A Theory of Specular Surface Geometry*

Michael Oren
Department of Computer Science
Columbia University
New York, NY 10027
oren@cs.columbia.edu

Shree K. Nayar
Department of Computer Science
Columbia University
New York, NY 10027
nayar@cs.columbia.edu

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 novel 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. 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 Specular Surfaces

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 ([Blake-1985],[Waldon and Dyer-1993]). 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 estimate the shape of a specular object (see [Ikeuchi-1981], [Nayar *et al.*-1990],[Sanderson *et al.*-1988] and [Schultz-1994]).

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[1] which travels over the surface when the observer moves. Given the fundamental nature of the difference between real and virtual features, they must be distinguished before they are used (or discarded) by existing vision algorithms.

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 in [Ullman-1976],[Symosek-1985],[Healey and Binford-1987], [Brelstaff and Blake-1988], and [Lee-1991], are limited in their applicability.

Since real and virtual features can be indistinguishable from a single image, we seek to investigate specular surfaces in the context of a moving observer. Two questions arise in this setting: First, how can real and virtual features be distinguished from their image trajectories? Second, what information regarding surface shape is contained in the image trajectory of a virtual feature? These questions have also intrigued other investigators in the past [Longuet-Higgens-1960], [Koenderink and van Doorn-1980], [Blake-1985], [Blake and Brelstaff-1988], [Blake and Bulthoff-1991],[Waldon and Dyer-1993]. However, we still lack a complete understanding of what information regarding scene geometry can be extracted from specularities.

It turns out that even for a moving observer, exploitation of virtual features is a difficult problem. Zisserman *et. al* [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 was only recently that Bellver-Cebreros

*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.

[1] It is worth noting that a virtual feature is not necessarily a highlight. It could be the reflection of any scene point.

and Rodriguez-Danta [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. Both of the above results are based on the assumption that the light source position is known. 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, remain unsolved problems.

The first half of our work 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 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* ([Cornbleet-1984],[Stavroudis-1972]). We demonstrate how caustics can be used 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 surface profiles from just two virtual features. In contrast to previous work, this approach does not require prior knowledge of scene features or points on the profile. Experimental results are presented that verify the effectiveness of both feature classification and 2D profile recovery.

In the second half of this 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 general problem of 3D profile recovery 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. This derivation was made possible by using a coordinate frame that is attached to the specularly reflected ray. Mathematical conciseness resulting from this idea also enables us to derive a novel 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 successfully verified via experiments.

2 Recovery of 2D Surface Profiles

We begin by analyzing the 2D case where camera motion, the scene feature, and the specular surface profile are all confined to a plane. We assume that the profile curve is smooth (has a tangent at all points) and does not consist of straight line segments. The case of a straight line (constant normal) is a special one that can be detected by other means; it causes the reflection of a well-defined point feature to be spread over a large part of the segment. This case is excluded in our analysis.

2.1 Curve Representation

First, we present a general represntation of curves which is based on the support function. This representation will play a critical role in the classification of real and virtual features as well as the recovery of specular surfaces.

2.1.1 Envelopes and the Legendre Transform

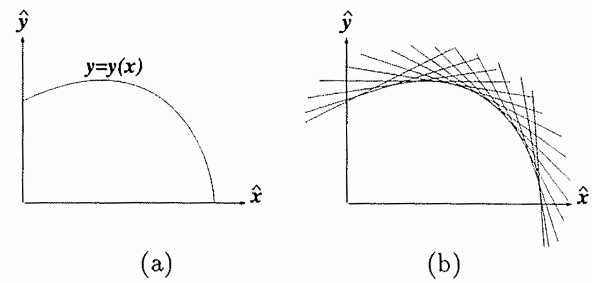


Figure 1: (a) Representation of a curve using its Cartesian coordinates (x, y). (b) Representation of a curve as an envelope of tangents.

The most common description of a 2D curve is its representation as a collection of points given by their Cartesian coordinates, (x, y) (see Figure 1(a)). Though this representation is conceptually simple, it results in complex equations when used to describe the geometry of specular reflection. The direction of the reflected ray changes with observer motion and hence there is no convenient Cartesian coordinate system that can be used to express the constraints on the slope and position of the reflecting point. A more natural choice is to parametrize the curve by its slope. The curve may then be viewed as an envelope of surface tangents rather than a collection of points, as illustrated in Figure 1(b).

What is the most suitable representation for an envelope of tangents? Bellver-Cebreros and Rodriguez-Danta [Bellver-Cebreros and Rodriguez-Danta-1992] suggested using the Legendre transform of the curve equation. The Legendre transform, $\Psi(\xi)$, of a differentiable function of one variable, $y(x)$, is:

$$\Psi(\xi) = \xi x - y(x) \quad \text{where} \quad \xi = y'(x) \tag{1}$$

A geometric interpretation of the Legendre transform is illustrated in Figure 2(a). If $(x, y(x))$ is the Cartesian representation of a curve, then, ξ 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.

2.1.2 Support Function of a Curve

While the Legendre transform of a curve simplifies the treatment of specular reflectance (see [Bellver-Cebreros and Rodriguez-Danta-1992]), it suffers from a few drawbacks. A major disadvantage is that neither Ψ nor ξ 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. This motivated us to represent the tangent line not with the slope and intersection point but with its distance to the origin, ρ_n, and the normal angle, θ_n, as shown in Figure 2(b). The function, $\rho_n(\theta_n)$, is called the *support function* of the curve [Guggenheimer-1977].

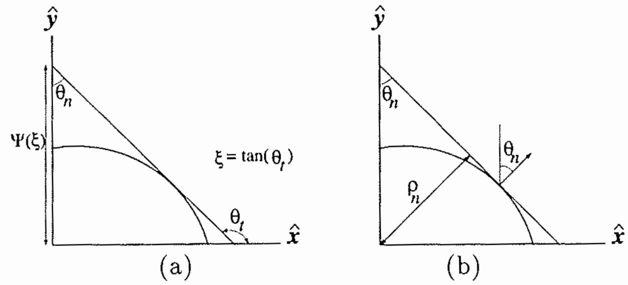


Figure 2: (a) Representation of a tangent using $(\Psi(\xi), \xi)$. (b) Representation of a tangent using the support function, i.e. the distance ρ_n from the origin and the normal angle θ_n.

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

$$\theta_n = \tan^{-1}(-1/y'(x)) \quad (2)$$

$$\rho_n = -x \sin\theta_n + y \cos\theta_n \quad (3)$$

Calculating the derivative of ρ_n in (3) with respect to θ_n and using (2) yields the following mapping:

$$\begin{pmatrix} \rho_n \\ \rho_n' \end{pmatrix} = \begin{pmatrix} \cos\theta_n & \sin\theta_n \\ -\sin\theta_n & \cos\theta_n \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (4)$$

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

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

This representation has the advantage that it depends explicitly on the slope of the curve. Further, under a rotation of the coordinate system, ρ_n remains unchanged while θ_n is only subjected to a simple linear shift.

2.2 2D Caustics and Feature Classification

We are now ready to address the classification of features into real and virtual ones. Figure 3 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). Despite these differences, the two features appear almost identical and hence are indistinguishable from a single image. We introduce an algorithm that allows a moving oberver to quickly discriminate between real and virtual features without making any assumption regarding the photometric properties of the features.

When the sensor moves around the object, the virtual feature travels on the specular surface producing a family of reflected rays. The envelope defined by this family is called a *caustic*. 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

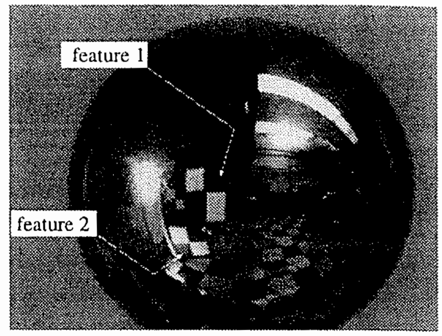


Figure 3: Image of a specular sphere with real and virtual features, both checkerboard corners (see text).

for a virtual feature the caustic will be a curve.[2] Therefore, to classify a feature, all we need is to compute the caustic and to test whether it is a point or a curve.

Pertinent information that the image of a virtual feature contains is the direction, θ_r, of the reflected ray relative to the world coordinate system and its signed distance, ρ_r, from the origin of the coordinate system. These parameters can be computed in a staightforward manner from the position of the feature in the image and the camera parameters (position, orientation and focal length).

The caustic that we seek to compute is tangent to each of the reflected rays. Hence, $\rho_r(\theta_r)$ (more precisely, $\rho_r(\theta_r + \frac{\pi}{2})$) represents the support function of the caustic. Therefore, we will use the terms *caustic* and *image trajectory* of a feature interchangeably since they convey the same information. Note that we have used ρ_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 ρ_n. The computation of the caustic curve (x_c, y_c) (parametrized by θ_r) in Cartesian coordinates is straightforward. Given $\rho_r(\theta_r)$, expression (5) can be used to get:

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} -\sin\theta_r & -\cos\theta_r \\ \cos\theta_r & -\sin\theta_r \end{pmatrix} \begin{pmatrix} \rho_r(\theta_r) \\ \rho_r'(\theta_r) \end{pmatrix} \quad (6)$$

where, ρ_r' is the derivative of ρ_r with respect to θ_r. As stated earlier, the compactness of the above caustic gives us a direct means of classifying real and virtual features. At times, however, virtual features may produce a compact caustic. Specifically, when the radius of curvature of the profile is very small, for example in a sharp corner [Koenderink and van Doorn-1980], the caustic will be compact making it hard in the presence of noise to determine that the feature is a virtual one. 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 effectively used in techniques such as stereo and motion.

[2]This assumption is violated in the case of a parabolic specular surface with a scene feature exactly on its axis. It is easy to convince oneself of this by using the shape recovery equations to be derived in section 2.4. This is clearly a special case that is unlikely from a practical perspective.

2.3 Experiments: Feature Classification

To illustrate classification we used the metallic sphere shown in Figure 3. The real and virtual features shown in the image were tracked while the sensor was moved around in a planar trajectory. The two features are tracked from one frame to the next using the sum of square difference (SSD) correlation operator. The computed support functions $\rho_r(\theta_r)$ of the two features are shown in Figure 4(a). From each support function a caustic curve was computed using expression (6).

In Figure 4(b), the two caustics 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 typical of caustics resulting from specular reflection). Several tests can be employed to determine the compactness of caustics. We found the second moment of the caustic computed with respect to the centroid to be a simple but effective measure. 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.

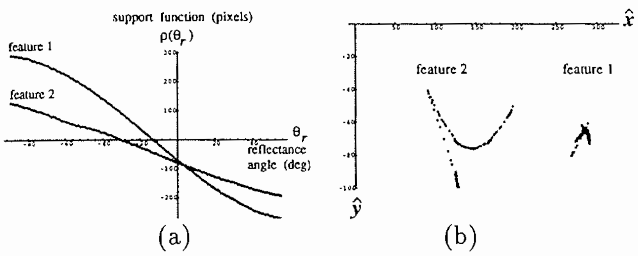


Figure 4: (a) Support functions produced by the two features shown in Figure 3. (b) Caustics computed from the support functions and plotted on the $\hat{x}$-$\hat{y}$ plane. The caustic of the real feature (1) is a compact cluster, while that of the virtual feature (2) is a curve with a cusp.

2.4 2D Profile Recovery

We now examine the problem of recovering 2D specular profiles by moving the sensor and tracking virtual features. In this section, we derive new 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 (or sources), but also to recover the profile of the specular surface without any ambiguity. This is shown to be possible without prior knowledge of any profile points.

2.4.1 The Profile Equation

Figure 5 shows a specular surface 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. 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, θ_i is nearly constant over the entire profile. As in section 2.2, we use the nota-

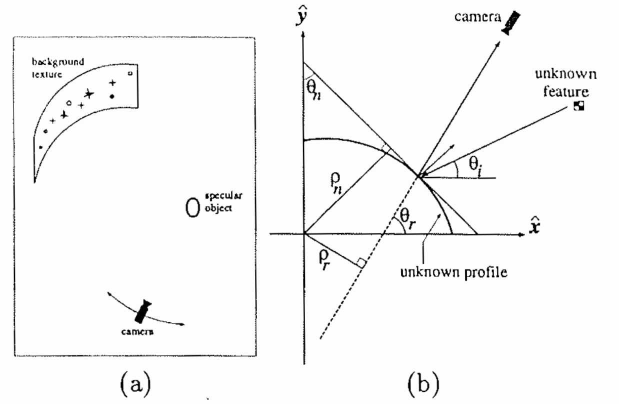


Figure 5: (a) The general setting of a specular object in an environment including a variety of visual features. The observer can view the object from different directions. (b) The geometry of feature reflection from a specular profile.

tion $\rho_n(\theta_n)$ for the profile support function and $\rho_r(\theta_r)$ for the caustic support function. 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 θ_n. From (4) 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 $\theta_r + \frac{\pi}{2}$ (see Figure 5). Again, using (4) we have:

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

From these two equations we get the following expression for the x-coordinate of the profile point:

$$x = \frac{\cos\theta_r\rho_n(\theta_n) - \sin\theta_n\rho_r(\theta_r)}{\cos(\theta_r - \theta_n)} \tag{7}$$

Also, from expression (5) we get:

$$x = \cos\theta_n\rho_n(\theta_n) - \sin\theta_n\rho_n'(\theta_n) \tag{8}$$

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

$$\theta_r = 2\theta_n - \theta_i \tag{9}$$

we get:

$$\rho_r(2\theta_n - \theta_i) = -\sin(\theta_n - \theta_i)\rho_n + \cos(\theta_n - \theta_i)\rho_n' \tag{10}$$

or:

$$\rho_r(2\theta_n - \theta_i) = \frac{d}{d\theta_n}\Big[\cos(\theta_n - \theta_i)\rho_n(\theta_n)\Big] \tag{11}$$

This differential equation is fundamental to our analysis as it relates the support function ρ_r of a feature to the surface profile ρ_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 θ_r and not with θ_n (the unknown normal direction) we substitute equation (9) in (11) and integrate over θ_r:

$$\rho_n(\frac{\theta_r + \theta_i}{2}) = \frac{\frac{1}{2}\int_{\theta_r^0}^{\theta_r} \rho_r(\theta_r')d\theta_r' + \rho_n(\frac{\theta_r^0+\theta_i}{2})\cos(\frac{\theta_r^0-\theta_i}{2})}{\cos(\frac{\theta_r-\theta_i}{2})} \tag{12}$$

where, θ_r^0 is the starting angle of the integration. This last expression 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. The Cartesian coordinates of the surface profile can be computed using equation (5), or alternatively, using equation (27) which was derived in section 3 and does not require the computation of the derivative of ρ_n. An important observation from equation (12) is that even if the feature direction θ_i is given, the surface profile cannot be determined completely due to the unknown constant of integration: $C = \rho_n(\frac{\theta_r^0+\theta_i}{2})\cos(\frac{\theta_r^0-\theta_i}{2})$. This implies that the surface profile is determined only up to a one-parameter family of curves. In the next section, we will show how this ambiguity can be resolved.

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 recovered uniquely. Consider two scene features in the directions θ_i^k $(k = 1, 2)$. We do not assume these angles are known, or that any point on the surface is given a-priori. By moving the camera around the profile in a known trajectory and tracking two features we get two caustics $\rho_r^k(\theta_r)$ $(k = 1, 2)$. Each caustic determines, based on equation (12), a two-parameter family of profile support functions, $\rho_n^k(\theta_n)$ $(k = 1, 2)$. Each support function depends on two parameters, the feature direction and the constant of integration: (θ_i^k, C_k) $(k = 1, 2)$, a total of 4 unknown parameters. A feature in direction θ_i^k will be reflected by points on the profile whose normals lie in the range $(\theta_i^k - 90°, \theta_i^k + 90°)$. Therefore, unless the two features are located exactly at diametrically opposite sides of the object, i.e. 180° apart, their profile trajectories must overlap. In the overlap region, the two recovered support functions $\rho_n^1(\theta_n)$ and $\rho_n^2(\theta_n)$ must agree, as they represent the same profile segment. In the recovery process, we search the $(\theta_i^1, C_1; \theta_i^2, C_2)$-paprameter space and find the 4 parameters that minimize the distance between $\rho_n^1(\theta_n)$ and $\rho_n^2(\theta_n)$ in the overlapping region. Once the 4 parameters are found, 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 to obtain a larger profile.

2.5 Experiments: Recovery of 2D Profile Using 2 Features

We conducted experiments on a variety of objects. Here, we present results on just two profiles, one circular and the other elliptical. The experimental setup used is shown in Figure 5(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. Two features were selected in the initial image (Figure 6(b)) and tracked through the image sequence using the SSD correlation operator. From the feature trajectories, the camera coordinates and focal length, support functions for each of the two features were computed (see Figure 7(a)). Next, the two unknown pairs of parameters (θ_i^1, C_1) and (θ_i^2, C_2) that minimize the distance between the two support function in the overlap region were found via search. We used the mean-squared-distance metric to formulate the search. Once the four parameters are determined, 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 7(b) shows the recovered profile of the sphere. Experimental results for an oval-shaped object are shown in Figure 8. In both experiments we see that the specular profiles are estimated with high accuracy.

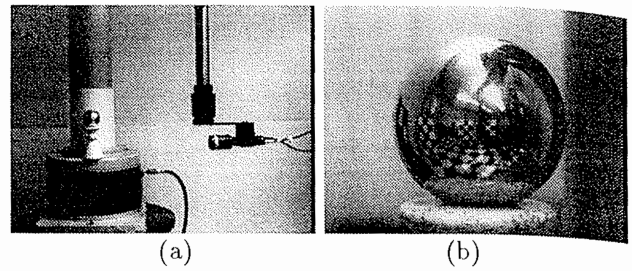


Figure 6: (a) The experimental setup. A camera is mounted on the end-effector of a 5 DOF robot. (b) A typical image including a large number of virtual features.

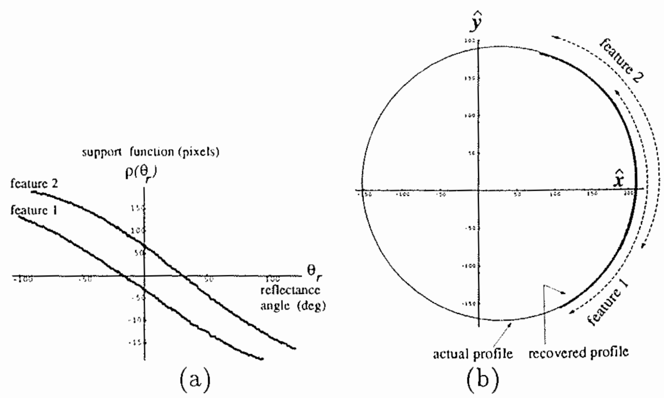


Figure 7: 2D profile of a sphere recovered by tracking two unknown features. (a) Support functions of the two features. (b) The recovered surface profile (thick line) and the actual profile (thin line). The directions of the two unknown features were found to be $\theta_i^1 = -28°$ and $\theta_i^2 = 34°$.

3 Recovery of 3D Surface Curves

We now generalize the results of the previous sections to 3D surfaces. The camera motion is no longer confined to a plane, and consequently, the surface profile can be any smooth space curve. Figure 9 shows the tracking

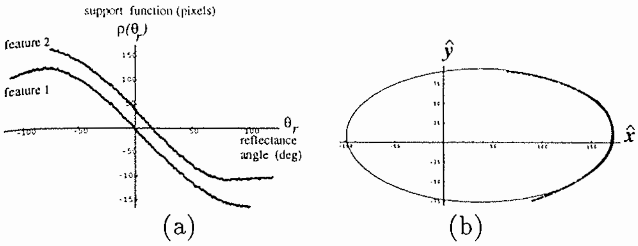


Figure 8: 2D profile of an ellipsoid recovered by tracking two unknown features. (a) Support functions of the two features. (b) The recovered surface profile (thick line) and the actual profile (thin line). The directions of the two unknown features were found to be $\theta_i^1 = -26°$ and $\theta_i^2 = 50°$.

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 generalize our results to the 3D case.

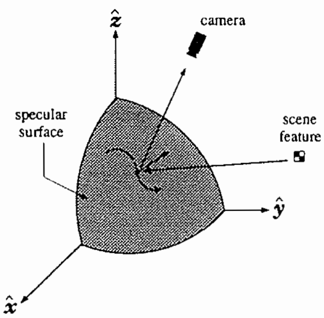


Figure 9: Recovery of surface geometry for general camera motion (along a 3D trajectory) cannot be decomposed into two 2D profile recovery problems.

3.1 3D Caustics and Feature Classification

Analysis of 3D specular surfaces requires us to use vectorial representations of incidence, reflection, and normal directions. For general camera motion, the family of reflected rays, $\hat{\boldsymbol{v}}(t)$, is no longer confined to a plane. Hence, the caustic corresponding to the reflected rays is an arbitrary space curve. We have parameterized the family of reflected rays by t, which may be viewed as a *time* parameter. The caustic curve is then given by $\vec{\boldsymbol{x}}_c(t)$. It is also parametrized by t such that point $\vec{\boldsymbol{x}}_c(t)$ on the caustic is tangent to the ray $\hat{\boldsymbol{v}}(t)$. We use the notation $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{n}}(t)$ for the feature direction vector and the surface normal vector, respectively. All vectors defined thus far, $\hat{\boldsymbol{v}}(t)$, $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{n}}(t)$, represent only directions and are assumed to be unit vectors.

The main idea behind the derivation of the 3D caustic curve is to decompose the caustic point position at any given instant t into two orthogonal components as follows:

$$\vec{\boldsymbol{x}}_c(t) = \vec{\boldsymbol{L}}(t) + \langle \vec{\boldsymbol{x}}_c(t), \hat{\boldsymbol{v}}(t)\rangle \hat{\boldsymbol{v}}(t) \tag{13}$$

The first component, $\vec{\boldsymbol{L}}(t)$, is the distance vector from the ray $\hat{\boldsymbol{v}}(t)$ to the origin $\vec{\boldsymbol{O}}$ of the world coordinate system. This vector is analogous to the support function, ρ_n, introduced in section 2.1.2. The vectors $\hat{\boldsymbol{v}}(t)$ and $\vec{\boldsymbol{L}}(t)$ are computed directly from the image trajectory of a virtual feature and the known camera parameters. Equation (13) expresses the unknown caustic curve $\vec{\boldsymbol{x}}_c(t)$ in a moving coordinate system that is attached to the ray $\hat{\boldsymbol{v}}(t)$. To determine $\vec{\boldsymbol{x}}_c(t)$ we need to find the unknown component $\langle \vec{\boldsymbol{x}}_c(t), \hat{\boldsymbol{v}}(t)\rangle$ along $\hat{\boldsymbol{v}}(t)$.

The first step in the caustic curve derivation is to differentiate (13) with respect to t. This yields the tangent to the caustic curve:

$$\dot{\vec{\boldsymbol{x}}}_c(t) = \dot{\vec{\boldsymbol{L}}}(t) + \langle \vec{\boldsymbol{x}}_c(t), \hat{\boldsymbol{v}}(t)\rangle \dot{\hat{\boldsymbol{v}}}(t) + \left(\langle \dot{\vec{\boldsymbol{x}}}_c(t), \hat{\boldsymbol{v}}(t)\rangle + \langle \vec{\boldsymbol{x}}_c(t), \dot{\hat{\boldsymbol{v}}}(t)\rangle\right)\hat{\boldsymbol{v}}(t) \tag{14}$$

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

$$\langle \dot{\vec{\boldsymbol{x}}}_c(t), \dot{\hat{\boldsymbol{v}}}(t)\rangle = \langle \dot{\vec{\boldsymbol{L}}}(t), \dot{\hat{\boldsymbol{v}}}(t)\rangle + \langle \vec{\boldsymbol{x}}_c(t), \hat{\boldsymbol{v}}(t)\rangle \|\dot{\hat{\boldsymbol{v}}}(t)\|^2 \tag{15}$$

Since $\vec{\boldsymbol{x}}_c(t)$ is the caustic curve, for any given t its tangent $\dot{\vec{\boldsymbol{x}}}_c(t)$ is parallel to $\hat{\boldsymbol{v}}(t)$: $\langle \dot{\vec{\boldsymbol{x}}}_c(t), \dot{\hat{\boldsymbol{v}}}(t)\rangle = 0$. If we plug this constraint in (15), we get an expression for $\langle \vec{\boldsymbol{x}}_c(t), \hat{\boldsymbol{v}}(t)\rangle$ that can be substituted back in (13). The result is the 3D caustic curve equation:

$$\vec{\boldsymbol{x}}_c(t) = \vec{\boldsymbol{L}}(t) - \frac{\langle \dot{\vec{\boldsymbol{L}}}(t), \dot{\hat{\boldsymbol{v}}}(t)\rangle}{\|\dot{\hat{\boldsymbol{v}}}(t)\|^2}\hat{\boldsymbol{v}}(t) \tag{16}$$

Comparing this to (6) we see that the 2D caustic is a special case of the 3D caustic, corresponding to planar motion when the curve parameter is $t = \theta_r$.

3.2 Experiments: 3D Caustic Curves and Feature Classification

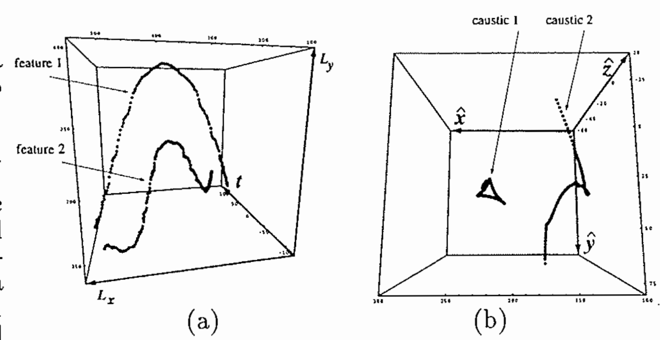


Figure 10: (a) Image trajectories, $\vec{\boldsymbol{L}}(t)$, of two features plotted as parameterized curves. (b) Their computed 3D caustic curves.

We conducted feature classification experiments on the object in Figure 3, which produces both real and virtual features. The camera motion is not planar but rather an arbitrary smooth 3D trajectory. The caustics shown in Figure 10(b) are computed from the image trajectories shown in Figure 10(a) using equation (16). As expected, the caustic curve of the real feature (1) is a small cluster of points centered around the actual feature position while the caustic curve of the virtual feature (2) is a space curve. The ratio of the second moments of the two caustics was found to be 12:1.

3.3 Surface Curve Equation for 3D Camera Motion

As the observer moves around a specular object, a virtual feature travels along a 3D profile on the object's surface. In this section, we show how this surface curve, $\vec{x}(t)$, can be recovered from the image trajectory of the virtual feature given by the pair $(\vec{L}(t), \hat{v}(t))$. As in section 3.1, the key step is to decompose the surface curve into two orthogonal components as:

$$\vec{x}(t) = \vec{L}(t) + \langle \vec{x}(t), \hat{v}(t) \rangle \hat{v}(t) \tag{17}$$

The unknown quantity is the component of $\vec{x}(t)$ along the reflected ray which is given by $\langle \vec{x}(t), \hat{v}(t) \rangle$. To recover this component we examine the support function of the surface curve which is the distance of the *tangent plane* from the world origin $\vec{O}$:

$$\rho_n(t) = \langle \vec{x}(t), \hat{n}(t) \rangle \tag{18}$$

where, $\hat{n}(t)$ is the surface normal at $\vec{x}(t)$. Differentiating the above expression with respect to t and using the fact that the vector $\dot{\vec{x}}$ is tangent to the surface gives:

$$\dot{\rho_n}(t) = \langle \vec{x}, \dot{\hat{n}} \rangle \tag{19}$$

Now, the law of specular reflection can be written in vectorial form as:

$$\hat{v} = 2\langle \hat{s}, \hat{n} \rangle \hat{n} - \hat{s} \tag{20}$$

Using this expression for $\hat{v}(t)$ in (17) and substituting $\vec{x}(t)$ back in (18) and (19) 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:

$$\begin{aligned} &\rho_n(t)\langle \hat{s}, \dot{\hat{n}}(t) \rangle + \dot{\rho_n}(t)\langle \hat{s}, \hat{n}(t) \rangle = \\ &\langle \vec{L}(t), \hat{n}(t) \rangle \langle \hat{s}, \dot{\hat{n}}(t) \rangle + \langle \vec{L}(t), \dot{\hat{n}}(t) \rangle \langle \hat{s}, \hat{n}(t) \rangle \end{aligned} \tag{21}$$

This result can be further simplified. To this end, we find the derivative of $\hat{v}$ with respect to t using (20). Since the feature direction $\hat{s}$ is a constant we get:

$$\frac{1}{2}\dot{\hat{v}}(t) = \langle \hat{s}, \dot{\hat{n}}(t) \rangle \hat{n}(t) + \langle \hat{s}, \hat{n}(t) \rangle \dot{\hat{n}}(t) \tag{22}$$

Note that the right hand side of the above expression figures explicitly in (21). Substituting the left hand side instead gives the following fundamental relationship:

$$\frac{d}{dt}\Big(\rho_n(t)\langle \hat{s}, \hat{n}(t) \rangle\Big) = \frac{1}{2}\langle \vec{L}(t), \dot{\hat{v}}(t) \rangle \tag{23}$$

Then, by integration we have:

$$\rho_n(t) = \frac{\frac{1}{2}\int_{t_0}^{t} \langle \vec{L}(t'), \dot{\hat{v}}(t') \rangle dt' - \rho_n(t_0)\langle \hat{s}, \hat{n}(t_0) \rangle}{\langle \hat{s}, \hat{n}(t) \rangle} \tag{24}$$

Note that even for a known feature direction the above solution is determined only up to an unknown parameter, namely, the constant of integration: $C = \rho_n(t_0)\langle \hat{s}, \hat{n}(t_0) \rangle$. 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 (24) 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 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 $\langle \vec{x}(t), \hat{v}(t) \rangle$. The inner product of (17) with the normal vector $\hat{n}(t)$ can be written as:

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

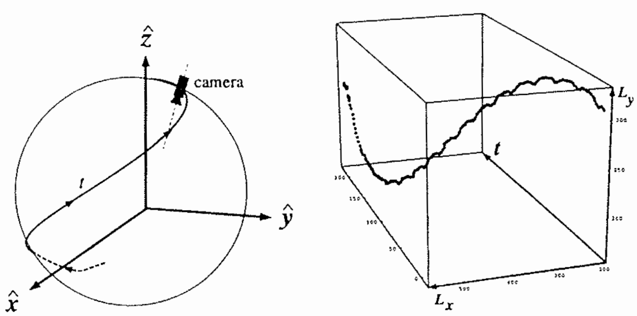
or:

$$\langle \vec{x}(t), \hat{v}(t) \rangle = \frac{\rho_n(t) - \langle \vec{L}(t), \hat{n}(t) \rangle}{\langle \hat{v}(t), \hat{n}(t) \rangle} \tag{26}$$

Substituting this back in (17) gives us the surface curve:

$$\vec{x}(t) = \vec{L}(t) + \Big(\frac{\rho_n(t) - \langle \vec{L}(t), \hat{n}(t) \rangle}{\langle \hat{v}(t), \hat{n}(t) \rangle}\Big)\hat{v}(t) \tag{27}$$

3.5 Experiments: Recovery of 3D Curves



(a) (b)

Figure 11: (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 other two coordinates represent the distance vector $\vec{L}(t)$.

Our last set of experiments are on recovery of 3D curves of specular objects. The recovery of a 3D profile is a much harder problem than that of 2D profiles since the surface trajectories traveled by different features are not guaranteed to overlap over large surface curve segments. This forces us to use one feature at a time but resolve shape ambiguity by tracking each feature from the occluding boundary of the surface as explained in the section 3.3.

In these experiments we tracked the reflection of a point light source (highlight) rather than the reflection of a scene feature. The two however are equivalent from a theoretical perspective. The camera direction trajectory used is plotted in Figure 11. The surface curve was recovered from the image trajectory of the virtual feature using equation (24). The recovered (dots) and actual (solid) surface profiles are displayed in Figure 12 using two different viewpoints. We see that the recovered curve is in strong agreement with the actual surface curve.

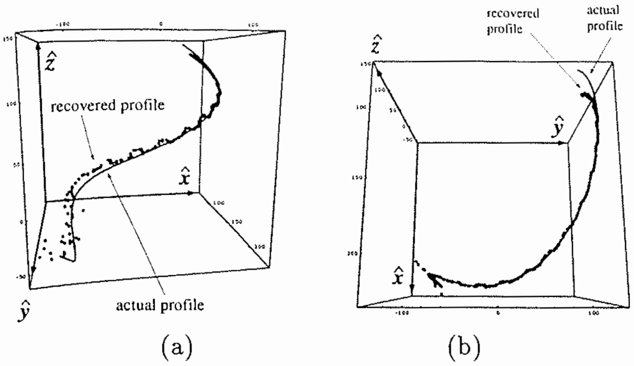


Figure 12: 3D surface curve on a chrome-plated ball recovered and displayed from two different viewpoints. The recovered surface profile (dots) is seen to closely match the actual profile (solid line).

4 Summary

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

In this work, we have introduced a comprehensive 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 representations for surface curves and reflected rays are carefully chosen. We invoked support functions and the notion of caustic curves to represent trajectories produced by image features. Caustics were shown to hold valuable information regarding scene geometry.

Image features were categorized into two basic classes: real and virtual. Real features are the only sort that should 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 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 a variety of vision techniques.

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 feasible and can be performed with reasonable accuracy.

References

[Bellver-Cebreros and Rodriguez-Danta, 1992] C. Bellver-Cebreros and M. Rodriguez-Danta. Caustics and the legendre transform. *Optics Communications*, 92(4-6):187–192, 1992.

[Blake and Brelstaff, 1988] A. Blake and G. Brelstaff. Geometry from specularity. *Proceedings of The Second ICCV*, pages 394–403, 1988.

[Blake and Bulthoff, 1991] A. Blake and H. Bulthoff. Shape from specularity: Computation and psychophysics. *Phil. Trans. Royal Society London B.*, 331:237–252, 1991.

[Blake, 1985] A. Blake. Specular stereo. *Proc. 9th IJCAI Conf.*, pages 973–976, 1985.

[Brelstaff and Blake, 1988] G. Brelstaff and A. Blake. Detecting specular reflections using lambertian constraints. *Proceedings of The Second ICCV*, pages 297–302, 1988.

[Cornbleet, 1984] S. Cornbleet. *Microwave and Optical Ray Geometry*. John Wiley and Sons, 1984.

[Guggenheimer, 1977] H. W. Guggenheimer. *Differential Geometry*. Dover Publications, 1977.

[Healey and Binford, 1987] G. Healey and T. O. Binford. Local shape from specularity. *Proc. of ICCV*, pages 151–160, 1987.

[Ikeuchi, 1981] K. Ikeuchi. Determining surface orientation of specular surfaces by using the photometric stereo method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):661–669, 1981.

[Koenderink and van Doorn, 1980] J. J. Koenderink and A. J. van Doorn. Photometric invariants related to solid shapes. *Optica Acta*, 27(7):981–996, 1980.

[Lee, 1991] S. W. Lee. *Understanding of Surface Reflections in Computer Vision by Color and Multiple Views*. PhD thesis, University of Pennsylvania, 1991.

[Longuet-Higgens, 1960] M.S. Longuet-Higgens. Reflection and refraction at a random moving surface: (i) pattern and paths of specular points. *Journal of the Optical Society of America*, 50(9):838–844, 1960.

[Nayar *et al.*, 1990] S. K. Nayar, K. Ikeuchi and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Transactions on Robotics and Automation*, 6(4):418–431, August 1990.

[Sanderson *et al.*, 1988] A. C. Sanderson, L. E. Weiss and S. K. Nayar. Structured highlight inspection of specular surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(1):44–55, 1988.

[Schultz, 1994] H. Schultz. Retrieving shape information from multiple images of a specular surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):195–201, February 1994.

[Stavroudis, 1972] O. N. Stavroudis. *The Optics of Rays, Wavefronts, and Caustics*. Academic Press, 1972.

[Symosek, 1985] P. F. Symosek. *Parameter Estimation and Classification of Machine Parts Based on Specular or Mirror-Like Image Data*. PhD thesis, Brown University, May 1985.

[Ullman, 1976] S. Ullman. On visual detection of light sources. *Biological Cybernetics*, 21:205–211, 1976.

[Waldon and Dyer, 1993] S. Waldon and C. R. Dyer. Dynamic shading, motion parallax and qualitative shape. *Proceedings of the IEEE Workshop on Qualitative Vision*, pages 61–70, June 1993.

[Zisserman *et al.*, 1989] A. Zisserman, P. Giblin and A. Blake. The information available to a moving observer from specularities. *Image and Vision Computing*, 7:287–291, 1989.