—the *epipolar plane*. This is also the *osculating* plane of the t-parameter curve on the surface, $\mathbf{r}(s_0, t)$ (figure 3). This is the infinitesimal analogue to epipolar-plane matching in stereo [2, 10]. For a general motion, however, the epipolar plane structure rotates continuously as the direction of translation, $\mathbf{U}$, changes and the space curve, $\mathbf{r}(s_0, t)$, generated by the movement of a contact point will be nonplanar.

Substituting (16) into (26), the tangents to the t-parameter curves on the spatiotemporal image, $\tilde{\mathbf{Q}}(s_0, t)$, are defined at $t = 0$ by

$$\tilde{\mathbf{Q}}_t = \frac{(\mathbf{U} \wedge \tilde{\mathbf{Q}}) \wedge \tilde{\mathbf{Q}}}{\lambda} - \mathbf{\Omega} \wedge \tilde{\mathbf{Q}} \tag{27}$$

Note that $\tilde{\mathbf{Q}}_t$ is equal to the image velocity of a point on the projection of a static space curve at depth λ [39]. This is not surprising since instantaneously image velocities are dependent only on depth and not surface curvature. Points on successive apparent contours are "matched" by searching along epipolar great circles on the image sphere (or epipolar lines for planar image geometry) defined by the viewer motion, $\mathbf{U}$, $\mathbf{\Omega}$ and the image position $\tilde{\mathbf{Q}}$. This induces a natural correspondence between the contour generators from successive viewpoints on the surface.

The contact point on a contour generator moves/slips along the line of sight $\mathbf{Q}$ with a speed, $\mathbf{r}_t$ determined by the distance and surface curvature.

$$\mathbf{r}_t = \left[\frac{\mathbf{Q}_t \cdot \mathbf{n}}{\kappa^t}\right] \mathbf{Q} \tag{28}$$

where κ^t is the normal curvature of the space curve, $\mathbf{r}(s_0, t)$:

$$\kappa^t = \frac{\mathbf{r}_{tt} \cdot \mathbf{n}}{\mathbf{r}_t \cdot \mathbf{r}_t} \tag{29}$$

Derivation 1. Substituting the matching constraint of (26) into the time derivative of (10) we obtain:

$$\mathbf{r}_t = (\lambda_t + \mathbf{Q} \cdot \mathbf{U})\mathbf{Q} \tag{30}$$

Differentiating (30) with respect to time and substituting this into (29) we obtain the relationship between surface curvature and viewer motion.

$$\kappa^t = \frac{\mathbf{Q}_t \cdot \mathbf{n}}{(\lambda_t + \mathbf{Q} \cdot \mathbf{U})} \tag{31}$$

Combining (31) and (30) gives the required result.

The numerator, $\mathbf{Q}_t \cdot \mathbf{n}$, of (28) is analogous to stereo disparity (as appears below in the denominator of the depth formula (32)). It depends only on the distance of the contact point and the "stereo baseline." The denominator is the curvature (normal) of the space curve generated as the viewer moves in time. The speed of the contact point is therefore inversely proportional to the surface curvature. The contour generator "clings" to points with high curvature and speeds up as the curvature is reduced. This property will be used later to distinguish surface markings or creases from the contour generators of curved surfaces (extremal boundaries).

4.3 Dynamic Properties of Apparent Contours

The choice of a suitable (although arbitrary) spatiotemporal parameterization permits us to make measurements on the spatiotemporal image, $\tilde{\mathbf{Q}}(s, t)$, and to recover an exact specification of the visible surface. This includes position, orientation and 3D shape as well as qualitative cues such as to which side of the image contour the occluding surface lies.

4.3.1 Recovery of Depth from Image Velocities. Depth λ (distance along the ray $\mathbf{Q}$—see figure 1) can be computed from the deformation ($\mathbf{Q}_t$) of the apparent contour under known viewer motion (translational velocity $\mathbf{U}$) [10]: From (26),

$$\lambda = -\frac{\mathbf{U}\cdot\mathbf{n}}{\mathbf{Q}_t\cdot\mathbf{n}} \tag{32}$$

This formula is an infinitesimal analogue of triangulation with stereo cameras (figure 3). The numerator is analogous to baseline and the denominator (the normal component of image velocity) to disparity. In the infinitesimal limit, stereo will, in principle, correctly determine the depth of the contour generator.

Equation (32) can also be reexpressed in terms of spherical image position $\tilde{\mathbf{Q}}$ and the normal component of image velocity $\tilde{\mathbf{Q}}_t\cdot\mathbf{n}$:

$$\lambda = -\frac{\mathbf{U}\cdot\mathbf{n}}{\tilde{\mathbf{Q}}_t\cdot\mathbf{n} + (\mathbf{\Omega}\wedge\tilde{\mathbf{Q}})\cdot\mathbf{n}} \tag{33}$$

Clearly, absolute depth can only be recovered if rotational velocity $\mathbf{\Omega}$ is known.

4.3.2 Surface Curvature from Deformation of the Apparent Contour. Surface curvature (3D shape) is to be expressed in terms of the first and second fundamental forms, $\mathbf{G}$ and $\mathbf{D}$ ((4) and (8)), which in the epipolar parameterization and for unit basis vectors can be simplified to

$$\mathbf{G} = \begin{bmatrix} 1 & \cos\theta \\ \cos\theta & 1 \end{bmatrix} \tag{34}$$

$$\mathbf{D} = \begin{bmatrix} \kappa^s & 0 \\ 0 & \kappa^t \end{bmatrix} \tag{35}$$

where κ^t is the normal curvature of the t-parameter curve $\mathbf{r}(s_0, t)$ and κ^s is the normal curvature of the contour generator $\mathbf{r}(s, t_0)$ at P. Equivalently κ^t is the curvature of the normal section at P in the direction of the ray, $\mathbf{Q}$.

Note that $\mathbf{D}$ is diagonal. This is a result of choosing, in the epipolar parameterization, basis directions, $\{\mathbf{r}_s, \mathbf{r}_t\}$ that are ***conjugate.*** From (21) it is easy to show that the off-diagonal components are both equal to zero:

$$\mathbf{r}_{ts}\cdot\mathbf{n} = -\mathbf{r}_t\cdot\mathbf{n}_s = -|\mathbf{r}_t|\mathbf{Q}\cdot\mathbf{n}_s = 0$$

How, in summary, can the components of $\mathbf{G}$ and $\mathbf{D}$ be computed from the deformation of apparent contours under viewer motion?

4.3.2.1 Angle Between Ray and Contour Generator, $\theta(s, t_0)$. First $\theta(s, t_0)$ can be recovered from the contour generator $\mathbf{r}(s, t_0)$ which is itself obtained from image velocities along the apparent contour via (33). This requires the numerical differentiation of depths along the contour generator. From (17) and simple trigonometry,

$$\tan\theta = \frac{\lambda|\mathbf{Q}_s|}{\lambda_s} \tag{36}$$

4.3.2.2 Normal Curvature Along the Contour Generator, κ^s. Then normal curvature along the contour generator, κ^s, is computed from the curvature κ^p of the apparent contour. Rearranging (23),

$$\kappa^s = \frac{\kappa^p\sin^2\theta}{\lambda} \tag{37}$$

4.3.2.3 Normal Curvature Along the Line of Sight, κ^t. Finally the normal curvature κ^t, along the line of sight, can be recovered from image accelerations, as explained below:

The normal curvature at P in the direction of the ray $\mathbf{Q}$, κ^t can be computed from the rate of deformation of the apparent contour under viewer motion. From (31) and (32),

$$\kappa^t = -\frac{\mathbf{U}\cdot\mathbf{n}}{\lambda(\lambda_t + \tilde{\mathbf{Q}}\cdot\mathbf{U})} \tag{38}$$

Since κ^t depends on λ_t, it is clear from (33) that κ^t is a function of viewer *acceleration* ($\mathbf{U}_t$ and $\mathbf{\Omega}_t$) and the *second* derivative of image position, $\tilde{\mathbf{Q}}_{tt}$, that is, image acceleration. By differentiating (33) and substituting (38) we find that the normal component of image acceleration at an apparent contour is determined by viewer motion (including translational and rotational accelerations) in addition to a dependency on depth and surface curvature:

$$\begin{aligned}\tilde{\mathbf{Q}}_{tt}\cdot\mathbf{n} = &-\frac{(\mathbf{U}\cdot\mathbf{n})^2}{(\lambda)^3}\left[\frac{1}{\kappa^t}\right] - 2\frac{(\tilde{\mathbf{Q}}\cdot\mathbf{U})(\mathbf{U}\cdot\mathbf{n})}{(\lambda)^2} \\ &-\frac{\mathbf{U}_t\cdot\mathbf{n}}{\lambda} - (\mathbf{\Omega}_t\wedge\tilde{\mathbf{Q}})\cdot\mathbf{n} \\ &-\frac{2(\tilde{\mathbf{Q}}\cdot\mathbf{U})(\mathbf{\Omega}\wedge\tilde{\mathbf{Q}})\cdot\mathbf{n}}{\lambda} \\ &+\frac{(\mathbf{\Omega}\wedge U)\cdot\mathbf{n}}{\lambda} + (\mathbf{\Omega}\cdot\tilde{\mathbf{Q}})(\mathbf{\Omega}\cdot\mathbf{n})\end{aligned} \tag{39}$$

The details of equation (39) are not important. It merely demonstrates that the recovery of κ^t requires knowledge of viewer motion (including translational and rotational accelerations) together with measurement of image accelerations. In section 5 it will be shown how to cancel the undesirable dependency on viewer accelerations and rotations.

Note two important points:

— As a result of the *conjugacy* relationship between the direction of the contour generator and the ray, surface curvature at a contour generator is completely determined by the *normal* curvatures in these two directions and the angle between them, θ. Compare this to a general surface point which requires the *normal* curvatures in three directions.

— Determining surface curvature usually requires the computation of second-order spatial derivatives of depth, λ. At extremal boundaries, however, only first-order spatial derivatives, λ_s, and temporal derivatives, λ_t, need be computed. One derivative is performed, effectively, by the physical system. This is also the case with specularities [8].

4.3.3 Sidedness of Apparent Contour and Contour Generator. In the static analysis of the apparent contour it was assumed that the "sidedness" of the contour generator—on which side of the image contour the obscuring surface lies—was known. Up to now in the dynamic analysis of apparent contours an arbitrary direction has been chosen for the s-parameter curve (and hence the image tangent $\mathbf{Q}_s$) and the surface orientation, $\mathbf{n}$, has been recovered up to an unknown sign from (19). The actual sign can now be determined from the deformation of the apparent contour. Equation (38) determines both a sign and magnitude for normal curvature along the ray, κ^t. This must, however, be convex and so its sign given by equation (38) allows us to infer the correct orientation of the tangent plane. This is an important qualitative geometric cue. The distinction between free space and solid surface is extremely useful in visual navigation and manipulation.

4.3.4 Gaussian and Mean Curvature. Although the first and second fundamental forms completely characterize the local 3D shape of the surface, it is sometimes more convenient to express the geometry of the surface by its *principal* curvatures and their geometric and arithmetic means: the Gaussian and mean curvature.

The Gaussian curvature, K, at a point is given by the product of the two *principal* curvatures [18]. With the epipolar parameterization, Gaussian curvature can be expressed as a product of two curvatures: the normal curvature κ^t and the curvature of the apparent contour, κ^p scaled by inverse-depth.[8]

$$K = \frac{\kappa^p \kappa^t}{\lambda}$$

This is the well-known result of Koenderink [29, 31] extended here to recover the magnitude as well as the sign of Gaussian curvature. The *mean curvature*, H, and the *principal* curvatures κ_1, κ_2 can similarly be expressed by

$$H = \frac{1}{2}\left[\frac{\kappa^p}{\lambda} + \kappa^t \operatorname{cosec}^2 \theta\right] \tag{40}$$

$$\kappa_{1,2} = H \pm \sqrt{H^2 - K}. \tag{41}$$

4.4 Degenerate Cases of the Epipolar Parameterization

In the previous section we introduced the epipolar parameterization and showed how to recover the 3D local shape of surfaces from the deformation of apparent contours.

There are two possible cases where degeneracy of the parameterization arises. These occur when $\{\mathbf{r}_s, \mathbf{r}_t\}$ fails to form a basis for the tangent plane.

Case 1. $\mathbf{r}_t = 0$

The contour generator does not *slip* over the surface with viewer motion but is fixed. It is therefore not an extremal boundary but a 3D rigid space curve (surface marking or discontinuity in depth or orientation).

An important advantage of the epipolar parameterization is its unified treatment of the image motion of curves. The projections of surface markings and creases can be simply treated as the limiting cases of apparent contours of surfaces with infinite curvature, κ^t (from (28)). In fact the magnitude of the curvature, κ^t, can be used to discriminate these image curves from apparent contours. The parameterization degrades gracefully and hence this condition does not pose any special problems.

Although the space curve $\mathbf{r}(s, t_0)$ can still be recovered from image velocities via (33) the surface orientation is no longer completely defined. The tangency conditions ((11) and (18)) are no longer valid and the surface normal is only constrained by (17) to be perpendicular to the tangent to the space curve, leaving one degree of freedom unknown.

Case 2. $\mathbf{r}_s \wedge \mathbf{r}_t = 0$ and $\mathbf{r}_t \neq 0$
Not surprisingly, the parameterization degenerates at the singularity of the surface-to-image mapping where $\mathbf{r}_s$ and $\mathbf{r}_t$ are parallel on the surface. A cusp ($|\mathbf{Q}_s| = 0$) is generated in the projection of the contour generator. For generic cases the image contour appears to come to a halt at isolated points.

Although the epipolar parameterization and equations (19) and (32) can no longer be used to recover depth and surface orientation at the isolated cusp point, this in general poses no problems. By tracking the contour-ending it is still possible in principle to recover the distance to the surface at a cusp point and the surface orientation [22].

5 Motion Parallax and the Robust Estimation of Surface Curvature

It has been shown that although it is feasible to compute surface curvature from the observed deformation of an apparent contour, this requires knowledge of the viewer's translational and rotational velocities and accelerations. Moreover the computation of surface curvature from the deformation of apparent contours is highly sensitive to errors in assumed ego-motion. This may be acceptable for a moving camera mounted on a precision robot arm or when a grid is in view so that accurate visual calibration of the camera position and orientation can be performed [47]. In such cases it is feasible to determine motion to the required accuracy of around 1 part in 1000 (see later). However, when only crude estimates of motion are available another strategy is called for. In such a case, it is sometimes possible to use the crude estimate to bootstrap a more precise visual ego-motion computation [24]. However this requires an adequate number of identifiable corner features, which may not be available in an unstructured environment. Moreover, if the estimate is too crude the ego-motion computation may fail; it is notoriously ill conditioned [48].

The alternative approach is to seek geometric cues that are much less sensitive to error in the motion estimate. In this section, it is shown that estimates of surface curvature based on the relative image motion of nearby points in the image—*parallax*-based measurements—have just this property. Such estimates are stable to perturbations of assumed ego-motion. Intuitively it is relatively difficult to judge, moving around a smooth, featureless object, whether its silhouette is extremal or not—whether curvature along the contour is bounded or not. This judgement is much easier to make for objects which have at least a few surface features. Under small viewer motions, features are "sucked" over the extremal boundary, at a rate which depends on surface curvature.

The theoretical findings of this section exactly reflect the intuition that the "sucking" effect is a reliable indicator of relative curvature, regardless of the exact details of the viewer's motion. It is shown that relative measurements of surface curvature across two adjacent points are entirely immune to uncertainties in the viewer's rotational velocity. This is somewhat related to earlier results showing that relative measurements of this kind are important for depth measurement from image velocities [32, 36, 30, 14], or stereoscopic disparities [52] and for curvature measurements from stereoscopically viewed highlights [8].

Furthermore, it will be shown that, unlike the interpretation of single-point measurements, differences of measurements at two points are insensitive to errors in rotation and in translational acceleration. Only dependence on translational velocity remains. Typically, the two feaures might be one point on an extremal boundary and one fixed surface point. The surface point has infinite curvature and therefore acts simply as a stable reference point for the measurement of curvature at the extremal boundary. The reason for the insensitivity of relative curvature measurements is that global additive errors in motion measurement are canceled out.

5.1 Motion Parallax

Consider two visual features whose projections on the image sphere (figure 4) are $\tilde{\mathbf{Q}}(s_i, t)$, $i = 1, 2$ (which we will abbreviate to $\tilde{\mathbf{Q}}^{(i)}$, $i = 1, 2$) and which have image velocities given by (27):

$$\tilde{\mathbf{Q}}_t^{(i)} = [(\mathbf{U} \wedge \tilde{\mathbf{Q}}^{(i)}) \wedge \tilde{\mathbf{Q}}^{(i)}] \quad \frac{1}{\lambda^{(i)}} \quad - \Omega \wedge \tilde{\mathbf{Q}}^{(i)} \qquad (42)$$

Point image velocities consist of two components. The first is due to viewer translation and it is this component

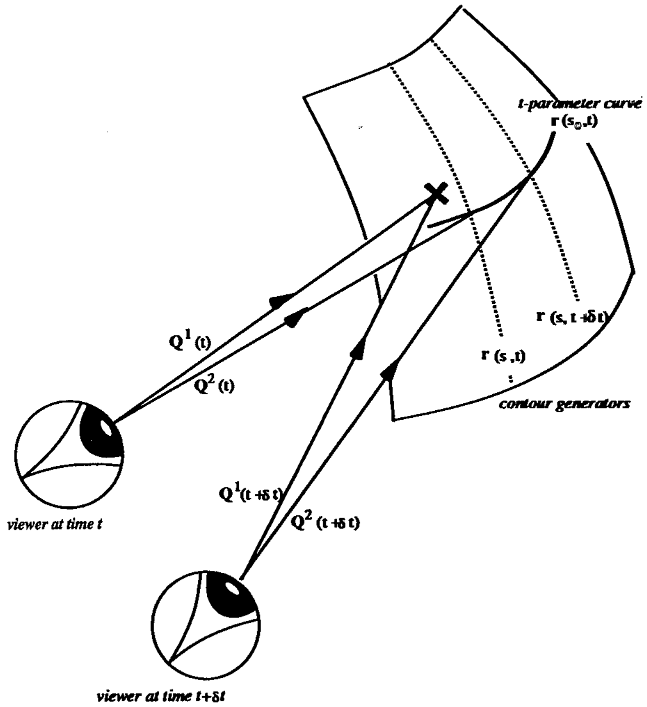

Fig. 4. Motion parallax. Consider the relative displacement between a point on an apparent contour and the image of a nearby surface feature (shown as a cross): $\delta = \mathbf{Q}^{(2)} - \mathbf{Q}^{(1)}$. The rate of change of relative image position—*parallax*, δ_t—has been shown to be a robust indicator of relative depth. In this section we show that its temporal derivative—the rate of parallax δ_{tt}—is a robust geometric cue for the recovery of surface curvature at extremal boundaries.

that encodes scene structure (depth). The other component is due to the rotational part of the viewer's motion and is independent of scene strucure. Clearly depth can only be recovered accurately if rotational velocity $\mathbf{\Omega}$ is known. The dependence on rotational velocity is removed if, instead of using raw image motion $\tilde{\mathbf{Q}}_t$, the difference of the image motions of a pair of points, $\tilde{\mathbf{Q}}^{(1)}$, $\tilde{\mathbf{Q}}^{(2)}$, is used. This is called *parallax* [25]. (See figure 4.)

The relative image position δ of the two points is

$$\delta(t) = \tilde{\mathbf{Q}}^{(2)} - \tilde{\mathbf{Q}}^{(1)} \qquad (43)$$

Parallax is the temporal derivative of δ, δ_t. If instantaneously the two points project to the same point on the image sphere, so that

$$\tilde{\mathbf{Q}}^{(1)}(0) = \tilde{\mathbf{Q}}^{(2)}(0) = \tilde{\mathbf{Q}}$$

then, from (42), the parallax δ_t depends only on their relative inverse-depths and on viewer translational velocity. It is independent of (and hence insensitive to errors in) angular rotation $\mathbf{\Omega}$:

$$\delta_t = [(\mathbf{U} \wedge \tilde{\mathbf{Q}}) \wedge \tilde{\mathbf{Q}}] \left[\frac{1}{\lambda^{(2)}} - \frac{1}{\lambda^{(1)}} \right] \qquad (44)$$

The use of "motion parallax" for robust determination of the direction of translation $\mathbf{U}$ and relative depths from image velocities was described by Longuet-Higgins and Prazdny [36] and Rieger and Lawton [45].

5.2 Rate of Parallax

Following from the well-known results about motion parallax, we derive the central result of this section—that the *rate* of parallax is a robust cue for surface curvature. The direct formula (39) for normal curvature κ^t in terms of image acceleration was sensitive to viewer translational acceleration and rotational velocity and acceleration. If, instead, *differences* of image accelerations are used, the undesirable sensitivity vanishes.

The relationship between image acceleration and normal curvature for points $\tilde{\mathbf{Q}}^{(1)}$, $\tilde{\mathbf{Q}}^{(2)}$ can be expressed as:

$$
\begin{aligned}
\tilde{\mathbf{Q}}^{(i)}_{tt} \cdot \mathbf{n}^{(i)} = &- \frac{(\mathbf{U} \cdot \mathbf{n}^{(i)})^2}{(\lambda^{(i)})^3} \left[\frac{1}{\kappa^{ti}} \right] \\
&-2 \, \frac{(\mathbf{U} \cdot \tilde{\mathbf{Q}}^{(i)})(\mathbf{U} \cdot \mathbf{n}^{(i)})}{(\lambda^{(i)})^2} \\
&- \frac{\mathbf{U}_t \cdot \mathbf{n}^{(i)}}{\lambda^{(i)}} - (\boldsymbol{\Omega}_t \wedge \tilde{\mathbf{Q}}^{(i)}) \cdot \mathbf{n}^{(i)} \\
&- \frac{2(\mathbf{U} \cdot \tilde{\mathbf{Q}}^{(i)}(\boldsymbol{\Omega} \wedge \tilde{\mathbf{Q}}^{(i)}) \cdot \mathbf{n}^{(i)}}{\lambda^{(i)}} \\
&+ \frac{(\boldsymbol{\Omega} \wedge \mathbf{U}) \cdot \mathbf{n}^{(i)}}{\lambda^{(i)}} \\
&+ (\boldsymbol{\Omega} \cdot \tilde{\mathbf{Q}}^{(i)})(\boldsymbol{\Omega} \cdot \mathbf{n}^{(i)})
\end{aligned}
\tag{45}
$$

where $\mathbf{n}^{(i)}$ is now taken to be a vector perpendicular to $\tilde{\mathbf{Q}}^{(i)}$ and will only be a surface normal if the point belongs to an apparent contour. The important point is that the two copies of this equation for the two positions $i = 1, 2$ can be subtracted, canceling the undesirable dependency on $\boldsymbol{\Omega}$, $\boldsymbol{\Omega}_t$ and on $\mathbf{U}_t$ if the images of the two points are instantaneously coincident.

Think of the two points as being the projection of extremal contour generators, which trace out curves with (normal) curvatures κ^{t1} and κ^{t2} as the viewer moves. Let us define the relative inverse curvature, ΔR, of the feature pair by

$$\Delta R = \frac{1}{\kappa^{t2}} - \frac{1}{\kappa^{t1}} \tag{46}$$

Note that it is simply the difference of the radii of curvature of the normal sections.

Consider the two features to be instantaneously spatially coincident, that is, initially, $\tilde{\mathbf{Q}}(s_1, 0) = \tilde{\mathbf{Q}}(s_2, 0)$. Moreover assume they lie at a common depth λ, and hence, instantaneously, $\tilde{\mathbf{Q}}^{(1)}_t = \tilde{\mathbf{Q}}^{(2)}_t$. In practice, of course, the feature pair will only coincide exactly if one of the points is a surface marking which is instantaneously on the extremal boundary (figure 4). The effect of a small separation is analyzed below. Now, taking the difference of equation (45) for $i = 1, 2$ leads to the following relation between the two relative quantities $\boldsymbol{\delta}_{tt}$ (rate of parallax) and ΔR (relative inverse curvature):

$$\boldsymbol{\delta}_{tt} \cdot \mathbf{n} = - \frac{(\mathbf{U} \cdot \mathbf{n})^2}{\lambda^3} \Delta R \tag{47}$$

where $\mathbf{n}$ is a vector perpendicular to the direction of the ray ($\tilde{\mathbf{Q}} \cdot \mathbf{n} = 0$), for example, the surface normal of one of the two points. The two points will not necessarily have the same surface normal. From this equation we can obtain relative inverse curvature, ΔR, as a function of depth λ, viewer velocity $\mathbf{U}$, and the second temporal derivative of $\boldsymbol{\delta}$. *Dependence on viewer motion is now limited to the velocity* $\mathbf{U}$. There is no dependence on viewer acceleration or rotational velocity. Hence the relative measurement should be much more robust. (Computationally higher derivatives are generally far more sensitive to noise.)

In the case that $\tilde{\mathbf{Q}}^{(1)}$ is known to be a fixed surface reference point, with $1/\kappa^{t1} = 0$, then $\Delta R = 1/\kappa^{t2}$ so that the relative inverse curvature ΔR constitutes an estimate, now much more robust, of the normal curvature κ^{t2} at the extremal boundary point $\tilde{\mathbf{Q}}^{(2)}$. Of course this can now be used in equations (24), (34), and (35) to obtain robust estimates of surface curvature. This is confirmed by the experiments in the next section.

Note that the use of the epipolar parameterization is not important in the above analysis. It can be shown that the normal component of the relative image acceleration $\boldsymbol{\delta}_{tt} \cdot \mathbf{n}$ between a distinct feature and an apparent contour is independent of viewer motion and can be determined completely from spatiotemporal measurements on the image (appendix A).

5.3 Degradation of Sensitivity with Separation of Points

The theory above relating relative curvature to the rate of parallax assumed that the two points $\tilde{\mathbf{Q}}^{(1)}$ and $\tilde{\mathbf{Q}}^{(2)}$ were instantaneously coincident in the image and at the same depth, $\lambda^{(1)} = \lambda^{(2)}$. In practice, point pairs used as features will not coincide exactly and an error limit on curvature (or, more conveniently, its inverse) must be computed to allow for this. The relative curvature can still be computed from the rate of parallax by taking into account an error,

$$\Delta R = \Delta \hat{R} + R^{\text{error}} \tag{48}$$

where $\Delta \hat{R}$ is the estimate of the relative inverse curvature computed from the rate of parallax.

$$\Delta \hat{R} = - \frac{\lambda^3}{(\mathbf{U} \cdot \mathbf{n})^2} \boldsymbol{\delta}_{tt} \cdot \mathbf{n} \tag{49}$$

The error in the radii of curvature, R^{error}, consists of errors due to the difference in depths of the two features, $\delta\lambda$; the finite separation in the image, $\boldsymbol{\delta} = \Delta\tilde{\mathbf{Q}}$ and the differences in tangent planes of the two features, $\Delta\mathbf{n}$. The magnitude of these effects can be easily computed from the difference of equation (45) for the two points (appendix B). For nearby points that are on the same surface and for fixation ($\mathbf{U} = \lambda\mathbf{\Omega} \wedge \tilde{\mathbf{Q}}$) the dominant error can be conveniently expressed as:

$$|R^{\text{error}}| \approx 9\lambda|\delta| + \lambda^3|\delta|\frac{|\mathbf{\Omega}_{\mathbf{t}}|}{(\mathbf{U}\cdot\mathbf{n})^2} + \lambda|\Delta\lambda|\frac{\mathbf{U}_t\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2} \tag{50}$$

Parallax-based measurements of curvature will usually be accurate and insensitive to errors in viewer motion if the separation between points on nearby contours satisfies

$$|\delta| \ll \frac{\Delta R}{9\lambda} \tag{51}$$

Equation (50) can also be used to predict the residual sensitivity to translational and rotational accelerations. The important point to notice is that sensitivity to viewer motion is still reduced. As an example consider the sensitivity of absolute measurements of surface curvature along the ray to error in viewer position. Think of this as adding an unknown translational acceleration, $\mathbf{U}_t$. For absolute measurements (45) the effect of this unknown error is amplified by a factor of $\lambda^2/(\mathbf{U}\cdot\mathbf{n})^2$ when estimating surface curvature. From appendix B and (50) we see that for parallax-based measurements the sensitivity is reduced to a factor of $2\Delta\lambda/\lambda$ of the original sensitivity. This sensitivity vanishes, of course, when the features are at the same depth. A similar effect is observed for rotational velocities and accelerations.

5.4 Qualitative Shape

Further robustness can be obtained by considering the *ratio* of *relative curvatures*. More precisely this is the ratio of differences in radii of curvature. Ratios of pairs of parallax-based measurements can, in theory, be completely insensitive to viewer motion. This is because the normal component of the relative image acceleration $\boldsymbol{\delta}_{tt}\cdot\mathbf{n}$ can be shown to be independent of the viewer motion and can be determined from spatiotemporal measurements on the image for a distinct point and apparent contour pair (appendix A). This is surprising because the epipolar parameterization has a hidden dependence on viewer velocity in the "matching" condition (26). This result is important because it demonstrates the possibility of obtaining robust inferences of surface geometry which are independent of any assumption of viewer motion.

In particular, if we consider the *ratio* of relative curvature measurements for two *different* point-pairs at similar depths, terms depending on absolute depth λ and velocity $\mathbf{U}$ are canceled out in equation (47). This result corresponds to the following intuitive idea. The rate at which surface features rush toward or away from an extremal boundary is inversely proportional to the (normal) curvature there. The constant of proportionality is some function of viewer motion and depth; it can be eliminated by considering only ratios of curvatures.

6 Implementation of Theory

In the previous sections a computational theory for the recovery of 3D shape from the deformation of apparent contours was presented. The implementation of this theory and the results of experiments performed with a camera mounted on a moving robot arm are now described. In particular this requires the accurate extraction of image curves from real images and tracking their temporal evolution to measure image velocities and accelerations. Uncertainty and sensitivity analysis is used to compute bounds on the estimates of surface curvature. This is critical in discriminating fixed features from extremal boundaries—deciding whether curvature along the ray is bounded or not—since with noisy measurements and poorly calibrated viewer motions, we must test by error analysis the hypothesis that the curvature is unbounded at a fixed feature. Sensitivity analysis is also used to substantiate the claim that *parallax* methods—using the relative image motion of nearby contours—allow the robust recovery of surface curvature.

6.1 Tracking Image Contours with B-Spline Snakes

Image contours can be localized and tracked using a variant of the well-known "snake" of Kass, Witkin, and Terzopoulos [28]. The snake is a computational construct, a dynamic curve able to track moving, deforming image features. Since many snakes can be active at once, each tracking its feature contour as a background process, they constitute a versatile mechanism for direction and focus of attention.

6.1.1 Active Contours—Snakes. Energy-minimizing active-countour models (snakes) were proposed by Kass et al. [28] as a top-down mechanism for locating features of interest in images and tracking their image motion, provided the feature does not move too fast.

The behavior of a snake is controlled by internal and external "forces."[9] The internal forces enforce smoothness and the external forces guide the active contour toward the image feature. In their implementation for image curve localization and tracking, these forces are derived by differentiating internal and external energies respectively.

- Internal energy
  The internal energy (per unit length), E_{internal}, at a point on the snake, $\mathbf{x}(s)$:
  $$E_{\text{internal}} = \frac{\alpha|\mathbf{x}_s|^2 + \beta|\mathbf{x}_{ss}|^2}{2} \qquad (52)$$
  is composed of first and second-order terms, forcing the active contour to act like a string/membrane (avoiding gaps) or a thin rod/plate (avoiding high curvatures) respectively. These effects are controlled by the relative values of α and β. The internal energy serves to maintain smoothness of the curve under changing external influences.
- External energy
  The external force is computed from the image intensity data $I(\mathbf{x}(s))$, where the position of the snake is represented by $\mathbf{x}(s)$, by differentiating an external energy (per unit length) E_{external}:
  $$E_{\text{external}} = -|\nabla G(\sigma) * I(\mathbf{x}(s))|^2 \qquad (53)$$
  which is computed after convolution of the image with the derivative of a Gaussian kernel, $\nabla G(\sigma)$, of size (scale) σ. Gaussian smoothing extends the search range of the snake by smearing out image edge features.

The goal is to find the snake (contour) that minimizes the total energy. This is achieved by the numerical solution of the elastic problem using techniques from variational calculus. The main step is the solution of a linear equation involving a banded matrix, typically in several hundred variables [28].

6.1.2 The B-Spline Snake. A more economical realization can be obtained by using far fewer state variables [46]. In [15] we proposed the use of cubic B-splines [20]. These are deformable curves represented by four or more state variables (control points). The curves may be open or closed as required. The flexibility of the curve increases as more control points are added; each additional control point allows either one more inflection in the curve or, when multiple knots are used [5], reduced continuity at one point.

B-spline snakes are ideally suited for representing, detecting, and tracking image curves. Their main advantages include:

- Local control—modifying the position of a data point or control point causes only a small part of the curve to change.
- Continuity control—B-splines are defined with continuity properties at each point.
- Compact representation—the number of variables to be estimated is reduced to the number of control points [40].

The B-spline is a curve in the image plane (figure 5)

$$\mathbf{x}(s) = \sum_i f_i(s)\mathbf{q}_i \qquad (54)$$

where f_i are the spline basis functions and $\mathbf{q}_i$ are the coefficients or control points. These are positioned so that the curve locates the desired image contour. In the original implementation the "external force" on a point $\mathbf{x}(s_j)$ was chosen to be

$$\mathbf{F}(s_j) = \nabla|\nabla G(\sigma) * I(\mathbf{x}(s_j))| \qquad (55)$$

so that, at equilibrium (when image forces vanish), the B-spline, $\mathbf{x}(s)$, stabilized close to a high-contrast contour.

In this section two major simplifications are introduced. The first concerns the control of spatial scale

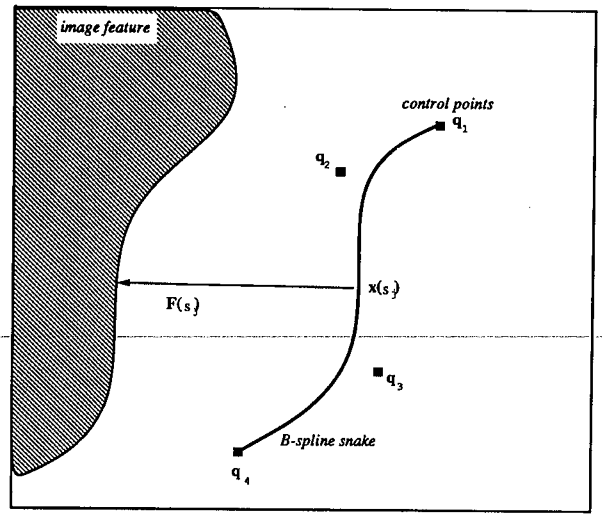

Fig. 5. The B-spline snake. The B-spline snake can be used for image contour localization and representation. A cubic B-spline can be represented by a minimum of 4 control points, $\mathbf{q}_i$, and these are positioned so as to locate a nearby contour. The snake moves under the influence of external forces, $\mathbf{F}(s_j)$, which guide it toward the image feature.

for tracking. The other simplification is that there is no need for internal forces since the B-spline representation maintains smoothness via hard constraints implicit in the representation.

6.1.2.1 Exernal Forces and Control of Scale. The "force" is chosen to depend on the distance between the feature of interest (an edge) and the approximation by the B-spline. For each sample point the "force" on the snake is found by a coarse-to-fine strategy. This is done by inspecting intensity gradients on either side of the snake (either along the normal or along a direction determined by hard constraints, for example, scanlines). Control of scale is achieved by inspecting gradients nearer to or further from the snake itself. Each point chooses to move in the direction of the largest intensity gradient (hence toward a contrast edge). If the intensity gradients either side of the contour have opposite signs the scale is halved. This is repeated until the edge has been localized to the nearest pixel.

The gradient is estimated by finite differences. Gaussian smoothing is not used.[10] Image noise is not, as might be thought, a major problem in the unblurred image since CCD cameras have relatively low signal-to-noise. Moreover, gradients are sampled at several places along the spline, and those samples combined to compute motions for the spline control points (described below, (57)). The combination of those samples itself has an adequate averaging, noise-defeating effect.

6.1.2.2 Positioning the Control Points. External forces are sampled at N points,

$$\mathbf{x}(s_j), \ j = 1, \ldots, N$$

along the curve—typically $N > 20$ has been adequate in our experiments. External forces are applied to the curve itself; but, for iterative adjustment of displacement, it is necessary to compute the force transmitted to each control point. This can be done by positioning the B-spline so that it minimizes the sum of the square of the distances between the discrete data points of the feature and the approximation by the B-spline. Effectively, each snake point is attached to a feature by an elastic membrane so that its potential energy is proportional to the distance squared. This technique has been used to represent image curves [41].

Derivation 2. If the desired feature position is given by $\mathbf{y}(s_j)$ for a point on the B-spline, $\mathbf{x}(s_j)$, we wish to minimize the potential energy:

$$\sum_j \left[\mathbf{y}(s_j) - \sum_i f_i(s_j)\mathbf{q}_i \right]^2 . \tag{56}$$

The new positions of the control points, $\mathbf{q}_i$, are chosen by solving (the least-squares solution):

$$\sum_i \mathbf{q}_i \sum_j f_i f_k = \sum_j f_k \mathbf{y}(s_j) \tag{57}$$

where k has the same range of values as the control points, i.

As the snake approaches the image contour it "locks on" and the scale is reduced to enable accurate contour localization. Since accurate measurements are required to compute image accelerations, care has been taken over subpixel resolution. At earlier stages of tracking, when coarse blurring (large scale) is used, the capture range of the snake is large but localization is poor—the snake may lag behind the contour. Once the snake has converged onto the contour, standard edge-detection techniques such as smoothing for subpixel resolution [13] are used to obtain accurate localization.

The snakes were either initialized by hand in the first frame near image contours of interest, which they then track automatically or wait in the image until they are swept by the motion of the camera over a feature for which they have an affinity have also been successful (figure 6). Tracking is maintained provided the contour does not move too quickly. The contour tracker can run at 15 Hz on a SUN4/260 [7]. By using a parallel MIMD architecture (based on 9 Transputers) and interframe constraints (simulating inertia and damping) to enhance tracking capability, the real-time tracking of 10 snakes has been achieved [17]. The adaptive choice of the number of control points; the control of the length of the contour, and the scale of the feature search are current topics of research.

6.2 *The Epipolar Parameterization*

In the *epipolar* parameterization of the spatiotemporal image and surface, a point on an apparent contour in the first image is "matched" to a point in successive images (in an infinitesimal sense) by searching along the corresponding epipolar lines. This allows us to extract a t-parameter curve from the spatiotemporal image. As shown in the previous sections, depth and surface curvature are then computed from first- and second-order temporal derivatives of this t-parameter image

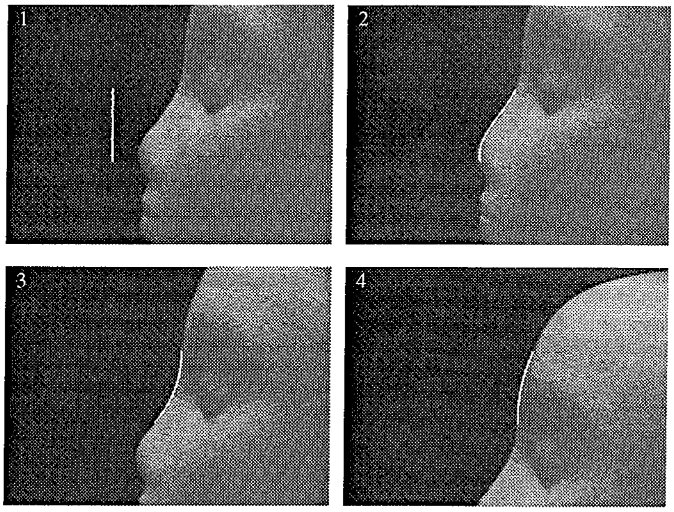

Fig. 6. Tracking image contours with B-spline snakes. A single span B-spline snake "hangs" in the image until it is swept by the motion of the camera into the vicinity of a high-contrast edge (top left). The snake then tracks the deforming image contour as the camera is moved vertically upward by the robot. Four samples of an image sequence are shown in which the robot moves with a speed of 20 mm/s. Tracking speeds of 15 Hz have been achieved without special-purpose hardware by windowing and avoiding Gaussian smoothing.

curve by equations (32) and (39). This is a nontrivial practical problem since the epipolar structure is continuously changing for arbitrary viewer motions. It requires a dense image sequence and knowledge of the geometrical and optical characteristics of the camera (the *intrinsic* parameters, for example, image center, pixel size, and focal length [19, 21, 50]) as well as the camera motion.

Estimates of the camera motion are either determined directly from the position and orientation of the gripper and its relationship with the camera center [49] or are obtained by visual calibration techniques [47]. Extraction of the t-parameter curve can be done in a number of ways. We have implemented two simple methods. The first is an extension of epipolar plane image analysis [10, 53] and allows the recovery of depth and surface curvature at a point. The second method analyses the case of extended displacements and arbitrary rotations of the viewer to recover constraints (bounds) on surface curvature.

The *epipolar* parameterization of the image is greatly simplified for simple motions. In particular, if we consider linear viewer motion perpendicular to the optical axis, epipolar lines are simply corresponding raster lines of subsequent images. Figure 7 shows the spatiotemporal image formed by taking a sequence of images in rapid succession and stacking these sequentially in time. For linear motions of the viewer, the t-parameter image curves are trajectories lying in horizontal slices of the spatiotemporal image. Each horizontal slice corresponds to a different epipolar plane. The trajectories of the image positions of points on an apparent contour (A) and a nearby surface marking (B) are shown as a function of time in the spatiotemporal cross-section of figure 7 and plotted in figure 8. Note that this is a simple extension of epipolar plane image analysis in which the trajectories of fixed, rigid features appear as straight lines in the spatiotemporal cross-section image with a gradient that is proportional to inverse depth [10]. For apparent contours however, the trajectories are no longer straight. It is shown below that the gradient of the trajectory still encodes depth. The curvature determines the curvature of the surface in the epipolar plane.

6.2.1 Estimation of Depth and Surface Curvature. For motion perpendicular to the optical axis and for an apparent contour which at time t is instantaneously aligned with the optical axis (image position in mm $X(0) = 0$), it is easy to show (from (27) and (39)) that the image velocity, $X_t(0)$, and acceleration, $X_{tt}(0)$, along the scan-line are given by

$$X_t(0) = -\frac{fU}{\lambda} \tag{58}$$

$$X_{tt}(0) = -\frac{fU^2R}{\lambda^3} \tag{59}$$

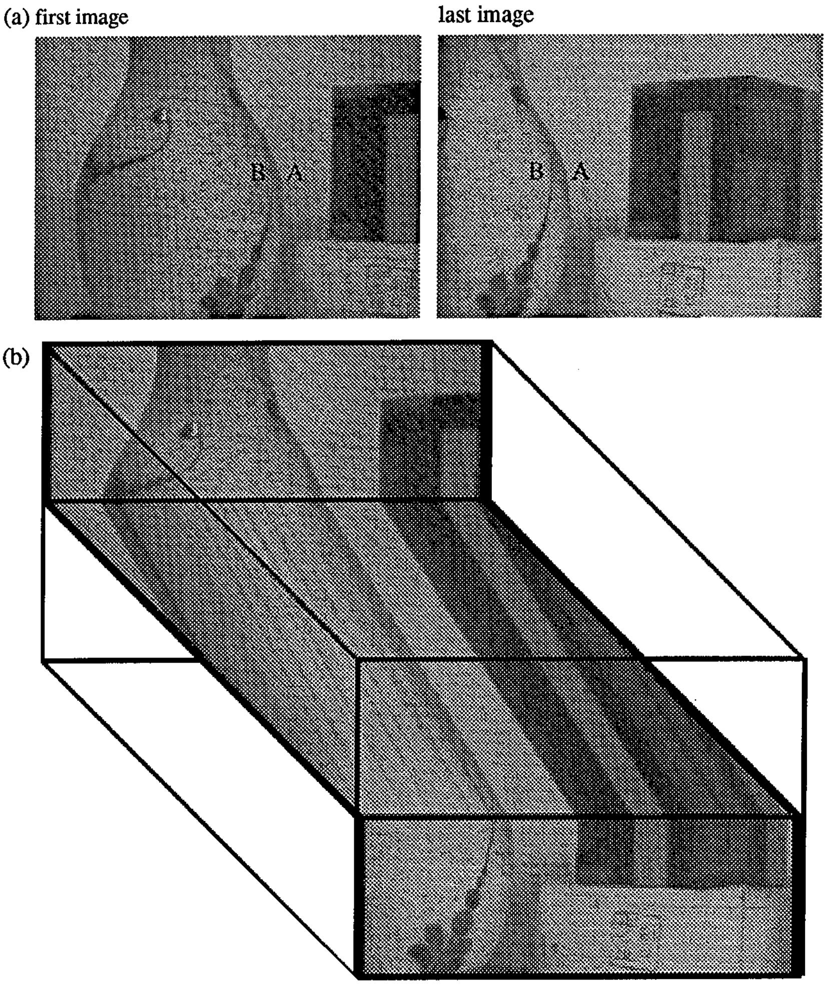

Fig. 7. 3D spatiotemporal image. (a) The first and last image from an image sequence taken from a camera mounted on a robot arm and moving horizontally from left to right without rotation. (b) The 3D spatiotemporal image formed from the image sequence piled up sequentially with time. The top of the first image and the bottom of the last image are shown along with the spatiotemporal cross-section corresponding to the same epipolar plane. For simple viewer motions consisting of camera translations perpendicular to the optical axis the spatiotemporal cross-section image is formed by storing the scan-lines (epipolar lines) for a given epipolar plane sequentially in order of time [16].

where U is the component of viewer translational velocity perpendicular to the optical axis and parallel to the scan line; f is the focal length of the CCD camera lens, and R is the radius of curvature of the t-parameter curve lying in the epipolar plane. The radius of curvature is related to the curvature of the normal section by Meusnier's formula [18]. The normal curvature along the ray is given by

$$\kappa^t = \frac{1}{R}\cos\phi \qquad (60)$$

where $\cos\phi$ is the angle between the surface normal and the epipolar plane. In this case, it is simply equal to the angle between the image curve normal and the horizontal scan line.

The estimate of depth and surface curvature follow directly from the first and second temporal derivatives of the t-parameter curve, $X(t)$ and equations (58, 59). Due to measurement noise and vibrations of the robot arm, the trajectory may not be smooth (figures 8, 9, 10) and so these derivatives are computed from the coefficients of a parabola fitted locally to the data by least-squares estimation. The uncertainty due to random image localization and ego-motion errors can be derived from analysis of the residual errors [44, 14].

In practice, the viewer will not execute simple translational motions perpendicular to the optical axis

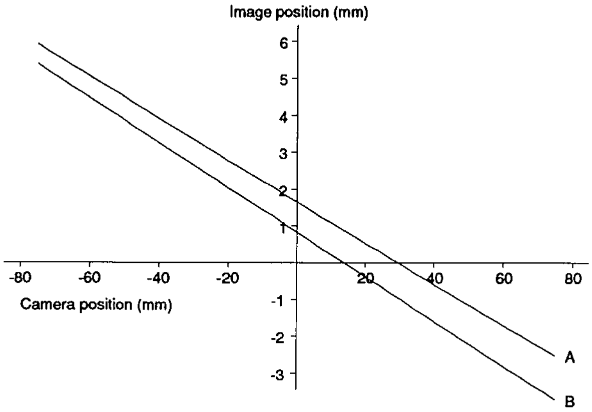

Fig. 8. Spatiotemporal cross-section image trajectories. For linear motion and epipolar parameterization the t-parameter surface curves lie in the epipolar plane. The t-parameter spatiotemporal image trajectory is also planar. The gradient and curvature of this trajectory encode depth to the contour generator and curvature in the epipolar plane respectively.

but will rotate to fixate on an object of interest. For linear translational viewer motions with known camera rotations, the analysis of epipolar plane images is still appropriate if we rectify the detected image curves. Rectification can be performed by a 3×3 rotation matrix relating measurements in the rotated coordinate frame to the standard parallel geometry frame.

For arbitrary curvilinear motions the t-parameter curves are no longer constrained to a single cross-section of the spatiotemporal image. Each time instant requires a different epipolar structure and so extracting the t-parameter curve from the spatiotemporal image poses a more difficult practical problem. From at least three discrete views, it is possible to determine whether or not a contour is extremal. For a surface marking or crease (discontinuity in surface orientation), the three rays should intersect at a point in space for a static scene. For an extremal boundary, however, the contact point slips along a curve, $\mathbf{r}(s_0, t)$ and the three rays will not intersect (figures 11 and 12). In [14, 51] a simple numerical method for estimating depth and surface curvatures from a minimum of three discrete views, by determining the *osculating* circle in each epipolar plane, is described.

6.2.2 Experimental Results—Curvature from the Spatiotemporal Image. Figure 7 shows the t-parameter trajectories for both a feature on an apparent contour (A) and a nearby surface marking (B). The trajectories are both approximately linear with a gradient that determines the distance for the feature. Depth can be estimated to an accuracy of 1 part in 1000 (table 1).

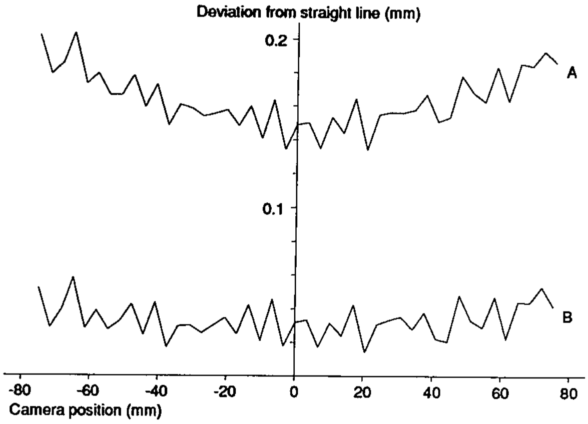

Fig. 9. Deviation from the straight line trajectory. The curvature of the spatiotemporal trajectories is used to estimate the curvature of the epipolar section. The trajectories are not smooth due to vibrations of the robot manipulator (amplitude 0.2 mm). Their effect on the estimation of curvature is reduced by a least-squares fit to the data. The surface curvatures (expressed as radii of curvature) at A and B are estimated as 51.4 ± 8.2 mm and 11.8 ± 7.3 mm respectively. B is not on an extremal boundary but is on a fixed curve. This is a degenerate case of the parameterization and should ideally have zero "radius of curvature," that is, the spatiotemporal trajectory should be a straight line.

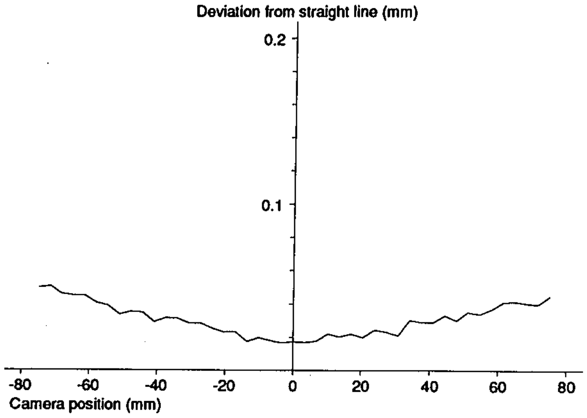

Fig. 10. Relative image positions. The effect of robot vibrations is greatly reduced if the *rate of parallax* (relative image positions) is used instead.

The effect due to surface curvature is very difficult to discern. This is easily seen, however, if we look at the deviation of image position away from the straight-line trajectory of a feature at a fixed depth (figure 9). Notice that the image position is noisy due to perturbations in the robot position. Typically the robot vibrations

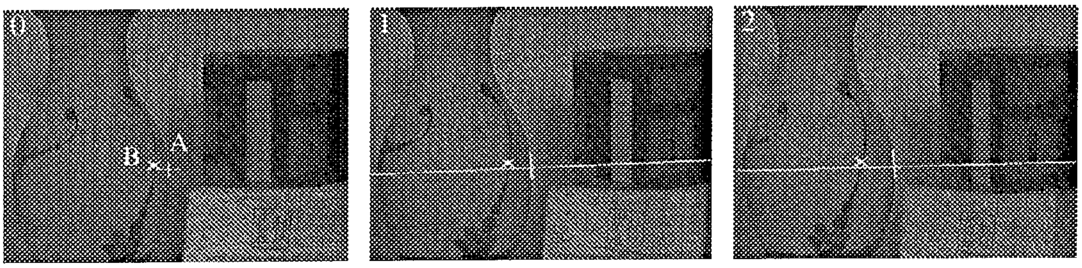

Fig. 11. Estimating surface curvatures from three discrete views. Points are selected on image contours in the first view (t_0), indicated by crosses A and B for points on an extremal boundary and surface marking respectively. For *epipolar parameterization* of the surface, corresponding features lie on epipolar lines in the second and third view (t_1 and t_2). Measurement of the three rays lying in an epipolar plane can be used to estimate surface curvatures (figure 12).

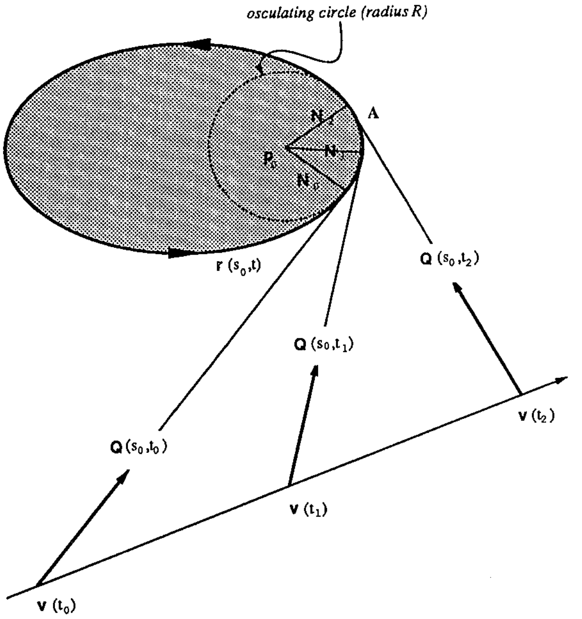

Fig. 12. The epipolar plane. Each view defines a tangent to $\mathbf{r}(s_0, t)$. For linear camera motion and epipolar parameterization the rays and $\mathbf{r}(s_0, t)$ lie in a plane. If $\mathbf{r}(s_0, t)$ can be approximated locally as a circle, it can be uniquely determined from measurements in three views.

Table 1. Radius of curvature of the *epipolar section* estimated from the spatiotemporal trajectory for a point on an extremal boundary (A) and on a surface marking (B).

	Measured Depth	Measured Curvature	Actual Curvature
Extremal boundary, A	424.3 ± 0.5 mm	51.4 ± 8.2 mm	37 ± 2 mm
Surface marking, B	393.9 ± 0.4 mm	11.8 ± 7.3 mm	0
Parallax measurement (A)	424.3 ± 0.5 mm	39.6 ± 2 mm	37 ± 2 mm

have amplitudes between 0.1 mm and 0.2 mm. From (39) we see that these vibrations are amplified by a factor depending on the square of the distance to the feature, and that this results in a large uncertainty in the estimate of surface curvature. Equations (58) and (59) are used to estimate the depth and curvature for a point on the extremal boundary of the vase (A) by fitting a parabola to the spatiotemporal trajectory. The method is repeated for a point that is not on an extremal boundary but is on a nearby surface marking (B). This is a degenerate case of the parameterization. A surface marking can be considered as the limiting case of a point with infinite curvature and hence ideally will have zero "radius or curvature." The estimates of depth and curvature are shown in table 1. The veridical values of curvature were measured using calipers. Note that there is a systematic error, not explained by the random errors in the data. This is possibly due to an error in the assumed ego-motion of the robot or focal length.

Figure 10 plots the relative image position between A and B against robot position (time). The curvature of this parabola also encodes the surface curvature at A as predicted by (47). The parabola is considerably smoother since the effects of robot "wobble" are attenuated when making relative measurements. This is because *the amplification of robot vibrations is reduced* by an order of magnitude. The exact factor depends on the difference of depths between the two features, as predicted by (50). In this experiment this corresponds to an order of magnitude. This results in a greatly reduced uncertainty in the estimate of relative inverse curvature. Better still, the estimate of surface curvature based on relative measurements is also more accurate. As predicted by the theory, the nearby point B acts as a stable reference point. Global additive errors in the robot motion effect the visual motion of both A and B and hence can be canceled out when the differences of positions are used.

6.3 Error and Sensitivity Analysis

The previous section showed that the visual motion of apparent contours can be used to estimate surface curvatures of a useful accuracy if the viewer ego-motion is known. However, the estimate of curvature is very sensitive to perturbations in the motion parameters and errors in image contour localization. The effect of small errors and uncertainties can be computed by first-order perturbation analysis [14].

The effects of small errors in the assumed ego-motion—position and orientation of the camera—are plotted in figures 13a and 13b (curves labeled I). Accuracies of 1 part in 1000 in the measurement of ego-motion are essential for surface curvature estimation.

Parallax based methods measuring surface curvature are in principle based on measuring the relative image motion of nearby points on different contours (47). In practice, this is equivalent (46) to computing the difference of radii of curvature at the two points, say A and B (figure 7). The radius of curvature measured at a surface marking is determined by errors in image measurement and ego-motion. (For a precisely known viewer motion and for exact contour localization the radius of curvature would be zero at a fixed feature.) It can be used as a reference point to subtract the global additive errors due to imprecise motion when estimating the curvature at the point on the extremal boundary. Figures 13a and 13b (curves labeled II) show how the sensitivity of the relative inverse curvature, ΔR, to error in position and rotation computed between points A and B (two nearby points at similar depths) is reduced by an order of magnitude. This is a striking decrease in sensitivity even though the features do not coincide exactly as the theory required.

Note that the sensitivity to image localization errors remains (figure 14). This sensitivity is not reduced by parallax and requires subpixel localization of image contours.

7 Applications

7.1 Discriminating Between Fixed Features and Extremal Boundaries

The magnitude of the estimate of the radius of curvature, R, can be used to determine whether a point on an image contour lies on an apparent contour or on the projection of a fixed surface feature such as a crease, shadow, or surface marking.

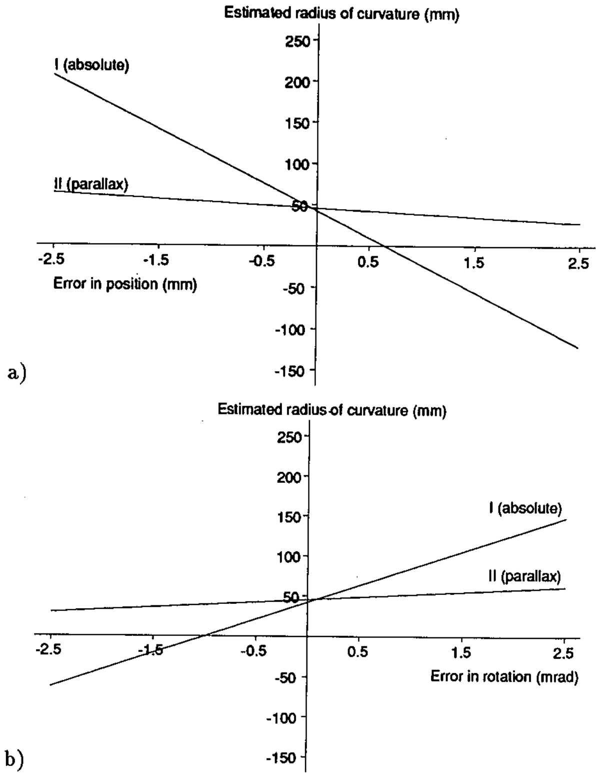

Fig. 13. Sensitivity of curvature estimated from absolute measurements and parallax to errors in motion. (a) The radius of curvature ($R = 1/\kappa^t$) for a point on the extremal boundary (A) is plotted as a function of errors in the camera position (a) and orientation (b). Curvature estimation is highly sensitive to errors in ego-motion. Curve I shows that a perturbation of 1 mm in position (in a translation of 100 mm) produces an error of 155% in the estimated radius of curvature. A perturbation of 1 mrad in rotation about an axis defined by the epipolar plane (in a total rotation of 200 mrad) produces an error of 100%. (b) However, if parallax-based measurements are used the estimation of curvature is much more robust to errors in ego-motion. Curve II shows the difference in radii of curvature between a point on the extremal boundary (A) and the nearby surface marking (B) plotted against error in the position (a) and orientation (b). The sensitivity is reduced by an order of magnitude, to 19% per mm error and 12% per mrad error respectively.

With noisy image measurements or poorly calibrated motion we must test by error analysis the hypothesis that R is not equal to zero for an extremal boundary. We have seen how to compute the effects of small errors in image measurement, and ego-motion. These are conveniently represented by the *covariance* of the estimated curvature, σ_R. The estimate of the radius of curvature and its uncertainty is then used to test the hypothesis of an extremal boundary. In particular if we assume that the error in the estimate of the radius has a normal distribution (as an approximation to the student-t

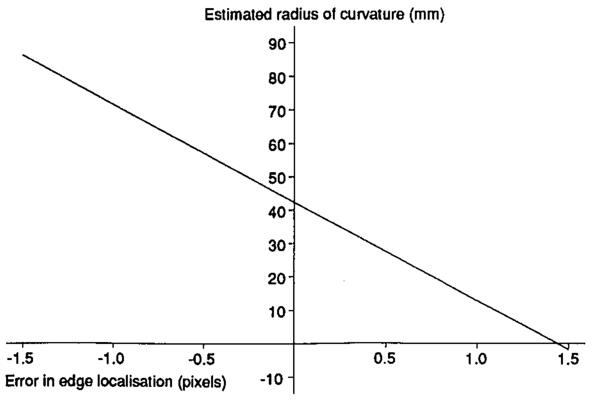

Fig. 14. Sensitivity of curvature estimate to errors in image contour localization.

distribution [44]), the image contour is assumed to be the projection of a fixed feature (within a confidence interval of 95%) if

$$-1.96\sigma_R < R < 1.96\sigma_R \tag{61}$$

Using *absolute measurements*, however, the discrimination between fixed and extremal features is limited by the uncertainties in robot motion. For the image sequence of figure 11 it is only possible to discriminate between fixed features and points on extremal boundaries with inverse curvatures greater than 15 mm. High curvature points ($R < 1.96\sigma_R$) cannot be distinguished from fixed features and will be incorrectly labeled.

By using parallax-based (*relative*) measurements the discrimination is greatly improved and is limited by the finite separation between the points as predicted by (50). For the example of figure 11 this limit corresponds to a relative curvature of approximately 3 mm. This, however, requires that we have available a fixed nearby reference point.

Suppose now that no known surface feature has been identified in advance. Can the robust relative measurements be made to bootstrap themselves without an independent surface reference? It is possible by relative (two-point) curvature measurements obtained for a small set of nearby points to determine pairs which are fixed features. They will have zero relative radii of curvature. Once a fixed feature is detected it can act as stable reference for estimating the curvature at extremal boundaries.

In detecting an apparent contour we have also determined on which side the surface lies and so can compute the sign of Gaussian curvature from the curvature of the image contour. Figure 15 shows a selected number of contours which have been automatically tracked and are correctly labeled by testing for the sign and magnitude of R.

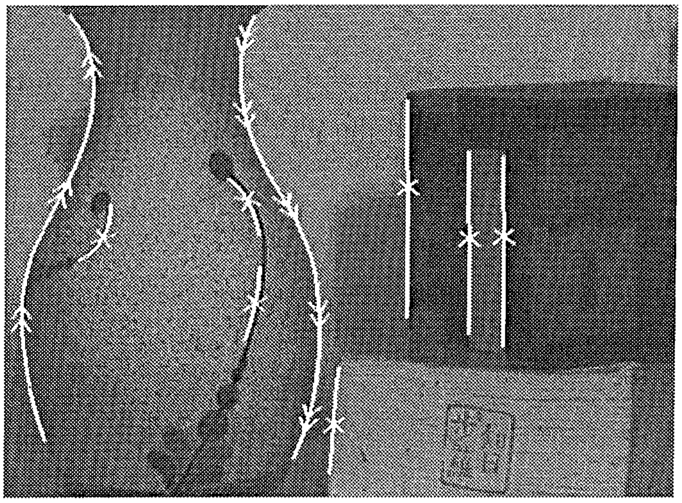

Fig. 15. Detecting and labeling extremal boundaries. The magnitude of the radius of curvature ($1/\kappa^t$, computed from 3 views) can be used to classify image curves as either the projection of extremal boundaries or fixed features (surface markings, occluding edges or orientation discontinuities). The sign of κ^t determines on which side of the image contour lies the surface. *Note*: a × label indicates a fixed feature. A $\gg$ label indicates an apparent contour. The surface lies to the right as one moves in the direction of the twin arrows [37]. The sign of Gaussian curvature can then be inferred directly from the sign of the curvature of the apparent contour.

7.2 Reconstruction of Surfaces

In the vicinity of the extremal boundary we can recover the two families of parametric curves. These constitute a *conjugate* grid of surface curves: s-parameter curves (the extremal contour generators from the different viewpoints) and t-parameter curves (the intersection of a pencil of epipolar planes defined by the first two viewpoints and the surface). The recovered strip of surface is shown in figure 16 projected into the image from a fourth viewpoint. The reconstructed surface obtained by extrapolation of the computed surface curvatures at the extremal boundary A of the vase is shown from a new viewpoint in figure 17.

7.3 Visual Navigation Around Curved Surfaces

In this section results are presented showing how a moving robot manipulator can exploit the visually derived 3D shape information in real time to plan a smooth, safe path around an obstacle placed in its path. The scenario of this work is that the start position and goal position for a mobile camera are fixed and the robot is instructed to reach the goal from the start position, skirting around any curved obstacles that would be encountered on a straight line path from the current position to the goal. The camera first localizes an apparent contour and makes a small sideways motion to generate visual motion. This allows it to compute the distance to the contour generator, the "sidedness" of the contour, and more importantly the curvature of the visible surface in the epipolar plane. A safe path around the curved object is then planned by extrapolating the computed curvatures with a correction to allow for the

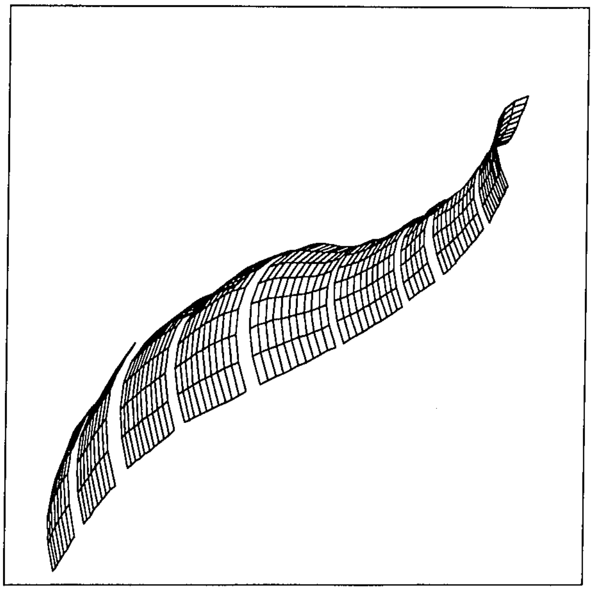

Fig. 17. Reconstructed surface. Reconstructed surface obtained by extrapolation of computed surface curvatures in the vicinity of the extremal boundary (A) of the vase, shown here from a new viewpoint.

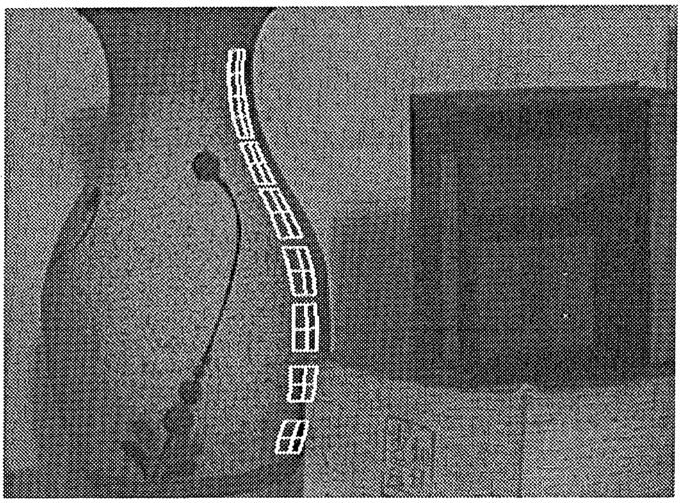

Fig. 16. Recovery of surface strip in vicinity of extremal boundary. From a minimum of three views of a curved surface it is possible to recover the 3D geometry of the surface in the vicinity of extremal boundary. The surface is recovered as a family of s-parameter curves—the contour generators—and t-parameter curves—portions of the *osculating* circles measured in each epipolar plane. The strip is shown projected into the image of the scene from a different viewpoint.

uncertainty so as to ensure safe clearances. The robot then steers the camera around the obstacle with a clearance of a few millimeters. Examples running with single image contours are shown in figures 18 and 19.

The path-planning algorithms for navigating around curved surfaces are further developed in [7]. Minimal paths are shown to be smooth splines composed of *geodesics* [18] and straight lines in free space. Computation of the geodesics, in general, requires the complete 3D surface. In the case where geometric information is imperfect, in that surface shape is not known a priori, they show that it is possible to compute a helical approximation to the sought geodesic, based only on the visible part of the surface near the extremal boundary. The information required for the helical approximation can be computed directly from the deformation of the apparent contour.

7.4 Manipulation of Curved Objects

Surface curvature recovered directly from the deformation of the apparent contour (instead of dense depth maps) yields useful information for path planning. This information is also important for grasping curved objects.

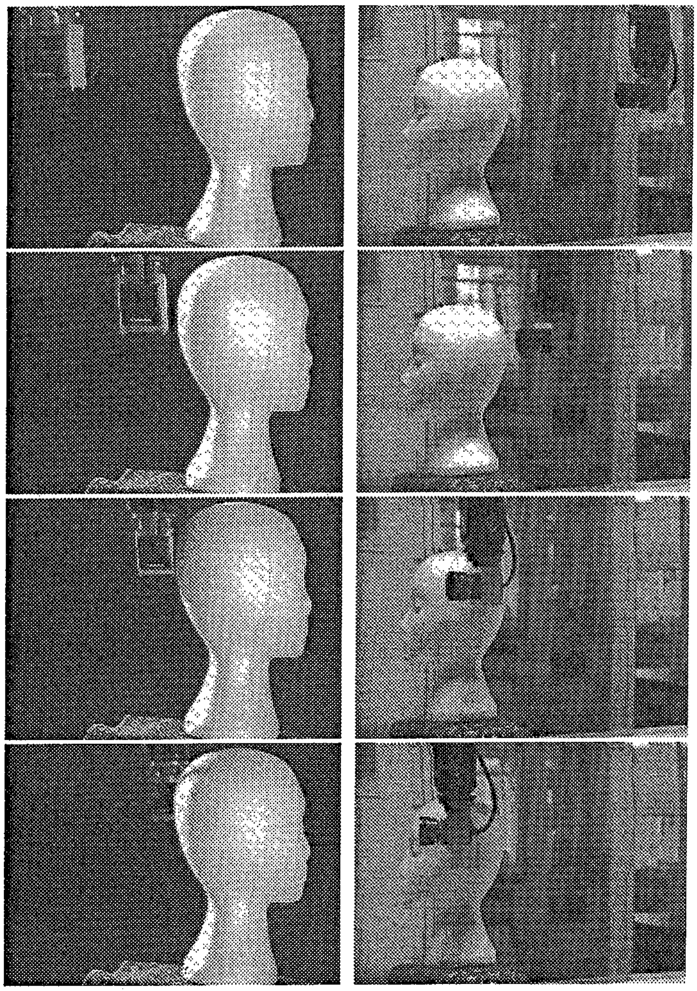

Fig. 18. Visually guided navigation around curved obstacles. The visual motion of an apparent contour under known viewer motion is used to estimate the position, orientation, and surface curvature of the visible surface. In addition to this quantitative information the visual motion of the apparent contour can also determine which side of the contour is free space. This qualitative and quantitative information is used to map out a safe path around the unmodeled obstacle. The sequence of images shows the robot manipulator's safe execution of the planned path, seen from two viewpoints.

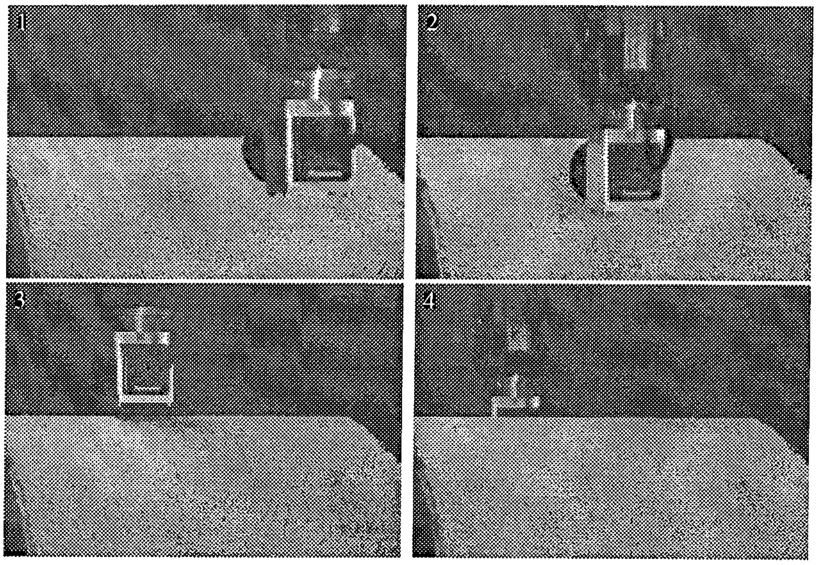

Fig. 19. Visually guided navigation over undulating surface. After detecting a horizontal image contour, the image motion due to a small local vertical viewer motion is used to estimate the distance to the contour generator. A larger extended motion is then used to estimate the surface curvature at the contour generator. This is used to map out and execute a safe path over the obstacle, shown in this sequence of images.

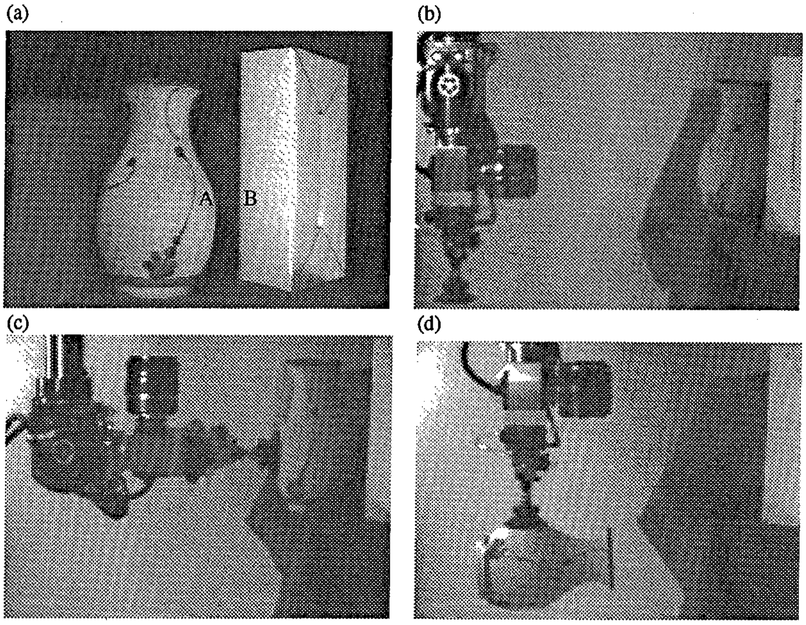

Fig. 20. Visually guided manipulation of piecewise curved objects. The manipulation of curved objects requires precise 3D shape (curvature) information. The accuracy of measurements of surface curvature based on the deformation of a single apparent contour is limited by uncertainty in the viewer motion. The effect of errors in viewer motion is greatly reduced and the accuracy of surface curvature estimates consequently greatly improved by using the rate of parallax. In the example shown the relative motion between the image of the projection of the crease of the box (B) and the apparent contour of the vase (A) is used to estimate surface curvature to an accuracy of ±5 mm (in a measurement of 40 mm) and the contour generator position to the nearest 1 mm (at a distance of 1 m). This information is used to guide the manipulator and suction gripper to a convenient location on the surface of the vase for manipulation.

Reliable estimates of surface curvature can be used to determine grasping points. Figure 20 shows an example of a scene with a vase placed approximately 1 m away from a robot manipulator equipped with a suction gripper. Estimates of surface curvature at the extremal boundary are used to position a suction gripper for manipulation. The robot initializes a snake which localizes a nearby high-contrast edge. In the example shown the snake initially finds the edge of the cardboard box (B). The robot then makes a small local motion of a few centimeters to estimate the depth of the feature. It uses this informtion so that it can then track the contrast edge over a larger baseline while fixating (keeping the edge in the center of the image). Before executing the larger motion of 15 cm the first snake (parent) spawns a child snake which finds a second nearby edge (A). The two edges are then tracked together, allowing the accurate estimation of surface curvature by reducing the sensitivity to robot "wobble" and systematic errors in the robot motion (see figures 9 and 10). The estimates of curvature are accurate to a few millimeters. This is in contrast to estimates of curvature based on the absolute motion of an apparent contour which deliver curvature estimates which are only correct to the nearest centimeter. The extrapolation of these surface curvatures allows the robot to plan a grasping position which is then successfully executed (figure 20).

8 Summary and Conclusions

This paper has

- related the geometry of apparent contours to the differential geometry of the visible surface and to the analysis of visual motion.
- shown how a moving monocular observer can recover an exact and complete description of the visible surface in the vicinity of a contour generator from the deformation of apparent contours. This requires the computation of spatiotemporal derivatives (up to second order) of the image and known viewer motion. The *epipolar* parameterization of the spatiotemporal image and surface was introduced. Its advantages include that it allows all image contours to be analyzed in the same framework. Image velocities allow the recovery of the contour generator while image accelerations allow the computation of surface curvature. A consequence of this is that the visual motion of curves can be used to detect extremal boundaries and distinguish them from rigid contour generators such as surface markings, shadows, or creases.
- shown how the relative motion of image curves (parallax-based measurements) can be used to provide robust estimates of surface curvature which are independent of (and hence insensitive to) the exact details of the viewer's motion.
- presented a simple, computationally efficient method for accurately extracting image curves from real images and tracking their temporal evolution. This was an extension of tracking with *snakes*—energy minimizing splines guided by "image forces" —which avoids computing the internal energies by representing sections of curves as cubic B-splines. Moreover real-time processing (15 frames per second) is achieved by windowing and avoiding Gaussian smoothing.
- analyzed the effect of errors in the knowledge of assumed viewer motion (camera positon and orientation) and in the localization of image contours on the estimates of depth and curvature. Uncertainty and sensitivity analysis was important for two reasons. First, it is useful to compute bounds on the estimates of surface curvature. This is critical in discriminating fixed features from extremal boundaries—deciding whether curvature along the ray is bounded or not—since with noisy measurements and poorly calibrated viewer motions, we must test by error analysis the hypothesis that the curvature is unbounded at a fixed feature. Second, sensitivity analysis was used to substantiate the claim that *parallax* methods—using the relative image motion of nearby contours—allow the robust recovery of surface curvature. It was shown that estimates of curvature based on absolute measurements of image position are extremely sensitive to motion calibration, requiring accuracies of the order of 1 part in 1000. Estimates of curvature based on *relative* measurements prove to be orders of magnitude less sensitive to errors in robot position and orientation. The sensitivity to image localization remains, however, but is reduced by integrating measurements from a large number of viewpoints.
- as an illustration of their power, these motion analysis techniques have been used to achieve something which has so far eluded analysis based on photometric measurements alone: namely reliable discrimination between fixed surface features and points on extremal boundaries. On which side of the image contour the obscuring surface lies can also be determined. As well as using these methods to detect and label extremal boundaries it was shown how they can recover strips of surfaces in the vicinity of extremal boundaries.
- presented results of the real-time implementation of these algorithms for use in tasks involving the active

exploration of the 3D geometry of visible surfaces. This demonstrated the utility and reliability of the proposed theories and methods. It was shown that the deformation of apparent contours under viewer motion is a rich source of geometric information which is extremely useful for visual navigation, motion planning and object manipulation. Examples exploiting the visually derived shape for navigation around and the manipulation of piecewise smooth curved objects were presented.

Acknowledgments

The authors acknowledge discussions with Professor Mike Brady, Dr. David Forsyth, Dr. Steve Maybank, and Dr. Andrew Zisserman and the contribution of Zhiyan Xie in programming the Adept Robot. This work was partially funded by Esprit BRA 3274 (FIRST) and the SERC. Roberto Cipolla gratefully acknowledges the support of the IBM UK Scientific Centre, St. Hugh's College, Oxford, and the Toshiba Corporation Research and Development Centre, Japan.

Notes

1. A summary of this theory was first presented by us in [9]. A similar approach was followed by [51, 1].
2. Subscripts denote differentiation with respect to the subscript parameter. Superscripts will be used as labels.
3. The *normal* curvature is the curvature of the planar section of the surface through the normal and tangent vector.
4. These are in fact the eigenvalues and respective eigenvectors of the matrix $\mathbf{G}^{-1}\mathbf{D}$. The determinant of this matrix (product of the two *principal* curvatures) is called the Gaussian curvature, K. It determines qualitatively a surface's shape. A surface patch that is locally hyperbolic (saddle-like) has *principal* curvatures of opposite sign and hence negative Gaussian curvature. Elliptic surface patches (concave or convex) have *principal* curvatures with the same sign and hence positive Gaussian curvature. A locally flat surface patch will have zero Gaussian curvature.
5. The *geodesic* curvature of a curve on a sphere is sometimes called the *apparent* curvature [31]. It measures how the curve is curving in the imaging surface. It is equal to the curvature of the perspective projection onto a plane defined by the ray direction.
6. The curvature, κ, and the Frenet-Serret normal, $\mathbf{N}$, for a space $\gamma(s)$ are given by ([20], p. 103): $\kappa\mathbf{N} = (\gamma_s \wedge \gamma_{ss}) \wedge \gamma_s / |\gamma_s|^4$. The normal curvature is the magnitude of the component of $\kappa\mathbf{N}$ in the direction of the surface normal (here $\mathbf{Q}$ since $\mathbf{Q}(s, t_0)$ is a curve on the image sphere); the geodesic curvature is the magnitude of the component in a direction perpendicular to the surface normal and the curve tangent (in this case $\mathbf{Q}_s$). For a curve on a sphere this direction is parallel to the curve normal ($\mathbf{n}$ for apparent contours).
7. If we define the surface normal as being outwards from the solid surface, the *normal* curvature will be negative in any direction for a convex surface patch.
8. In general the Gaussian curvature can be determined from the determinant of $\mathbf{G}^{-1}\mathbf{D}$ or equivalently the ratio of the determinants of the matrixes of coefficients of the second and first fundamental forms. From (34) and (35) it is trivial to show that Gaussian curvature can be expressed by

$$K = \frac{\kappa^s \kappa^t}{\sin^2 \theta}$$

Substituting (23) for κ^s allows us to derive the result.

9. The forces are derived from an arbitrary field. They are not natural forces.
10. Small amounts of smoothing can be achieved economically by defocusing the image until the desired degree of blur is achieved. Whilst this worked satisfactorily, it was found that the tracker continued to operate just as well when the lens was sharply focused.

References

1. E. Arbogast, Modélisation automatique d'objets non polyédriques par observation monoculaire, Ph.D. thesis, Institut National Polytechnique de Grenoble, 1991.
2. S.T. Barnard and M.A. Fischler, Computational stereo, *ACM Computing Surveys* 14(4):553–572, 1982.
3. H.G. Barrow and J.M. Tenenbaum, Recovering intrinisic scene characteristics from images. In A. Hanson and E. Riseman, eds., *Computer Vision Systems.* Academic Press: New York, 1978.
4. H.G. Barrow and J.M. Tenenbaum, Interpreting line drawings as three-dimensional surfaces, *Artificial Intelligence* 17:75–116, 1981.
5. R.H. Bartels, J.C. Beatty, and B.A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann: Los Altos, CA, 1987.
6. L.M.H. Beusmans, D.D. Hoffman, and B.M. Bennett, Description of solid shape and its inference from occluding contours, *J. Opt. Soc. Amer. A* 4:1155–1167, 1987.
7. A. Blake, J.M. Brady, R. Cipolla, Z. Xie, and A. Zisserman, Visual navigation around curved obstacles, *Proc. IEEE Intern. Conf. Robotics Automat.* 3:2490–2495, 1991.
8. A. Blake and H. Bulthoff, Shape from specularities: Computation and psychophysics, *Phil. Trans. Roy. Soc. London* 331:237–252, 1991.
9. A. Blake and R. Cipolla, Robust estimation of surface curvature from deformation of apparent contours. O. Faugeras, ed., *Proc. 1st Europ. Conf. Comput. Vis.*, Antibes, Fr., pp. 465–474, Springer-Verlag: New York, 1990.
10. R.C. Bolles, H.H. Baker, and D.H. Marimont, Epipolar-plane image analysis: An approach to determining structure, *Intern. J. Comput. Vis.* 1:7–55, 1987.
11. M. Brady, J. Ponce, A. Yuille, and H. Asada, Describing surfaces, *Comput. Vis. Graph. Image Process.* 32:1–28, 1985.
12. J. Callahan and R. Weiss, A model for describing surface shape, *Proc. Conf. Comput. Vis. Patt. Recog.*, San Francisco, pp. 240–245, 1985.
13. J.F. Canny, A computational approach to edge detection, *IEEE Trans. Patt. Anal. Mach. INtell.* 8:679–698, 1986.

14. R. Cipolla, Active visual inference of surface shape, Ph.D. thesis, University of Oxford, 1991.
15. R. Cipolla and A. Blake, The dynamic analysis of apparent contours, *Proc. 3rd Intern. Conf. Comput. Vis.*, pp. 616-623, 1990.
16. R. Cipolla and M. Yamamoto, Stereoscopic tracking of bodies in motion, *Image Vis. Comput.* 8(1):85-90, 1990.
17. R.M. Curwen, A. Blake, and R. Cipolla, Parallel implementation of lagrangian dynamics for real-time snakes, *Proc. 2nd British Mach. Vis. Conf.*, 1991.
18. M.P. DoCarmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall: Englewood Cliffs, NJ, 1976.
19. O.D Faugeras and G. Toscani, The calibration problem for stereo, *Proc. Conf. Comput. Vis. Patt. Recogn.*, Miami Beach, pp. 15-20, 1986.
20. I.D. Faux and M.J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis-Horwood, 1979.
21. S. Ganapathy, Decomposition of transformation matrices for robot vision, *Proc. IEEE Conf. Robotics*, pp. 130-139, 1984.
22. P.J. Giblin and M.G. Soares, On the geometry of a surface and its singular profiles, *Image Vis. Comput.* 6(4):225-234, 1988.
23. P.J. Giblin and R. Weiss, Reconstruction of surfaces from profiles, *Proc. 1st Intern. Conf. Comput. Vis.*, London, pp. 136-144, 1987.
24. C.G. Harris, Determination of ego-motion from matched points, *3rd Alvey Vis. Conf.*, pp. 189-192, 1987.
25. H. von Helmholtz, *Treatise on Physiological Optics*, Dover: New York, 1925.
26. B.K.P. Horn, *Robot Vision*. McGraw-Hill, NY, 1986.
27. B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer.* A4(4):629-642, 1987.
28. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: active contour models, *Proc. 1st Intern. Conf. Comput. Vis.*, London, pp. 259-268, 1987.
29. J.J. Koenderink, What does the occluding contour tell us about solid shape? *Perception* 13:321-330, 1984.
30. J.J. Koenderink, Optic flow, *Vision Research* 26(1):161-179, 1986.
31. J.J. Koenderink, *Solid Shape*, MIT Press: Cambridge, MA, 1990.
32. J.J. Koenderink and A.J. Van Doorn, Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer, *Optica Acta* 22(9):773-791, 1975.
33. J.J. Koenderink and A.J. Van Doorn, The singularities of the visual mapping, *Biological Cybernetics* 24:51-59, 1976.
34. J.J. Koenderink and A.J. Van Doorn, The shape of smooth objects and the way contours end, *Perception* 11:129-137, 1982.
35. M.M. Lipschutz, *Differential Geometry*, McGraw-Hill: New York, 1969.
36. H.C. Longuet-Higgins and K. Pradzny, The interpretation of a moving retinal image, *Proc. Roy. Soc. London* B208:385-397, 1980.
37. J. Malik, Interpreting line drawings of curved objects, *Intern. J. Comput. Vis.* 1:73-103, 1987.
38. D. Marr, Analysis of occluding contour, *Proc. Roy. Soc. London* 197:441-475, 1977.
39. S.J. Maybank, The angular velocity associated with the optical flow field arising from motion through a rigid environment, *Proc. Roy. Soc. London* A401:317-326, 1985.
40. G. Medioni and Y. Yasumoto, Corner detection and curve representation using curve b-splines, *Proc. Conf. Comput. Vis. Patt. Recog.*, Miami Beach, pp. 764-769, 1986.
41. S. Menet, P. Saint-Marc, and G. Medioni, B-snakes: implementation and application to stereo, *Proc. DARPA*, pp. 720-726, 1990.
42. R.S. Millman and G.D. Parker, *Elements of Differential Geometry*, Prentice-Hall: Englewood Cliffs, NJ, 1977.
43. B. O'Neill, *Elementary Differential Geometry*, Academic Press: San Diego, CA, 1966.
44. C.R Rao, *Linear Statistical Inference and Its Applications*, Wiley: New York, 1973.
45. J.H. Rieger and D.L. Lawton, Processing differential image motion, *J. Opt. Soc. Amer.* A2(2):354-360, 1985.
46. G. Scott, The alternative snake—and other animals, *Proc. 3rd Alvey Vis. Conf.*, pp. 341-347, 1987.
47. R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robot. Automat.* RA-3(4):323-344, 1987.
48. R.Y. Tsai and T.S. Huang, Uniqueness and estimation of three-dimensional motion parameters of a rigid object with curved surfaces, *IEEE Trans. Patt. Anal. Mach. Intell.* 6(1):13-26, 1984.
49. R.Y. Tsai and R.K. Lenz, A new technique for fully autonomous and efficient 3D robotics hand-eye calibration, *4th Intern. Symp. Robotics Res.*, Santa Cruz, CA, pp. 287-297, 1987.
50. R.Y. Tsai and R.K. Lenz, Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology, *IEEE Trans. Patt. Anal. Mach. Intell.* 10(5):713-720, 1988.
51. R. Vaillant and O.D. Faugeras, Using occluding contours for recovering shape properties of objects. (Submitted to *Trans. PAMI*), 1992.
52. D. Weinshall, Qualitative depth from stereo, with applications, *Comput. Vis. Graph. Image Process.* 49:222-241, 1990.
53. M. Yamamoto, Motion analysis using the visualized locus method, *Trans. Inform. Process. Soc. Japan* 22(5):442-449, 1981 (in Japanese).

Appendix A. Determining $\delta_{tt} \cdot n$ from the Spatiotemporal Image $Q(s, t)$

If the surface marking is a discrete point (image position $\mathbf{Q}^*$) it is possible in principle to measure the image velocity, $\mathbf{Q}_t^*$ and acceleration, $\mathbf{Q}_{tt}^*$ directly from the image without any assumption about viewer motion. This is impossible for a point on an image curve. Measuring the (real) image velocity $\mathbf{Q}_t$ *(and accleration* $\mathbf{Q}_{tt}$) for a point on an image curve requires knowledge of the viewer motion—equation (27). Only the normal component of image velocity can be obtained from local measurements at a curve. It is shown below however that for a discrete point-curve pair, $\boldsymbol{\delta}_{tt} \cdot \mathbf{n}$—the normal component of the relative image acceleration—is completely determined from measurements on the spatiotemporal image. This result is important because it demonstrates the possibility of obtaining robust inferences of surface geometry that are independent of any assumption of viewer motion.

The proof depends on reparameterizing the spatiotemporal image so that it is independent of knowledge of viewer motion. In the epipolar parameterization of the spatiotemporal image, $\mathbf{Q}(s, t)$, the s-parameter curves were defned to be the image contours while the t-parameter curves were defined by equation (27) so that at any instant the magnitude and direction of the tangent to a t-parameter curve is equal to the (real) image velocity, $\tilde{\mathbf{Q}}_t$—more precisely, $(\partial\tilde{\mathbf{Q}}/\partial t)|_s$.

A parameterization which is completely independent of knowledge of viewer motion, $\tilde{\mathbf{Q}}(\bar{s}, t)$, where s is a function of s and t, $\bar{s}(s, t)$ can be chosen. Consider, for example, a parameterization where the t-parameter curves (with tangent $(\partial\tilde{\mathbf{Q}}/\partial t)|_{\bar{s}}$) are chosen to be orthogonal to the $\bar{s}$-parameter curves (with tangent $(\partial\tilde{\mathbf{Q}}/\partial\bar{s})|_t$)—the image contours. Equivalently the t-parameter curves are defined to be parallel to the curve normal $\mathbf{n}$,

$$\frac{\partial\tilde{\mathbf{Q}}}{\partial t}\bigg|_{\bar{s}} = \beta\mathbf{n} \tag{62}$$

where β is the magnitude of the normal component of the (real) image velocity. Such a parameterization can always be set up in the image. It is now possible to express the (real) image velocities and acclerations in terms of the new parameterization.

$$\tilde{\mathbf{Q}}_t = \frac{\partial\tilde{\mathbf{Q}}}{\partial t}\bigg|_{s} \tag{63}$$

$$= \frac{\partial\bar{s}}{\partial t}\bigg|_s \frac{\partial\tilde{\mathbf{Q}}}{\partial\bar{s}}\bigg|_t + \frac{\partial\tilde{\mathbf{Q}}}{\partial t}\bigg|_{\bar{s}} \tag{64}$$

$$\tilde{\mathbf{Q}}_{tt} = \frac{\partial^2\tilde{\mathbf{Q}}}{\partial t^2}\bigg|_s \tag{65}$$

$$= \frac{\partial^2\bar{s}}{\partial t^2}\bigg|_s \frac{\partial\tilde{\mathbf{Q}}}{\partial\bar{s}}\bigg|_t + \left(\frac{\partial\bar{s}}{\partial t}\bigg|_s\right)^2 \frac{\partial^2\tilde{\mathbf{Q}}}{\partial\bar{s}^2}\bigg| + 2\frac{\partial\bar{s}}{\partial t}\bigg|_s \frac{\partial}{\partial\bar{s}}\left(\frac{\partial\tilde{\mathbf{Q}}}{\partial t}\bigg|_{\bar{s}}\right)\bigg|_t + \frac{\partial^2\tilde{\mathbf{Q}}}{\partial t^2}\bigg|_{\bar{s}} \tag{66}$$

$$\tilde{\mathbf{Q}}_{tt}\cdot\mathbf{n} = \left(\frac{\partial\bar{s}}{\partial t}\bigg|_s\right)^2 \frac{\partial^2\tilde{\mathbf{Q}}}{\partial\bar{s}^2}\bigg|_t\cdot\mathbf{n} + 2\frac{\partial\bar{s}}{\partial t}\bigg|_s \frac{\partial}{\partial\bar{s}}\left(\frac{\partial\tilde{\mathbf{Q}}}{\partial t}\bigg|_{\bar{s}}\right)\bigg|_t\cdot\mathbf{n} + \frac{\partial^2\tilde{\mathbf{Q}}}{\partial t^2}\bigg|_{\bar{s}}\cdot\mathbf{n} \tag{67}$$

From (64) we see that $(\partial\bar{s}/\partial t)|_s$ determines the magnitude of the tangential component of image-curve velocity and is not directly available from the spatiotemporal image. The other quantities in the right-hand side of the (67) are directly measurable from the spatiotemporal image. They are determined by the curvature of the image contour, the variation of the normal component of image velocity along the contour and the variation of the normal component of image veloctiy perpendicular to the image contour respectively.

However the discrete point (with image position $\tilde{\mathbf{Q}}^*$) which is instantaneously aligned with the extremal boundary has the same image velocity, $\tilde{\mathbf{Q}}_t^*$, as the point on the apparent contour. From (27),

$$\tilde{\mathbf{Q}} = \tilde{\mathbf{Q}}^* \tag{68}$$

$$\tilde{\mathbf{Q}}_t = \tilde{\mathbf{Q}}_t^* \tag{69}$$

Since $\tilde{\mathbf{Q}}_t^*$ is measurable it allow us to determine the tangential component of the image velocity

$$\frac{\partial\bar{s}}{\partial t}\bigg|_s = \frac{\tilde{\mathbf{Q}}_t\cdot\frac{\partial\tilde{\mathbf{Q}}}{\partial\bar{s}}\Big|_t}{\left|\frac{\partial\tilde{\mathbf{Q}}}{\partial\bar{s}}\Big|_t\right|^2} \tag{70}$$

and hence $\tilde{\mathbf{Q}}_{tt}\cdot\mathbf{n}$ and $\delta_{tt}\cdot\mathbf{n}$ from spatiotemporal image measurements.

Appendix B. Correction for Parallax-Based Measurements when Image Points are not Coincident

The theory relating relative inverse curvatures to the rate of parallax assumed that the two points $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ were actually coincident in the image, and that the underlying surface points were also coincident and hence at the same dept $\lambda^{(1)} = \lambda^{(2)}$. In practice, point pairs used as features will not coincide exactly. We analyze below the effects of a finite separation in image positions $\Delta\mathbf{Q} = \delta$, and a difference in depths of the 2 features, $\Delta\lambda$. $\mathbf{n}$ is the surface normal at the second point, $\mathbf{Q}^{(2)}$.

$$\mathbf{Q}^{(2)} = \mathbf{Q}$$

$$\mathbf{Q}^{(1)} = \mathbf{Q} + \Delta\mathbf{Q}$$

$$\lambda^{(2)} = \lambda$$

$$\lambda^{(1)} = \lambda + \Delta\lambda$$

$$\mathbf{Q}^{(2)}\cdot\mathbf{n} = 0$$

$$\mathbf{Q}^{(1)}\cdot\mathbf{n} = \Delta\mathbf{Q}\cdot\mathbf{n}$$

If the relative inverse curvature is computed from (47),

$$\Delta\hat{R} = \frac{(\mathbf{U}\cdot\mathbf{n})^2}{\lambda^3}\frac{1}{\delta_{tt}\cdot\mathbf{n}} \tag{71}$$

an error is introduced into the estimate of surface curvature due to the fact that the features are not instantaneously aligned nor at the same depth nor in the same tangent plane:

$$\Delta R = \Delta \hat{R} + R^{\text{error}} \tag{72}$$

where R^{error} consists of errors due to the 3 effects mentioned above.

$$R^{\text{error}} = R^{\Delta\lambda} + R^{\Delta \mathbf{Q}} + R^{\mathbf{n}} \tag{73}$$

These are easily computed by looking at the differences of equation (45) for the two points. Only first-order errors are listed.

$$R^{\Delta\lambda} = \Delta\lambda \left[\frac{3}{\lambda\kappa^{t2}} + \frac{4\mathbf{U}\cdot\mathbf{Q}}{\mathbf{U}\cdot\mathbf{n}} + \frac{\lambda\mathbf{U}_t\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2} \right.$$
$$\left. + \frac{2\lambda(\mathbf{U}\cdot\mathbf{Q})(\boldsymbol{\Omega}\wedge\mathbf{Q})\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2} + \frac{\lambda(\mathbf{U}\wedge\boldsymbol{\Omega})\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2} \right]$$

$$R^{\Delta \mathbf{Q}} = \left[\frac{-2\lambda(\mathbf{U}\cdot\boldsymbol{\delta})}{\mathbf{U}\cdot\mathbf{n}} - \frac{\lambda^3(\boldsymbol{\Omega}_t\wedge\boldsymbol{\delta})\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2} \right.$$
$$- \frac{2\lambda^2(\mathbf{U}\cdot\boldsymbol{\delta})(\boldsymbol{\Omega}\wedge\mathbf{Q})\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2}$$
$$- \frac{2\lambda^2(\mathbf{U}\cdot\mathbf{Q})(\boldsymbol{\Omega}\wedge\boldsymbol{\delta})\cdot\mathbf{n}}{(\mathbf{U}\cdot\mathbf{n})^2}$$
$$\left. + \frac{\lambda^3(\boldsymbol{\Omega}\cdot\boldsymbol{\delta})(\boldsymbol{\Omega}\cdot\mathbf{n})}{(\mathbf{U}\cdot\mathbf{n})^2} \right]$$

$$R^{\mathbf{n}} = \boldsymbol{\delta}\cdot\mathbf{n} \left[\frac{\lambda^2\mathbf{U}_t\cdot\mathbf{Q}}{(\mathbf{U}\cdot\mathbf{n})^2} - \frac{\lambda|\mathbf{U}|^2}{(\mathbf{U}\cdot\mathbf{n})^2} \right.$$
$$+ \frac{(\mathbf{U}\cdot\mathbf{Q})}{\mathbf{U}\cdot\mathbf{n}}\frac{1}{\kappa^{t2}} + \frac{2\lambda(\mathbf{U}\cdot\mathbf{Q}^2}{(\mathbf{U}\cdot\mathbf{n})^2}$$
$$\left. - \frac{\lambda^2(\boldsymbol{\Omega}\wedge\mathbf{Q})\cdot\mathbf{U}}{(\mathbf{U}\cdot\mathbf{n})^2} - \frac{\lambda^3|\boldsymbol{\Omega}|^2}{(\mathbf{U}\cdot\mathbf{n})^2} \right]$$