Model Based Recognition Of 3-D Surfaces
Using
Curvature Parameterization

by

S. J. Garnier
G. L. Bilbro
W. E. Snyder

Center for Communications and Signal Processing
Department of Electrical and Computer Engineering
North Carolina State University

CCSP-TR-89/6

March 15, 1989
Abstract

A new approach to rigid object recognition in range imagery is presented. This approach is based on a special re-parameterization of a smooth surface that is independent of rotation, translation, or surface parameterization. An observed surface point always transforms to identical coordinates in the special parameter system. This property allows a correspondence between points on two surfaces and permits comparison of the surfaces by correlation techniques.

*Keywords:* surface recognition, curvature, 3-D surfaces, range images, altitude images, viewpoint-invariance.
1 Introduction

This paper investigates a recently developed [6] approach to rigid object recognition in range imagery, and is based on a special re-parameterization of a smooth surface that does not depend on the conditions of rotation, translation, or parameterization of the surface.

The special re-parameterization utilizes a differential geometric formalism, and is based on determining the principal curvatures of the points of interest, and then using the principal curvatures as axes of the parameterized coordinate space.

For a continuous surface, an arbitrary point on the surface will always transform to numerically identical coordinates in the special parameter system, independent of observation pose. This amounts to a continuous correspondence between points on two surfaces and permits the matching of surfaces by correlation techniques.

In section 2, a general derivation for the parameterization is given, utilizing a continuous formulation. In section 3, the concept is formulated in a discrete manner amenable to implementation on a digital computer. This formulation is stated as an algorithm in section 4. The technique is tested on range images in section 5, and shown to be an effective visible-invariant means for recognizing surfaces.
2 Problem Formulation

We consider a surface in 3-space as a mapping, $X$, from a set of points in a parameterizing plane $U \subseteq \mathbb{R}^2$, to a set of points in 3-space

$$X : U \to \mathbb{R}^3.$$  \hfill (1)

In this paper, we assume $X$ is a three times differentiable manifold. That is, $X$ is specified by three parametric equations

$$\mathbf{x}(u) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix} \quad \text{where} \quad u = (u, v).$$  \hfill (2)

The actual surface $X(U)$ is invariant to observation viewpoint, but the details of the parameterization are not unique. The same surface may be represented by $M(V)$ where $M : V \to \mathbb{R}^3$, and $M$ is some triple of functions different from those of $X$ defined on some different domain set $V$. Or, the surface $M(V)$ may be a rotated or translated duplicate of $X(U)$. Or it may be another surface entirely. If we denote the image or observed surface as $X(U)$ and the model surface as $M(V)$, the question of interest is whether $X(U)$ is congruent to $M(V)$. That is to say, does there exist a rigid body motion, $R$, such that $X(U) = RM(V)$?

We will employ the surface curvature to answer this question. We are particularly interested in curvature since it is viewpoint-invariant. That is, the curvature of a surface at a point is a property of the surface, and not a property of how the surface is observed. We will make extensive use of this property.

Given some parameterization of the surface as determined, for example, by a
range camera [2], we determine the principal curvatures through the use of the first and second fundamental forms [14, pp.107-112] [19, pp.67-93] of differential geometry. The first fundamental form is the determinant of $G$ where

$$G_{uv} = \begin{bmatrix}
\frac{\partial r}{\partial u} \cdot \frac{\partial r}{\partial u} & \frac{\partial r}{\partial u} \cdot \frac{\partial r}{\partial v} \\
\frac{\partial r}{\partial v} \cdot \frac{\partial r}{\partial u} & \frac{\partial r}{\partial v} \cdot \frac{\partial r}{\partial v}
\end{bmatrix}$$

(3)

and where the parametric surface $r = r(u,v) = [f(u,v), g(u,v), h(u,v)]^T$. $G$ (also known as "the metric of the surface") may also be written in terms of the $[x, y, z]^T$ Cartesian coordinate vector where $z = z(x,y)$,

$$G_{xy} = \begin{bmatrix}
(\frac{\partial z}{\partial x})^2 + 1 & \frac{\partial z}{\partial x} \frac{\partial z}{\partial y} \\
\frac{\partial z}{\partial x} \frac{\partial z}{\partial y} & (\frac{\partial z}{\partial y})^2 + 1
\end{bmatrix}.$$

(4)

The second fundamental form is the determinant of $D$ where

$$D_{uv} = \begin{bmatrix}
n \cdot \frac{\partial^2 r}{\partial u^2} & n \cdot \frac{\partial^2 r}{\partial u \partial v} \\
n \cdot \frac{\partial^2 r}{\partial v \partial u} & n \cdot \frac{\partial^2 r}{\partial v^2}
\end{bmatrix},$$

(5)

and where $n$ is the surface normal at the point of interest. Again we may write $D$ in terms of the $[x, y, z]^T$ Cartesian coordinate vector,

$$D_{xy} = \frac{1}{S_{xy}} \begin{bmatrix}
\frac{\partial^2 z}{\partial x^2} & \frac{\partial^2 z}{\partial x \partial y} \\
\frac{\partial^2 z}{\partial x \partial y} & \frac{\partial^2 z}{\partial y^2}
\end{bmatrix}$$

(6)

where $S$, the square root of the determinant of the first fundamental matrix, is the differential surface area [25, pp.437-441] at the point of interest:

$$S_{xy} = \left( (\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 + 1 \right)^{\frac{1}{2}}.$$

(7)
We determine the principal curvatures, \( k_1 \) and \( k_2 \), by solving the quadratic equation [14, pp.107-112]

\[
\det (D - kG) = 0. \tag{8}
\]

The solutions, \( k = k_1 \) and \( k = k_2 \) are the minimum and maximum normal curvatures at the point of interest. For convenience, we introduce the notation \( k(x,y,z) \) to represent the process of solving Equation 8 at a particular \((x,y,z)\) on the surface. We refer to the mapping from the measurement parameterization to principal curvatures as the \textit{curvature map}

\[
K_I : U \rightarrow \mathbb{R}^2. \tag{9}
\]

We will further restrict our attention to those surfaces (or segments of surfaces) which satisfy an additional property:

Consider a domain \( U \) of the imaged surface, on which the curvature map has a nonsingular Jacobian everywhere

\[
\det J = \det \left[ \frac{\partial(k_1, k_2)}{\partial(u,v)} \right] = \det \begin{bmatrix} \frac{\partial k_1}{\partial u} & \frac{\partial k_1}{\partial v} \\ \frac{\partial k_2}{\partial u} & \frac{\partial k_2}{\partial v} \end{bmatrix} \neq 0. \tag{10}
\]

We shall call any region on which \( J \) is never singular, and where the region is the image of a rectangle by a map which is one-to-one, a \textit{regular segment}. On a regular segment, \( K_I \) has a local inverse everywhere. [28]

**Postulate:** On a regular segment, the curvature map has a global inverse.

\(^1\)Later, we will distinguish between the curvature map of an image, \( K_I \), and the map of the corresponding model, \( K_M \). We use the subscript here in anticipation.
We have shown [6] this postulate to be true on the additional condition that the regular segment is part of a quadric surface. We believe that the quadric condition is unnecessary but we have not been able to prove our conjecture. The conjecture simplifies the theory and we will assume it to be true. Even if it is not true, the theory could be modified by using the notion of multiple valued functions borrowed [22] from the study of complex variables.

If $K_I$ is invertible, then the composite function $X \circ K_I^{-1} : K_I(U) \rightarrow X(U)$ is a single-valued visible-invariant parameterization of the segment, that is, for each pair of principal curvatures, a unique point on the surface is determined. A similar map can be constructed on any model segment $M \circ K_M^{-1} : K_M(V) \rightarrow M(V)$. Thus, since this type of composite map takes a pair of principal curvatures to a Cartesian triple on the surface, the entire surface is exactly represented and can be accessed by curvature pairs. As an immediate result of the curvature mapping, we have:

**Corollary:** If regular surfaces $\Phi_1$ and $\Phi_2$ have identical curvature maps, then $\Phi_1$ is a similarity transform of $\Phi_2$ and vice-versa.

Thus, recognition of a regular surface becomes a simple comparison of curvature maps. We are not limited to regular surfaces, however, as we will see in section 3, by taking account of the area subtended by a particular curvature map, we can distinguish even umbilic surfaces such as spheres.

A small area on the imaged surface, $dA$, is related to a small area on the
curvature plane by
\[ dA = \rho_I(k) dk^2. \]  
(11)
\( \rho_I(k) \) is determined from \( G_I \), the first fundamental form of the imaged surface by
\[ \rho_I(k) = \sqrt{\text{det} G_I} \]  
(12)
and the Jacobian is as defined in Equation 10. The metric \( G_I \) is positive definite \cite{28} in this parameterization just as is the metric in the original parameterization, \( G_{Io} \), from which it is derived. \cite{28} Here \( G_{Io} \) is simply the metric in the original (camera frame) parameterization, put forth by Equation 4.

**Definition (1):** An imaged segment is said to be *congruent* with a model segment if and only if the curvature image of the former is contained within the curvature image of the latter, that is,

1. \( K_I(U) \subseteq K_M(V) \),

2. \( \forall k \in K_I(U), \rho_I(k) = \rho_M(k) \). From 2) it follows that,

3. \( \int \int_{K_I} \rho_I(k) dk = \int \int_{K_M} \rho_M(k) dk \).

We refer to condition 1) above as the *domain test*. Simply put, no point on the imaged surface may have a principal curvature pair which is not in the model surface. Conditions 2) and 3) comprise the *range test*, the area densities of the
two segments must be identical whenever the imaged surface is observed. Note that since this comparison is only over the visible portions of the imaged surface, occlusions are automatically handled. Similarly, condition 3) requires that the total area subtended by the two segments be the same, again, considering only visible area. To see that condition 3) represents the area, consider the different parameterizations based upon, say, \((u, v)\) and \((x, y)\) of the same surface, \(\Phi\), and let \(\Phi_{uv}\) and \(\Phi_{zy}\) be the domain of the surface in the \((u, v)\) and \((x, y)\) parameterizations, respectively.

\[
S = \int \int_{(u, v) \in \Phi_{uv}} (\det G_{uv})^{\frac{1}{2}} dudv = \int \int_{(x, y) \in \Phi_{zy}} (\det G_{zy})^{\frac{1}{2}} dx dy = \quad (14)
\]

\[
= \int \int_{k \in K_I} (\det G_{zy})^{\frac{1}{2}} \frac{\partial (x, y)}{\partial (k_1, k_2)} dk = \quad (15)
\]

\[
= \int \int_{k \in K_I} (\det G_{zy})^{\frac{1}{2}} (\det J)^{-1} dk = \int \int_{k \in K_I} \rho(k) dk . \quad (16)
\]

Thus, if the imaged surface consists of exactly one regular segment, the curvature parameterization provides a transformation which will exactly match (in the absence of noise) the transformed model segment.

Note that in this work, we use a mapping of the entire surface into curvature space. Such work may be compared with that of Ponce and Brady [23] and Faugeras [12] who both noted the invariant nature of curvature and used the zero crossings as distinctive features.
3 Implementation

3.1 Effects of Sampling

In order to apply the strategy of Section 2 to realistic problems, we must first consider the discrete nature of the data. Let \( M_S \) be the finite set of points resulting from a sampling of the model surface \( M \). Let \( I = RM \) represent the (unknown) rigid body motion applied to \( M \) to result in the observation \( I \), and similarly, let \( I_S \) be the sampled version of \( I \). Due to the sampling it is a virtual certainty that there do not exist any two points \( x_I \in I_S \) and \( x_M \in M_S \) which satisfy exactly \( x_I = Rx_M \). Due to the sampling of the segments, we simply are not looking at the same point on both the imaged and modeled surfaces.

Sampling causes no significant problems with the domain test. The sampled representation is in error by, at most, the displacement of one pixel, and curvature changes very little between adjacent pixels (except near a singularity, and we avoid such points). The range test is violated by sampled data, but for the most part, such violations nearly cancel out. When the violations do not cancel out, they are dealt with robustly by the recognition algorithm discussed in Section 4.

We define the curvature histogram of the image as a doubly-indexed array

\[
S_I = S_I(k_1, k_2) \text{ where } k_1, k_2 = 0, \pm \Delta k, \pm 2\Delta k, \ldots, \pm n\Delta k. \quad (17)
\]

(We use notation analogous to that of Section 2, to represent analogous concepts, however since in this section we consider only sampled data, there should be no
Model Based Recognition Of 3-D Surfaces Using Curvature Parameterization

We calculate $S_I$ by

$$S_I(k_1, k_2) = \sum_{x,y \in I} \rho_I(k(x,y,z)) \text{ where } z = z(x,y).$$

(18)

That is, over the surface, we sum the area subtended by all pixels in the range image whose principal curvatures fall within the span defined by $k_1 \pm \frac{1}{2} \Delta k, k_2 \pm \frac{1}{2} \Delta k$.

**Theorem (1):** The volume of the curvature histogram is equal to the surface area of the imaged segment.

This follows directly from Equations 14, 15 and 16. It is only approximately true for sampled data, however it illustrates the strength of the method. Since the curvature histogram is visible-invariant, and the computation of its volume is an integral process, we have a method which is relatively robust against noise.

**Definition (2):** Let $x = (x_1, x_2)$ and $y = (y_1, y_2)$ be two vectors in the plane $\mathbb{R}^2$. A vector $z = (z_1, z_2)$ is said to be *between* $x$ and $y$ if and only if $x_1 < z_1 < y_1$ and $x_2 < z_2 < y_2$.

Suppose two points, $a = (x_a, y_a, z_a)$ and $b = (x_b, y_b, z_b)$ lie on the same regular surface. Let $k_a$ and $k_b$ be the curvature mappings of $a$ and $b$. Now, suppose $k_c$ is between $k_a$ and $k_b$. We ask the question, can $S(k_c) = 0$? That is, can there be curvatures in this range which correspond to no points on the surface? Since the curvature map is continuous over a regular surface, this is clearly impossible, if we
think of the problem as a continuous mapping. However, since both the surface
and the curvature space are sampled, we should more properly ask, can holes exist
in the curvature histogram?

The answer is certainly yes. We could force this to happen in a trivial way by
simply making $\Delta k$ very small, thus producing a curvature map of fine granularity.
Since each sampled point on the surface contributes to exactly one bin in $S_I$, by
sampling $S_I$ more finely, we guarantee some empty cells. Clearly, the details of
the conditions under which these holes appear depend upon the sampling rate
used in Cartesian space, the sampling rate in curvature space, and the rates at
which the surface bends. Certainly, if the surface can undergo an abrupt change
in curvature between two pixels, then with respect to a discrete curvature space
mapping, there will exist at least one curvature histogram cell between the two
pixels curvature histogram cells which receives no contribution. We quantify this
notion with a one-dimensional argument. Let the image consist of one cycle of
the function $y = \sin(x)$, and let this function be uniformly sampled by $N$ samples
spaced $\delta$ units apart, thus $\delta = \frac{2\pi}{N}$. We will now derive a relationship between $N$, $\delta$, and allowable values of $\Delta k$. Given the vector $\mathbf{r} = [x, y, z]^T$ where

$$\mathbf{r} = [x, \sin(x), 0]^T, \quad (19)$$

$$\dot{\mathbf{r}} = [1, \cos(x), 0]^T, \quad (20)$$

$$\ddot{\mathbf{r}} = [0, - \sin(x), 0]^T, \quad (21)$$
the curvature, $\kappa$, of $\mathbf{r}$ is [28]

$$\kappa(x) = \frac{\left| \dot{\mathbf{r}} \times \ddot{\mathbf{r}} \right|}{(\dot{\mathbf{r}} \cdot \ddot{\mathbf{r}})^{\frac{3}{2}}} = \frac{\sin(x)}{(1 + \cos^2(x))^{\frac{3}{2}}} \cdot \tag{22}$$

The rate of change of curvature with respect to $x$ is then

$$\dot{\kappa}(x) = \frac{2 \cos(x)[1 + \sin^2(x)]}{(1 + \cos^2(x))^{\frac{3}{2}}} \tag{23}$$

and the curvature extrema occur when

$$\dot{\kappa}(x) = \frac{2\sin(x)\left\{2\cos(x)\left[4 + \sin^2(x)\right] - \sin^2(x) - 1\right\}}{(1 + \cos^2(x))^{\frac{3}{2}}} = 0. \tag{24}$$

By solving Equation 24, we determine the location of the maximum rate of change of curvature to occur at $\theta_{k_{\text{max}}} = 64.285$ degrees. If dropouts are to occur, they will occur in those areas of the function where $\kappa$ changes most rapidly. One such point is $\theta_{k_{\text{max}}}$. Moving one pixel away, we have

$$\Delta \kappa = |\kappa(\theta_{k_{\text{max}}} + \delta) - \kappa(\theta_{k_{\text{max}}})| \tag{25}$$

$$\approx \left| \sin(\theta_{k_{\text{max}}} + \delta) - \sin(\theta_{k_{\text{max}}}) \right| \frac{1}{(1 + \cos^2(\theta_{k_{\text{max}}}))^{\frac{3}{2}}} \tag{26}$$

$$= \left| \sin(\theta_{k_{\text{max}}})\cos(\delta) + \sin(\delta)\cos(\theta_{k_{\text{max}}}) - \sin(\theta_{k_{\text{max}}}) \right| \approx \left| \frac{\delta \cos(\theta_{k_{\text{max}}})}{(1 + \cos^2(\theta_{k_{\text{max}}}))^{\frac{3}{2}}} \right| = \frac{2.1}{N}. \tag{27}$$

Thus, if we have no variations in our image which change faster that a sine of $N$ pixels duration, we may quantize our curvature array no finer than $\frac{2.1}{N}$ units per partition. If curvature may change more rapidly (as is usually the case), coarser quantization may be required to avoid dropouts.
4 Algorithm

In this section, we discuss the development of the model based curvature space histogram library, and the recognition algorithm which uses this library.

4.1 Histogram Library Generation

1. For all model range images, $Z_m(x, y)$ where $m = 1, \ldots, M$, compute $k_{1m}(x, y)$, $k_{2m}(x, y)$, and $\rho_m(x, y)$.

2. For model range image $Z_m$, determine $\max k_{1m} = \max_{x,y \in Z_m} |k_{1m}(x, y)|$.

   Similarly determine $\max k_{2m}$.

3. Find the global maximum curvature $\max k = \max_{m=1,\ldots,M} (\max k_{1m}, \max k_{2m})$

4. Using $\max k$, linearly partition the principal curvature space into $N \times N$ partitions, which span a range of $(2n + 1)\Delta k$, from $-\max k$ to $+\max k$. (Note the redefinition of $N$, as distinguished from that used in Section 3.) We specified $N$ as a positive odd integer, $N = 2n + 1$, in order to force all planar surfaces to map exactly onto an origin-centered partition, all spherical surfaces to map exactly onto partitions on the $k_1 = k_2$ line, and all minimal surfaces to map exactly onto partitions on the $k_1 = -k_2$ line.

5. Given a pixel at $x = [x, y, z]^T$ in the model range image, compute $k_1$ and $k_2$, and update the curvature histogram for that model using Equation 18.
In solving Equation 8 for $k_1$ and $k_2$, we arbitrarily choose the sign of the square root so that $k_1 \geq k_2$, and choose $k_1$ and $k_2$ to be the vertical and horizontal axes, respectively. This assignment maps all surfaces onto or above the 45 degree line. If this line, originally defined by $k_1 = k_2$ (the line of umbilic surfaces), is rotated into the horizontal axis, then the line $k_1 = -k_2$ (the line of minimal surfaces) is also rotated into the vertical axis. The new horizontal sum axis is proportional to mean curvature, whereas the new vertical axis represents a difference curvature. This new representation requires roughly half the storage of the original $k_1, k_2$ representation, since all points below the 45 degree line are of zero value. A range image view of the exterior of an ellipsoid transforms to the second quadrant of this new sum-difference curvature plane, and an interior view of the same object would transform to the first quadrant, as a mirror reflection about the vertical axis.

4.2 Recognition

Surfaces are recognized using a minimum error classifier, which is based on maximum likelihood probability estimation. The maximum likelihood estimator we utilize compares the curvature-space histograms of the observation with the histograms of the models. Therefore, this comparison provides a visible-invariant method for recognition.

We wish to find the model $Z_m$ which maximizes the conditional probability $P(Z_m|I)$, given the measurement, $I$. Since the measurement may be corrupted,
we define two sources of such corruption:

- accretion error

If, at \((k_1, k_2)\), we find \(S_I(k_1, k_2) > S_{Z_m}(k_1, k_2)\), then somehow, for a particular curvature value, more area has occurred in the image than in the model. The larger the accreted area, the lower the probability that the observed surface belongs to the model.

- occlusion error

If, for some point in curvature space \((k_1, k_2)\), we find \(S_I(k_1, k_2) < S_{Z_m}(k_1, k_2)\), this implies that some part of the observed surface is occluded, either by the surface itself, or by some other surface. The larger the occluded surface, the lower the probability that the observed surface belongs to the model.

### 4.2.1 Likelihood of Accretion

Accretion error occurs when a bin in the image curvature histogram violates either the range test or the domain test; that is, the bin’s surface area is greater than the model’s surface area. We characterize this error in the following manner:

Using Bayes rule, given an image \(I\), the probability of accretion of a particular model \(m\) is

\[
P_A(Z_m|I) = \frac{P_A(I|Z_m)P_A(Z_m)}{\sum_m P_A(I|Z_m)P_A(Z_m)} \propto P_A(I|Z_m)P_A(Z_m). \tag{29}
\]
Model Based Recognition Of 3-D Surfaces Using Curvature Parameterization

We ignore the constant denominator and define the likelihood of accretion of a particular model to be

$$L_A(Z_m|I) \equiv P_A(I|Z_m)P_A(Z_m). \quad (30)$$

Considering only the conditional probability, and assuming independence,

$$P_A(I|Z_m) = P_A(\Phi_1|Z_m)P_A(\Phi_2|Z_m) \cdots = \prod_{\Phi_i \in \Phi} P_A(\Phi_i|Z_m)W_i \quad (31)$$

where $\Phi$ is the set of curvature bins which violate the range or domain tests. From this we define the probability of accretion within one bin,

$$P_A(\Phi_i|Z_m) = \frac{1}{N_{S_m}} \sum_{k_j \in S_{Z_m}} P_A(\Phi_i|k_j \in S_{Z_m}), \quad (32)$$

where $N_{S_m}$ is the number of non-zero curvature bins in $S_{Z_m}$. Finally, we choose a form for $P_A(\Phi_i|k_j \in S_{Z_m})$. We choose a zero mean, unit variance model,

$$P_A(\Phi_i|k_j \in S_{Z_m}) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left\{-\frac{1}{2} \left(\frac{d_{ij}}{\sigma}\right)^2\right\}, \quad (33)$$

where $d_{ij}$ is the Euclidean distance from bin $j$ to bin $i$ in the curvature histogram.

In Equation 31, $W_i$ is a weight which reflects our intuitive understanding that the magnitude of the range test violation should affect the probability of occurrence:

$$W_i = \exp\{-|S_I(k_i) - S_{Z_m}(k_i)|\}. \quad (34)$$

The probability of accretion within one bin should reflect the relative size of each donor bin. All other parameters (such as distance) held equal, a larger model bin
has a greater probability of contributing accretion area to an image bin than does a smaller model bin. We modify Equation 32 in order to weight the probability of occurrence with respect to the relative sizes of each donor bin:

$$P_A(\Phi_i|Z_m) = \frac{1}{N_{s_m}[\sum_{k_j \in S_{Z_m}(k_j)} S_{Z_m}(k_j)]} \sum_{k_j \in S_{Z_m}} S_{Z_m}(k_j) \cdot P_A(\Phi_i|k_j \in S_{Z_m}). \quad (35)$$

### 4.2.2 Likelihood of Occlusion

The probability of occlusion of a particular model is determined using Bayes rule, and is similar to that expressed in Equation 29:

$$P_O(Z_m|I) = \frac{P_O(I|Z_m)P_O(Z_m)}{\sum_m P_O(I|Z_m)P_O(Z_m)} \propto P_O(I|Z_m)P_O(Z_m). \quad (36)$$

Again ignoring the constant denominator, we define the likelihood of occlusion of a particular model to be

$$L_O(Z_m|I) = P_O(I|Z_m)P_O(Z_m). \quad (37)$$

Assuming independence of the individual occluded curvature bins, we obtain

$$P_O(I|Z_m) = \prod_{\Theta_i \in \Theta} P_O(\Theta_i|Z_m) \quad (38)$$

where $\Theta$ is the set of curvature bins which exhibit occlusion. We define the probability of occlusion within one bin,

$$P_O(\Theta_i|Z_m) \equiv \alpha \exp\left\{-[S_{Z_m}(k_i) - S_I(k_i)]^2\right\} \quad (39)$$

where $\alpha$ is an appropriate normalization. Here we assume that for each curvature cell, $k_i \in \{S_I\}$ where $\Theta_i \in \Theta$, the probability that it resulted from an occlusion
error applied to the corresponding pixels in the model is Gaussian. Since \( S_{Z_m}(k_i) - S_I(k_i) \) is strictly positive for all \( \Theta_i \in \Theta \) (by definition of occlusion error), and we will momentarily take the logarithm, we omit the square and define a non-normalized likelihood:

\[
L_0(\Theta_i | Z_m) \equiv \exp\{-[S_{Z_m}(k_i) - S_I(k_i)]\}.
\] (40)

Hence, by reflecting this likelihood into Equation 38, we have

\[
L_0(I | Z_m) = \prod_{\Theta_i \in \Theta} \exp\{-[S_{Z_m}(k_i) - S_I(k_i)]\}.
\] (41)

### 4.2.3 Minimum Error Classification

Finally, we define total likelihood as the product of the independent probabilities and likelihoods due to occlusion and accretion:

\[
L(Z_m | I) = L_0(Z_m | I) L_A(Z_m | I).
\] (42)

We derive a minimum error classifier from the maximum likelihood estimator by taking the negative of the natural logarithm of Equation 42. We obtain the total error:

\[
Error(m) = Error_0(m) + Error_A(m).
\] (43)

The model with the minimum error is selected as the most likely match for the image.

We now develop the accretion error by taking the negative logarithm of the accretion likelihood. Our notation for the number of image curvature bins that
exhibit accretion with respect to each model is $N_{\Phi_m}$.

$$Error_A(m) = -\log \{ P_A(Z_m) \} + N_{\Phi_m} \log \{ N_{S_m} \sigma \sqrt{2\pi} \sum_{k_j \in S_{Z_m}} S_{Z_m}(k_j) \} +$$

$$+ \sum_{\Phi_i \in \Phi} [S_I(k_i) - S_{Z_m}(k_i)] - \sum_{\Phi_i \in \Phi} \log \left[ \sum_{k_j \in S_{Z_m}} \exp \left\{ -\frac{1}{2} \left( \frac{d_{ij}}{\sigma} \right)^2 \right\} S_{Z_m}(k_j) \right]$$

Similarly, we develop the occlusion error by taking the negative logarithm of the occlusion likelihood:

$$Error_O(m) = -\log \{ P_O(Z_m) \} + \sum_{\Phi_i \in \Phi} [S_{Z_m}(k_i) - S_I(k_i)] .$$

In Section 5, we assume the a priori probabilities, $P_A(Z_m)$ and $P_O(Z_m)$, to be identical for all models, and therefore ignore this term while calculating the total error.

4.3 Recognition Algorithm

1. The curvature histogram of the range image to be recognized is computed in a manner similar to the method described in Section 4.1.

2. For each model, $m$, the total error is computed as shown in Equation 43.

3. The image is recognized as belonging to the model with the minimum error.

5 Performance

The recognition algorithm discussed in Section 4 was tested using both synthetic and real range images. To test the recognition algorithm and empirically deter-
mine the sensitivity of the algorithm to noise, data with well-controlled statistical properties was required. Therefore, synthetic images were generated [20] which represent rotated and translated octant and sub-octant patches from one of the ten ellipsoid octant models (specifically ellipsoid model 5). These ten ellipsoid models vary in only one of their major axes. We note from Tables 1 and 2 that models 4, 6 and 7 are very similar to model 5 (indistinguishable to a human observer), and the algorithm matching corrupted images extracted from model 5 can be expected to fail most often by classifying the unknown as case 4, 6 or 7. Indeed, the experiments confirm this expectation. The synthetic image patches were corrupted with white Gaussian noise. After preprocessing, the resulting range image was matched using the total error measure defined by Equation 43.

To build the model histogram library, a range image synthesis program [20] was used to generate three quarter-cylinder models, ten different ellipsoid-octant models, and cone, hyperboloid, and sphere models. The model-based curvature histogram library is developed using the algorithm described in Section 4. Even though these surfaces are used in our experiments, it should be emphasized that the objective of this work is not to recognize only quadric surfaces. (See Faugeras [13] for an excellent approach to quadric surface recognition.) Ellipsoids and other quadric surfaces were used in these experiments for ease of synthesis and analytic tractability. However, there is nothing in the algorithm that requires analytic representations.
Table 1: The Quadric Models

<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellipsoid model 1</td>
<td>66.6667</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 2</td>
<td>66.6667</td>
<td>100</td>
<td>141.421</td>
</tr>
<tr>
<td>ellipsoid model 3</td>
<td>66.6667</td>
<td>100</td>
<td>115.47</td>
</tr>
<tr>
<td>ellipsoid model 4</td>
<td>66.6667</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ellipsoid model 5</td>
<td>66.6667</td>
<td>100</td>
<td>89.4427</td>
</tr>
<tr>
<td>ellipsoid model 6</td>
<td>66.6667</td>
<td>100</td>
<td>81.6497</td>
</tr>
<tr>
<td>ellipsoid model 7</td>
<td>66.6667</td>
<td>100</td>
<td>75.5929</td>
</tr>
<tr>
<td>ellipsoid model 8</td>
<td>66.6667</td>
<td>100</td>
<td>70.7107</td>
</tr>
<tr>
<td>ellipsoid model 9</td>
<td>66.6667</td>
<td>100</td>
<td>66.6667</td>
</tr>
<tr>
<td>ellipsoid model 10</td>
<td>66.6667</td>
<td>100</td>
<td>63.2456</td>
</tr>
<tr>
<td>truncated cone model 11</td>
<td>50 to 100</td>
<td>50 to 100</td>
<td>-</td>
</tr>
<tr>
<td>hyperboloid model 12</td>
<td>100</td>
<td>89.4427</td>
<td>-</td>
</tr>
<tr>
<td>sphere model 13</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>cylinder model 1</td>
<td>16.1937</td>
<td>16.1937</td>
<td>-</td>
</tr>
<tr>
<td>cylinder model 2</td>
<td>14.7479</td>
<td>14.7479</td>
<td>-</td>
</tr>
<tr>
<td>cylinder model 3</td>
<td>12.6491</td>
<td>12.6491</td>
<td>-</td>
</tr>
</tbody>
</table>

The sensitivity of the algorithm was tested with respect to *cardinality* and to noise. By *cardinality*, we mean the number of pixels in the observed portion of the image surface. Cardinality is distinguished from area since a single pixel can subtend a vast range of possible areas, depending on its orientation with respect to the viewpoint. In the following sections, the term *noise* will refer to the variance of white Gaussian noise added to the image. Tests were run with preprocessing described in Section 5.1.

\(^2\)See Table 2 for additional clarification.

\(^3\)See Table 2 for additional clarification.
### Table 2: Coefficients of the Quadric Models

\[ Ax^2 + By^2 + Cz^2 = D^2 \]

<table>
<thead>
<tr>
<th>Models</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellipsoid model 1</td>
<td>9</td>
<td>4</td>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 2</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 3</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 4</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 5</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 6</td>
<td>9</td>
<td>4</td>
<td>6</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 7</td>
<td>9</td>
<td>4</td>
<td>7</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 8</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 9</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>200</td>
</tr>
<tr>
<td>ellipsoid model 10</td>
<td>9</td>
<td>4</td>
<td>10</td>
<td>200</td>
</tr>
<tr>
<td>truncated cone(^4)model 11</td>
<td>4</td>
<td>-1</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>hyperboloid(^5)model 12</td>
<td>-9</td>
<td>4</td>
<td>5</td>
<td>200</td>
</tr>
<tr>
<td>sphere model 13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>cylinder model 1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>16.1937</td>
</tr>
<tr>
<td>cylinder model 2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>14.7479</td>
</tr>
<tr>
<td>cylinder model 3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>12.6491</td>
</tr>
</tbody>
</table>

#### 5.1 Preprocessing

Prior to mapping a noisy surface into principal curvature space, we must preprocess the noisy range image in order to extract meaningful first and second order derivatives from the underlying surface of the data. This is accomplished in this paper by using various combinations of three techniques:

1. Iterative Gaussian smoothing, or
2. Mean Field restoration, or

\(^4\)The cone is truncated such that \(50 \leq \sqrt{x^2 + z^2} \leq 100\).

\(^5\)The hyperboloid suboctant patch is constrained by \(x > 0, y > 0,\) and \(z > 0\). Then, in order, it is pitched 35.26 degrees about the y-axis, yawed 45 degrees about the x-axis, translated 10, 70, and 60 units in the x, y, and z directions, respectively, and finally constrained within a 256 by 256 image by \(0 \leq z \leq 255, 0 \leq y \leq 255\).
3. MSE fitting of the data to some particular surface type, such as a quadric [13].

Iterative Gaussian smoothing [10] uses the smallest smoothing kernel possible [26], thus preserving resolution, and produces floating point output. Iterating the process results in the equivalent of a large kernel.

In a separate work [5] [16], the authors of this paper set up a restoration problem which required that

1. the restored data should be similar (in the MSE sense) to the data, and

2. the restored data should be locally smooth.

This problem formulation resulted in a minimization problem which was solved using Mean Field Annealing [7] [16] (MFA). The algorithm performed spectacularly well on range images of polyhedra, removing noise while preserving edges. When applied to quadric surfaces, the algorithm produced excellent, although not perfect, restorations. This same software was used in some experiments described in this section.

In Figures 1 through 4, we show the error generated when comparing rotated, translated, noisy subimages of ellipsoid model 5 with each of the ten ellipsoids, truncated cone, hyperboloid, and sphere models. In each graph, the error resulting from comparing the model 5 subimages with model $i$ is indicated by the number $i$ immediately adjacent to the corresponding error curve. Thus, whenever the error curve corresponding to model 5 exceeds one of the other curves, a misclassification
occurs. In Figures 1 through 4, note the relatively large separation between the recognition error measurements of model 5 and models 11, 12, and 13 (truncated cone, hyperboloid, and sphere, respectively).

5.2 Results: Preprocessing Using Gaussian Smoothing

Noisy synthetic quadric data was processed with iterative Gaussian smoothing. Curvatures were determined from the smoothed surface.

5.2.1 Sensitivity to Cardinality

Figure 1 illustrates the recognition results after smoothing by showing the error measure versus cardinality for a fixed Gaussian noise standard deviation, $\sigma_n = 0.30$. Figure 1 clearly shows how important it is to have relatively large image segments to work with, when trying to classify very similar objects. Error for the correct class drops, while erroneous matches have errors which grow relatively rapidly (note log scale) with cardinality. The error "spikes" which occur at lower cardinalities are due to the relative placement of a small image with respect to ellipsoid model 5. Extracting a small image from a highly identifiable portion of a model will cause the algorithm to produce relatively large errors for models not possessing those image curvatures. Conversely, extracting a small image from an area which is not uniquely identifiable will cause similar error values for several models. Lower cardinality images were extracted from different regions of model 5. This accounts for all the low cardinality error "spikes" present in Figures 1 and 3.
Figure 1: Error vs. Cardinality, $\sigma_n = 0.30$

5.2.2 Sensitivity to Noise

Figure 2 illustrates the recognition results after smoothing by showing the error measure versus the Gaussian noise standard deviation for a fixed cardinality of 8530. Here, the error associated with grossly incorrect matches actually declines with increasing noise. This rather anomalous result is explained by the fact that an increase in the standard deviation of noise was dealt with by a linear increase in the number of smoothing steps.
5.3 Results: Preprocessing Using Quadric Fitting

Noisy synthetic quadric data was MSE fitted with a quadric surface. Curvatures were determined from the fitted surface.

5.3.1 Sensitivity to Cardinality

Figure 3 illustrates the recognition results after quadric surface fitting by showing the error measure versus cardinality for a fixed Gaussian noise standard deviation of 0.30. Again, we note that the decline in error rate for larger images is dramatic, and that error "spikes" (explained in Section 5.2.1) occur for lower cardinalities.
5.3.2 Sensitivity to Noise

Figure 4 illustrates the recognition results after quadric surface fitting by showing the error measure versus the Gaussian noise standard deviation for a fixed cardinality of 8530.

5.4 Tests with Industrial Images

To further test the algorithm, two images of cylinders were obtained, using the GE [1] range camera. In addition to the high levels of noise found in present laser triangulation range scanners, these images (Figure 5) exhibited a large number of dropouts (points where the scanner simply returned no data), and some geometric distortion. Dropouts apparently occurred often in this data because the surfaces were metallic and somewhat specular.

In order to correct for dropouts in these cylinder images, prior to submittal to the recognition algorithm, these images were first preprocessed by a median filter. Noise was removed from the median filter output by the MFA restoration algorithm mentioned in Section 5.1. The MFA output was then operated upon by quadric surface fitting. After fitting, the resulting range images were compared to cylindrical models using the total error measure, defined by Equation 43. The cylinder images have cardinalities of 1595 and 1424 pixels. Gaussian smoothing was not used since the images were too small. The error measures for the cylinders
Figure 3: Error vs. Cardinality, $\sigma_n = 0.30$

![Figure 3: Error vs. Cardinality, $\sigma_n = 0.30$](image)

Figure 4: Error vs. $\sigma_n$, Cardinality = 8530

![Figure 4: Error vs. $\sigma_n$, Cardinality = 8530](image)
compared to all the models are presented in Table 3.6

Table 3: Cylinder Recognition Error Measurements

<table>
<thead>
<tr>
<th>Models</th>
<th>total error for cylinder 1</th>
<th>total error for cylinder 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellipsoid 1</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 2</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 3</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 4</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 5</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 6</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 7</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 8</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 9</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>ellipsoid 10</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>truncated cone 11</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>hyperboloid 12</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>sphere 13</td>
<td>1.70e+38</td>
<td>1.70e+38</td>
</tr>
<tr>
<td>cylinder 1</td>
<td>2.06e+03</td>
<td>2.09e+03</td>
</tr>
<tr>
<td>cylinder 2</td>
<td>2.00e+03</td>
<td>1.59e+03</td>
</tr>
<tr>
<td>cylinder 3</td>
<td>1.70e+38</td>
<td>1.64e+03</td>
</tr>
</tbody>
</table>

In Table 3, we note that while the algorithm clearly distinguishes between the cylinders and the other surfaces, it confuses the range image of cylinder 1 with the model of cylinder 2. To see the reason for this, observe Figure 6 which presents orthographic views of cylinder 1, before and after preprocessing. It is clear from these images that, due to geometric distortion in the camera, the image is simply not cylindrical.

6In Table 3, the value 1.70e+38 is not the actual error, but rather the largest floating point number representable on the computer used. The error was actually in excess of this value.
Figure 5: Dropouts in Range Data

Figure 6: Orthographic Views of Cylinder 1, Before and After
6 Conclusion

In this paper, we have presented a viewpoint-invariant method for recognition of curved surfaces in range images. Although we have given considerable attention to regular surfaces, the method is by no means limited to such surfaces. A sphere, for example, will produce a sharp peak somewhere along the line \( k_1 = k_2 \). Furthermore, two spheres may be distinguished by the location of the peak along that line, which is determined by the radius.

The method is tolerant of partial occlusion. Such an occlusion simply reduces the height of some bins in the histogram. This fact contributes significantly to the power of the result: the method does not depend on an accurate segmentation. Segmentation errors which cause region fragmentation [30] or which, due to boundary effects, reduce region size, are automatically accommodated. The construction of the curvature histogram is fundamentally an integral process, making the method somewhat robust under such errors. Region blending [3] [17], the other type of common segmentation error, will cause erroneous results. However, most segmentation algorithms may be tuned to favor either region fragmentation or region blending. To use this algorithm, one should tune the segmenter to be prone to fragmentation (also referred to as over segmentation). The curvature histogram may then be used to not only recognize the region, but will also serve to identify the fragmented components of the same region. When used in this way, the algorithm functions in much the same way, philosophically, as Hough-based
strategies [11], or as mappings onto a parameter space such as the Gauss sphere [18, 365-391].

The principal criticism which may be leveled against this technique has nothing to do with the technique at all: it is the fact that given a noise-corrupted surface, the input (curvature of the underlying surface) is difficult to accurately estimate. The purpose of this paper is not to address how to calculate curvature. In this sense, we philosophically follow the lead of researchers in optical flow [21] [24] [27] who assume that a disparity map [27] or motion field [21] [24] is available, and show that a wide variety of powerful conclusions may be drawn. Accurate estimation of disparity maps and motion fields is often exceedingly difficult since it requires solution of the correspondence problem.

We are not discouraged by the difficulty inherent in curvature calculation. In this paper we used three different preprocessing techniques to smooth the images and then computed curvatures in a rather straightforward way. We had reasonable success in distinguishing between surfaces which were very similar (recall Table 1). Other researchers in our own laboratory, as well as others [9] [23], are actively pursuing improved methods of curvature estimation. Guckin [15] has shown that fitting a paraboloid can provide invariant local estimates of curvature, provided that fit is performed in the locally tangent frame. (Of course, the tangent frame is also unknown, requiring an iterative algorithm). Guckin's work is similar to the invariant reconstruction of Blake and Zisserman [8, 90-96] with the exception
that Guckin does not explicitly model discontinuities. Guckin finds this adequate for invariant detection of step and roof edges using a single constant threshold. Other researchers have posed restoration problems which remove noise while preserving edges, using assumptions that the surface is locally constant [16] [29] [8] or locally planar [4] [8], with excellent results. Although we have not yet had the opportunity to perform the experiment, it seems straightforward to formulate such a restoration which assumes the surface is locally homogeneous in curvature. It is likely that such a restoration technique would further reduce the frequency of image misclassification.

Finally, if one is fortunate enough to be dealing with surfaces of a known analytic form (e.g. quadrics) [13], truly superior curvature estimates may be produced by first fitting the entire surface with a single function.

Thus we have a 3-D surface recognition method which is robust and viewpoint-invariant, provided relatively good estimates of curvature may be determined, and we have indications that such determinations are feasible.
References


