

# A Theory of Specular Surface Geometry

MICHAEL OREN AND SHREE K. NAYAR

*Department of Computer Science, Columbia University, New York, N.Y. 10027*

oren@cs.columbia.edu

nayar@cs.columbia.edu

*Received August, 1995; Accepted September, 1995*

**Abstract.** A theoretical framework is introduced for the perception of specular surface geometry. When an observer moves in three-dimensional space, real scene features such as surface markings remain stationary with respect to the surfaces they belong to. In contrast, a virtual feature which is the specular reflection of a real feature, travels on the surface. Based on the notion of caustics, a feature classification algorithm is developed that distinguishes real and virtual features from their image trajectories that result from observer motion. Next, using support functions of curves, a closed-form relation is derived between the image trajectory of a virtual feature and the geometry of the specular surface it travels on. It is shown that, in the 2D case, where camera motion and the surface profile are coplanar, the profile is uniquely recovered by tracking just two unknown virtual features. Finally, these results are generalized to the case of arbitrary 3D surface profiles that are traveled by virtual features when camera motion is not confined to a plane. This generalization includes a number of mathematical results that substantially enhance the present understanding of specular surface geometry. An algorithm is developed that uniquely recovers 3D surface profiles using a single virtual feature tracked from the occluding boundary of the object. All theoretical derivations and proposed algorithms are substantiated by experiments.

## 1. Introduction

This paper focuses on mirror-like reflection from smooth surfaces like glass, ceramic, polished metal, and plastic. Although the physics and geometry that govern specular reflection are well understood, visual interpretation of specular surfaces remains an open problem. Two major issues are associated with specular reflection. The first is detection of specularity. How can we determine whether an image feature corresponds to an actual scene point or whether it is the specular reflection of another scene point? This ambiguity poses a problem for all vision techniques that are based on feature detection and matching, such as, binocular stereo and structure from motion. At present these techniques simply produce incorrect results when confronted with specular surfaces. The second problem, which is even more challenging, is shape recovery of specular surfaces. This problem is hard from

an analytical perspective and currently only structured (active) illumination techniques can recover the shape of a specular object.

The ambiguity that specular surfaces introduce into image analysis arises from the existence of two distinctly different types of image features: real and virtual. A *real feature* corresponds to a physical scene point such as a surface marking or a surface texture element. On the other hand, a *virtual feature* is the reflection by a specular surface of another physical scene point<sup>1</sup>. While a real feature remains stationary with respect to the surface it belongs to, a virtual feature shifts on the specular surface when either the reflected scene point or the observer moves. Further, as the observer moves, the virtual feature produces an image trajectory that varies with the geometry of the specular surface.

Given the fundamental nature of the difference between real and virtual features, they must be distinguished before they are used (or discarded) by existing

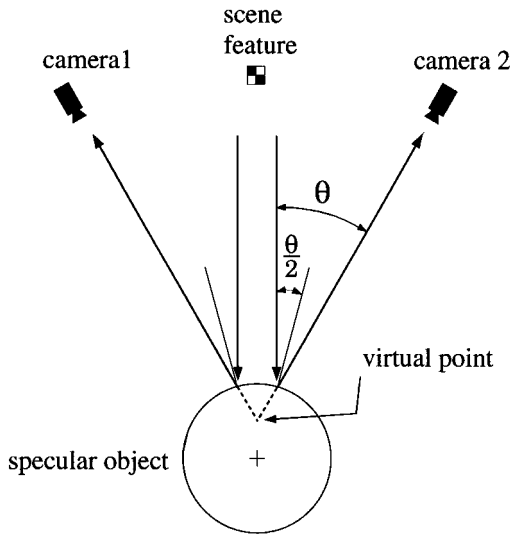


Figure 1. A simple example to illustrate that specular surfaces can cause existing shape recovery algorithms to produce results that do not lie even close to the desired result. The case at hand is binocular stereo. The same scene feature produces two virtual features, one in each sensor. The depth computed by matching these features does not lie on the physical surface (also see (Blake, 1985; Lee, 1991)). Interestingly, even as the stereo vergence ( $2\theta$ ) approaches zero, the computed virtual point does not approach the physical surface. It converges to a point midway between the sphere's center and its surface.

vision algorithms. As an example, Fig. 1 shows a specular object in a binocular stereo setting. We see that the same scene feature is reflected into the two sensors by different physical points on the surface. The resulting virtual features in the two images, when matched, produce an erroneous depth estimate; a virtual point is estimated that can be either above or below the surface (depending on whether the surface is concave or convex (Blake, 1985; Lee, 1991)). Even if the vergence of the stereo pair is reduced, the virtual point fails to approach the physical surface. In the case of a specular sphere, it can be shown to converge exactly midway between the center of the sphere and its surface. Virtual features also pose problems for motion estimation. This problem was investigated by Waldon and Dyer (1993) who showed that the optical flow field for specular surfaces does not yield the correct motion parallax but instead the flow of virtual features on the surface, called *specular flow*.

These examples illustrate that current vision algorithms that were designed to handle only real features fail for specular objects. Therefore, a technique that determines whether a feature is real or virtual is

clearly desirable. This classification problem is non-trivial since the photometric properties of a virtual feature could be identical to those of a real one. Consequently, brightness based methods for identifying specular highlights (reflections of light sources), such as Ullman's *S*-operator (Ullman, 1976) or the retinex-based test and the cylinder test proposed by Brelstaff and Blake (1988), are limited in their applicability. In recent years, several photometric methods have been developed for detection of specular reflections using color (Klinker et al., 1988), polarization (Wolff and Boult, 1991), color and multiple views (Lee, 1991), and color and polarization (Nayar et al., 1996). These techniques work reasonably well in partially controlled scenes but are not guaranteed to resolve ambiguities posed by virtual features in unstructured settings (see discussion in (Nayar et al., 1996)).

The fundamental difference between real and virtual features raises two questions: First, what geometric information is contained in virtual features? Second, how can virtual features be exploited to recover shapes of specular surfaces? Before we address these questions and introduce our results, a survey of previous work is in order. Several techniques have been developed that use active illumination to recover shapes of specular surfaces. These include the photometric stereo based approach of Ikeuchi (1981), the photometric sampler developed by Nayar et al. (1990) and the SHINY system implemented by Sanderson et al. (1988). Recently, Schultz (1994) suggested using multiple viewpoint images of a specular surface to iteratively recover shape. This technique uses a known irradiance map of the illumination to compute the surface that minimizes discrepancies between rendered and measured images.

Others have analyzed specular surfaces from the perspective of an active observer. Koenderink and van Doorn (1980) described the qualitative behavior of highlights under observer motion. It was shown that highlights travel freely in elliptic or hyperbolic regions and speed up near parabolic lines. Blake (1985), Blake and Brelstaff (1988), and Blake and Bulthoff (1991) have studied the effect of specular reflection on binocular stereo. Stereo disparities produced by a real feature (surface texture for instance) and a virtual feature (a highlight) were compared. They showed that a specular highlight can yield additional information about the local curvature of the surface. For instance, it can be used to resolve convex/concave ambiguities. Healey and Binford (1988) analyze monocular images to estimate local shape within a highlight region. They use

the Torrance-Sparrow model (Torrance and Sparrow, 1967) to relate the brightness distribution of a highlight to local surface curvature. Thrift and Lee (1983) studied the problem of localizing and determining the dimensions of spheres and straight-axis generalized cylinders from specular highlights. A similar problem of surface geometry classification based on the reflection of a rectangular light source was studied by Symosek (1985) who developed estimators for recognition and pose estimation of three quadric surfaces—planes, cylinders and spheres. A novel direction was pursued by Longuet-Higgins (1960) who studied specular reflections from random surfaces (such as a sea surface) and explored the creation and annihilation of specular points.

Although specular surfaces have intrigued many researchers, we still lack a complete understanding of what information regarding scene geometry can be extracted from specularities. It is clear that virtual features in a single image do not reveal sufficient information. Hence, we pursue an active approach where the observer (sensor) moves along a known trajectory in the scene. Image features are tracked through the acquired image sequence. It turns out that even for a moving observer, exploitation of virtual features is a difficult problem. Zisserman et al. (1989) showed that a moving observer can determine a surface profile by tracking the reflection of a known light source, but only up to a one-parameter family of curves. In other words, even with a known source, shape cannot be uniquely recovered. While their result shows the existence of a family of curves, it does not provide a closed-form expression for this family. It is only recently that Bellver-Cebreros and Rodriguez-Danta (1992) derived equations for the family of curves in the 2D case, where camera motion, a known source, and the reflecting surface profile are all assumed to be coplanar. Recent work by Zheng et al. (1995) also addressed the problem of 2D recovery using single and multiple light sources. All of the above schemes assume that the light source position is known a-priori. They also lead to the conclusion that the exact surface profile remains ambiguous if a single virtual feature is used. Moreover, profile recovery using unknown features and relating image trajectories to surface profiles for general 3D camera motion have remained unsolved problems.

In our work, we address all of the problems stated above. The first half of this paper focuses on the 2D case, where camera motion, the reflected scene feature, and the specular profile curve are confined to the same

plane. It is observed that analysis of specular surface geometry is possible only if the surface curve representation used is chosen with care. We begin by analyzing the envelope of the family of reflected rays produced by sensor motion. This envelope is referred to as the *caustic of reflection*. Caustics play an important role in geometrical optical design (Cornbleet, 1984; Stavroudis, 1972) and here we show that they can be exploited for classification of image features into real and virtual, based solely on their image trajectories and not their brightness properties. Next, we proceed to address the recovery of 2D surface profiles from image trajectories. A differential equation is derived that relates surface profile to the caustic of a virtual feature. Based on this result, an algorithm is devised for unique recovery of 2D specular profiles from just two *unknown* virtual features. In contrast to all previous work, this approach does not require prior knowledge of scene features or markings on the profile. Experimental results are presented that verify the effectiveness of both feature classification and 2D profile recovery.

In the second half of the paper, we extend our ideas to the more challenging 3D case where the camera motion and the surface profile can be arbitrary space curves. The problem of profile recovery in this case is more difficult than in the 2D case. A solution to this problem requires a non-trivial mathematical generalization of the concepts and the techniques developed for the 2D case. We begin with an analysis of 3D caustic curves. This involves the derivation of a new set of explicit caustic equations that do not appear in previous literature. This derivation was made possible using the idea of a coordinate frame that is attached to the specularly reflected ray. Mathematical conciseness resulting from this idea enables us to derive a differential equation that relates the 3D caustic of a virtual feature to the 3D specular surface profile. We find that, when a virtual feature can be tracked from the occluding boundary of an object, the 3D surface profile can be recovered without ambiguity. The feasibility of this technique is demonstrated via experiments.

The paper is organized as follows. In Section 2, we study 2D specular surface profiles. We first describe a curve representation based on the support function (Guggenheimer, 1963) and derive a set of fundamental expressions, including, the caustic equation and the profile recovery equation. Next, unique profile recovery using two unknown features is presented. In addition, we present experiments that demonstrate practical feasibility of the feature classification and

profile recovery algorithms. In Section 3, we study the more general case of arbitrary 3D camera motions. The caustic curve equation for a family of reflected rays in 3D is derived and used for feature classification. Next, the 3D curve recovery equation is derived and experiments on curve recovery are shown. Finally, in Section 4, we conclude with a summary of our results and their implications.

## 2. Recovery of 2D Surface Profiles

In this section, we will analyze the 2D case where the camera motion, the scene feature, and the specular surface profile are all confined to the same plane. In our analysis, we assume that the profile curve is smooth (twice differentiable) and its curvature does not vanish. The case of zero curvature (a straight line segment) is a special one that may be detectable by other means; it causes the reflection of a well-defined point feature to be spread over a large part of the segment.

### 2.1. Curve Representation

Appropriate representation of curves and surfaces is critical for most shape recovery algorithms and object recognition techniques. A carefully chosen representation can simplify analysis of the problem and sometimes even determine its tractability. The same is true in the case of specular surfaces. In this section, we discuss limitations of popular representations of 2D curves, introduce a representation based on the support function of a curve, and demonstrate its pertinence to the analysis of specular reflection.

**2.1.1. Envelopes and the Legendre Transform.** The most common description of a 2D curve is its representation as a collection of points given by their Cartesian coordinates,  $(x, y)$  (see Fig. 2(a)). Though this

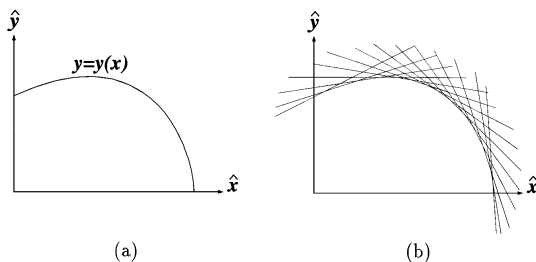


Figure 2. (a) Representation of a curve using its Cartesian coordinates  $(x, y)$ . (b) Representation of a curve as an envelope of tangents.

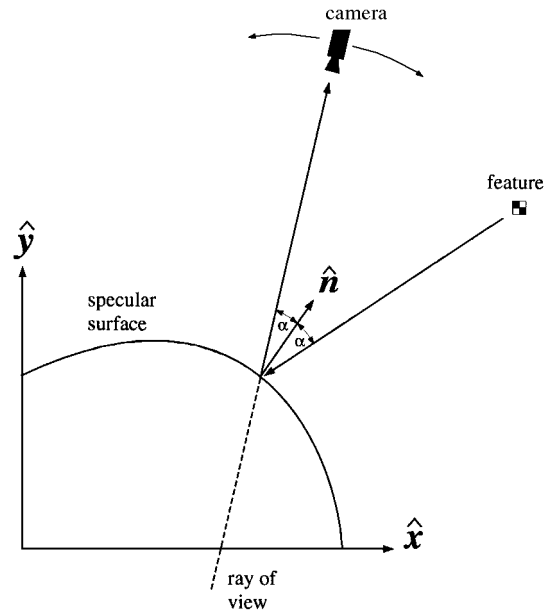


Figure 3. A ray reflected from a specular surface constrains the position of the reflecting surface point to a line in space. It also imposes constraints on the slope of the surface. These constraints are hard to formulate when the curve is described in a Cartesian coordinate system, since the reflected ray and the reflecting surface point both vary with camera location.

representation is conceptually simple, it results in complex equations when used to describe the geometry of specular reflection. The reason is that specular reflection depends not only on the spatial coordinates of the reflecting surface profile but also its local slope. The constraints on the reflecting profile point are that its normal is the bisector of the angle between the incident and reflected rays and that the point lies on the reflected ray. This is illustrated in Fig. 3. As an observer moves around the surface, the direction of the reflected ray and the location of the reflecting point both vary. There is no convenient Cartesian coordinate system that can be used to express the above constraints. This problem suggests that the analysis of specular reflection requires the use of an alternative curve representation. Since the information available to the observer is restricted to reflected rays, it is more meaningful to treat the curve as an envelope of surface tangents rather than a collection of points. This concept is illustrated in Fig. 2(b). As we will see shortly, the two representations in Fig. 2 are equivalent.

What is the most suitable representation for our envelope of tangents? Bellver-Cebreros and Rodriguez-Danta (1992) suggested using the Legendre transform of the curve equation. The Legendre transform,  $\Psi(\xi)$ ,

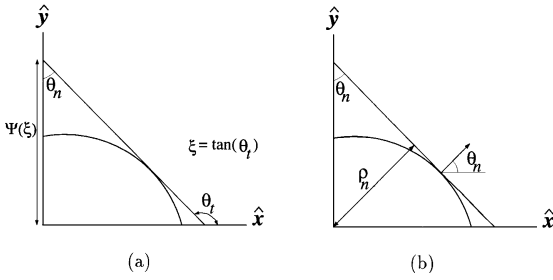


Figure 4. (a) Representation of a tangent using  $(\Psi(\xi), \xi)$ . (b) Representation of a tangent using the support function, i.e., the distance  $\rho_n$  from the origin and the normal angle  $\theta_n$ .

of a differentiable function of one variable,  $y(x)$ , is:

$$\xi = dy/dx \quad (1)$$

$$\Psi(\xi) = y(x) - \xi x \quad (2)$$

A geometric interpretation of the Legendre transform is illustrated in Fig. 4(a). If  $(x, y(x))$  is the Cartesian representation of a curve then  $\xi$  is the tangent of the angle between the tangent line and the  $\hat{x}$ -axis, and  $\Psi(\xi)$  is the intersection of the tangent line with the  $\hat{y}$ -axis. If we apply the Legendre transform to the function  $\Psi(\xi)$ , the resulting function is nothing but the original function  $y(x)$ . This shows that the Legendre transform preserves all the information contained in the original function.

### 2.1.2. Support Function Representation of Curves.

As described above, the Legendre transform of a curve is a representation of its tangent lines by their slopes and  $\hat{y}$ -intercepts (see Fig. 4(a)). While this representation simplifies the treatment of specular reflectance, as noted by Bellver-Cebreros and Rodriguez-Danta (1992), it suffers from a few drawbacks. The main disadvantage is that neither  $\Psi$  nor  $\xi$  undergo simple transformations under rotation of the coordinate system, a property that is highly desirable in our work for reasons that will become clear in due course. To achieve such an invariance, we represent the tangent line not with the slope and intersection point but with its distance to the origin,  $\rho_n$ , and the normal angle,  $\theta_n$ , as shown in Fig. 4(b). The function,  $\rho_n(\theta_n)$ , is called the *support function* of the curve (see (Guggenheimer, 1963; Horn, 1986)).

Given the Cartesian representation  $(x, y(x))$  of a curve, the support function representation is:

$$\theta_n = \tan^{-1} \left( -1 \left/ \frac{dy}{dx} \right. \right) \quad (3)$$

$$\rho_n = x \cos \theta_n + y \sin \theta_n \quad (4)$$

Calculating the derivative of  $\rho_n$  in (4) with respect to  $\theta_n$  and using  $x' \cos \theta_n + y' \sin \theta_n = 0$  from Eq. (3) yields the following transformation:

$$\begin{pmatrix} \rho_n \\ \rho'_n \end{pmatrix} = \begin{pmatrix} \cos \theta_n & \sin \theta_n \\ -\sin \theta_n & \cos \theta_n \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (5)$$

From the above relation, the inverse transform is easily determined as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \theta_n & -\sin \theta_n \\ \sin \theta_n & \cos \theta_n \end{pmatrix} \begin{pmatrix} \rho_n \\ \rho'_n \end{pmatrix} \quad (6)$$

This representation has the advantage that it depends explicitly on the slope of the curve, and that under a rotation of the coordinate system  $\rho_n$  remains unchanged while  $\theta_n$  is only subjected to a simple linear shift.

## 2.2. Caustics and Feature Classification

Before addressing shape recovery, we study the problem of feature classification and introduce the important notion of caustics. As stated in the introduction, several vision algorithms rely on robust detection and tracking of scene features. While *real* scene features (such as surface markings) cling to the surface they belong to, specular reflection produces *virtual* features that do not directly reveal the locations of the real features that produce them and tend to travel on the surface as the observer moves in the world. Techniques ranging from motion estimation to structure from motion produce incorrect results when a virtual feature is taken to be a real one. Therefore, a technique for distinguishing between real and virtual features would serve as a valuable preprocessing step for almost any vision algorithm that relies on feature tracking.

Figure 5 shows a single image taken from a sequence obtained by a moving sensor. Two image features (1 and 2) are highlighted. Feature 1 is a real surface marking (a checkerboard corner pasted on the surface of the sphere) while feature 2 is the reflection of a scene feature (also a checkerboard corner) that is located outside the field of view of the sensor. Despite these differences, the two features appear almost identical and hence are indistinguishable from a single image. However, if the observer moves around the object, each of the above features traces a trajectory in image space. We devise a technique for quickly discriminating between real and virtual features based solely on the geometrical properties of their image trajectories.

Pertinent information that the image of a virtual feature contains is the direction and position of the

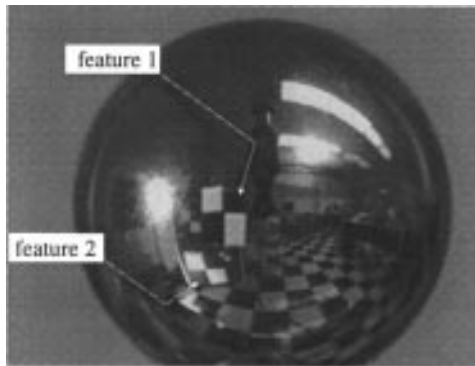


Figure 5. Image of a specular sphere with two features (both checkerboard corners). One is a real feature (surface marking, (1)) while the other is a virtual feature (reflection of a scene feature, (2)). The features are similar in appearance and cannot be distinguished from just their brightness properties. However, they can be discriminated by tracking them along an image sequence taken by a moving sensor and computing their caustics.

reflected ray relative to the world coordinate system. Given the position of a virtual feature in the image and the camera parameters (position, orientation and focal length), the direction of the reflected ray,  $\theta_r$ , and its signed distance,  $\rho_r$ , from the origin of the world coordinate system are easily computed. In the special case of orthographic projection, the direction of the reflected ray is simply the direction of the optical axis, and the distance  $\rho_r$  is just the distance in the image between the projection of the feature and the projection of the origin of the world coordinate system.

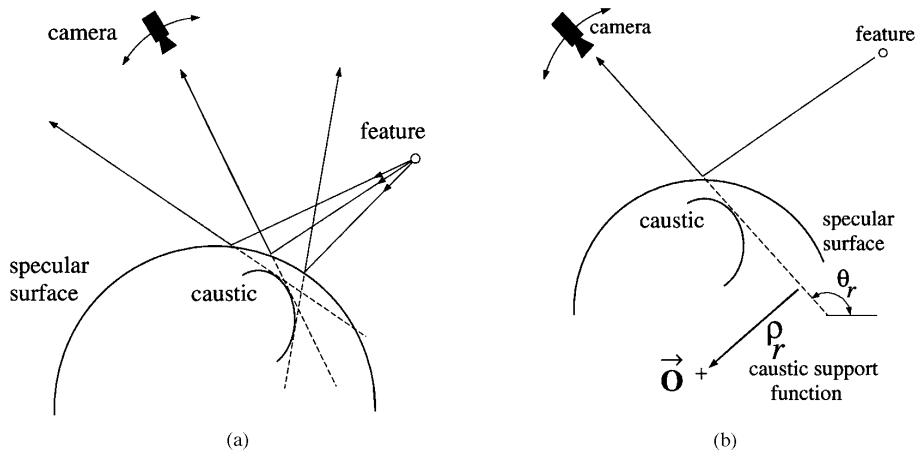


Figure 6. (a) The caustic of a feature is the envelope of the reflected rays. (b) The support function of the caustic is given by the distance of the reflected ray from origin of the world coordinate system. This distance is computed from the feature position in the image and the camera parameters.

When the sensor moves around the object, the virtual feature travels on the specular surface producing a family of reflection rays. The envelope defined by this family, illustrated in Fig. 6(a), is called a *caustic*. The caustic is tangent to each one of the reflected rays, and the normal to the caustic for a reflection ray in direction  $\theta_r$  is  $\theta_r + \frac{\pi}{2}$ . From Fig. 6(b), we see that the support function representation of the caustic is given simply by  $\rho_r(\theta_r + \frac{\pi}{2})$ . (We could also choose the normal angle to be  $\theta_r - \frac{\pi}{2}$ , in which case,  $\rho_r(\theta_r - \frac{\pi}{2}) = -\rho_r(\theta_r + \frac{\pi}{2})$ ). In the remaining of this paper, we use  $\rho_r$  for the support function of the caustic, to distinguish it from the support function of the specular surface profile which will be denoted by  $\rho_n$ . For simplicity of notation, we parameterize  $\rho_r$  by  $\theta_r$  but the normal angle to the caustic is really  $\theta_r + \frac{\pi}{2}$ .

From the above discussion, we see that the image trajectory traced by a feature is nothing but the support function of the feature's caustic. Therefore, hereafter, we will use the terms *caustic* and *image trajectory* of a feature interchangeably since they convey the same information.

Returning to the problem of feature classification, we note that the caustic of a real feature is nothing but a point, the actual position of the feature in the scene where all the reflected rays intersect. But for a virtual feature the caustic will be a curve, (except for the special case of a parabolic specular surface with a scene feature exactly on its axis<sup>2</sup>, clearly a special case that is unlikely in most practical applications). Therefore, to classify a feature, all we need is to compute the caustic

and to test whether it is a point or a curve. The computation of the caustic curve  $(x_c, y_c)$  (parameterized by  $\theta_r$ ) in Cartesian coordinates is straightforward. Given  $\rho_r(\theta_r)$ , and using expression (6) with  $\theta_n = \theta_r + \frac{\pi}{2}$ , the caustic curve is determined as:

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} -\sin \theta_r & -\cos \theta_r \\ \cos \theta_r & -\sin \theta_r \end{pmatrix} \begin{pmatrix} \rho_r(\theta_r) \\ \rho_r'(\theta_r) \end{pmatrix} \quad (7)$$

where  $\rho_r'$  is the derivative of  $\rho_r$  with respect to  $\theta_r$ .

When the radius of curvature of the profile is very small, for example at a sharp corner (Koenderink and van Dorn, 1980), the caustic will be compact making it hard in the presence of noise to determine whether it is a virtual feature or a real feature. However, such virtual features are almost fixed in space and behave like real feature points. As a result, they can be treated as real features and directly used in techniques such as stereo and motion. The above described test on the compactness of caustics can be invoked while tracking image features. Then, only those trajectories that are found to be produced by real features are used as input to vision algorithms such as structure from motion. Meanwhile, detected virtual features can be used as an additional source of scene geometry, as we will see shortly.

### 2.3. Experiments: Feature Classification

To illustrate classification we used the metallic sphere shown in Fig. 5. The real and virtual features shown

in the image were tracked while the sensor was moved around in a planar trajectory using a 5 degree of freedom robot manipulator. The viewing direction of the sensor was varied over a range of approximately  $150^\circ$ ; images were taken every  $1^\circ$ . The two features are tracked along the sequence, using the sum of square difference (SSD) correlation operator. The support functions  $\rho_r(\theta_r)$  of the two features are shown in Fig. 7(a). From each support function a caustic curve was computed using expression (7), as outlined in the previous section. The computation of the caustic involves finding the local derivative  $\rho_r'(\theta_r)$  which is typically noise sensitive. To minimize the effect of noise we smoothed the data by fitting a quadratic polynomial to each data point and its close neighbors (we used 7 neighbors on either side). The values of  $\rho_r(\theta_r)$  and  $\rho_r'(\theta_r)$  were then evaluated from the local polynomial coefficients. This smoothing technique considerably improved the quality of the computed caustic curves. Another factor that must be consider is the perspective projection of the camera. We found that the deviation of the reflected ray direction from that of the optical axis of the camera must be taken into consideration, even if the deviation is as small as  $3^\circ$ . If not, the accumulated error in caustic computation can be substantial.

The caustics computed for the two features are shown in Fig. 7(b). They are plotted as parameterized curves in the  $\hat{x}$ - $\hat{y}$  plane. As expected, the caustic of the real feature is a small cluster centered around the actual position of the feature. In contrast, the caustic of the virtual feature is a curve with a cusp (which is common

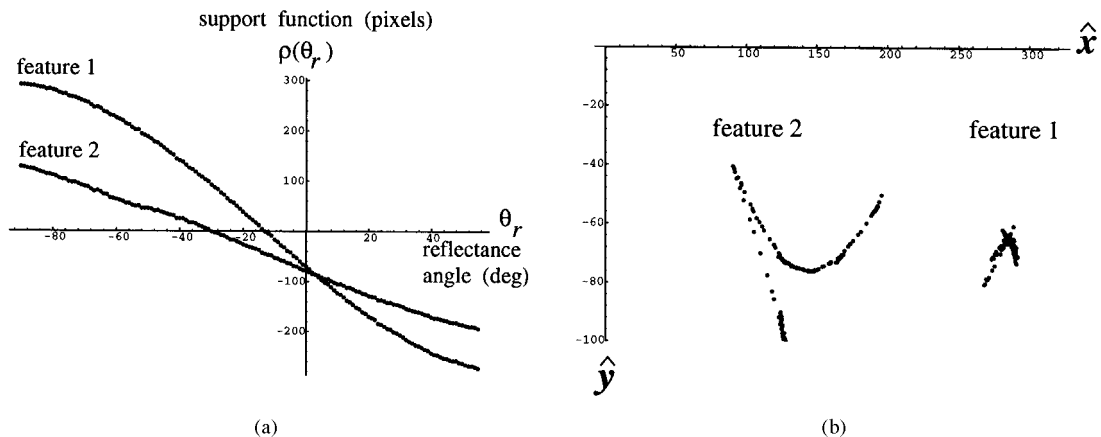


Figure 7. (a) Support functions produced by the two features shown in Fig. 5. (b) Caustics computed from the support functions and plotted on the  $\hat{x}$ - $\hat{y}$  plane. The caustic of the real feature (1) is a compact cluster, while that of the virtual feature (2) is a curve with a cusp. Real and virtual features are distinguished by using a simple measure of caustic compactness. For instance, the second moments of the two caustics have a ratio of 30:1.

in caustics resulting from specular reflection). We used the second moment of the caustic, computed with respect to the axis through the centroid perpendicular to caustic plane, as a simple measure of caustic compactness. In the above experiment, second moments of the real and virtual caustics were found to have a ratio of 30:1, clearly sufficient for reliable classification. In case a significant level of noise causes the caustic of the real feature to spread out in a larger random cluster, additional tests based on the spatial distributions of the two caustics (a well-defined continuous curve versus a random scatter of points) can be employed.

#### 2.4. 2D-Profile Recovery

We now examine the problem of recovering 2D specular profiles by moving the sensor and tracking virtual features. Zisserman et al. (1989) showed that a single feature trajectory yields surface profile only up to a one-parameter family of curves even when the position of the light source (scene feature) is known. This means that there are an infinite number of possible profiles that can yield the same image trajectory. Bellver-Cebreros and Rodriguez-Danta (1992) used the

Legendre transform to derive the equation of this family of curves. Their technique was designed for applications such as inspection of cracks from profile deformations. A single light source of known position was used to generate a specular highlight which is treated as a virtual feature. The expressions they have derived are hard to extend to the analysis of profiles when the locations of scene features are *unknown*. Moreover, their expressions cannot be generalized to 3D specular surfaces, which are also of interest to us.

In this section, we derive a new set of expressions that relate surface profile to image trajectory using the support function representation. We also show how tracking of two or more unknown virtual features enables us not only to find the position of the corresponding scene features but also to recover the profile of the specular surface without any ambiguity. In contrast to previous work, this is shown to be possible without prior knowledge of any profile points.

**2.4.1. Trajectory Equation.** Figure 8 shows a specular profile and scene features reflected by it in the direction of the camera. The camera is moved in the plane of the profile and each feature is tracked in image space.

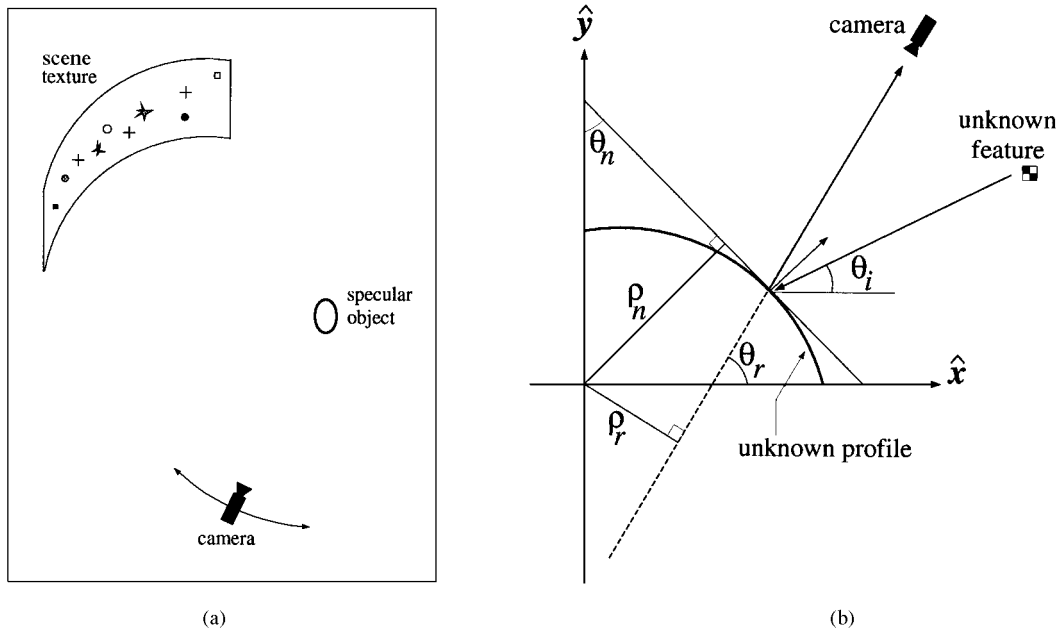


Figure 8. (a) The general setting of a specular object in an environment including a variety of visual features. The observer can view the object from different directions. (b) The geometry of feature reflection from a specular profile.



We assume that all scene features are relatively far, so that any given feature's direction is the same for all points on the surface profile, i.e., for each feature,  $\theta_i$  is nearly constant over the entire profile. As in Section 2.2, we use the notation  $\rho_n(\theta_n)$  for the profile support function and  $\rho_r(\theta_r)$  for the caustic of the feature. By  $(x, y)$  we denote the Cartesian coordinates of the reflecting point on the profile. The reflecting point lies on two tangents. The first is a tangent to the profile curve whose normal is  $\theta_n$ . From (5) we get:

$$\rho_n(\theta_n) = x \cos \theta_n + y \sin \theta_n$$

The second tangent is the reflected ray which is a tangent to the caustic. The normal to the caustic at the tangential point is  $\theta_r + \frac{\pi}{2}$ . Again, using (5) we have

$$\rho_r(\theta_r) = x \cos\left(\theta_r + \frac{\pi}{2}\right) + y \sin\left(\theta_r + \frac{\pi}{2}\right)$$

These two equations can be written together as:

$$\begin{pmatrix} \rho_n(\theta_n) \\ \rho_r(\theta_r) \end{pmatrix} = \begin{pmatrix} \cos \theta_n & \sin \theta_n \\ -\sin \theta_r & \cos \theta_r \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (8)$$

By matrix inversion, we get the following expression for the profile coordinates  $x$  and  $y$ :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{\cos(\theta_r - \theta_n)} \begin{pmatrix} \cos \theta_r & -\sin \theta_n \\ \sin \theta_r & \cos \theta_n \end{pmatrix} \begin{pmatrix} \rho_n(\theta_n) \\ \rho_r(\theta_r) \end{pmatrix} \quad (9)$$

Also, from expression (6) we get:

$$x = \cos \theta_n \rho_n(\theta_n) - \sin \theta_n \rho_n'(\theta_n) \quad (10)$$

Equating the above two expressions for  $x$ , and using the law of specular reflection:

$$\theta_r = 2\theta_n - \theta_i \quad (11)$$

we get

$$\rho_r(2\theta_n - \theta_i) = -\sin(\theta_n - \theta_i)\rho_n + \cos(\theta_n - \theta_i)\rho_n' \quad (12)$$

or

$$\rho_r(2\theta_n - \theta_i) = \frac{d}{d\theta_n} [\cos(\theta_n - \theta_i)\rho_n(\theta_n)] \quad (13)$$

This differential equation is fundamental to our analysis as it relates the support function  $\rho_r$  of a feature to the surface profile  $\rho_n$  that we seek to recover. We need to integrate this equation to retrieve the desired profile support function  $\rho_n(\theta_n)$ . Since in practice we work with the angle of reflectance  $\theta_r$  and not with  $\theta_n$  (which depends on the unknown scene feature direction) we substitute Eq. (11) in (13) and integrate over  $\theta_r$ :

$$\rho_n\left(\frac{\theta_r + \theta_i}{2}\right) = \frac{\frac{1}{2} \int_{\theta_r^0}^{\theta_r} \rho_r(\theta_r') d\theta_r' + \rho_n\left(\frac{\theta_r^0 + \theta_i}{2}\right) \cos\left(\frac{\theta_r^0 - \theta_i}{2}\right)}{\cos\left(\frac{\theta_r - \theta_i}{2}\right)} \quad (14)$$

where  $\theta_r^0$  is the starting angle of the integration. This equation gives the support function of the unknown profile as an integral of the support function of the caustic which is measured by the moving observer. But we see that even if the feature direction  $\theta_i$  is given, the surface profile cannot be determined completely due to the unknown constant of integration:

$$C = \rho_n\left(\frac{\theta_r^0 + \theta_i}{2}\right) \cos\left(\frac{\theta_r^0 - \theta_i}{2}\right) \quad (15)$$

This implies that the surface profile is determined only up to a one-parameter family of curves. However, we can make an important observation: If we can start the integration from an occluding boundary of the profile, the constant of integration vanishes since  $\cos\left(\frac{\theta_r^0 - \theta_i}{2}\right) = 0$ . This facilitates unambiguous recovery of the support function and hence the surface profile. Since reflections from occluding boundaries are not always guaranteed, we will pursue an alternative approach in the next section that uses multiple features. Either way, once the support function  $\rho_n(\theta_n)$  of the profile is determined, the actual shape of the profile is computed using Eq. (6). Alternatively, as we will see in Section 3, we can recover the profile by just using  $\rho_n$  and  $\rho_r$ , without computing the derivative of  $\rho_n$  in (6).

#### 2.4.2. Recovery of a 2D-Profile Using Multiple Features.

As we saw in the previous section, even if we track a feature whose position is known, the surface profile cannot be recovered uniquely unless we start tracking the feature from the occluding boundary, a requirement that cannot always be satisfied in practice. We show in this section that the image trajectories of two unknown virtual features impose powerful constraints. The trajectories of two different features

not only enable us to determine the positions of the unknown features in the scene, but also estimate the constant of integration in (15) and thus unique profile recovery. In the following discussion we confine ourselves to two features, though the approach easily generalizes to any larger number of features.

Consider two scene features in the directions  $\theta_i^k$  ( $k = 1, 2$ ). If these directions are known, the observer can compute the range of surface normals ( $\theta_n = \frac{1}{2}(\theta_i^k + \theta_r)$ ) along the profile traversed by each of the features. Furthermore, unless the two features are located exactly at diametrically opposite sides of the object, i.e.,  $180^\circ$  apart, their profile trajectories must overlap on some region. This overlap region, although its shape is unknown as yet, is defined by an interval  $I$  of surface normals which can be computed easily from  $\theta_i^1$  and  $\theta_i^2$  and the range of viewing angles. The result of the previous section, Eq. (14), gives the support function of the surface profile  $\rho_n^k(\theta_n; \theta_i^k, C_k)$  ( $k = 1, 2$ ) as a function of the surface normal which depends on two parameters: the feature directions  $\theta_i^k$  and the constants of integration  $C_k$ . In the interval  $I$  of surface normals, the two support functions must agree, as they represent the same profile segment:

$$\rho_n^1(\theta_n; \theta_i^1, C_1) = \rho_n^2(\theta_n; \theta_i^2, C_2), \quad \theta_n \in I \quad (16)$$

The above constraint will not be satisfied exactly due to noise, quantization and calibration errors. To quantify the above constraint, we define a distance function  $R(\cdot)$  that measures how close the two support functions are in the interval  $I$ :

$$R(\theta_i^1, C_1; \theta_i^2, C_2) = \text{Dist}[\rho_n^1(\theta_n; \theta_i^1, C_1), \rho_n^2(\theta_n; \theta_i^2, C_2); \theta_n \in I] \quad (17)$$

There are several ways of defining  $R(\cdot)$ . Our choice is discussed in the following experimental section. Once  $R(\cdot)$  is defined, the idea is to search through  $(\theta_i^1, C_1; \theta_i^2, C_2)$ -space and find the four parameters that minimize  $R(\cdot)$ . Using the recovered parameters, the surface profile is reconstructed over the entire range of measurements, i.e., not just the overlap region, but all points on the profile traveled by either of the two features. The above described approach is easily extended to larger numbers of features; the distance metric is minimized using pairs of successive features and all computed profile segments are concatenated to obtain a larger profile.

## 2.5. Experiments: Recovery of 2D Profile Using 2 Unknown Features

We now present experimental results on 2D profile recovery by tracking two unknown features. We conducted experiments on a variety of objects. Here, we present results on two profiles, one circular and the other elliptical. The experimental setup used is shown in Fig. 9(a). A specular object is positioned in a robot's workspace. A textured background at a large distance from the object produces virtual features on the specular surface of interest. A camera attached to the end-effector of the robot is moved around the object. In these experiments, the features, the camera, and object are all in the same plane. All objects used are approximately 6 cm in diameter, the camera is at a distance of approximately 40 cm from the object, and the textured background at a distance of about 200 cm. The range of viewing angles used is about  $250^\circ$  and images were taken for every  $1^\circ$  increment to ensure continuity of the support function. Two features were selected in the initial image (Fig. 9(b)) along the equator and tracked through the image sequence. Each feature was tracked independently by matching a small rectangular window around it to windows in the consecutive image in the sequence, using the SSD correlation operator. The search for the feature in the second image was pruned by estimating its new position based on its motion between the previous pair of images. Since the sensor, the feature and the object are in the same plane, self-occlusion of the feature by the sensor can occur. This problem is not critical since it occurs in no more than a couple of images in the sequence. However, we avoided such occlusions completely by raising the sensor slightly above the feature plane. Once the image trajectories were obtained, using the camera coordinates and focal length, support functions for each of the two tracked features were computed (see Fig. 10(a)).

Next, the two pairs of unknown parameters  $(\theta_i^1, C_1)$  and  $(\theta_i^2, C_2)$  were determined by minimizing the distance function  $R(\theta_i^1, C_1; \theta_i^2, C_2)$ . The current implementation uses the following distance function, which is simply a mean-squared-distance metric:

$$R(\theta_i^1, C_1; \theta_i^2, C_2) = \left( \frac{1}{N} \sum_{\theta_n \in I} [\rho_n^1(\theta_n; \theta_i^1, C_1) - \rho_n^2(\theta_n; \theta_i^2, C_2)]^2 \right)^{\frac{1}{2}} \quad (18)$$

Each term in the sum corresponds to a pair of measurements (one for each feature) in the overlap region

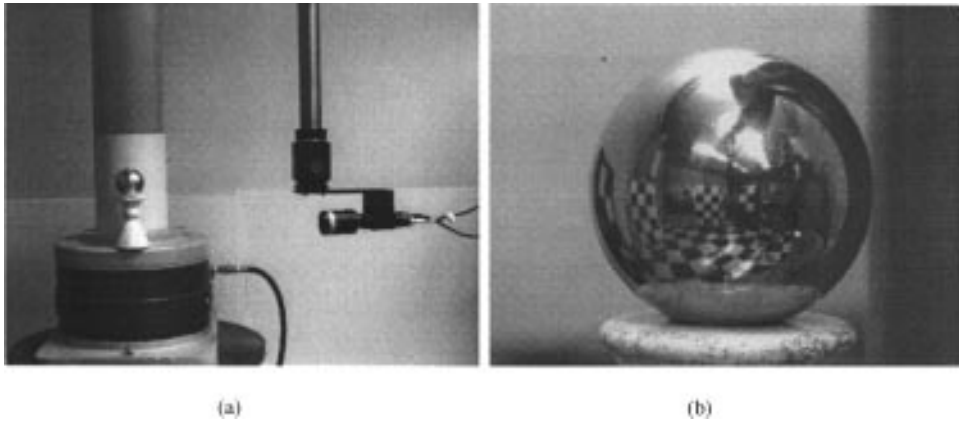


Figure 9. (a) The experimental setup. A camera is mounted on the end-effector of a 5 DOF robot. It is moved around the object and virtual features are continuously tracked in image space using correlation. (b) A typical image including a large number of virtual features.

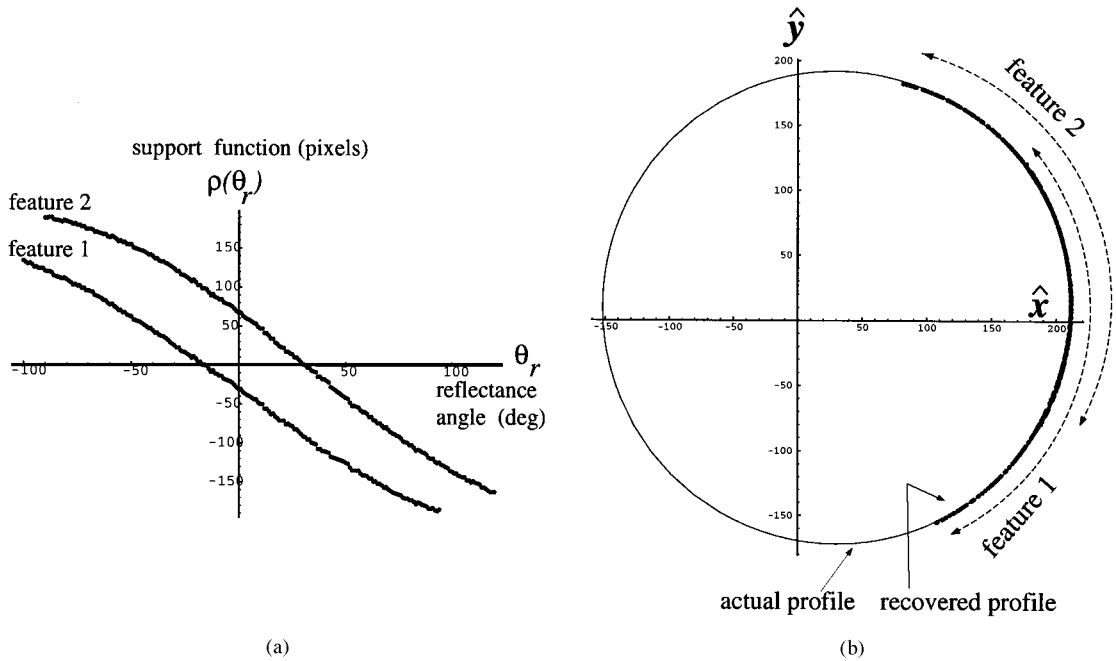


Figure 10. 2D profile of a sphere recovered by tracking two unknown features. (a) Support functions of the two features computed from their image trajectories. (b) The recovered surface profile. The dots represent the computed profile and the solid line is the actual profile. The directions of the two unknown features were found to be  $\theta_1^1 = -28^\circ$  and  $\theta_2^2 = 34^\circ$ .

$I$  determined by the parameters  $(\theta_i^1, \theta_i^2)$  chosen during search.  $N$  is the total number of measured pairs in the overlap region. The surface support function  $\rho_n$  is computed for each chosen set of parameter values by using Eq. (14). This process is made efficient by storing all the partial sums of the integral  $\int_{\theta_0^i}^{\theta_r} \rho_r(\theta_r') d\theta_r'$  in a look-up table indexed by  $\theta_r$ .

During the minimization process, to avoid getting trapped in local minima, we employ exhaustive search in the 4-parameter space. To limit the search, we use rough estimates of the upper and lower bounds on the 4 parameters. Given the physical constraints of our experimental setting, we used  $-150^\circ \leq \theta_i^1 \leq 150^\circ$  and  $|\theta_i^2 - \theta_i^1| \leq 120^\circ$ . Estimates for the search ranges

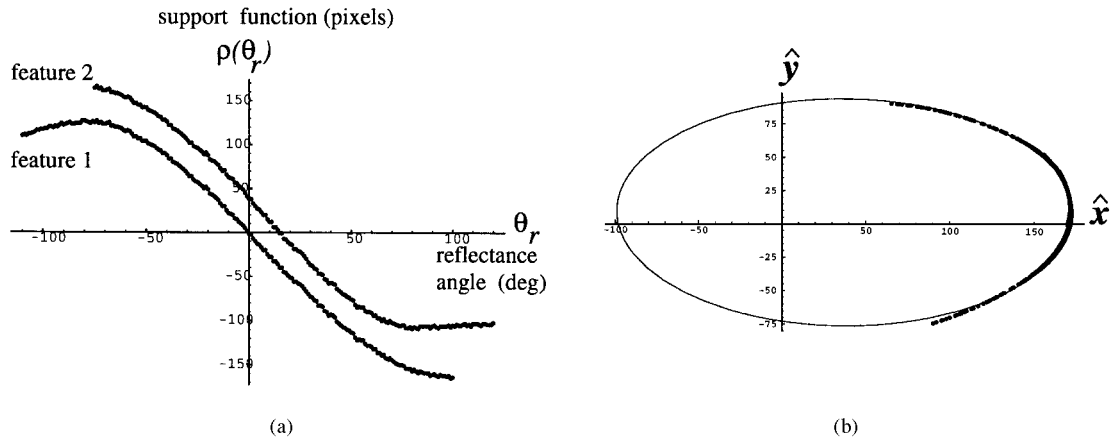


Figure 11. 2D profile of an ellipsoid recovered by tracking two unknown features. (a) Support functions of the two features computed from their image trajectories. (b) The recovered (dots) and actual (solid line) profiles. The directions of the two unknown features were found to be  $\theta_i^1 = -26^\circ$  and  $\theta_i^2 = 50^\circ$ .

of  $C_1$  and  $C_2$  can be obtained using rough approximations for the size and location of the object. Note that none of the above approximations are critical to profile recovery; they are only used to expedite the search process. Once the two pairs of parameters,  $(\theta_i^1, C_1)$  and  $(\theta_i^2, C_2)$ , are determined, the support functions  $\rho_n^1(\theta_n)$  and  $\rho_n^2(\theta_n)$  can be computed without ambiguity. From the support functions the surface profile corresponding to each feature trajectory is independently recovered and then the two profiles are fused together to obtain a larger reconstructed profile. Figure 10(b) shows the recovered profile of the sphere. Experimental results for an oval-shaped object (elliptical profile with ratio between major and minor axes of 1.6) are shown in Fig. 11. In both experiments we see that the specular profiles are estimated with high accuracy.

The tracking of virtual features is harder than the tracking of real features. One practical problem is related to the quality of the surface finish. A poor finish can blur reflections which limit the types of features which can be tracked. A second problem arises at regions of very low curvature. In such cases, the distortion of the feature can be significant even for small viewpoint changes. The experiments reported here demonstrate that, when virtual features can be tracked with precision, caustics and surface profiles can be computed with high accuracy.

### 3. Recovery of 3D Surface Curves

We now generalize the results of the previous sections to 3D surfaces. The camera motion is no longer

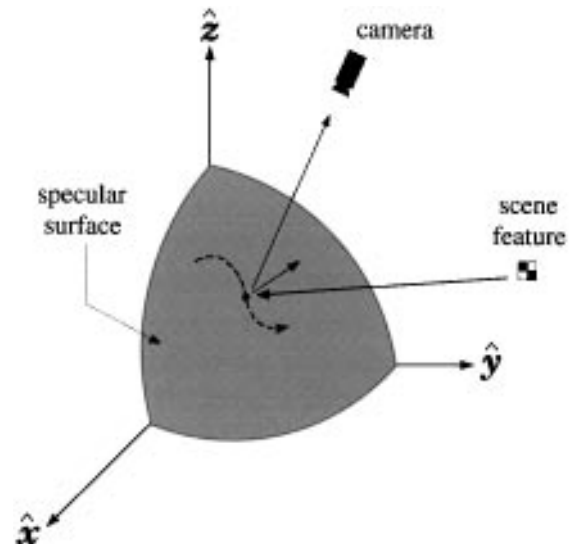


Figure 12. Recovery of surface geometry for general camera motion (along a 3D trajectory) cannot be decomposed into two 2D profile recovery problems.

confined to a plane, and consequently, the surface profile can be any smooth space curve. Before we proceed with our 3D analysis, it is worth elaborating on the problems we face. Figure 12 shows the tracking of a virtual feature as it travels on a 3D curve. If we project the curve onto a 2D plane, for instance the  $\hat{x}$ - $\hat{z}$  plane, the projection of any surface normal on the curve is not necessarily the bisector of the projections of the incidence and reflection angles. In general, there is no projection plane that obviates this problem. In short,

the problem of 3D curves *cannot* be reduced to a finite number of 2D profile problems. However, our results on 2D profiles have given us the basic tools and understanding necessary to proceed with the 3D case.

We begin by generalizing the notion of caustics to a family of reflected rays in 3D space. For the derivation of this caustic equation, we introduce a coordinate system that is attached to reflected rays. This simple idea has made the 3D case tractable. Using this moving coordinate system, we derive equations that relate the image trajectory of a virtual feature to the reflecting surface curve. We conclude this section with experimental verification of feature classification and 3D profile recovery.

In the following analysis, we assume that the camera motion is continuously differentiable and that the surface is twice differentiable. We also assume that the Gaussian curvature does not vanish along intervals on the surface profile traversed by the virtual feature. In regions with zero Gaussian curvature (parabolic lines and planar regions) all the points in the region will reflect a single feature in the same viewer direction. Such regions need to be analyzed by other means. For instance, regions with zero Gaussian curvature may be identified by the spread of virtual features.

### 3.1. 3D Caustic Curve and Feature Classification

In Section 2, the caustic was defined for a family of coplanar lines. In the current scenario, as the camera moves, reflected rays are no longer confined to a plane. Therefore, the family of lines is parameterized not by their direction (which is defined now by two angles) but rather by a parameter  $t$  which may be viewed as a *time* parameter. Other (allowable) parameters can be used too. If the envelope of this family of 3D rays exists, it is known as a 3D *caustic curve*.

For arbitrary camera motion in 3D, the family of reflected rays is given by the direction of each line, the unit vector  $\hat{\mathbf{v}}(t)$ , and its distance vector  $\vec{\mathbf{L}}(t)$  from the origin of the world coordinate system as illustrated in Fig. 13. At any given instance  $t$ , these two vectors are computed from the feature position in the image and the camera parameters. The vector  $\vec{\mathbf{L}}(t)$  is analogous to the support function<sup>3</sup>,  $\rho_n$ , introduced in Section 2.1.2. In the case of orthographic projection,  $\hat{\mathbf{v}}(t)$  is the optical axis and  $\vec{\mathbf{L}}(t)$  is simply a vector on the image plane between the image projections of the feature and the origin of the world coordinate system  $\vec{\mathbf{O}}$ . In the following derivation, we assume the existence of the caustic curve  $\vec{\mathbf{x}}_c$  and derive its equation. The caustic curve is

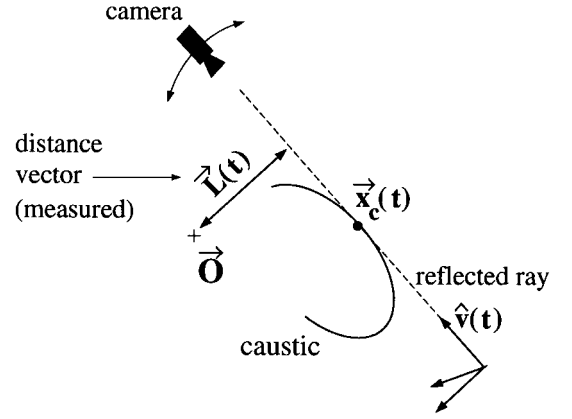


Figure 13. The 3D-caustic of a feature is the envelope of the reflected rays. The reflected ray is defined by its direction  $\hat{\mathbf{v}}(t)$  and by its distance  $\vec{\mathbf{L}}(t)$  from the origin of the world coordinate system. Both vectors,  $\vec{\mathbf{L}}(t)$  and  $\hat{\mathbf{v}}(t)$ , are determined by the position of the feature in the image and the camera parameters.

also parameterized by  $t$  such that the point  $\vec{\mathbf{x}}_c(t)$  on the caustic is tangent to the ray  $\hat{\mathbf{v}}(t)$ . The main idea behind the derivation of the caustic curve, is to decompose the caustic point position at any given  $t$  into two orthogonal components as follows:

$$\vec{\mathbf{x}}_c(t) = \vec{\mathbf{L}}(t) + \langle \vec{\mathbf{x}}_c(t), \hat{\mathbf{v}}(t) \rangle \hat{\mathbf{v}}(t) \quad (19)$$

The above decomposition expresses the unknown caustic curve  $\vec{\mathbf{x}}_c(t)$  in a moving coordinate system that is attached to the ray  $\hat{\mathbf{v}}(t)$ . In this coordinate system, the two orthogonal components of  $\vec{\mathbf{x}}_c(t)$  which are given by  $\vec{\mathbf{L}}(t)$  are known, and the third component along  $\hat{\mathbf{v}}(t)$  needs to be determined.

The first step is to differentiate (19) with respect to  $t$ . This yields the tangent to the caustic curve:

$$\begin{aligned} \dot{\vec{\mathbf{x}}}_c(t) &= \dot{\vec{\mathbf{L}}}(t) + \langle \vec{\mathbf{x}}_c(t), \hat{\mathbf{v}}(t) \rangle \dot{\hat{\mathbf{v}}}(t) \\ &\quad + (\langle \dot{\vec{\mathbf{x}}}_c(t), \hat{\mathbf{v}}(t) \rangle + \langle \vec{\mathbf{x}}_c(t), \dot{\hat{\mathbf{v}}}(t) \rangle) \hat{\mathbf{v}}(t) \end{aligned} \quad (20)$$

Since  $\vec{\mathbf{x}}_c(t)$  is the caustic curve, for any given  $t$  its tangent  $\dot{\vec{\mathbf{x}}}_c(t)$  is parallel to  $\hat{\mathbf{v}}(t)$  and therefore perpendicular to  $\dot{\hat{\mathbf{v}}}(t)$ :

$$\langle \dot{\vec{\mathbf{x}}}_c(t), \dot{\hat{\mathbf{v}}}(t) \rangle = 0 \quad (21)$$

Taking the inner product of both sides of Eq. (20) with  $\dot{\hat{\mathbf{v}}}(t)$  and using  $\langle \hat{\mathbf{v}}(t), \dot{\hat{\mathbf{v}}}(t) \rangle = 0$ , we get:

$$0 = \langle \dot{\vec{\mathbf{L}}}(t), \dot{\hat{\mathbf{v}}}(t) \rangle + \langle \vec{\mathbf{x}}_c(t), \hat{\mathbf{v}}(t) \rangle \|\dot{\hat{\mathbf{v}}}(t)\|^2 \quad (22)$$

From the above equation, we get an expression for  $(\vec{x}_c(t), \hat{v}(t))$  that can be substituted back in (19). The result is the caustic curve equation:

$$\vec{x}_c(t) = \vec{L}(t) - \frac{\langle \dot{\vec{L}}(t), \hat{v}(t) \rangle}{\|\dot{\hat{v}}(t)\|^2} \hat{v}(t) \quad (23)$$

Comparing this to (7) we see that the 2D caustic is a special case of the 3D caustic, i.e., for coplanar motion when the curve parameter<sup>4</sup> is  $t = \theta_r$ . It may be noted that the caustic does not always exist for a one-parameter family of lines<sup>5</sup>. Since the tangent vector  $\dot{\vec{x}}_c(t)$  must be parallel to  $\hat{v}(t)$ , for a caustic to exist, the following condition must also hold true in addition to Eq. (21):

$$\langle \dot{\vec{x}}_c(t), \hat{v}(t) \times \dot{\hat{v}}(t) \rangle = 0 \quad (24)$$

However, this additional condition may not always be satisfied<sup>6</sup>.

As in the 2D case, Eq. (23) can be used for feature classification. For a real feature, the caustic exists and it is a point. In contrast, for a virtual feature, the curve defined by Eq. (23), will be a space curve. In Appendix B, we show that the only case where a virtual feature produces a point caustic is the case of a paraboloid

surface whose axis is parallel to the feature direction. From a practical perspective, this is clearly a pathological case. Clearly, not a common case in practical applications.

### 3.2. Experiments: 3D Caustic Curves and Feature Classification

We now present experimental results on classification of real and virtual features based on their caustic curves. The object used is the one shown in Fig. 5, it has a single surface marking (real feature) and a virtual feature that is almost identical in appearance to the real one. The camera motion is not planar but rather an arbitrary smooth 3D trajectory. The caustics shown in Fig. 14(b) are computed from the image trajectories shown in Fig. 14(a) using Eq. (23). As expected, the caustic curve of the real feature is a small cluster of points centered around the actual feature position, while the caustic curve of the virtual feature is a space curve. Again, the caustics can be distinguished by measuring their compactness. The measure we used is the sum of squares of distances of the caustic's points from its centroid. The ratio of the sums of squares of distances of points on the two caustics was found to be 12:1.

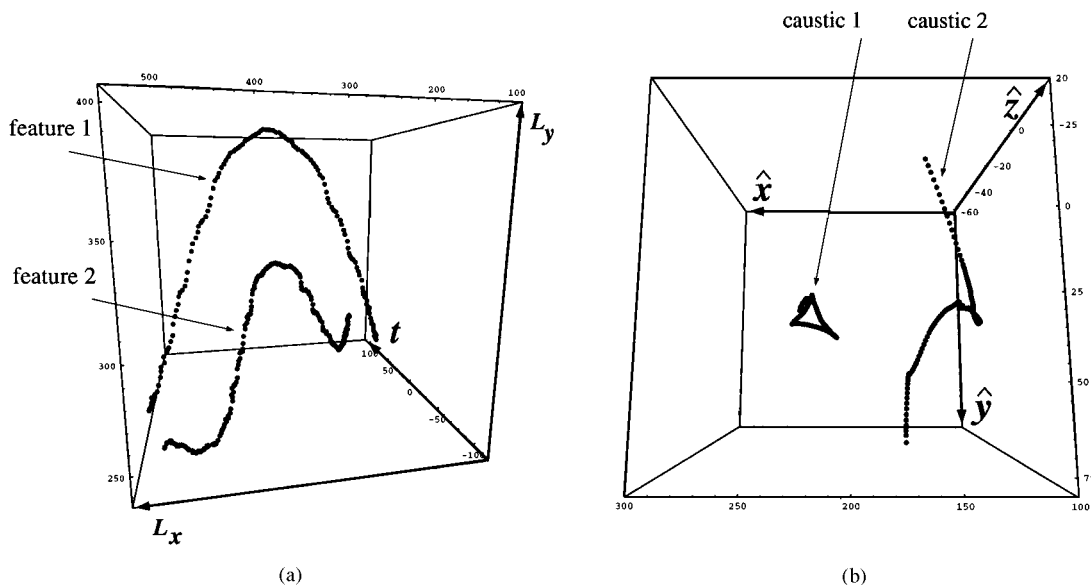


Figure 14. (a) Image trajectories,  $\vec{L}(t)$ , of two features plotted as parameterized curves. (b) The computed 3D caustic curves. The caustic of the real feature (1) is a compact cluster while that of the virtual feature (2) is a space curve. The two are easily distinguished using a measure of compactness.

### 3.3. Surface Curve Equation for 3D Camera Motion

As the observer moves around a specular object, each virtual feature travels along a 3D profile on the object's surface. The recovery of the surface profile  $\vec{x}(t)$  directly from the image trajectory, given by the pair  $(\vec{L}(t), \hat{v}(t))$ , is a very difficult problem. Therefore, as with the analysis of the 2D-profile recovery problem in Section 2.4, our goal here is to first recover the support function  $\rho_n(t)$ , which is the distance of the *tangent plane* at the point  $\vec{x}(t)$  from the world origin  $\vec{O}$ . As shown in Appendix A, the support function serves as a dual representation not only in the case of planar curves, but also surfaces. These results suggest that the study of the support function and its derivative would be worthwhile also in the case of 3D-profile recovery.

As in Section 3.1, the key step is to decompose the surface curve into two orthogonal components as:

$$\vec{x}(t) = \vec{L}(t) + \langle \vec{x}(t), \hat{v}(t) \rangle \hat{v}(t) \quad (25)$$

The unknown quantity is the component of  $\vec{x}(t)$  along the reflected ray which is given by  $\langle \vec{x}(t), \hat{v}(t) \rangle$ . The fundamental equation that defines the relationship between the support function and the surface is:

$$\rho_n(t) = \langle \vec{x}(t), \hat{n}(t) \rangle \quad (26)$$

where, the unit vector  $\hat{n}(t)$  is the surface normal at  $\vec{x}(t)$ . Differentiation of the above expression with respect to  $t$  gives:

$$\dot{\rho}_n(t) = \langle \dot{\vec{x}}(t), \hat{n}(t) \rangle + \langle \vec{x}(t), \dot{\hat{n}}(t) \rangle$$

The vector  $\dot{\vec{x}}(t)$  is tangent to the surface and therefore orthogonal to  $\hat{n}(t)$ . This simplifies the above expression to:

$$\dot{\rho}_n(t) = \langle \vec{x}(t), \dot{\hat{n}}(t) \rangle \quad (27)$$

It can be shown that (26) and (27) reduce to Eq. (5), for the special case of a 2D profile with  $t = \theta_r$ . Now, the law of specular reflection can be written in vectorial form as,

$$\hat{v}(t) = 2\langle \hat{s}, \hat{n}(t) \rangle \hat{n}(t) - \hat{s} \quad (28)$$

where the unit vector  $\hat{s}$  is the feature direction. Using this expression for  $\hat{v}(t)$  in (25) and substituting  $\vec{x}(t)$

back in (26) and (27) we get:

$$\begin{aligned} \rho_n(t) &= \langle \vec{x}(t), \hat{v}(t) \rangle \langle \hat{s}, \hat{n}(t) \rangle + \langle \vec{L}(t), \hat{n}(t) \rangle \\ \dot{\rho}_n(t) &= -\langle \vec{x}(t), \hat{v}(t) \rangle \langle \hat{s}, \dot{\hat{n}}(t) \rangle + \langle \vec{L}(t), \dot{\hat{n}}(t) \rangle \end{aligned}$$

Multiplying the first equation by  $\langle \hat{s}, \dot{\hat{n}}(t) \rangle$ , the second by  $\langle \hat{s}, \hat{n}(t) \rangle$ , and adding the results we have:

$$\begin{aligned} \rho_n(t) \langle \hat{s}, \dot{\hat{n}}(t) \rangle + \dot{\rho}_n(t) \langle \hat{s}, \hat{n}(t) \rangle &= \langle \vec{L}(t), \hat{n}(t) \rangle \langle \hat{s}, \dot{\hat{n}}(t) \rangle + \langle \vec{L}(t), \dot{\hat{n}}(t) \rangle \langle \hat{s}, \hat{n}(t) \rangle \\ &= \langle \vec{L}(t), \langle \hat{s}, \dot{\hat{n}}(t) \rangle \hat{n}(t) + \langle \hat{s}, \hat{n}(t) \rangle \dot{\hat{n}}(t) \rangle \quad (29) \end{aligned}$$

This result can be further simplified. To this end, we find the derivative of  $\hat{v}(t)$  with respect to  $t$  using (28). Since the feature direction  $\hat{s}$  is a constant we get:

$$\frac{1}{2} \dot{\hat{v}}(t) = \langle \hat{s}, \dot{\hat{n}}(t) \rangle \hat{n}(t) + \langle \hat{s}, \hat{n}(t) \rangle \dot{\hat{n}}(t) \quad (30)$$

Note that the right hand side of the above expression figures explicitly in (29). Substituting the left hand side in its place gives the following fundamental relationship between the image trajectory  $\vec{L}(t)$  and the support function  $\rho_n(t)$  of the surface profile:

$$\frac{d}{dt} (\rho_n(t) \langle \hat{s}, \hat{n}(t) \rangle) = \frac{1}{2} \langle \vec{L}(t), \dot{\hat{v}}(t) \rangle \quad (31)$$

Then, by integration we have:

$$\rho_n(t) = \frac{\frac{1}{2} \int_{t_0}^t \langle \vec{L}(t'), \dot{\hat{v}}(t') \rangle dt' + \rho_n(t_0) \langle \hat{s}, \hat{n}(t_0) \rangle}{\langle \hat{s}, \hat{n}(t) \rangle} \quad (32)$$

Note that even for a known feature direction the above solution is determined only up to an unknown parameter, namely, the constant of integration:

$$C = \rho_n(t_0) \langle \hat{s}, \hat{n}(t_0) \rangle \quad (33)$$

The above ambiguity is inherent to the recovery problem;  $C$  is determined by  $\rho_n(t_0)$  which is itself unknown. However, if the integration in (32) is started from an occluding boundary, the ambiguity is eliminated; the unknown constant vanishes since  $\langle \hat{s}, \hat{n}(t_0) \rangle = 0$ .

### 3.4. Curve Equation from the Support Function

Our final result is a closed-form solution that enables the unique recovery of a surface curve from the above support function,  $\rho_n(t)$ . The unknown quantity we seek

is  $\langle \vec{\mathbf{x}}(t), \hat{\mathbf{v}}(t) \rangle$ . Our first step is to obtain an expression for the support function by taking the inner product of (25) with the normal vector  $\hat{\mathbf{n}}(t)$ :

$$\langle \vec{\mathbf{x}}(t), \hat{\mathbf{n}}(t) \rangle = \langle \vec{\mathbf{x}}(t), \hat{\mathbf{v}}(t) \rangle \langle \hat{\mathbf{v}}(t), \hat{\mathbf{n}}(t) \rangle + \langle \vec{\mathbf{L}}(t), \hat{\mathbf{n}}(t) \rangle \quad (34)$$

The left hand side of the above equation is the support function  $\rho_n(t)$ . After rearranging terms, we get the desired unknown quantity:

$$\langle \vec{\mathbf{x}}(t), \hat{\mathbf{v}}(t) \rangle = \frac{\rho_n(t) - \langle \vec{\mathbf{L}}(t), \hat{\mathbf{n}}(t) \rangle}{\langle \hat{\mathbf{v}}(t), \hat{\mathbf{n}}(t) \rangle} \quad (35)$$

Substituting this back in (25) gives us the surface curve:

$$\vec{\mathbf{x}}(t) = \vec{\mathbf{L}}(t) + \left( \frac{\rho_n(t) - \langle \vec{\mathbf{L}}(t), \hat{\mathbf{n}}(t) \rangle}{\langle \hat{\mathbf{v}}(t), \hat{\mathbf{n}}(t) \rangle} \right) \hat{\mathbf{v}}(t) \quad (36)$$

For the special case of 2D camera motion, if the caustic and the surface profile are parameterized by  $t = \theta_r$ , the surface profile equation is reduced to:

$$\begin{aligned} \begin{pmatrix} x(\theta_r) \\ y(\theta_r) \end{pmatrix} &= \rho_r(\theta_r) \begin{pmatrix} -\sin \theta_r \\ \cos \theta_r \end{pmatrix} \\ &+ \frac{\rho_n\left(\frac{\theta_r + \theta_i}{2}\right) + \rho_r(\theta_r) \sin\left(\frac{\theta_r - \theta_i}{2}\right)}{\cos\left(\frac{\theta_r - \theta_i}{2}\right)} \begin{pmatrix} \cos \theta_r \\ \sin \theta_r \end{pmatrix} \end{aligned} \quad (37)$$

Eq. (37) can be used instead of the inverse transform in Eq. (6) to reconstruct surface profiles. This is advantageous since the above expression does not require computation of the derivative of  $\rho_n(\theta_n)$ , which is noise sensitive and hence tends to introduce errors in profile recovery.

### 3.5. Experiments: Recovery of 3D Curves

Our last set of experiments are on recovery of 3D specular curves. In the 2D case, we were able to effectively use multiple features since the surface trajectories traveled by different features were guaranteed to overlap. This unfortunately is not the case with 3D surfaces; feature trajectories may at best intersect but not likely to overlap over large surface curve segments. This forces us to use one feature at a time and resolve shape ambiguity by tracking each feature from the occluding boundary of the surface. As explained in the Section

3.3, this forces the constant of integration to vanish and results in unambiguous recovery of 3D curves.

In these experiments, we tracked the reflection of a point light source (highlight) rather than the reflection of a scene feature. However, the two are equivalent from a theoretical perspective. The object used is a chrome-plated sphere of radius 2.5 cm. The camera mounted on the robot was moved around at a distance of about 20 cm from the object. This distance was selected based on the robot's limited workspace; it enables us to use a large range of viewing directions. The camera trajectory used is plotted in Fig. 15. Since feature tracking near the occluding boundary is very difficult due to the distortion of the feature, we tracked the feature from the middle of the sequence toward the occluding boundary. The surface curve was recovered from the image trajectory of the virtual feature using Eq. (32). The recovered (dots) and actual (solid line) surface profiles are displayed in Fig. 16 using two different viewpoints. We see that the recovered curve is in strong agreement with the actual surface curve.

## 4. Summary

Specular reflection is a phenomenon that is ubiquitous in the real world; it is exhibited to some degree by most real surfaces. A clear understanding of specular surfaces and their appearance in brightness images is therefore fundamental to progress in computational vision. In this paper, we explored the information regarding specular surface geometry that is available to a moving observer.

We have introduced a mathematical framework for analyzing the relation between specular surface geometry and image trajectories of scene features reflected by the surface. It was shown that analysis of specular surfaces is tractable only if the representations for surface curves and reflected rays are carefully chosen. We invoked the notion of caustic curves to represent trajectories produced by image features. Caustics were shown to hold valuable information regarding scene geometry.

A simple but effective algorithm was presented for classifying all image features into two basic categories: real and virtual. Real features are scene features such as surface markings and texture that cling to their surfaces. These features are the only sort that must be directly used by vision algorithms such as structure from motion. In contrast, virtual features are reflections of scene features by a specular surface. Unlike



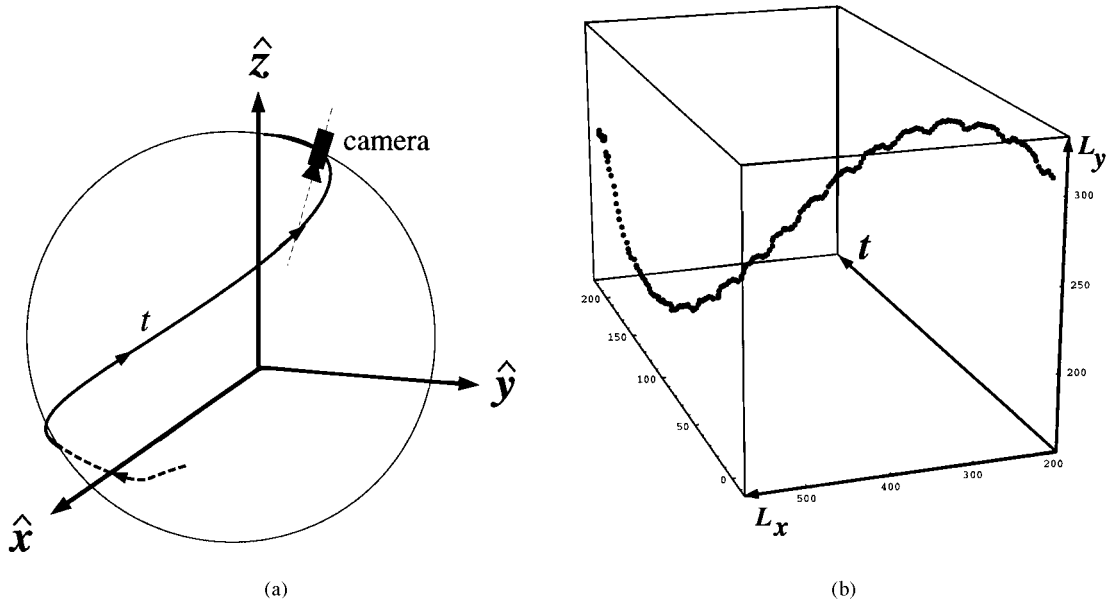


Figure 15. (a) Actual trajectory of camera motion plotted on a sphere. (b) The tracking of a virtual feature. The trajectory is parameterized by time  $t$  shown here as the third dimension. The remaining two coordinates represent the distance vector  $\vec{L}(t)$ .

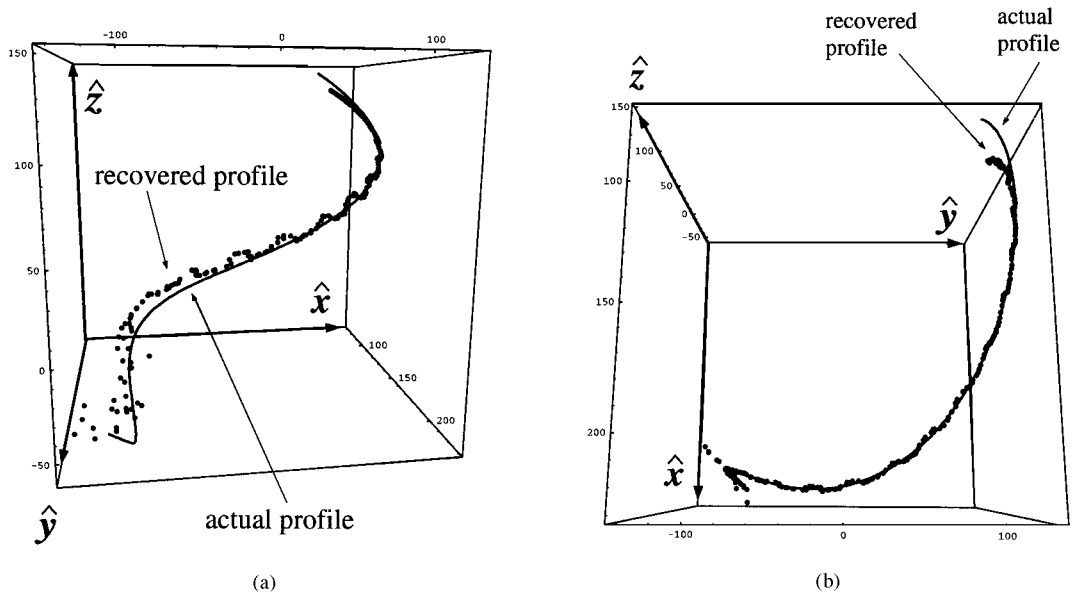


Figure 16. 3D surface curve on a chrome-plated ball recovered and displayed from two different viewpoints. The center of the sphere is at  $(20, -5, -50)$  in the cube (wireframe) and its radius is 225 pixels. The recovered surface profile (dots) is seen to closely match the actual profile (solid line). This shape information is obtained from the image trajectory of a single virtual feature.

real features, they travel on specular surfaces when the observer changes his/her viewpoint. We showed that the caustics of real and virtual features have distinctly different anatomies; one is a compact cluster while the other is an arbitrary space curve. These properties of

feature caustics were used to develop a classification algorithm. This algorithm can serve as a useful precursor to vision techniques that deal with motion sequences.

Finally, we showed that virtual features must not be quickly discarded as they contain valuable information

regarding the shapes of specular objects in the scene. In the case of pure specular surfaces (smooth metals, glass, etc.), virtual features are the only available source of visual information. We derived shape recovery equations that relate the image trajectory of a virtual feature to the profile of the reflecting surface. Though the problem of specular profile recovery was considered severely under-constrained in the past, we demonstrated that it is possible and can be performed with reasonable accuracy. The first recovery technique we introduced was for 2D profiles. It exploits multiple virtual features. We showed that even when the positions of the reflected scene features are unknown, and no point on the profile is given, the profile can be uniquely reconstructed and the feature directions determined. The second recovery technique we introduced is for 3D surface profiles. It uses only one feature but requires that the feature be tracked from the occluding boundary of the object. Both techniques were demonstrated via experiments.

A problem that future work may address is that of combining constraints from multiple features for general 3D camera motion. Another interesting problem is the recognition and pose estimation of specular objects from a motion sequence. It is not entirely clear as to what further information is contained in the caustics of virtual features and whether image trajectories can be directly used to assign surfaces to different parametric classes. It is hoped that the results reported in this paper will aid in the investigation of these problems.

## A. Surface Equation from the Support Function

For any regular surface (with the appropriate smoothness conditions), the support function  $\rho$  is defined for every point. In this appendix, we show that the support function  $\rho(\hat{\mathbf{n}})$  parameterized by the surface normal  $\hat{\mathbf{n}}$  defines the surface completely. More precisely, we show that at any surface point with non-zero Gaussian curvature, the support function determines the surface uniquely in the neighborhood of the point. Furthermore, we will derive an explicit formula, analogous to Eq. (6), that expresses the surface equation as a function of the support function and its partial derivatives.

Given the support function  $\rho(\hat{\mathbf{n}})$  in the neighborhood of a point with non-zero Gaussian curvature, we would like to find the surface equation  $\vec{\mathbf{x}}(\hat{\mathbf{n}})$ . The fundamental relationship between the support function and the

surface is given by Eq. 26 which is restated below:

$$\langle \vec{\mathbf{x}}, \hat{\mathbf{n}} \rangle = \rho(\hat{\mathbf{n}}) \quad (38)$$

We can express the normal vector  $\hat{\mathbf{n}}$  using partial derivatives of the the unknown surface  $\vec{\mathbf{x}}$  and expand the left side of the above equation. However, this expansion results in a nonlinear partial differential equation. Rather than solving this equation directly, we pursue an alternative approach that requires only differentiation and matrix inversion. Let  $\vec{\mathbf{x}}(u, v)$  be a parameterization of the surface. We can differentiate Eq. (38) with respect to the parameter  $u$ , and use the fact that  $\langle \vec{\mathbf{x}}_u, \hat{\mathbf{n}} \rangle = 0$  to obtain:

$$\langle \vec{\mathbf{x}}, \hat{\mathbf{n}}_u \rangle = \rho_u(\hat{\mathbf{n}}) \quad (39)$$

where, the subscript denotes the partial derivative with respect to  $u$ . Similarly, we differentiate Eq. (38) with respect to the parameter  $v$ , and write the result together with Eqs. (38) and (39) in matrix form as follows:

$$\begin{pmatrix} \hat{\mathbf{n}} \\ \hat{\mathbf{n}}_u \\ \hat{\mathbf{n}}_v \end{pmatrix} \vec{\mathbf{x}}^T = \begin{pmatrix} \rho \\ \rho_u \\ \rho_v \end{pmatrix} \quad (40)$$

The vectors  $\hat{\mathbf{n}}_u$  and  $\hat{\mathbf{n}}_v$  are perpendicular to  $\hat{\mathbf{n}}$  and are independent of each other since the Gaussian curvature is non-zero and  $(u, v)$  is a parameterization of the surface (do\_Carmo, 1976). Therefore, all the three rows of the matrix are linearly independent and the matrix is invertible. Hence, the surface equation is obtained as:

$$\vec{\mathbf{x}}^T = \begin{pmatrix} \hat{\mathbf{n}} \\ \hat{\mathbf{n}}_u \\ \hat{\mathbf{n}}_v \end{pmatrix}^{-1} \begin{pmatrix} \rho \\ \rho_u \\ \rho_v \end{pmatrix} \quad (41)$$

The above expression shows that the support function in the neighborhood of a point with non-zero Gaussian curvature determines the surface uniquely. Furthermore, it provides an explicit formula for the calculation of the surface equation. The above analysis excludes points with zero Gaussian curvature—parabolic and planar points—since, in the neighborhood of these points, the mapping between surface points and their normals (the Gauss map) is not one-to-one. However, this restriction is not very severe since, using the continuity of the surface, we can extend the reconstruction over parabolic and planar regions.

## B. Single-Point Caustics

In this appendix, we study the conditions under which a virtual feature in direction  $\hat{s}$  will produce a single-point caustic. If the computed caustic<sup>7</sup> is a fixed point,  $\vec{p}$ , then we can, without loss of generality, translate the origin of the coordinate system such that it coincides with the point  $\vec{p}$ :

$$\vec{x}_c(t) = \vec{0} \quad (42)$$

From Eq. (23) we get:

$$\vec{L}(t) = \frac{\langle \dot{\vec{L}}(t), \dot{\hat{v}}(t) \rangle}{\|\dot{\hat{v}}(t)\|^2} \hat{v}(t) \quad (43)$$

On the other hand, the distance vector  $\vec{L}(t)$ , by its definition, must be orthogonal to the reflected ray:

$$\vec{L}(t) \perp \hat{v}(t) \quad (44)$$

From the last two equations, we find that the distance vector must be the zero vector:

$$\vec{L}(t) = \vec{0} \quad (45)$$

Substituting this result into the recovery expression (31), shows that the support function of the surface  $\rho_n(t)$  must satisfy:

$$\frac{d}{dt}(\rho_n(t) \langle \hat{s}, \hat{n}(t) \rangle) = 0 \quad (46)$$

or equivalently:

$$\rho_n(t) = \frac{C}{\langle \hat{s}, \hat{n}(t) \rangle} \quad (47)$$

for some constant  $C$ .

The above equation must be satisfied by all points on the surface traversed by the virtual feature. It has to be satisfied by all surface points if the caustic is a point for any arbitrary camera motion. Our next step is to find the surface. Without loss of generality, we can rotate the coordinate system such that the feature direction  $\hat{s}$  coincides with the  $\hat{z}$ -axis of the coordinate system. The unknown surface  $(x, y, z(x, y))$  can be parametrized using the partial derivatives:

$$p = z_x \quad q = z_y \quad (48)$$

This results in the following expressions for the surface normal and the support function:

$$\hat{n}(p, q) = \frac{(p, q, -1)}{\sqrt{p^2 + q^2 + 1}} \quad (49)$$

$$\rho(p, q) = -C\sqrt{p^2 + q^2 + 1} \quad (50)$$

Differentiating each of the above equations by  $p$  and  $q$  and using Eq. (41), we get:

$$(x, y, z) = (-2Cp, -2Cq, C(1 - p^2 - q^2)) \quad (51)$$

which is a parametric representation of the family of paraboloids:

$$z = -\frac{1}{4C}(x^2 + y^2) + C \quad (52)$$

In summary, the above analysis shows that the caustic of a virtual feature is a point when the reflecting surface is a paraboloid whose axis is parallel to the feature direction. This result holds true for the 2D case. A 2D caustic will be point if the specular curve is a parabola whose axis is parallel to the feature direction. It is interesting to note that the inverse result is well-known: A light source placed at the focus of a paraboloid mirror will be reflected as a beam of parallel rays. Here, we have provided a proof for the uniqueness of the family of surfaces with this property using the caustic Eq. (23) and the shape recovery Eq. (31).

## Acknowledgment

This work was supported in part by an NSF National Investigator Award and in part by ARPA contract DACA-76-92-C-007. Michael Oren was supported by an IBM Graduate Fellowship. We thank Peter Giblin of the University of Liverpool for his comments on the existence of caustics, and Ruud Bolle of IBM for his constant support of this work. Thanks also to Kristin Dana and the anonymous referees for their careful reviews of the paper.

## Notes

1. Note that a virtual feature is not necessarily a highlight. It could be the reflection of any scene point.
2. It is easy to convince oneself of this by using the shape recovery equations to be derived in Section 2.4. A more detailed treatment of 3D caustics is given in Appendix B.

3. It is important to note that  $\bar{L}$  is not the support function. For a surface, the support function  $\rho_n$  is the distance of the *tangent plane* from the origin (see (Guggenheimer, 1963)).
4. Using the following notation:  $\hat{\nu} = (\cos \theta_r, \sin \theta_r)$  and  $\bar{L} = (\rho_r \cos(\theta_r + \frac{\pi}{2}), \rho_r \sin(\theta_r + \frac{\pi}{2}))$ .
5. This fact was pointed out by P.J. Giblin (1995).
6. The 3D-caustic exists if the reflected rays form a developable surface which is not a generalized cylinder or a cone (see also (Bruce and Giblin, 1992)).
7. In this discussion, we only use the term *caustic*, since, if the curve defined by Eq. (23) is a point then it is a caustic, although a degenerate one.

## References

- Bellver-Cebreros, C. and Rodriguez-Danta, M. 1992. Caustics and the Legendre transform. *Optics Communications*, 92(4–6):187–192.
- Blake, A. 1985. Specular stereo. *Proc. 9th IJCAI Conf.*, pp. 973–976.
- Blake, A. and Brelstaff, G. 1988. Geometry from specularity. *Proceedings of ICCV*, Florida, pp. 394–403.
- Blake, A. and Bulthoff, H. 1991. Shape from specularity: Computation and psychophysics. *Phil. Trans. Royal Society London B.*, 331:237–252.
- Brelstaff, G. and Blake, A. 1988. Detecting specular reflections using Lambertian constraints. *Proceedings of ICCV*, Florida, pp. 297–302.
- Bruce, J.W. and Giblin, P.J. (Eds.) 1992. *Curves and Singularities*. Cambridge University Press, second edition.
- Cornbleet, S. 1984. *Microwave and Optical Ray Geometry*. John Wiley and Sons.
- do Carmo, M.P. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall Inc.
- Giblin, P.J. 1995. Personal communication.
- Guggenheimer, H.W. 1963. *Differential Geometry*. McGraw-Hill Book Company, Reprinted by Dover Publications (1977).
- Healey, G. and Binford, T.O. 1988. Local shape from specularity. *CVGIP*, pp. 62–86.
- Horn, B.K.P. 1986. *Robot Vision*. The MIT Press.
- Ikeuchi, K. 1981. Determining surface orientation of specular surfaces by using the photometric stereo method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):661–669.
- Klinker, G.J., Shafer, S.A., and Kanade, T. 1988. The measurement of highlights in color images. *International Journal of Computer Vision*, 2:7–32.
- Koenderink, J.J. and van Doorn, A.J. 1980. Photometric invariants related to solid shapes. *Optica Acta*, 27(7):981–996.
- Lee, S.W. 1991. Understanding of surface reflections in computer vision by color and multiple views. Ph.D. Thesis, University of Pennsylvania.
- Longuet-Higgins, M.S. 1960. Reflection and refraction at a random moving surface: (i) Patterns and paths of specular points. *Journal of the Optical Society of America*, 509:838–844.
- Nayar, S.K., Ikeuchi, K., and Kanade, T. 1990. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Transactions on Robotics and Automation*, 6(4):418–431.
- Nayar, S.K., Fang, X.S., and Boulton, T. 1996. Separation of reflection components using color and polarization. *International Journal of Computer Vision*, To appear in 1996.
- Sanderson, A.C., Weiss, L.E., and Nayar, S.K. 1988. Structured high-light inspection of specular surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(1):44–55.
- Schultz, H. 1994. Retrieving shape information from multiple images of a specular surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):195–201.
- Stavroudis, O.N. 1972. *The Optics of Rays, Wavefronts, and Caustics*. Academic Press.
- Symosek, P.F. 1985. Parameter estimation and classification of machine parts based on specular or mirror-like image data. Ph.D. Thesis, Brown University.
- Thrift, P. and Lee, C.-H. 1983. Using highlights to constrain object size and location. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3):426–431.
- Torrance, K. and Sparrow, E. 1967. Theory for off-specular reflection from rough surfaces. *Journal of the Optical Society of America*, 57:1105–1114.
- Ullman, S. 1976. On visual detection of light sources. *Biological Cybernetics*, 21:205–211.
- Waldon, S. and Dyer, C.R. 1993. Dynamic shading, motion parallax and qualitative shape. *Proceedings of the IEEE Workshop on Qualitative Vision*, pp. 61–70.
- Wolff, L.B. and Boulton, T.E. 1991. Constraining object features using a polarization reflectance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(6):635–657.
- Zheng, J.Y., Fugakawa, Y., and Abe, N. 1995. Shape and model from specular motion. *Proceedings of ICCV*, pp. 72–79.
- Zisserman, A., Giblin, P., and Blake, A. 1989. The information available to a moving observer from specularities. *Image and Vision Computing*, 7:287–291.