Invariant Surface Characteristics for 3D Object Recognition in Range Images*

PAUL J. BESL AND RAMESH C. JAIN

Electrical Engineering and Computer Science Department, The University of Michigan, Ann Arbor, Michigan 48109-1109

Received January 12, 1985; revised March 26, 1985 and June 26, 1985

In recent years there has been a tremendous increase in computer vision research using range images (or depth maps) as sensor input data. The most attractive feature of range images is the explicitness of the surface information. Many industrial and navigational robotic tasks will be more easily accomplished if such explicit depth information can be efficiently obtained and interpreted. Intensity image understanding research has shown that the early processing of sensor data should be data-driven. The goal of early processing is to generate a *rich description* for later processing. Classical differential geometry provides a complete local description of smooth surfaces. The first and second fundamental forms of surfaces provide a set of differential-geometric shape descriptors that capture domain-independent surface information. *Mean curvature* and *Gaussian curvature* are the fundamental second-order surface characteristics that possess desirable invariance properties and represent *extrinsic* and *intrinsic* surface geometry respectively. The signs of these surface curvatures are used to classify range image regions into one of eight basic viewpoint-independent surface types. Experimental results for real and synthetic range images show the properties, usefulness, and importance of differential-geometric surface characteristics. © 1986 Academic Press, Inc.

1. INTRODUCTION

The quality of the digitized range data available from active and passive sensors has improved steadily with time. As a result, there has been a tremendous increase in computer vision research using range data in recent years. Range data is often obtained as an array of numbers, which is referred to as a *range image* (or *depth map*), where the numbers quantify the distances from the focal plane of the sensor to the surfaces of objects within the field of view along rays emanating from a regularly spaced grid. Range images are obtained using a variety of different techniques described in [1, 4, 26, 28, 29, 35, 47, 49, 53, 54]. The most attractive feature of range images is the explicitness of object surface information. Indeed, the three-dimensional (3-D) *shape* of range image regions directly approximates the 3-D shape of the corresponding *visible object surfaces* in the field of view. Because of the *explicitness* of this type of information, recognizing objects by their geometric shape should be much easier using range images instead of the intensity images conventionally used in computer vision. Many industrial and navigational robotic tasks will be more easily accomplished if such explicit depth information can be efficiently obtained and interpreted.

Contrary to common belief, the problems confronted by range data systems are not that much different from those of visual systems. Intensity images and range images have identical image formats; only the interpretation of pixel values differs.

*This work was supported in part by IBM Corp., Data Systems Div., Kingston, N.Y., under the monitorship of Dr. Jack Contino and in part by the Air Force Office of Scientific Research under Contract F49620-82-C-0089.

0734-189X/86 $3.00
Copyright © 1986 by Academic Press, Inc.
All rights of reproduction in any form reserved.

Several recent techniques for processing range images are reminiscent of those used by vision researchers in the early 1970's. Intensity image understanding research has shown that the early processing of sensor data should be completely data-driven. The goal of early processing is to generate a *rich description* of the sensor data for later processing by higher level modules of a vision system. The data-driven rich-description concept is equally valid for range images.

Our approach to range image description is motivated by a detailed analysis of the range image object recognition problem. This analysis suggests that objects in range images should be recognized by the characteristics of the family of depth map surface functions associated with an object. Classical differential geometry provides a complete local description of smooth surfaces and guides our selection of surface characteristics. *Mean curvature* and *Gaussian curvature* are identified as the local second-order surface characteristics that possess several desirable invariance properties and represent *extrinsic* and *intrinsic* surface geometry respectively. The signs of these surface curvatures are used to classify range image surface regions into one of eight basic types. We have identified a set of differential-geometric descriptors that capture domain-independent, view independent surface information based on the first and second fundamental forms of surfaces. An approach for computing these surface characteristics using window operators is presented, and experimental results of this computational approach are included to demonstrate its performance on real range images and synthetic depth maps with varying amounts of noise.

This paper is structured as follows. First, the object recognition problem is defined. This general problem is only partially addressed by existing computer vision systems. Second, the recognition problem is given a precise mathematical interpretation as a generalized inverse set mapping. Third, our approach for computing this mapping is outlined, and the concept of "visible-invariance" is discussed. Next, we review the relevant differential geometry of surfaces to point out the importance of the selected surface curvature characteristics. Other related techniques from the literature are briefly reviewed at this point in light of the previously discussed topics. Our method for computing various surface characteristics is then presented followed by experimental results. We conclude with a brief list of future research directions.

2. DEFINITION OF OBJECT RECOGNITION PROBLEM

Three-dimensional object recognition is a rather nebulous term. A survey of the literature on this subject demonstrates this point [4]. Therefore, we attempt to give a reasonably precise definition to the object recognition problem. We first discuss desirable human visual capabilities and then describe how these relate to computer vision.

The *real world* that we see and touch is primarily composed of solid *objects*. When people are given a new object they have never seen before, they are able to gather information about that object from many different *viewpoints*. The process of gathering detailed object information and storing that information is referred to as *model formation*. Once we are familiar with many objects, we can normally identify those objects from an *arbitrary stationary viewpoint* without further investigation.

People are also to *identify, locate, and qualitatively describe the orientation* of objects in black-and-white photographs. This basic capability is significant to computer vision research because it involves the spatial variation of only a *single* parameter within a framed rectangular region corresponding to a fixed view of the real world. Human color vision is more difficult to analyze and generally involves a *three* parameter color variation within a large, almost hemispherical solid angle corresponding to a continually changing viewpoint. Because we are interested in an automatic, computerized recognition process, input data must be compatible with digital computers. The term *digitized sensor data* refers to any input matrix of numerical values (which can represent intensity, range, or any other scalar parameter) and associated auxiliary information about how the matrix of values was obtained.

The above discussion motivates the following definition of the autonomous single-arbitrary-view 3-D object recognition problem:

(1) **Given** any collection of labeled rigid solid objects:

(a) These objects may be examined in any way desired (automatically or manually) as long as the objects are not deformed.

(b) Labeled object models may be formed using information from this examination.

(2) **Given** digitized sensor data corresponding to one particular, but arbitrary, field-of-view of the real world as it existed at the time of data acquisition;

Given any data stored previously during the model formation process; and

Given a list of distinguishable objects; the following issues must be addressed for each object in the list using the capabilities of an autonomous processing unit:

(a) Does the object appear in the digitized sensor data? (That is, does the object permit a consistent interpretation of a subset of the sensor data?)

(b) If so, how many times does the object occur?

(c) For each occurrence of the object:

(i) determine the *location* within the sensor data;

(ii) determine the 3-D location (or translation parameters) of that object with respect to a known coordinate system (if possible with the given sensor); and

(iii) determine the 3-D orientation (or rotation parameters) of that object with respect to a known coordinate system (if possible with the given sensor).

(3) If there are any regions in the sensor data that do not correspond to objects in the known list, characterize these regions in a way that will make them recognizable should they occur again in future images.

We refer to the problem of successfully completing these assigned tasks using real world sensor data while obeying the given constraints as the 3-D object recognition problem. This problem is *not* successfully addressed in many of the object recognition systems discussed in the literature (see [4]); more constrained problems, which are limited to particular surface types or particular applications, are typically addressed. If the stated 3-D object recognition problem could be solved successfully by a vision system, that system would be extremely useful in a wide variety of applications, including automatic inspection and assembly, and autonomous vehicle navigation. The problem is stated so that it may be feasible to use computers to solve the problem, and it is also clearly solvable by human beings.

3. MATHEMATICAL PROBLEM FORMULATION

We now leave the general problem to focus on range images as sensor data. It is often beneficial to define a problem in a stricter mathematical form to eliminate possible problem ambiguities. For example, we have not yet discussed how a system should respond if several distinct objects appear to be identical from a given viewpoint. Therefore, we redefine *range image object recognition* in precise mathematical terms as a *generalized inverse set mapping*.

First, we consider world modeling issues. We approximate the world as consisting of N_{tot} objects. The number of "distinguishable" objects is N_{obj}. Hence, $N_{\text{obj}} \leq N_{\text{tot}}$ in general. Two objects are considered distinguishable if a human being can tell them apart without reading surface markings of any sort. We give each distinguishable object an index i, and we refer to that object as A_i. The number of occurrences, or instances, of that object is denoted as N_i. This yields the general relationship

$$N_{\text{tot}} = \sum_{i=1}^{N_{\text{obj}}} N_i.$$

People can recognize an enormous number of objects depending on personal experience. The number of objects to be recognized by an object recognition system depends on the application and the appropriate system training.

In certain cases, it is difficult to decide what is an object and what is an assembly of objects. Each object should therefore be considered as possessing its own coordinate system and list of sub-objects for complete generality. In this discussion, we consider only simple objects with no sub-parts and with only one occurrence, or instance. That is, we let $N_{\text{obj}} = N_{\text{tot}}$ and $N_i = 1$ for simplicity. The general case of multiple instances of objects with sub-parts is not conceptually different with respect to the surface characterization problem discussed subsequently. Nevertheless, it does present important notation problems and implementation difficulties for higher level recognition processing. We define the origin of the object coordinate system at the center of mass of the object with three orthogonal axes aligned with the principal axes of the object because these parameters can be uniquely determined for any given solid object.

Each object occupies space, and at most one object can occupy any given point in space. It is necessary to describe the spatial relationships between each object and the rest of the world. One way to describe spatial relationships is with coordinate systems. For reference purposes, we assume the existence of a world coordinate system that is placed at any convenient location. Objects are positioned in space relative to this coordinate system using translation and rotation parameters. We refer to the translation parameters of an object as the vector $\boldsymbol{\alpha}$, and the rotation parameters of an object as the vector $\boldsymbol{\theta}$. The number of parameters for each vector depends on the *dimension* of the range image recognition problem. For example, the 2-D problem would require only three parameters total. For the 3-D case, we write the six 3-D parameters as follows:

$$\boldsymbol{\alpha} = (\alpha, \beta, \gamma) \qquad \boldsymbol{\theta} = (\theta, \phi, \psi).$$

See Fig. 1 for the graphical illustration of these parameters. We can now precisely

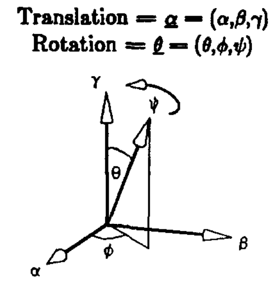
Translation = $\underline{\alpha}$ = (α,β,γ)
Rotation = $\underline{\theta}$ = (θ,ϕ,ψ)
γ
ψ
θ
φ
β
α

FIG. 1. 3-Dimensional translation and rotation parameters.

define our world model W as a set of ordered triples (object, translation, rotation) in the world reference coordinate system:

$$W = \{(A_i, \boldsymbol{\alpha}_i, \boldsymbol{\theta}_i)\}_{i=0.}^{N_{\text{obj}}}$$

We consider object A_0 to be the sensor object with position $\boldsymbol{\alpha}_0$ and orientation $\boldsymbol{\theta}_0$. If a time-varying world model is required, all parameters can be functions of time. For our current purposes of single-view object recognition, we concern ourselves with only static parameter values. We denote the set of all objects (the object list) as $L = \{A_i\}$. The set of all translations is denoted $\mathbf{R}^t$, and the set of all rotations is denoted $\mathbf{R}^r$. In the 3-D problem, $t = 3$ and $r = 3$. $\mathbf{R}$ is the set of all real numbers.

A depth sensor (or range finder) obtains a depth map projection of a scene. We model this projection as a mathematical operator P that maps the set $\Omega = L \times \mathbf{R}^t \times \mathbf{R}^r$ into the set of all scalar functions of $t - 1$ variables, which we denote as F:

$$P\colon \Omega \to F.$$

These real-valued functions are referred to as depth map surface functions (or range image functions). This projection operator could be *orthographic* or *perspective*, but we consider only the orthographic projection in this discussion. We write the projection operation as

$$f(\mathbf{x}) = g_{A,\boldsymbol{\alpha},\boldsymbol{\theta}}(\mathbf{x}) = P(A, \boldsymbol{\alpha}, \boldsymbol{\theta})$$

where $\mathbf{x}$ is the vector of $t - 1$ spatial variables of the focal plane of the sensor. The spatial parameters of the sensor object (the location $\boldsymbol{\alpha}_0$ and the orientation $\boldsymbol{\theta}_0$) are *implicitly* assumed arguments of the projection operator. This is done to simplify our expressions because we only have one sensor in this formalism. We refer to the depth-map surface function as f when the identity of the object and its parameters are unknown. The symbol g with subscripts refers to the depth-map surface function of a known object at a known location and orientation. This formalism points out that the set of depth map functions associated with a single object is an *infinite family of functions*. Two of the rotation parameters in the $\boldsymbol{\theta}$ vector have a particularly profound effect on this family of functions: the 3-D shape of the depth map function changes as the object rotates. Translation parameters have no effect on shape whatsoever under orthographic projection, and they have minimal effect under the perspective projection unless the sensor is very close to the object of interest (e.g., closer than 10 times the maximum object width).

Since objects do not occupy all space, we need a convention for the value of the depth map function for values of the spatial vector $\mathbf{x}$ that do not correspond to object surface points. If the point $(\mathbf{x}, f(\mathbf{x}))$ cannot lie on an object surface, we assign the value of $-\infty$ to $f(\mathbf{x})$. Hence, we can write the projection of a set of M objects as

$$f(\mathbf{x}) = \max_{1 \le i \le M} g_{A_i, \boldsymbol{\alpha}_i, \boldsymbol{\theta}_i}(\mathbf{x}).$$

The depth map object recognition problem is now rephrased as follows: Given a depth map function $f(\mathbf{x})$ that results from depth map projection of a 3-D world scene, determine the sets of possible objects with the corresponding sets of translation and rotation parameters that could be projected to obtain the given depth map function. That is, determine the set of all ω_J, subsets of Ω, such that $\omega_J = \{(A_j, \boldsymbol{\alpha}_j, \boldsymbol{\theta}_j)\}_{j \in J}$ projects to the depth map $f(\mathbf{x})$, where J is an index set that depends on the possible valid interpretations of the depth map.

We can write these ideas more precisely using inverse set mappings. For every *single object* depth-map function, there is a corresponding inverse set mapping that yields all single objects that could have created that object depth-map function. We denote the inverse set mapping of P as P^{-1}, where

$$P^{-1}(f(\mathbf{x})) = \{(A, \boldsymbol{\alpha}, \boldsymbol{\theta}) \in \Omega | P(A, \boldsymbol{\alpha}, \boldsymbol{\theta}) = f(\mathbf{x})\}.$$

In general, an inverse set mapping takes sets from the power set of the range into sets in the power set of the domain:

$$P^{-1}: 2^F \to 2^\Omega.$$

For our purposes, we restrict the input sets in 2^F to be singletons (i.e., single depth-map functions); therefore, we can replace 2^F with F. For multiple object depth-map functions, we must generalize P^{-1} one step further. A generalization is necessary because of the possible combinations of objects that can occur, as shown in Fig. 2. Hence, given $f(\mathbf{x}) \in F$, we seek a "generalized" inverse set mapping $\mathbf{P}^{-1}$ such that

$$\mathbf{P}^{-1}(f(\mathbf{x})) = \left\{ \omega_J \subseteq 2^\Omega | \max_{j \in J} P(A_j, \boldsymbol{\alpha}_j, \boldsymbol{\theta}_j) = f(\mathbf{x}) \right\}.$$

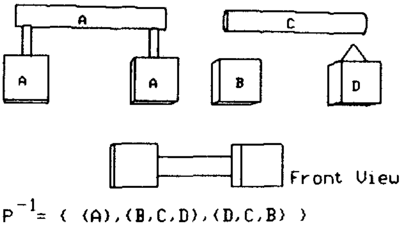

FIG. 2. Different, valid interpretations of a simple scene.

This mapping takes elements of the depth map function space into the power set of the power set of Ω:

$$\mathbf{P}^{-1}\colon F \to 2^{2^{\Omega}}.$$

The dimensionality of such a range space is huge even for relatively small finite sets, and the task of computing such an inverse for infinite sets appears formidable. We know that human beings can do similar "computations" quickly, easily, and accurately, using extremely compact biological "equipment." These computations are possibly being performed at several different levels using a variety of features and a large knowledge base. Knowing that the problem is solvable, we search for methods so that computers can handle such tasks.

The depth-map object recognition problem can now be stated in terms of the modeled world as follows: given the world model W with N_{obj} simple objects and given any realizable depth-map function $f(\mathbf{x})$, compute the inverse projection set mapping $\mathbf{P}^{-1}(f(\mathbf{x}))$ to obtain all possible explanations of the function $f(\mathbf{x})$ in terms of the world model. In many cases, there is only one valid scene interpretation. Nonetheless, ambiguous situations, such as that in Fig. 2, should be recognized as ambiguous when given only a single view. A general-purpose single-view vision system must generate the list of all valid scene interpretations.

Since there is no general theory regarding how to efficiently compute such a mapping, we suggest a computational surface-recognition theory based on *surface characterization and surface matching* constrained by known object structures. We propose that the generalized inverse projection set mapping should be computed by matching the *individual, isolated surface regions* with the $g_{A,\alpha,\theta}(\mathbf{x})$ *depth-map surface function families* of the individual objects in such a way that viewpoint-dependent effects are accounted for. The differential geometry of surfaces provides us with features that are useful for this purpose. We believe that it is necessary to do the matching required by the recognition process on data with lower dimensionality and with more explicit invariance properties than the discrete depth-map surface data itself. Invariance properties are necessary to reduce view-independent surface matching computations.

It is interesting to note that the formalism above can be augmented to state the *object recognition problem for intensity images*. By adding an illumination-reflectance operator I to the depth map function f, we obtain an intensity image $\mathbf{i}(\mathbf{x})$ given by

$$\mathbf{i}(\mathbf{x}) = I\Big(\max_{i} P(A_i, \alpha_i, \theta_i)\Big).$$

The intensity image object recognition problem is generally more difficult because of the additional inversion of the I operator. Note that the *max* computation is the multiple object occlusion operator. To expand our world model for understanding intensity images, we would also need to add objects that generate light in addition to the single sensor object that receives light. Shape from shading [24], shape from photometric stereo [12, 56], and shape from texture [55] techniques attempt to uniquely invert the I operator producing the depth map function f.

It is also interesting to note that people can understand images even when other shading operators besides illumination-reflectance are used to color visible surfaces. For example, people can correctly interpret photographic negatives or pseudo-colored images where the usual color and/or light-dark relationships are completely distorted.

The intensity-image object recognition problem is generally regarded as an ill-posed problem because of the lack of knowledge about light sources and surface reflectance functions even if possible object shapes are known. In contrast, we feel that the range-image object recognition problem as stated is a well-posed problem. In this paper, we focus on the viewpoint-invariant surface characteristics of these object depth-map surface-function families.

4. OVERVIEW OF OBJECT RECOGNITION APPROACH

Computing surface characteristics is an intermediate step toward the final goals of object recognition and range image understanding. Several approaches to the range image 3-D object recognition problem have already been developed [6, 8, 14, 25, 33, 44, 46, 51, 52]. A literature survey by Besl and Jain [4] discusses these approaches and points out that no general purpose vision systems have been developed that can use range-image input in a completely satisfactory manner. We are addressing the range-image object recognition problem with the aim of developing a *general purpose* approach that handles arbitrary surface shapes and arbitrary viewing directions. To have any hope of doing this, we require view-invariant surface characteristics that are robust enough to describe both polyhedra and curved objects. This notion of invariance is extremely important and is often confused in the vision literature. We therefore elaborate on this topic.

The term "invariant" refers to a quantity that does not change under a specified group of transformations. Opaque physical objects do not, in general, possess explicit surface or edge features that are visible from any viewing angle. There are almost always degenerate viewing angles in which object features are radically different. Consider, for example, an object as simple as a cylinder. One sees a flat planar surface with a circular boundary when one looks down the axis of a cylinder. On the other hand, one sees a curved surface with a rectangular projected boundary when one looks perpendicular to the axis direction. See Fig. 3 for the two views under consideration. There are *no explicit* invariant features even in this simple case. (For example, we do not consider the minimum projected silhouette area as an explicit feature.) The *roundness* of the cylinder manifests itself in both views, but in different ways: in the first case, we get an explicit *circular arc* depth-discontinuity boundary surrounding a flat region while, in the second case, we get a constant-negative-mean-curvature, zero-Gaussian-curvature surface bounded by a projected rectangular depth-discontinuity boundary. There are papers in the computer vision literature that use the term "invariant" to mean "invariant if visible." The term "visible-invariant" should be used to be more specific. A "visible-invariant" surface characteristic is a quantitative feature of a surface region that does not change under viewpoint transformations that do not affect the visibility of that region. *A general*

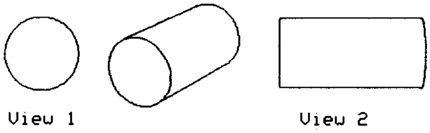

FIG. 3. Two views of a cylinder with no common features.

purpose vision system must be aware that key object features may not be visible even when an object is present and visible in the field of view. In situations where it is not possible to take intelligent actions based on what is currently visible, the general purpose system should automatically request the acquisition of image data from new vantage points [32].

The visible-invariant surface characteristics that we have decided to use are the *Gaussian curvature* (K) and the *mean curvature* (H), which are referred to collectively as *surface curvature*. We abbreviate this term as *S-curvature*. When a surface region is visible, its S-curvature is invariant to *changes in surface parameterization* and to *translations and rotations* of object surfaces. In addition, mean curvature is an *extrinsic* surface property whereas Gaussian curvature is *intrinsic*. These terms are discussed later. Differential geometry emphasizes that these are quite reasonable surface features to consider.

Since we can seldom obtain perfect sensor data from the real world, it is desirable to compute a "rich" characterization of a surface that preserves the surface structure information and is insensitive to noise. Noise insensitivity may be achieved by computing redundant, or at least "overlapping," information about a surface. In order to have a very rich geometric representation, we propose to combine surface critical points (local maxima, minima, and saddle points) and large metric determinant points (depth-discontinuities) with the surface curvature information to characterize a depth map surface in more detail. They provide useful complementary information and can be computed for a small additional cost. Given a depth map surface characterization, we suggest that depth map surface region characteristics can be matched against pre-computed object model surface region characteristics guided by depth-discontinuity and critical point information to achieve object recognition.

The matching algorithm of a robust 3-D object recognition system must be view-independent. One could use multiple view ideas similar to those of Koenderink and van Doorn (visual potential) [30, 31] or Chakravarty and Freeman (characteristic views) [10], but we are pursuing a new, more compact, scheme that does not increase its storage requirements so dramatically as object complexity increases. After the matching algorithm has produced a list of possible objects and their respective locations and orientations, we can use a depth-buffer algorithm to create a synthetic depth map using the world model. Verification matching could be done directly between the synthetic depth map and the sensor data, or we may run the surface characterization algorithm on the synthetic data to yield a synthetic scene description that could be matched against the surface characterization scene description computed from the sensor data. If major discrepancies exist, the system should try to remedy the problems in its understanding automatically. It may also be necessary to compute our surface characterization using different window sizes (scales) and correlate features in this scale-space dimension to help overcome the effects of noise. The matching algorithm, the matching object representation, the feedback process, and scale-space ideas require further study.

5. REVIEW OF DIFFERENTIAL GEOMETRY OF SURFACES

In Section 3, we discussed how range-image object recognition might be decomposed into a surface recognition problem. We assume that surfaces can be recognized by their characteristics. But what does this term "surface characteristic" mean?

We define a *characteristic* of a mathematical entity, such as a surface function, to be any well-defined feature that can be used to distinguish between different mathematical entities of the same type. We may consider characteristics that uniquely determine a corresponding entity or characteristics that are many-to-one although the former are more desirable. One simple example of a characteristic that uniquely determines a function is a detailed description of the function itself. Another simple example of a many-to-one characteristic is the following: A circle and an ellipse are *round* figures. This round characteristic distinguishes them from rectangles, triangles, and other polygons; however, it does not distinguish between circles and ellipses. In this section, we aim to find a good mathematical characterization of depth-map function surfaces.

It is well known that *curvature*, *torsion*, *and speed* uniquely determine the shape of 3-D space curves [5, 15, 23, 37, 45]. We must assume that the reader is familiar with these basic concepts. These characteristics are the ideal type of characteristic for a mathematical entity. They are invariant to coordinate transformations and they have a one-to-one relationship with curve shapes. We now discuss surface characteristics with similar properties.

We first write down the explicit parametric form of a general surface S with respect to a known coordinate system:

$$S = \{(x, y, z): x = d(u,v),\ y = e(u,v),\ z = f(u,v),\ (u,v) \in D \subseteq \mathbf{R}^2\}.$$

We refer to this general parametric representation as $\mathbf{x}(u,v)$, where the x component of the $\mathbf{x}$ function is $d(u,v)$, the y component of $\mathbf{x}$ is $e(u,v)$, and the z component is $f(u,v)$. In a later section, we use the graph surface (Monge patch surface) form to describe depth map surface functions. In the graph surface case, $d(u,v) = u$ and $e(u,v) = v$, which are extremely simple functions. We consider only *smooth* surfaces, where all three parametric functions possess continuous second partial derivatives.

There are two basic mathematical entities that are considered in the classical analysis of smooth surfaces. They are referred to as the first and second fundamental forms of a surface [23, 37, 45]. It is shown subsequently how complete knowledge of these forms uniquely characterizes and quantifies general smooth surface shape. Modern mathematics favors an equivalent formulation of this knowledge in terms of the metric tensor and the Weingarten mapping (the "shape" operator), which we also discuss. We begin our review by defining the fundamental forms of a surface in terms of the general explicit surface parameterization $\mathbf{x}(u,v)$.

The first fundamental form I of a surface defined by $\mathbf{x}(u,v)$ is given by the following quadratic form:

$$I(u,v,du,dv) = d\mathbf{x} \cdot d\mathbf{x} = \begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = d\mathbf{u}^{\mathrm{T}}[g]\, d\mathbf{u}$$

where the $[g]$ matrix elements are defined to be

$$g_{11} = E = \mathbf{x}_u \cdot \mathbf{x}_u \qquad g_{22} = G = \mathbf{x}_v \cdot \mathbf{x}_v \qquad g_{12} = g_{21} = F = \mathbf{x}_u \cdot \mathbf{x}_v$$

where the subscripts denote partial differentiation

$$\mathbf{x}_u(u,v) = \frac{\partial \mathbf{x}}{\partial u} \qquad \mathbf{x}_v(u,v) = \frac{\partial \mathbf{x}}{\partial v}.$$

$\mathbf{x}_u$ and $\mathbf{x}_v$ are referred to as the *u tangent vector* and the *v tangent vector* functions, respectively, and they may or may not be orthogonal to each other. These two tangent vectors are shown in Fig. 4 and are said to lie in (and form a basis for) the tangent plane $T(u, v)$ of the surface at the point $\mathbf{x}(u, v)$. We refer to the $[g]$ matrix as the first fundamental form matrix or, more simply, as the *metric* (or metric tensor) of the surface. Since the vector dot product is commutative, this $[g]$ matrix is symmetric and only has three independent components. We have used the E, F, G notation of Gauss along with the matrix element subscript notation because both are useful in different circumstances, and both occur often in the literature of differential geometry.

The first fundamental form $I(u, v, du, dv)$ measures the small amount of movement $|d\mathbf{x}|^2$ on the surface at a point (u, v) for a given small movement in the parameter space (du, dv) as shown in Fig. 4. This function is invariant to surface parameterization changes and to translations and rotations of the surface. The first fundamental form and the metric depend only on the surface itself. They do not depend on how the surface is embedded in 3-D space, and are therefore referred to as *intrinsic* properties of a surface. In fact, the metric functions E, F, G determine

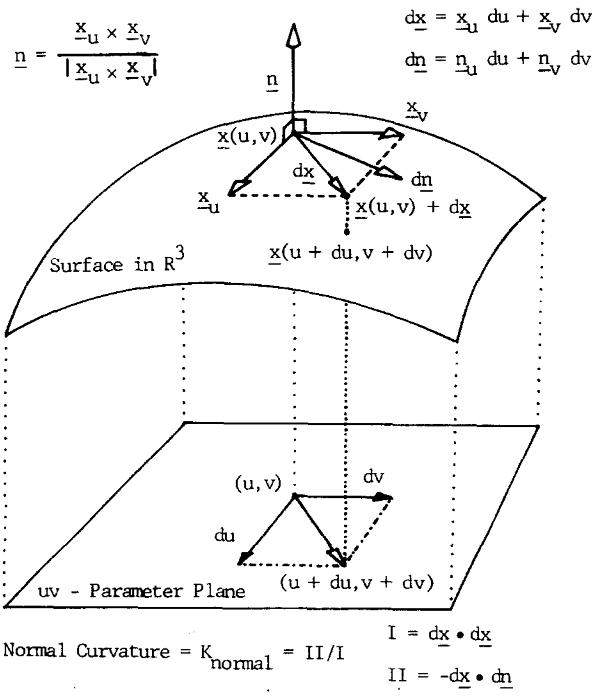
$\underline{n} = \frac{\underline{x}_u \times \underline{x}_v}{|\underline{x}_u \times \underline{x}_v|}$
$\underline{n}$
$\underline{dx} = \underline{x}_u\, du + \underline{x}_v\, dv$
$\underline{dn} = \underline{n}_u\, du + \underline{n}_v\, dv$
$\underline{x}_v$
$\underline{x}(u,v)$
$\underline{dx}$
$\underline{dn}$
$\underline{x}_u$
$\underline{x}(u,v) + \underline{dx}$
Surface in R^3
$\underline{x}(u + du, v + dv)$
(u,v)
dv
du
uv - Parameter Plane
(u + du,v + dv)
Normal Curvature = K_{normal} = II/I
I = $\underline{dx} \bullet \underline{dx}$
II = $-\underline{dx} \bullet \underline{dn}$

FIG. 4. Local coordinate frame at surface point.

all intrinsic properties of a surface. The *metric* 2×2 matrix function plays the same role as the scalar *speed* function does for curves. The intrinsic geometry of a curve is one-dimensional whereas that of a surface is two-dimensional.

In contrast, the second fundamental form of a surface is dependent on the embedding of the surface in 3-D space and is therefore considered as an *extrinsic* property of the surface. The second fundamental form II is given by

$$II(u, v, du, dv) = -d\mathbf{x} \cdot d\mathbf{n} = [du \quad dv] \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = d\mathbf{u}^{\mathrm{T}}[b]d\mathbf{u}$$

where the $[b]$ matrix elements are defined to be

$$b_{11} = L = \mathbf{x}_{uu} \cdot \mathbf{n} \qquad b_{22} = N = \mathbf{x}_{vv} \cdot \mathbf{n} \qquad b_{12} = b_{21} = M = \mathbf{x}_{uv} \cdot \mathbf{n}$$

$$\mathbf{n}(u, v) = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|} = \text{unit normal vector}$$

where the double subscript implies second partial derivatives

$$\mathbf{x}_{uu}(u, v) = \frac{\partial^2 \mathbf{x}}{\partial u^2} \qquad \mathbf{x}_{vv}(u, v) = \frac{\partial^2 \mathbf{x}}{\partial v^2} \qquad \mathbf{x}_{uv}(u, v) = \frac{\partial^2 \mathbf{x}}{\partial u \partial v} = \mathbf{x}_{vu}.$$

The $[b]$ matrix is the second fundamental form matrix and is also symmetric if the surface is well behaved (i.e., if the mixed partial derivatives are equal). The Gauss-like L, M, N notation is introduced again as above. These definitions allow us to discuss the "state" equation for surfaces.

The second fundamental form measures the correlation between the change in the normal vector $d\mathbf{n}$ and the change in the surface position $d\mathbf{x}$ at a surface point (u, v) as a function of a small movement (du, dv) in the parameter space. This is also indicated in Fig. 4. The differential normal vector $d\mathbf{n}$ always lies in the tangent plane $T(u,v)$. The ratio of $II\ (u, v, du, dv)/I(u, v, du, dv)$ is known as the the normal curvature function κ_{normal}. Normal curvature at a surface point varies only as a function of the direction of the differential vector (du, dv) in the parameter space. If $d\mathbf{n}$ and $d\mathbf{x}$ are aligned for a particular direction of (du, dv), that direction is called a *principal direction* of the surface at that surface point. The extrema of the normal curvature function at a given point occur in these directions and are referred to as *principal curvatures*.

The Gauss-Weingarten equations for 3-D surfaces play the same role as the Frenet-Serret equations for 3-D curves. We write the Gauss-Weingarten equations as a matrix differential equation where the differential operator is a type of gradient operator that acts on the normal, u-tangent, v-tangent coordinate frame field:

$$\begin{bmatrix} \mathbf{x}_{uu} \\ \mathbf{x}_{uv} \\ \mathbf{x}_{vu} \\ \mathbf{x}_{vv} \\ \mathbf{n}_u \\ \mathbf{n}_v \end{bmatrix} = \begin{bmatrix} \dfrac{\partial}{\partial u} \\ \dfrac{\partial}{\partial v} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{x}_u(u, v) \\ \mathbf{x}_v(u, v) \\ \mathbf{n}(u, v) \end{bmatrix} = \begin{bmatrix} \Gamma_{11}^1 & \Gamma_{11}^2 & b_{11} \\ \Gamma_{12}^1 & \Gamma_{12}^2 & b_{12} \\ \Gamma_{21}^1 & \Gamma_{21}^2 & b_{21} \\ \Gamma_{22}^1 & \Gamma_{22}^2 & b_{22} \\ -\beta_1^1 & -\beta_1^2 & 0 \\ -\beta_2^1 & -\beta_2^2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_u(u, v) \\ \mathbf{x}_v(u, v) \\ \mathbf{n}(u, v) \end{bmatrix}.$$

The "transition" matrix in this state equation contains twelve coefficient functions that we need to define. The Christoffel symbols of the second kind Γ_{ij}^{k} (connection coefficients) depend only on the metric functions $g_{ij}(u, v)$ and are defined as follows:

$$\Gamma_{ij}^{k}(u, v) = \frac{1}{2} \sum_{m=1}^{2} g^{km} \left[\frac{\partial g_{jm}}{\partial u^{i}} + \frac{\partial g_{mi}}{\partial u^{j}} - \frac{\partial g_{ij}}{\partial u^{m}} \right]$$

where g^{km} is the matrix inverse of g_{km}, which is the tensor notation for the metric $[g]$ already defined, and where $u^1 = u$ and $u^2 = v$. Note that $\Gamma_{ij}^{k} = \Gamma_{ji}^{k}$ since the metric is a symmetric matrix.

The last two row equations of the matrix equation are referred to as the Weingarten equations for 3-D surfaces. The Weingarten equations' coefficients β_j^i depend on both the first and second fundamental form matrices:

$$\beta_j^i = \sum_{k=1}^{2} b_{jk} g^{ki} \qquad \text{or} \qquad [\beta] = [g^{-1}][b].$$

The $[\beta]$ matrix is referred to as the " shape operator" matrix [45] or the *Weingarten mapping* matrix. The Weingarten mapping maps tangent vectors to other tangent vectors in the tangent plane $T(u, v)$ associated with each point $\mathrm{x}(u, v)$. For example, $\mathbf{n}_u(u, v)$ is specified as a linear combination of the u- and v-tangent vectors. One can view the $[\beta]$ matrix as the entity that determines surface shape by relating the intrinsic geometry of the surface to the Euclidean geometry of 3-D space. It is the generalization of the *curvature* of plane curves. In summary, we have now seen that all of the sixteen non-zero "state" matrix coefficient functions depend on only six scalar functions of two variables:

$$g_{11}(u, v) \qquad g_{12}(u, v) \qquad g_{22}(u, v) \qquad b_{11}(u, v) \qquad b_{12}(u, v) \qquad b_{22}(u, v).$$

Assuming we can solve the first-order linear homogeneous space-varying partial differential matrix equation for the $\mathbf{x}_u, \mathbf{x}_v, \mathbf{n}$ coordinate frame, we can also solve for the parametric surface function in the neighborhood of a point (u_0, v_0) using the following 3-D surface reconstruction formulae:

$$\mathbf{x}(u, v) = \int_{u_0}^{u} \mathbf{x}_u(\alpha, v_0)\, d\alpha + \int_{v_0}^{v} \mathbf{x}_v(u, \beta)\, d\beta \qquad \mathbf{x} \in \mathbf{R}^3.$$

There is a *fundamental existence and uniqueness theorem* for 3-D surfaces (which is credited to O. Bonnet):

(1) **Existence.** Let $g_{11}(u, v)$, $g_{12}(u, v)$, $g_{22}(u, v)$ be continuous functions with continuous second partial derivatives. Let $b_{11}(u, v)$, $b_{12}(u, v)$, $b_{22}(u, v)$ be continuous functions with continuous first partial derivatives. Assume all six functions are defined in an open set D containing the point (u_0, v_0). If all six functions satisfy the following set of compatibility equations and sign restrictions, then there exists a unique surface patch defined in the neighborhood of (u_0, v_0) such that g_{ij} and b_{ij} are the first and second fundamental form matrices respectively. Uniqueness is

determined up to a translation and rotation. The sign restrictions are as follows:

$$g_{11} > 0 \qquad g_{22} > 0 \qquad \det[g] = \left(g_{11}g_{22} - (g_{12})^2\right) > 0.$$

The compatibility equations are as follows:

$$\begin{aligned}
(b_{11})_v - (b_{12})_u &= b_{11}\Gamma^1_{12} + b_{12}\left(\Gamma^2_{12} - \Gamma^1_{11}\right) - b_{22}\Gamma^2_{11} \\
(b_{12})_v - (b_{22})_u &= b_{11}\Gamma^1_{22} + b_{12}\left(\Gamma^2_{22} - \Gamma^1_{21}\right) - b_{22}\Gamma^2_{21} \\
\left(b_{11}b_{22} - (b_{12})^2\right) &= g_{12}\left(\left(\Gamma^2_{22}\right)_u - \left(\Gamma^2_{12}\right)_v + \Gamma^1_{22}\Gamma^2_{11} - \Gamma^1_{12}\Gamma^2_{12}\right) \\
&\quad + g_{11}\left(\left(\Gamma^1_{22}\right)_u - \left(\Gamma^1_{12}\right)_v + \Gamma^1_{22}\Gamma^1_{11} + \Gamma^2_{22}\Gamma^1_{12} - \Gamma^1_{12}\Gamma^1_{12} - \Gamma^2_{12}\Gamma^1_{22}\right).
\end{aligned}$$

The first two compatibility equations are often referred to as the Mainardi–Codazzi equations. The third compatibility equation is a statement that the determinant of the second fundamental form matrix is a function only of the metric and is therefore an intrinsic property of the surface. This equation may be written in several different forms. It is referred to as the Gauss equation because it proves the *Theorema Egregium* of Gauss, which states that Gaussian curvature is a function of only E, F, G and their derivatives.

(2) **Uniqueness.** If two surfaces S and S^* possess fundamental form matrices g_{ij} and b_{ij} and g^*_{ij} and b^*_{ij}, respectively, such that the following matrix equalities hold:

$$g_{ij} = g^*_{ij} \qquad b_{ij} = b^*_{ij},$$

then there exists an appropriate translation and rotation such that S and S^* coincide exactly.

This tells us that *arbitrary smooth surface shape is captured by six scalar functions*: g_{11}, g_{12}, g_{22}, b_{11}, b_{12}, b_{22}. We also refer to these as the E, F, G, L, M, N functions.

It is difficult to interpret what each of these functions are individually telling us about surface shape however. It would be advantageous if there were combinations of these functions that would give us easily interpretable surface shape characteristics. Fortunately, there are two curvature functions that combine the information in the six E, F, G, L, M, N functions in two different ways. These two curvature functions do not, in general, contain all the "information" contained in the six E, F, G, L, M, N functions, but they do contain a substantial amount of useful information, which we describe subsequently. However, if we use only two functions as surface characteristcs, we are using characteristics that may not uniquely determine surface shape in general. However, we note that for compact *convex* surfaces ($LN > M^2$ at every point), there is a *single scalar function* (the Gaussian curvature function $K(u, v)$) that uniquely specifies surface shape [11, 23, 41]. It can be shown that if simply-connected bounded regions of positive Gaussian curvature are isolated, surface shape is uniquely determined within those regions if the Gaussian curvature function of the surface is known. In Section 7.1, we also discuss conditions where the mean curvature function uniquely determines surface shape.

In conclusion, we have looked at general smooth 3-D surfaces and have identified six functions that uniquely characterize the shape of these geometric entities. This review was intended to motivate the following simple qualitative statement on which we base our work:

> *If one wants to characterize the shape of a geometric entity, such as a curve or surface, the characteristics that one choose should have a well-defined relationship to those functions that uniquely determine shape according to the mathematics of differential geometry.*

6. SURFACE CURVATURE

It is established that surfaces are uniquely characterized by six functions that completely determine surface shape and intrinsic surface geometry. These six functions are the independent elements of two 2 × 2 symmetric matrix functions of the surface. We now examine two curvature functions that combine the information from the six E, F, G, L, M, N functions.

We defined the shape operator (Weingarten mapping) matrix $[\beta]$ in the previous section as the matrix product $[g^{-1}][b]$. Hence, the $[\beta]$ matrix combines the first and second fundamental form matrices into one matrix. This matrix is a linear operator that maps vectors in the tangent plane to other vectors in the tangent plane at each point on a surface. The metric $[g]$ is the generalization of the speed of a planar curve whereas the shape operator $[\beta]$ is a generalization of the curvature of a planar curve. The *Gaussian curvature* function K of a surface can be defined from the first and second fundamental form matrices as the determinant of the shape operator matrix function as follows:

$$K = \det[\beta] = \det\left(\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1}\right)\det\left(\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}\right).$$

The *mean curvature* function of a surface can be defined similarly as half the trace of the shape operator matrix function as follows:

$$H = \tfrac{1}{2}\mathrm{tr}[\beta] = \tfrac{1}{2}\mathrm{tr}\left(\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}\right).$$

Hence, we see that these two different surface curvature functions are obtained by mapping the two fundamental form matrix functions into one scalar function. The surface curvature functions (H and K) are the natural algebraic invariants (characteristic polynomial coefficients) of the shape operator, which is the generalization of curvature of a planar curve. Since a 2 × 2 matrix only has two natural algebraic invariants (the trace and determinant), we see that these two surface curvature functions arise naturally in a detailed analysis of surface curvature.

There are other ways of looking at surface curvature based on the curves that lie in the surface. At each point on a surface, there is a direction of maximum normal curvature and a direction of minimum normal curvature for all space curves that (1) lie in the surface, (2) pass through that point, and (3) have (curve) normals that align with the surface normal at that point. If we let κ_1 be the maximum principal curvature (the maximum of the normal curvature function) and let κ_2 be the minimum principal curvature (the minimum of the normal curvature function), then one can compute the Gaussian and mean curvature in terms of these principal curvatures:

$$K = \kappa_1 \kappa_2 \qquad H = \frac{(\kappa_1 + \kappa_2)}{2}.$$

Note that κ_1 and κ_2 are the two roots of the quadratic equation:

$$\kappa^2 - 2H\kappa + K = 0.$$

Hence, if K and H are known at each point in a depth map, it is straightforward to determine the two principal curvatures:

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

If $H^2 = K$ at a surface point, the point is referred to as an *umbilic* point to denote that the principal curvatures are equal and every direction is a principal direction (i.e., the normal curvature function at an umbilic point is constant). A surface must be either flat or spherical in the neighborhood of an umbilic point.

The principal curvatures κ_1 and κ_2 are a perfectly valid pair of surface curvature descriptors, which are analytically equivalent to the mean and Gaussian curvature pair. *The principal curvatures are the two eigenvalues of the* 2×2 *matrix shape operator*. They are also the extrema of the normal curvature function. They specify the curvature of surface curves in the directions of maximal and minimal normal curvature at each point. We now compare the surface curvatures $\{H, K\}$ to the principal surface curvatures $\{\kappa_1, \kappa_2\}$:

(1) Principal curvatures are associated with certain directions whereas mean and Gaussian curvature are direction-free quantities.

(a)

		$\kappa 2$	
$\kappa 1$	−	0	+
−	Peak	Ridge	Saddle
0	Ridge	Flat	Valley
+	Saddle	Valley	Pit

(b)

		K	
H	+	0	−
−	Peak	Ridge	Saddle ridge
0	*(none)*	Flat	Minimal surface
+	Pit	Valley	Saddle valley

FIG. 5. Surface types determined by sign of surface curvatures: (a) table of surface shapes from principal curvature signs; (b) table of surface shapes from Gaussian (K) and mean (H) curvature signs.

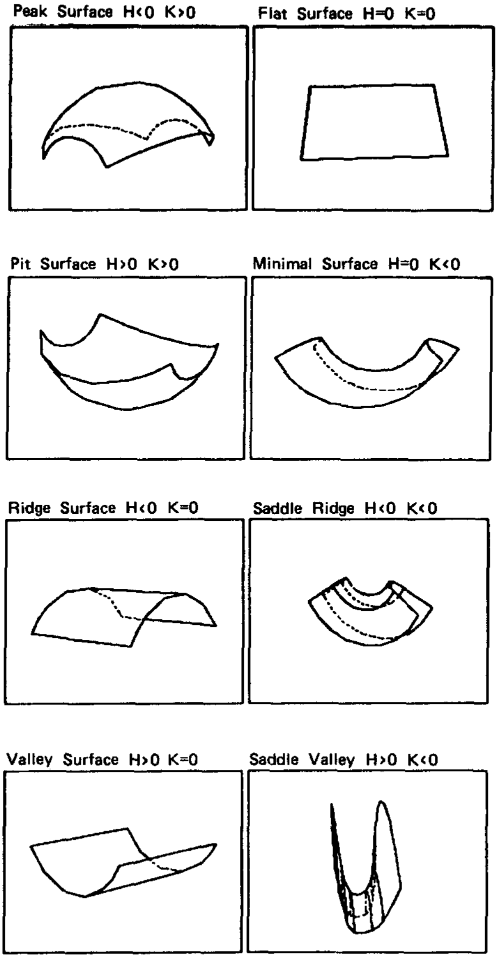

FIG. 6. Eight basic visible-invariant surface types.

(2) If only the signs of the principal curvatures are used to determine basic surface types, six surface types result: peak, pit, ridge, valley, flat, and saddle as shown in Fig. 5a. The signs of mean and Gaussian curvature yield *eight* basic surface types, as shown in Fig. 5b, because saddle surfaces can be resolved into saddle ridge, saddle valley, and minimal surfaces. Figure 6 shows the shapes of the eight surfaces. Note that K cannot be strictly positive when H is zero.

(3) Gaussian curvature exhibits isometric invariance properties. That is, Gaussian curvature is an intrinsic property of a surface. Both principal curvatures and the mean curvature are extrinsic properties of a surface. Isometric invariance and intrinsic and extrinsic properties are discussed in more detail later in this section.

(4) The mean curvature is the average of the principal curvatures. Therefore, it is slightly less sensitive to noise in numerical computations than the principal curvatures. Gaussian curvature is slightly more sensitive to noise.

(5) As mentioned earlier, the Gaussian curvature function of a convex surface uniquely determines the surface. A single principal curvature function does not permit a comparable theorem because of its directional nature.

(6) As discussed in Section 7, the mean curvature function of a graph surface taken together with the boundary curve of a graph surface uniquely determines the graph (Monge patch) surface from which it was computed. Range images are sampled graph surfaces. A single principal curvature function does not permit a comparable theorem because of its directional nature.

(7) It appears that a few more numerical computations are required to compute principal curvatures as compared to mean and Gaussian curvature.

To summarize, the pair $\{\kappa_1, \kappa_2\}$ contain the exact same surface curvature information as the pair $\{H, K\}$. There are slight advantages or disadvantages working with either pair depending on the application. Researchers have also worked with other pairs of surface curvatures, such as $\{\kappa_1, K\}$ [40]. The same surface curvature information is maintained. For our purposes, we have found that sign of the mean and Gaussian curvatures can be computed easily yielding the best classification of surface types in a range image.

The mathematical properties of the Gaussian curvature K and the mean curvature H are now discussed in more detail to stress the importance of these quantities to surface characterization and to give a better, more complete understanding of their qualities:

(1) Gaussian and mean curvature are invariant to arbitrary transformations of the (u, v)-parameters of a surface as long as the Jacobian of the (u, v)-transformation is always non-zero. In contrast, the six E, F, G, L, M, N functions all *vary* with (u, v)-transformations. This means that the E, F, G, L, M, N functions depend directly on the choice of the u, v coordinate system even though they uniquely characterize the 3-D shape of the surface. Therefore, it is not desirable to use these six functions as shape characteristics because of their dependence on parameterization.

(2) Gaussian and mean curvature are invariant to arbitrary rotations and translations of a surface. This is due to the fact that E, F, G, L, M, N are invariant to rotations and translations. This is clear from the definition of these functions and the properties of dot products and cross products. *Rotational and translational invariance* are extremely important properties for *view-independent* shape characteristics.

(3) Gaussian curvature is an *isometric invariant* of a surface. An isometric invariant is a surface property that depends only on the E, F, G functions(and possibly their derivatives). Consider that any surface S with Gaussian curvature K may be mapped to any other surface S^* with Gaussian curvature K^*. If the mapping is a distance-preserving (isometric) bijection, then $K = K^*$ at corresponding points on the two surfaces. An isometric mapping of surfaces is a continuous mapping where corresponding arcs on the surfaces have the same length.

(4) Isometric invariants are also referred to as intrinsic surface properties. Therefore, Gaussian curvature is an intrinsic surface quantity. Intrinsic properties have important interpretations. For example, the Gaussian curvature function K of

a surface does not "care" how the surface is embedded in a higher dimensional space. In contrast, the mean curvature function H does "care" about the embedding; it is an extrinsic surface quantity and is not an isometric invariant. The surface defined by a sheet of paper is readily used to demonstrate these ideas: If the paper lies flat on a desk top, we have $K = 0$ and $H = 0$ at each point on the sheet of paper. If we bend the paper making sure that no kinks occur, we still have $K = 0$ but now $H \neq 0$. When we bend the paper, we change how the surface is embedded in 3-D space, but we do not change the metric (intrinsic) properties of the surface. The within-surface distances between points on the paper remain the same, and the interior angles of a triangle still sum to π radians. In this example, Gaussian curvature is seen to be intrinsic whereas mean curvature is seen to be extrinsic. If the paper were deformed as if it were made of rubber, then Gaussian curvature would change as well as mean curvature. It should be clear that surface area is also an intrinsic surface property (can only depend on E, F, G).

(5) Another way of looking at intrinsic properties is that they do not change sign when the direction of the normal vector of the surface is reversed. Outward-pointing normals are usually chosen for surfaces of objects. If the surface is just an orientable surface patch floating in space, we could choose the normal to point in either direction. It should be clear that Gaussian curvature maintains its sign when the direction of the normal vector is flipped whereas mean curvature flips its sign. This is because the first fundamental form does not change sign while the second fundamental form does when the sign of the normal is flipped (see definitions above).

(6) Gaussian curvature indicates *surface shape* at individual surface points. When $K(u, v > 0$ at the surface point $\mathrm{x}(u, v)$, then the surface is *shaped like an ellipsoid* in the neighborhood of that point. When $K(u, v) < 0$, the surface is *locally saddle-shaped*. When $K(u, v) = 0$, the surface is locally flat, ridge-shaped, or valley-shaped. If $K = 0$ at every point on a surface, then that surface is referred to as a *developable* surface. Mean curvature also indicates surface shape at individual surface points when considered together with the Gaussian curvature. Figure 6 shows drawings of the eight basic surface shapes. If $H < 0$ and $K = 0$, the surface is locally ridge shaped. If $H > 0$ and $K = 0$, the surface is locally valley shaped. If $H = 0$ and $K = 0$, the surface is locally flat or planar. If $H < 0$ and $K > 0$, the surface is locally ellipsoidal and peaked (i.e., the surface bulges in the direction of the surface normal). If $H > 0$ and $K > 0$, the surface is locally ellipsoidal and cupped (i.e., the surface bulges in the direction opposite that of the surface normal). If $K > 0$, we can never have $H = 0$. When $K < 0$, $H \neq 0$ indicates if the saddle surface is predominantly valley shaped ($H > 0$) or ridge shaped ($H < 0$). When $H = 0$ at every point on a surface, then that surface is referred to as a *minimal* surface. Minimal surfaces have interesting mathematical properties. The minimal surface equation is studied in detail in many texts discussing partial differential equations.

(7) Gaussian and mean curvature are *local* surface properties. This allows surface curvature to be used in situations where *occlusion* is a problem because K and H do not depend on global properties of a surface.

(8) As a final note of comparison between H and K, a spherical surface of radius a has $H = \pm 1/a$ at every point on the surface where the sign depends on the

direction of the normal (outward or inward pointing) whereas $K = 1/a^2$ at every point *independent* of the direction of the normal vector. This also points out the dimensions of the curvature quantities and indicates how these quantities will change under 3-D scale transformations.

There are many other interesting properties of Gaussian and mean curvature, but the above list highlights the important ones for our needs. For example, one can use these facts in a surface matching algorithm as follows: Two surfaces that can be made to coincide exactly via a rotation and a translation are said to be *congruent*. Congruence implies that an isometry exists between the two surfaces *and* that the shape operators of the two surfaces are equivalent. If the two surfaces are congruent, then there exists a matching between the mean and Gaussian curvature values on the two surfaces. This implies that there is a matching between regions of constant sign of the mean and Gaussian curvatures on the two surfaces. Therefore, *if there does not exist a matching between regions of constant sign of the mean and Gaussian curvatures of the two surfaces, then the two surfaces are cannot have the same* 3-D *shape and therefore cannot be congruent.* Since a combined mean and Gaussian curvature sign image has only eight levels, it should be possible to easily discard surfaces that do not have similar shape.

Gaussian curvature and/or mean curvature can be defined and/or computed in several different ways:

(1) **Gauss Map Area Derivative Definition for Convex Surfaces.** In order to give this definition, we need first to describe the Gauss map, which is shown pictorially in Fig. 7. The Gauss mapping takes areas on surfaces to areas on the unit sphere. The unit surface normals at the surface points within the area ΔS on the surface are arranged in the unit sphere so that the tail of each normal vector is located at the sphere's center and the tip of the normal vector lies on the unit sphere's surface while preserving the direction of the normal vector. The surface area on the unit sphere (solid angle) subtended by these corresponding normal vectors is denoted ΔA. We can define the Gaussian curvature using the limit of the ratio of these two areas when the surface is convex:

$$K = \lim_{\Delta S \to 0} \frac{\Delta A}{\Delta S}.$$

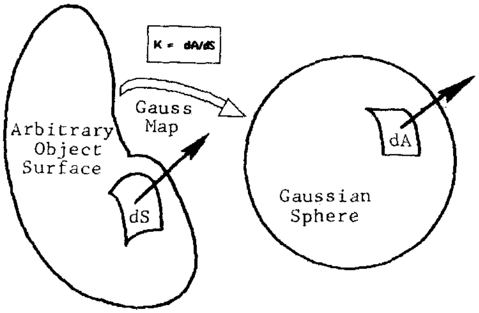

FIG. 7. Gaussian curvature = Gauss mapping derivative.

This popular definition seems to imply that K is a dimensionless quantity, which is not true of course. The quantity ΔA should be measured in solid angle units, such as steradians, rather than area units. This definition can be extended to handle non-convex surfaces, but one needs to be careful about how the solid angle on the unit sphere is computed.

(2) **Parallel Transport Definition** [42]. We start at a point P on a surface holding a vector that always points in the same direction in space (similar to a gyroscope). That direction is marked in a permanent fashion at the starting point. We go for a walk where we leave the point P and later return to it without crossing our path. Our path has enclosed an area we call ΔS. When we arrive back at P, we compare the direction of our vector with the reference direction that was marked when we left and obtain the angle $\Delta\alpha$. We can define the Gaussian curvature of the surface at the point P to be

$$K = \lim_{\Delta S \to 0} \frac{\Delta\alpha}{\Delta S}.$$

Note that the *sign* of K is given correctly here. There is a definite relationship between angles, area, and curvature on a surface.

(3) **Gauss Map Jacobian Definition.** This definition is closely related to the area derivative definition, but in this case we can define K in terms of the surface normal and u- and v-tangent vectors:

$$K = \frac{|\mathbf{n}_u \times \mathbf{n}_v|}{|\mathbf{x}_u \times \mathbf{x}_v|} \qquad \text{where } \mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{|\mathbf{x}_u \times \mathbf{x}_v|}.$$

(4) **Fundamental Form Matrix Coefficients Definitions.** These definitions of K and H are a restatement of the first matrix definitions given above, but may appear more familiar. We define $g = \det[g]$ and$b = \det[b]$:

$$K = \frac{b}{g} = \frac{b_{11}b_{22} - (b_{12})^2}{g_{11}g_{22} - (g_{12})^2} = \frac{LM - N^2}{EG - F^2}$$

$$H = \frac{g_{11}b_{22} + g_{22}b_{11} - 2g_{12}b_{12}}{2\left(g_{11}g_{22} - (g_{12})^2\right)} = \frac{EN + GL - 2FM}{2(EG - F^2)}.$$

(5) **Partial Derivative Expressions.** We can express K and H directly in terms of partial derivatives of the parameterization if desired. We introduce the triple-product notation $[\mathbf{a}\,\mathbf{b}\,\mathbf{c}] \equiv \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$ to simplify the expressions:

$$K = \frac{[\mathbf{x}_{uu}\mathbf{x}_u\mathbf{x}_v][\mathbf{x}_{vv}\mathbf{x}_u\mathbf{x}_v] - [\mathbf{x}_{uv}\mathbf{x}_u\mathbf{x}_v]^2}{|\mathbf{x}_u \times \mathbf{x}_v|^4}$$

$$H = \frac{(\mathbf{x}_v \cdot \mathbf{x}_v)[\mathbf{x}_{uu}\mathbf{x}_u\mathbf{x}_v] + (\mathbf{x}_u \cdot \mathbf{x}_u)[\mathbf{x}_{vv}\mathbf{x}_u\mathbf{x}_v] - 2(\mathbf{x}_u \cdot \mathbf{x}_v)[\mathbf{x}_{uv}\mathbf{x}_u\mathbf{x}_v]^2}{2|\mathbf{x}_u \times \mathbf{x}_v|^3.}$$

A few of the many different ways of looking at the mean and Gaussian curvature of a surface have been summarized. This list was intended to further stress the properties of these functions as shape descriptors and indicate how they are computed given a general surface parameterization. Note that the two curvature functions are both *nonlinear* combinations of all six E, F, G, L, M, N functions.

The most important invariance properties of surface curvature for view independent range image object recognition are the following: (1) invariance under changes in (u, v)-parameterization and (2) invariance under 3-D translations and 3-D rotations. In addition, we see that mean curvature information significantly complements Gaussian curvature information and vice versa in determining surface shape because H is extrinsic while K is intrinsic. Only *eight* basic local surface types are possible as discussed above, and these types are determined solely from the signs of the mean and Gaussian curvature. Note also that when K and H are considered together, we have a good generalization of the curvature function of space curves. There exist other functions of the E, F, G, L, M, N functions that are also useful, but we have attempted to show that the emphasis on K and H is reasonable. Henceforth, we consider only mean curvature and Gaussian curvature as surface curvature characteristics. Remember that we can compute principal curvatures, if needed, if we already know mean and Gaussian curvature. We have not proved that K and H are the "optimal" surface characteristics in any sense, but we have given substantial justification for their use as surface characteristics. Computation of surface characteristics is discussed in Section 7, but first we briefly review surface characterization research done by others.

6.1. *Literature Review*

Surface characterization and surface matching have been addressed in the literature by many [9, 14, 20, 22, 27, 34, 36, 40, 43, 48, 50] and are surveyed in [4]. Many different approaches have been used. One or both principal curvatures have often been used to characterize surface shape by finding surface edges. The approaches most related to our own technique are discussed here:

(1) Dreschler and Nagel [13] proposed the use of Gaussian curvature to find corner points in intensity images. They compute Gaussian curvature using the 5 × 5 window operators of Beaudet [3] to estimate the necessary partial derivatives. This is basically the same method that we use to compute the Gaussian curvature. Our emphasis, however, is on the collective use of different surface descriptors to provide surface characterization of range-image surfaces.

(2) Extended Gaussian images (EGIs) have been studied in detail by Horn and Ikeuchi *et al.* [22, 25]. This orientation histogram approach provides unique rotationally invariant shape description for convex objects [38, 41], but does not maintain this property for non-convex objects. This concept has been extended by Ikeuchi *et al.* [25] so that each discrete view of a non-convex objects has its own orientation histogram. The properties of Gaussian curvature are important to the EGI concept because the EGI is a discrete approximation of the Gaussian curvature function over all latitudes and longitudes on the unit sphere. This is discussed in detail in [22]. The EGI approach requires only surface orientation information. Hence, photometric stereo is used to obtain sensor data for input to the EGI algorithm, but depth information can also be used to provide such data. Object surface regions are assumed to be pre-segmented by another process.

(3) Faugeras [14] uses an approach that is similar to the EGI approach in its dependence on surface normal information and its assumption that object surface

regions have been pre-segmented by another process. Planar patches are isolated in range data using a region growing approach. Rotational matching on planar components of an object surface is performed using a quaternion-based algorithm that finds the best-fit rotation matrix to align a set of planar normal vectors. Mathematics for quadric surface set matching is also described.

(4) Medioni and Nevatia [40] present their shape description ideas in the context of differential geometry, but limit themselves to the zero crossings of the Gaussian curvature and the maximum principal curvature, and the maxima of the maximum principal curvature. This set of shape descriptors is a subset of the descriptors that we compute. Their derivative operators are primitive one-dimensional sums along a single row or column and are therefore more sensitive to noise than our approach.

(5) Brady *et al.* [9] use differential geometric features to describe surfaces, but they concentrate only on lines of curvature, asymptotes, bounding contours, surface intersections, and planar and spherical (umbilic) surface patches. By relying on surface curves to constrain surface shapes, the curvature primal sketch work of Asada and Brady [2] is used for planar curve shape description of surface curves. One problem is that lines of curvature are not necessarily planar. Also, while lines are necessary for most 3-D surface plots, the implicit premise that lines are necessary for a rich 3-D surface description does not necessarily follow. They compute principal curvatures and principal directions, but rely on an *ad hoc* scheme using a breadth-first search to link principal directions at each point into lines of curvature. They also propose a *surface primal sketch* that combines information on significant surface discontinuities.

(6) Nackman [43] discusses the use of critical point configuration graphs for surface characterization. He isolates four canonical types of slope districts of smooth surfaces. He suggests the use of curvature districts, but does not pursue it in his paper. Slope districts are bounded by the ridge and course (valley) lines that can be computed as the zero crossings of the first partial derivatives of a surface. This research does not address viewpoint-independent surface characterization.

(7) Marimont [39] identifies patterns of curvature sign changes as view-independent properties of *planar curves* in 3-D space. Our emphasis on sign of mean and Gaussian curvature generalizes these ideas for surfaces.

(8) The topographic primal sketch (TPS) proposed by Haralick *et al.* [20] labels each pixel of an intensity image surface with one of ten possible topographic labels: peak, pit, flat, ridge, ravine (valley), saddle, convex hillside, concave hillside, saddle hillside, or slope; or an edge (step-edge) label. Gradients, Hessians, and first and second directional derivatives are computed and used for labeling, but not in the context of surface differential geometry. That is, an intensity image function is viewed more as a function of two variables than as a movable geometric surface in the TPS approach. Our *HK*-sign surface categorization combined with critical point classifications is different than the TPS labeling in that more than four different types of hillsides are resolved. TPS quantities are proven to be invariant to monotonic grayscale transformations because this technique was originally intended for intensity images, not range images. However, all TPS labels are, in general, not invariant to even slight changes in viewpoint relative to a surface. The viewpoint

Surface-Type Pixel Labels for Digital Images (Range or Intensity)

HK-Sign + critical point labels	Topographic primal sketch label
Peak critical point pixel	→ Peak
Peak region pixel	→ Concave hillside
Ridge critical point pixel	→ Ridge
Ridge region pixel	→ Concave hillside
Saddle ridge critical point pixel	→ Saddle
Saddle ridge region pixel	→ Saddle hillside
Flat critical point pixel	→ Flat
Flat region pixel	→ Slope
Minimal critical point pixel	→ Saddle
Minimal region pixel	→ Saddle hillside
Saddle valley critical point pixel	→ Saddle
Saddle valley region pixel	→ Saddle hillside
Valley critical point pixel	→ Ravine (valley)
Valley region pixel	→ Convex hillside
Pit critical point pixel	→ Pit
Pit region pixel	→ Convex hillside

FIG. 8. TPS Pixel labels for HK-Sign + critical point labels.

dependent nature of the TPS labels is the key to understanding the difference between the TPS approach and our approach.

In an attempt to clarify the differences of these two similar approaches, we have constructed a list of the 16 possible pixel labels from our surface characterization technique discussed in Section 7. This list forms the left-hand column of the table in Fig. 8. The eight basic viewpoint-independent surface-type labels (see Fig. 6) are the most important pixel labels for our approach. We also compute the zero crossings of the first partial derivatives to find the critical points of a surface (discussed in Sect. 7). Critical point labels are *viewpoint-dependent* labels. If these labels are used to distinguish between non-critical points and critical points of a given surface type, we get the 16 labels mentioned above. In the table in Fig. 8, we list our label and the corresponding TPS label that it is mapped to. Hence, 16 labels are being mapped to 10 labels. However, the TPS labeling scheme incorporates additional viewpoint-dependent non-critical-point ridge (convex-roof-edge) labels, ravine or valley (concave-roof-edge) labels, and step-edge labels that we have not included in this comparison table. Figure 9 shows different types of edges that occur in range images. We have not attempted to include our edge labeling scheme into the above table

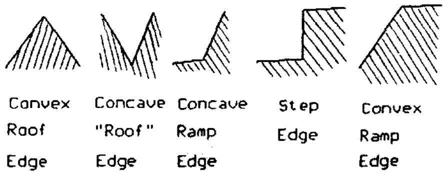

FIG. 9. Range image edge types.

because (1) edge types are viewpoint-dependent, (2) the table becomes too complicated when the five different edge types in Fig. 9 are also included, and (3) edges are boundaries of visible surface regions, and in our opinion, edge-type pixel labeling and surface-type pixel labeling are two distinctly different issues. To complete this detailed comparison, it is sufficient to note that (1) the TPS ridge and ravine labels include other pixels where the first directional derivative is zero besides just the critical points noted in the table in Fig. 8, and (2) the TPS step-edge label at the zero crossings of the second directional derivative is a key part of the TPS labeling scheme and is not mentioned in the table. We summarize this discussion by pointing out that although the TPS approach and our approach appear to be similar, (1) the properties of the labels under viewpoint transformations are quite different, and (2) the TPS mixes edge and surface labels together whereas the edge labeling process of our approach is a completely separate issue not included in this comparison.

7. RANGE IMAGE COMPUTATIONS

7.1. *Surface Curvature from Partial Derivatives*

We have seen that 3-D surface shape is well characterized by two scalar functions, Gaussian curvature and mean curvature, which are independent of parameterization and invariant to rotations and translations. Given a "2.5-D" range image with only discrete data, how can we compute surface curvature? To compute surface curvature of range images, we need estimates of the first and second partial derivatives of the depth map. In order to see this, we simplify the expressions for K and H for graph (Monge patch) surfaces because all range images are sampled graph surfaces.

First, we recall that the parameterization for a graph surface takes a very simple form: $\mathbf{x}(u, v) = [u \ \ v \ \ f(u, v)]^{\mathrm{T}}$. The T superscript indicates transpose so that $\mathbf{x}$ is column vector. This yields the following formulas for the surface partial derivatives and the surface normal:

$$\mathbf{x}_u = [1 \ \ 0 \ \ f_u]^{\mathrm{T}} \qquad \mathbf{x}_v = [0 \ \ 1 \ \ f_v]^{\mathrm{T}}$$

$$\mathbf{x}_{uu} = [0 \ \ 0 \ \ f_{uu}]^{\mathrm{T}} \qquad \mathbf{x}_{vv} = [0 \ \ 0 \ \ f_{vv}]^{\mathrm{T}} \qquad \mathbf{x}_{uv} = [0 \ \ 0 \ \ f_{uv}]^{\mathrm{T}}$$

$$\mathbf{n} = \frac{1}{\sqrt{1 + f_u^2 + f_v^2}}[-f_u \ \ -f_v \ \ 1]^{\mathrm{T}}.$$

These vectors are combined using the dot product definitions given earlier to form the six fundamental form coefficients:

$$g_{11} = 1 + f_u^2 \qquad g_{22} = 1 + f_v^2 \qquad g_{12} = f_u f_v$$

$$b_{11} = \frac{f_{uu}}{\sqrt{1 + f_u^2 + f_v^2}} \qquad b_{12} = \frac{f_{uv}}{\sqrt{1 + f_u^2 + f_v^2}} \qquad b_{22} = \frac{f_{vv}}{\sqrt{1 + f_u^2 + f_v^2.}}$$

Hence, we see that the five partial derivatives $f_u, f_v, f_{uu}, f_{uv}, f_{vv}$ are all we need to compute the six fundamental form coefficient functions for a graph surface.

Next, we recall that we can compute Gaussian curvature as the ratio of the determinants of the two fundamental form matrices. That expression is written directly in terms of the depth-map function (graph surface) derivatives as follows:

$$K = \frac{f_{uu}f_{vv} - f_{uv}^2}{\left(1 + f_u^2 + f_v^2\right)^2} = \frac{\det(\nabla \nabla^{\mathrm{T}} f)}{|\nabla f|^4};$$

∇ is the two-dimensional (u, v) gradient operator, and $\nabla \nabla^{\mathrm{T}}$ is the Hessian matrix operator. Hence, if we are given a depth-map function (u, v) that possesses first and second partial derivatives, we can compute the Gaussian curvature directly.

We also recall that we can compute mean curvature as half the trace of the shape operator. This expression can also be written directly in terms of the depth-map function derivatives as follows:

$$2H = \frac{f_{uu} + f_{vv} + f_{uu}f_v^2 + f_{vv}f_u^2 - 2f_uf_vf_{uv}}{\left(1 + f_u^2 + f_v^2\right)^{3/2}} = \nabla \cdot \left(\frac{\nabla f}{\sqrt{1 + |\nabla f|^2}} \right).$$

$(\nabla\ \cdot)$ is the divergence operator of vector calculus. Again, if we are given a depth map function $f(u, v)$ that possesses first and second partial derivatives, we can compute the mean curvature directly. It is important to note that the above equation (where H is known but f is not) is a second-order elliptic quasilinear partial differential equation. If D is a subset of $\mathbf{R}^2$ and H is an arbitrary function of two variables with continuous first partial derivatives defined over D, it is not possible to say whether or not a solution to the differential equation above even exists. By imposing certain restrictions, it is sometimes possible to prove the existence and the uniqueness of solutions. For example, Guisti [18] has proven that under certain extremal conditions, H alone *without Dirichlet boundary conditions* can *uniquely* determine f up to an additive constant (translations in depth). Also, Gilbarg and Trudinger [17] show that, under a set of certain other conditions (which include the restriction that the *boundary curve's* curvature must be greater than or equal to the absolute value of the sum of the principal curvatures of the surface at the boundary of the domain), then there exists a unique solution f to the Dirichlet boundary value problem defined by H plus the function f restricted to the boundary of the region D. However, there is separate uniqueness theorem [17] (that does not address existence) that states that if (1) H is continuously differentiable, (2) f_1 and f_2 are both solutions to the partial differential equation above in D, and (3) $f_1 = f_2$ on the boundary of the domain D, then $f_1 = f_2$ throughout that domain. In this sense, a smooth surface function $f(u, v)$ defined over a compact domain D with a simple closed contour boundary ∂D is essentially *equivalent* to that surface's mean curvature function H taken together with the boundary curve of the surface f restricted to ∂D. Hence, H plus f on ∂D constitute an ideal type of surface characteristic; all "information" present in the original smooth depth map function is maintained in the characteristic data. Given $f(u, v)$, one can compute $H(u, v)$, and, in theory, the Dirichlet problem can be solved to reproduce $f(u, v)$. The mean curvature surface characteristic is valuable to an object recognition algorithm because of its invariance properties. We stress that this property of the mean curvature is only valid for graph surfaces, but since all range images (and all intensity images) are sampled graph surfaces, this is an extremely important concept for digital surface characterization.

Principal curvatures do not have similar uniqueness theorems because of their directional nature.

Since any sampled depth map function encountered in practice may be approximated arbitrarily well by a sufficiently smooth function that possesses first and second partial derivatives, our next problem is how to compute estimates of these partial derivatives given the sampled data.

7.2. Estimating Partial Derivatives of Depth Maps

In general, direct numerical differentiation is discouraged if a problem can be addressed using other means. The basic approach for our current method is the following: (1) given discrete sample data, determine a continuous differentiable function that "best" fits the data, and (2) compute the derivatives of the continuous function analytically and evaluate them at the corresponding discrete points. Ideally, it might be desirable to fit all data with one smooth surface. This problem is computationally intensive and should only be used as a last resort if no simpler methods work. Instead, we compute only a local surface fit within each $N \times N$ window of discrete depth map surface data. Our experimental results show that this approach is adequate. This method uses a local quadratic surface model that is discussed in detail in several papers [3, 5, 7, 19]. For this reason, we state only the final results required to implement our approach.

Each data point in a given $N \times N$ window is associated with a position (u, v) from the set $U \times U$ where N is odd:

$$U = \left\{ \frac{-(N-1)}{2}, \dots, -1, 0, 1, \dots, \frac{(N-1)}{2} \right\}.$$

The following discrete orthogonal polynomials provide the quadratic surface fit:

$$\phi(u) = 1 \qquad \phi_1(u) = u \qquad \phi_2(u) = \left(u^2 - \frac{M(M+1)}{3} \right)$$

where $M = (N-1)/2$. The $b_i(u)$ functions are normalized versions of the orthogonal polymomials:

$$b_0(u) = \frac{1}{N} \qquad b_1(u) = \frac{3}{M(M+1)(2M+1)} u$$

$$b_2(u) = \frac{1}{P(M)} \left(u^2 - \frac{M(M+1)}{3} \right)$$

where $P(M)$ is a fifth-order polynomial in M:

$$P(M) = \tfrac{8}{45}M^5 + \tfrac{4}{9}M^4 + \tfrac{2}{9}M^3 - \tfrac{1}{9}M^2 - \tfrac{1}{15}M.$$

The recipe for computing derivatives at a sample point using odd size data windows is simple since the $b_i(u)$ vectors are precomputed and stored for any given window size. We can obtain a surface function estimate $\hat{f}(u, v)$ of the form

$$\hat{f}(u,v) = \sum_{i,j=0}^{2} a_{ij}\phi_i(u)\phi_j(v)$$

that minimizes the mean square error term

$$\varepsilon = \sum_{(u,v)\in U^2} (f(u,v) - \hat{f}(u,v))^2.$$

The solution for the unknown coefficients is given by

$$a_{ij} = \sum_{(u,v)\in U^2} f(u,v)b_i(u)b_j(v).$$

The first and second partial derivative estimates are then given by

$$f_u = a_{10} \qquad f_v = a_{01} \qquad f_{uv} = a_{11} \qquad f_{uu} = 2a_{20} \qquad f_{vv} = 2a_{02}.$$

The fit error is computed after the a_{ij} coefficients are determined:

$$\varepsilon = \sum_{(u,v)\in U^2} f^2(u,v) - \sum_{i,j} a_{ij}\left(\sum_u \phi_i^2(u)\right)\left(\sum_v \phi_j^2(v)\right).$$

Since the discrete orthogonal quadratic polynomials over the 2-D window are separable in u and v as shown in the above equations, we can compute our partial derivative estimates for an entire depth map using a separable convolution operator. This is quite efficient. These derivative estimates can then be plugged into the equations for the Gaussian curvature and the mean curvature. This describes all the mathematical details necessary to compute curvature functions $K(u, v)$ and $H(u, v)$ given samples from a continuous depth-map function $f(u, v)$.

The disadvantages of this local surface fit method are:

(1) A different quadratic surface is fitted to the neighborhood of each point. No compatibility constraints are imposed on these surfaces so that the net continuous surface interpretation is meaningful. Unfortunately, one may need to make *a priori* assumptions about the surface in order to correct this, and making these kind of assumptions is contrary to our goal of using as few *a priori* assumptions as possible in our data-driven processing.

(2) It seems counter-intuitive that all columns (or rows) in a least squares derivative window operator are weighted equally. If one asks for a 9 × 9 window least squares estimate of the first derivative of a depth map at a particular pixel, the data that runs four pixels away has the same impact on the final estimate as does the data that runs directly through the pixel at which we are estimating the derivative. This situation can be modified using weighted least squares techniques. The question then arises: What is the best assignment of weights? A particular triangular weight assignment was quickly tried on a few particular depth maps in experiments and was found to give different, but neither better nor worse results. One might argue that

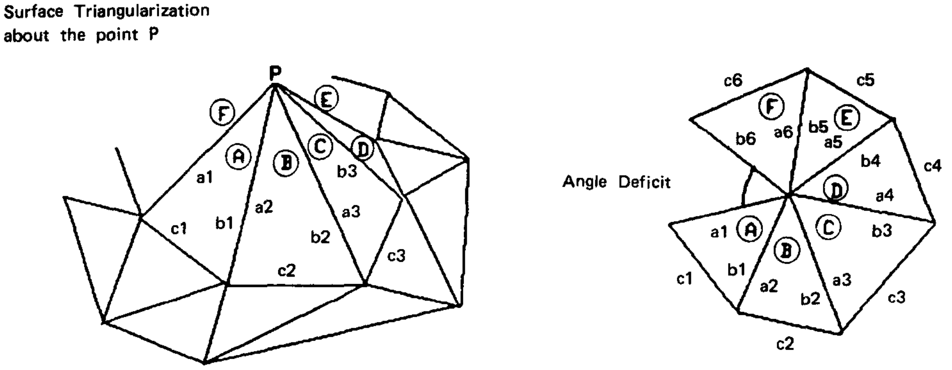

FIG. 10. Discrete Gaussian curvature at a point.

Gaussian weights should be used, but this is not likely to change our results too much. Haralick [21] is addressing this issue.

7.3. *Computing Gaussian Curvature without Partial Derivatives*

It is interesting to note that there is at least one way to compute Gaussian curvature without using explicit derivative estimates [36]. This technique originates in the Regge calculus of general relativity in which geometry is analyzed without coordinates. One must first obtain a discrete triangularization of a surface to use this technique. If we look at a particular point $\mathbf{x}_k$ on the triangularized surface, it is the vertex for N different triangles. See Fig. 10 for an example of the geometry. We assume that the lengths of sides of the ith triangle are a_i, b_i, c_i where c_i is the length of the side opposite the point of interest and where $a_i + 1 = b_i$. The angle deficit Δ_k at the point $\mathbf{x}_k$ is then given by

$$\Delta_k = 2\pi - \sum_{i=1}^{N} \phi_i \qquad \text{where } \phi_i = \cos^{-1}\left(\frac{a_i^2 + b_i^2 - c_i^2}{2a_i b_i} \right).$$

Gaussian curvature at a point is computed based on angle deficit as

$$K(\mathbf{x}) = \frac{2\Delta_k \cdot \delta(\mathbf{x} - \mathbf{x}_k)}{\left(\sum_{i=1}^{N} A_i \right)}$$

where

$$A_i = \sqrt{s(s - a_i)(s - b_i)(s - c_i)} \qquad \text{and} \qquad s = \frac{a_i + b_i + c_i}{2}$$

and where $\delta(\cdot)$ is the Dirac delta function. This method is related to our parallel transport definition of Gaussian curvature mentioned earlier. We have obtained amazingly accurate estimates of the Gaussian curvature of a sphere using only five points on the surface of a hemisphere. However, we still prefer our current method over this method because our method determines Gaussian curvature in addition to many other characteristics with comparable results and comparable noise sensitivity. It is impossible to compute mean curvature with a similar approach because of its extrinsic nature.

7.4. Critical Points

The use of surface critical points for surface characterization has been discussed by Nackman [43]. He notes that critical points and ridge and course (valley) lines surround slope districts in only four canonical ways. The critical points of a function $f(u, v)$ are those points (u, v), where $f_u(u, v) = 0$ and $f_v(u, v) = 0$. Since we have to compute f_u and f_v functions to compute K and H anyway, it is a trivial mathematical step to additionally determine the critical points of the given depth map by detecting the zero crossings of the first partial derivatives. For surfaces, there are seven kinds of *non-degenerate* critical points where $f_u = f_v = 0$ and $K \neq 0 \neq H$:

(1) peak critical points: $H < 0$ and $K > 0$,
(2) ridge critical points: $H < 0$ and $K = 0$,
(3) saddle ridge critical points: $H < 0$ and $K < 0$,
(4) minimal critical points: $H = 0$ and $K < 0$,
(5) saddle valley critical points: $H > 0$ and $K < 0$,
(6) valley critical points: $H > 0$ and $K = 0$, and
(7) pit critical points: $H > 0$ and $K > 0$.

In addition, there is one kind of *degenerate* critical point where $f_u = f_v = 0$ and $K = 0$ and $H = 0$: planar critical points. Hence, if we compute the zero crossings of the first partial derivatives in addition to Gaussian and mean curvature, then we have a richer structural description of the range image surface.

The proposed characterization is a generalization of the 1-D function characterization techniques. Computing critical points is the generalization of computing the zeros of the first derivative of a function of one variable. Computing the sign of Gaussian and mean curvature is a generalization of computing the sign of the second derivative to see if the function is concave up or down.

7.5. Other Features

It is also convenient to compute four other quantities that are of interest in surface characterization and range image segmentation. The first of these quantities is the square root of the determinant of the first fundamental form matrix:

$$\sqrt{g} = \sqrt{EG - F^2} = \sqrt{1 + f_u^2 + f_v^2}.$$

This *metric determinant* quantity can be summed over depth map regions to obtain the approximate surface area of the region. This summation corresponds to the continuous formulation

$$\text{Surface Area} = \iint \sqrt{1 + f_u^2 + f_v^2}\, du\, dv.$$

It can also be considered as an edge magnitude map since it is approximately equal to the square root of the sum of the squares of the first partial derivatives. This output is similar to the output of many edge detectors. It could be thresholded using a minimum depth separation distance to create a simple binary edge image. In depth maps, these edges generally correspond to depth discontinuities, which generally correspond to the occluding boundary of an object. Thus, the existence of a depth discontinuity and a surface region boundary along a curve will reinforce the interpretation of that curve as an occluding object boundary for segmentation purposes.

A second extrinsic quantity that is easy to compute given the derivatives already computed is the *quadratic variation*

$$Q = f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2.$$

When this function is integrated (summed) over a depth map region, the integral (sum) is a measure of the *flatness* of that region. This image function and the metric determinant image could be computed in parallel with the Gaussian and mean curvature using the computed derivative information and could be used to quickly provide surface area and flatness measures of the surface regions segmented later in the processing.

A third (intrinsic) quantity is the *coordinate angle function* Θ, which is defined as

$$\Theta = \cos^{-1}\left(F/\sqrt{EG}\right) = \cos^{-1}\left(\frac{f_u f_v}{\sqrt{1 + f_u^2 + f_v^2 + f_u^2 f_v^2}}\right).$$

This function measures the non-orthogonality of the u, v parameterization at each point: $\Theta = \pi/2$ when the u- and v-tangent vectors are orthogonal and ranges between 0 and π when they are not orthogonal. Also $\cos\Theta = 0$ implies that at least one of the first partial derivatives is zero in the graph surface formulation. It is not clear how this intrinsic surface feature can contribute in general to segmentation. However, the zeros of this function form the ridge and course lines discussed by Nackman [43].

The last quantities that we mention are the *principal directions* of the surface at each point. The principal direction vectors of the surface along with H and K completely determine the shape operator of the surface. We can compute the principal direction angles in the u–v plane as follows:

$$\Phi_{1,2} = \tan^{-1}\left(\frac{-B \pm \sqrt{B^2 - AC}}{C}\right)$$

where

$$A = EM - FL \qquad 2B = EN - GL \qquad C = FN - GM.$$

Note that these directions are in general *not orthogonal in the* (u, v) *plane* even though the 3-D principal direction vectors in the tangent planes are orthogonal. We do not currently propose to use these angles as surface descriptors as Brady *et al.* [9] do because they appear to be rather noisy to us. Nethertheless, we have included them in the discussion for completeness of the continuous surface description. We conjecture that the H, K, g, Θ, Φ_1, Φ_2 function description of a surface is equivalent to the E, F, G, L, M, N function description as related to the fundamental existence and uniqueness theorem of general surfaces. The main difficulty in proving this conjecture lies in the complexity of the differential equation for K in terms of E, F, G and their derivatives. If this conjecture is true, there would be an interesting split between angular and non-angular functions that describe a surface.

7.6. *Summary of Computational Approach*

We summarize the computational characterization process prescribed above in the continuous and discrete cases:

(1) **Continuous Case.** *Input.* A function $f(u, v)$ defined on a subset of the plane.

Process. (a) Compute f_u, f_v, f_{uv}, f_{uu}, f_{vv} using analytical techniques,
(b) Compute K, H, $\sqrt{g}$, $\cos\Theta$, and Q using the formulas given above,
(c) Compute zeros of f_u, f_v, K, and H.

Output. (a) Two three-level functions $\operatorname{sgn}(K)$ and $\operatorname{sgn}(H)$ where sgn( ) is the signum function that yields 1 if the argument is positive, 0 if the argument is zero, and -1 if the argument is negative. These two functions can be combined into one eight-level function.

(b) Three non-negative functions $|H|$, $|K|$, and $\sqrt{H^2 - K}$ that describe the magnitude of the two surface curvatures and the magnitude of the difference of the principal curvatures, respectively.

(c) A binary $c(u, v)$ that is 1 when (u, v) is critical and 0 when it not. Peak, pit, saddle ridge, saddle valley, and minimal critical points of smooth surfaces are always isolated critical points. Ridge and valley critical points can form planar curves. Planar critical points can form planar areas. Each resulting point, curve, or area should be labeled with its appropriate classification.

(d) Two non-negative functions $\sqrt{g}$ and Q. These functions are integrated in later processing to provide descriptive features of segmented surface regions.

(e) The binary image functions denoting the zeros of K, H, and $\cos\Theta$.

(2) **Discrete case.** *Input.* A matrix of values $\hat{f}(i, j)$ where $0 \leq i \leq (N_u - 1)$ and $0 \leq j \leq (N_v - 1)$ and $0 \leq \hat{f} \leq 2^{N_{\text{bits}}} - 1$ where N_{bits} is the number of bits used for sensor data quantization.

Process. (a) Compute $\hat{f}_u$, $\hat{f}_v$, $\hat{f}_{uv}$, $\hat{f}_{uu}$, $\hat{f}_{vv}$ matrices using window convolution techniques described above,
(b) Compute K, H, $\sqrt{g}$, $\cos\Theta$, Q, and ε matrices using the analytical formulas given above,
(c) Compute the zeros of $\hat{f}_u$, $\hat{f}_v$, K, and H.

Output. (a) Two three-level images, or matrices, $\operatorname{sgn}(K)$ and $\operatorname{sgn}(H)$. These two images can be combined into one eight-level image.

(b) Three non-negative images $|H|$, $|K|$, and $\sqrt{H^2 - K}$ that describe the magnitude of the two surface curvatures and the magnitude of the difference of the principal curvatures, respectively.

(c) A binary image matrix $c(i, j)$ that is 1 when (i, j) is critical and 0 when it is not. Each critical pixel is classified into one of the eight categories listed above using the corresponding S-curvature values.

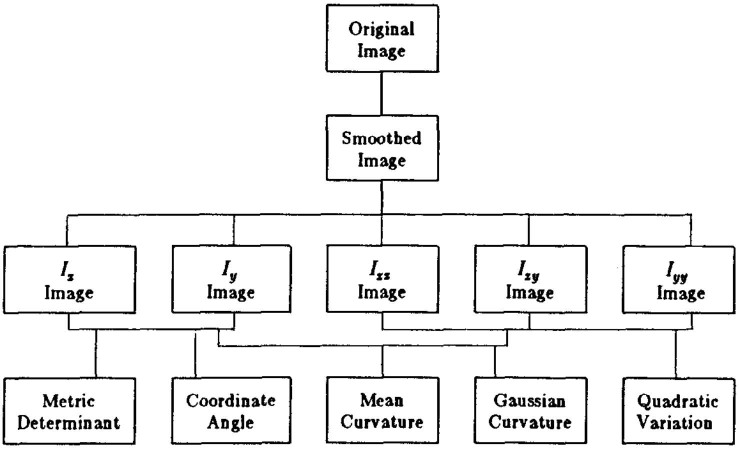

FIG. 11. Range image processing algorithm structure.

(d) Three non-negative images $\sqrt{g}$, Q, and ε are useful for computing region features, such as surface area and flatness. Fit error indicates the reliability of the partial derivative estimates.

(e) The binary images denoting the zeros of K, H, and $\cos\Theta$.

Note that we can "compress" much useful structural surface information about an N_{bits}- digitized depth map into eight levels (*only three bits*) if we use the signs of the Gaussian and mean curvature. This second-order sign information *substantially constrains* the possibilities of visible surfaces. Also, the sign of a second-order quantity computed from digital sensor data is more reliable than the magnitude. In addition, we can create a classified list of critical points that constrains the surface. This list normally contains only a small number of points compared to the total number of pixels in the range image and could be used as starting points for a matching algorithm as described in [5]. The other images provide additional useful, overlapping information.

8. EXPERIMENTAL RESULTS

A range image processing program has been written (in C) to do the surface characterization computations. The potentially parallel computational structure of this program is shown in Fig. 11. All five derivative images could be computed simultaneously after initial smoothing. Subsequently, all surface characteristics could be computed simultaneously after the derivative estimation stage. Our software currently accepts a square range image (with 8-bits of depth) as input and generates the following images as output:

(1) smoothed range image: f_{smooth};

(2) square root of metric determinant (edge magnitude) image: $\sqrt{g}$;

(3) quadratic variation (flatness measure) image: Q;

(4) local quadratic surface fit error image: ε;

(5) zeros of the mean curvature: $H = 0$;

(6) zeros of the Gaussian curvature: $K = 0$;
(7) zeros of the cosine-of-the-coordinate-angle function: $\cos \Theta = 0$;
(8) cosine-of-the-coordinate-angle function: $\cos \Theta$;
(9) sign regions of mean curvature: $\operatorname{sgn}(H)$;
(10) sign regions of Gaussian curvature: $\operatorname{sgn}(K)$;
(11) magnitude of principal curvatures difference: $\sqrt{H^2 - K}$;
(12) principal direction angle: Φ_1;
(13) magnitude of Gaussian curvature: $|K|$;
(14) magnitude of mean curvature: $|H|$;
(15) non-degenerate critical points image: $f_u = f_v = 0 \neq Q$; and
(16) critical points image: $f_u = f_v = 0$.

This output data characterizes the input depth map in a way that is quite useful both for segmentation of sensor data and recognition of objects. We get a decomposition of a range image into eight surface-type regions by using a combination of the $\operatorname{sgn}(H)$ and $\operatorname{sgn}(K)$ images. We can get step edges and surface area from the $\sqrt{g}$ image. It is possible to detect roof edges and ramp edges using the $|H|$ and $\sqrt{H^2 - K}$ images. Critical point configurations describe surfaces as discussed in Nackman [43]. Data-driven processing can yield rich, interrelated surface, edge, and point information.

In the experimental results that follow, several computational steps were performed that have not been mentioned yet. The following points should be made:

(1) Our original depth map data is quantized to eight bits. Quantization noise alone caused many problems in our first tests on analytically computed surfaces. To fix this problem, the original image is smoothed using a window operator that is two pixels larger than the window operators used to do the derivative estimation, and the results are stored in floating point form. This smoothing also compensates for random noise in the two or three least significant bits of range data.

(2) The output curvature images are smoothed using the same smoothing operator that was used on the input to even out the variations in the S-curvature output.

(3) The S-curvature sign images are obtained using a threshold about zero. That is, $\operatorname{sgn}(K) = 0$ if $|K| < \varepsilon_K |K|_{\max}$. Also, $\operatorname{sgn}(H) = 0$ if $|H| < \varepsilon_H |H|_{\max}$. The two thresholds were set to 1% for synthetic range images with no noise. This seems like a fairly reasonable number because the original 8-bit data can only measure zero to within 0.4% of the maximum value. Noisy images required larger percentage thresholds to obtain good HK-sign images.

These items are critical to the results displayed in this paper. Different smoothing schemes will create different S-curvature results. Different thresholds create different S-curvature sign images. Our future research will analyze and experiment with different smoothing and threshold-setting alternatives.

Experimental results for different object depth maps are shown in Figs. 13 through 28. Each depth map is briefly discussed individually below. The results are shown in the sixteen subimage format. The contents of each subimage are noted in Fig. 12.

f_{smooth}	$\sqrt{g}$	Q	ε
zeros(H)	zeros(K)	zeros($\cos \Theta$)	$\cos \Theta$
sgn(H)	sgn(K)	$\sqrt{H^2 - K}$	Φ_1
$\lvert H \rvert$	$\lvert K \rvert$	$Q \neq 0$ $\nabla f = 0$	$\nabla f = 0$

FIG. 12. Surface characterization results format: f = surface function; H = mean curvature; K = Gaussian curvature; g = metric determinant; Q = quadratic variation; ε = local surface fit error; Θ = coordinate angle; Φ_1 = principal direction angle.

Zeros images are white when the quantity is zero and black otherwise. Surface curvature sign images are coded as follows: *white for positive*, *gray for zero*, *black for negative*. Other images are scaled so that the image *minimum is black* and the image *maximum is white*. The exception to this rule is the depth map itself: *white* is used for pixels *closest* to the observer (depth is a minimum), whereas *black* is used for pixels *farthest* from the observer (depth is a maximum). This convention is sometimes reversed; our experience is that it is generally more difficult to interpret such reversed range images. There are odd-looking convolution-window effects near the edges of these 16 subimages; please ignore these artifacts. For several range images, we also provide a surface plot of the range data.

We obtained the range images used in this paper in two different ways. We can generate synthetic range images of arbitrary 3-D object models from arbitrary views using a combination of the SDRC/GEOMOD solid modeler [16] developed by Structural Dynamics Research Corporation (which is used to create object models) and our own software (which uses a depth-buffer algorithm to create the depth maps.) We have also obtained real range images from the Environmental Research Institute of Michigan (ERIM), which were obtained using the ERIM laser rangefinder [53]. The range-image points in these images are obtained using equal-increment azimuth and elevation *angle* sampling. This equal-angle-increment sampling causes flat surfaces in the real world to be mapped into slightly warped surfaces in range images.

We now consider the results for each object individually. Our first object is a coffee cup. Two gray scale images of two depth maps of this object are shown in Fig. 13 along with a surface plot of one of them. The two depth maps were obtained from the ERIM laser rangefinder. The quality of these depth maps is almost comparable to what we obtain from our model-based synthetic depth map generation program. The surface curvature characterization of these depth maps are shown in Fig. 14; a 7 × 7 derivative window operator was used. We find that the *zeros of the mean curvature* form a good line drawing of the object shape. The square root of the metric determinant, the quadratic variation, and the quadratic surface fit error provide an interesting sequence of edge-detector-like images. Clusters of local maxima and saddle ridge critical points are found at the closer and farther rims of the cup respectively whereas local minima critical points are found on the inside of the cup. Despite the noise present in this real image, very few spurious critical points are found. The magnitude of the principal curvature difference image shows that the principal curvatures on the surface differ the most on the rim of the cup.

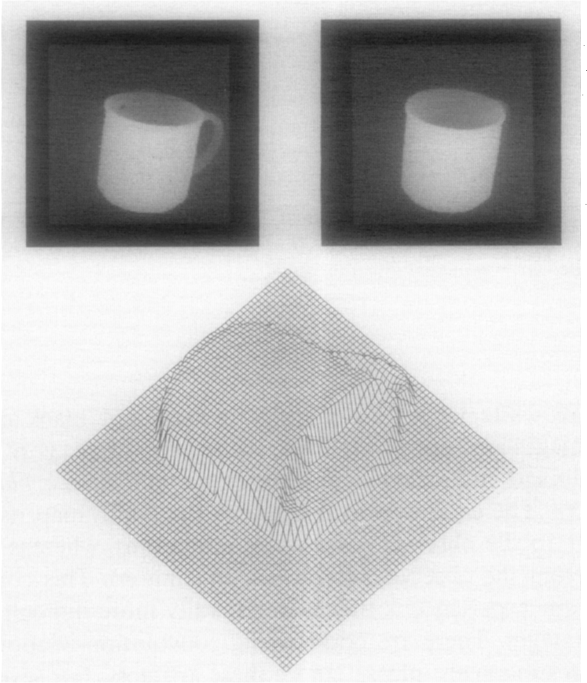

FIG. 13. Coffee cup range images and surface plot (128 × 128 ERIM range images).

A second depth map from the laser rangefinder is shown in Fig. 15 along with a surface plot of the same data. A histogram of this image shows that all 8-bit depth values are confined to the 32 to 128 range with most values in the 96 to 128 range. This image represents a portion of a keyboard. Two S-curvature characterizations are shown in Fig. 16. The top characterization was computed using a 3×3 derivative window operator and the bottom was computed using a 5×5 window. The reader can easily see the effect of an increase in window size. The concave shape of the top surface of the keyboard keys is detected by the small white regions in the mean curvature sign image. Again the zeros of the mean curvature image yields a very good line drawing of the keyboard. There are a large number of critical points on this surface as one might expect. The critical points are fairly well clustered into groups for the 5×5 window operator case.

We now discuss two views of a road scene extracted from a range image sequence acquired by the laser rangefinder. Figure 17 shows the original range images (with phase wraparound lines at 32 and 64 ft), the processed range images with the first wraparound transition removed, and a surface plot of one of the processed images. The 64 ft line was not removed because most of the data beyond that level is excessively noisy. Figure 18 shows the S-curvature characterization results for a 9×9 derivative window operator. The zeros of the mean curvature effectively isolate the ditches at the side of the road. The mean curvature sign image points out that the surface corresponding to the road itself is not flat in this image as expected from the angular sampling. The zeros of the cosine-of-the-coordinate-angle occur whenever either of the first partial derivatives is zero. Because of the equal-angle-increment sampling, the flat road samples are slightly warped yielding a line right up

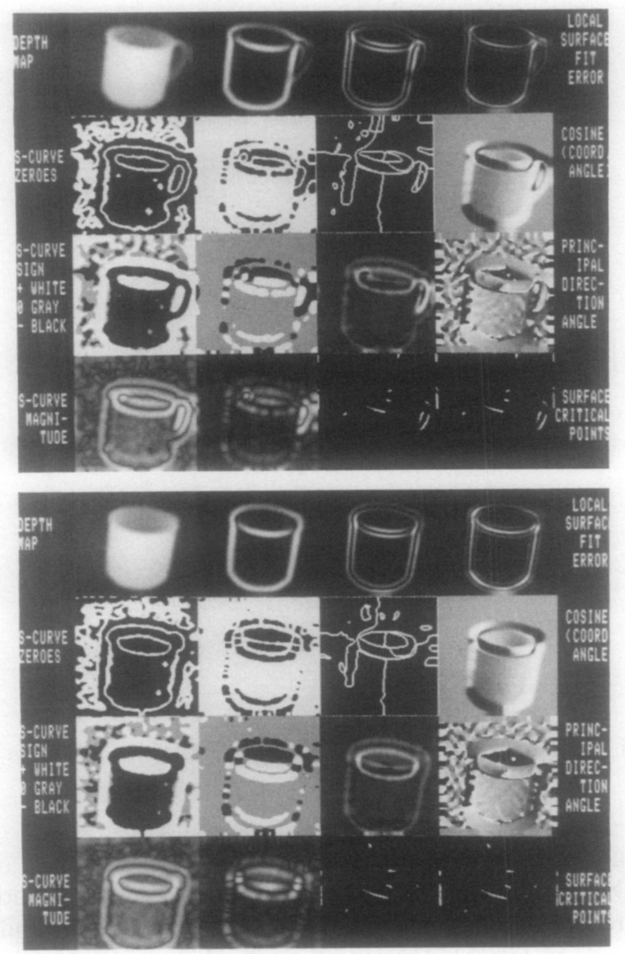

FIG. 14. S-curvature characterizations of two views of coffee cup (7 × 7 derivative window operator, zero threshold = 4%).

the center of the cos Θ zeros image. Because the noise beyond the 64 ft line was left in the image unwrapped, the maximum curvature points and the critical points all occur in this region but have no physical meaning.

Figure 19 is a surface plot of the range image of a tilted torus. The S-curvature results for two different synthetic range views of the torus are shown in Fig. 20. In the other view, the torus is tilted only five degrees. Note how well the surface critical points were detected. The structure of the ridge and course lines in the zeros of the cosine-of-the-coordinate-angle image gives important view-dependent information about the surface in terms of slope districts. The irregularities in the curvature magnitude images occur because the object model from which the range image was generated is a faceted polyhedral model. Note that the mean-curvature sign-image is almost exactly correct. The Gaussian curvature sign image shows that, to within the 1% threshold, many parts of the surface are approximately flat.

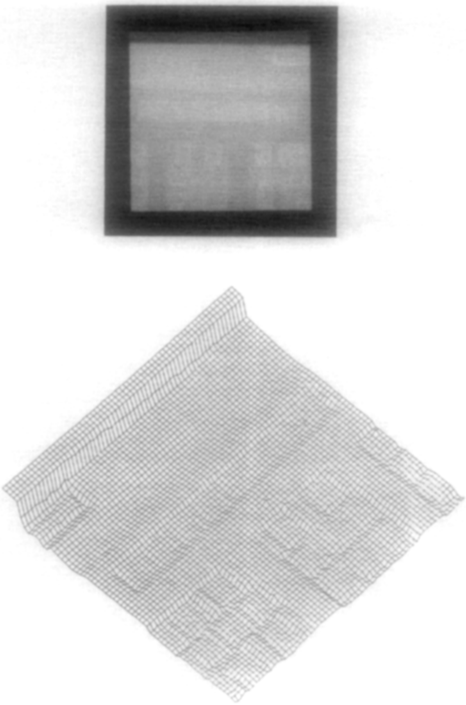

FIG. 15. Keyboard range image and surface plot (128 × 128 ERIM range image).

A free-form undulating surface was created by "stretching a skin" over a series of curves using SDRC/GEOMOD. The depth map for this surface is shown in Fig. 21. The S-curvature results for two different views using 5×5 window derivative operator are shown in Fig. 22. The critical points that are also maximum Gaussian curvature points tend to line up along the joining curve in the center of the range image when we look straight down on the surface. These points move predictably in the second view. These depth maps have no substantial depth-discontinuities; therefore, detailed slope magnitude variations are seen in the scaled edge map ($\sqrt{g}$) image. Note the slight changes in S-curvature sign images between the two views.

To give the reader an idea of how window size and noise level affect the results of the surface characterization algorithm, we use a synthetic image of a cube with three holes in it. We added pseudo-random pseudo-Gaussian noise (rounded to the nearest integer) to the original image to create four different synthetic noisy images. These four images are shown in Fig. 23 and correspond to noise standard deviations (σ's) of 2.3, 9.2, 16.0, and 22.9 gray levels (depth levels) added to an original image with a dynamic range of 256 levels. The resulting noisy images were rescaled to fit into the 8-bit depth range. These images were then processed with 5×5, 7×7, 9×9, 11×11, and 13×13 derivative window operators. We have selected the following five figures to demonstrate the noise performance:

- Figure 24. 5×5 operator applied to $\sigma = 2.3$ noisy image.
- Figure 25. 7×7 operator applied to $\sigma = 9.2$ noisy image.

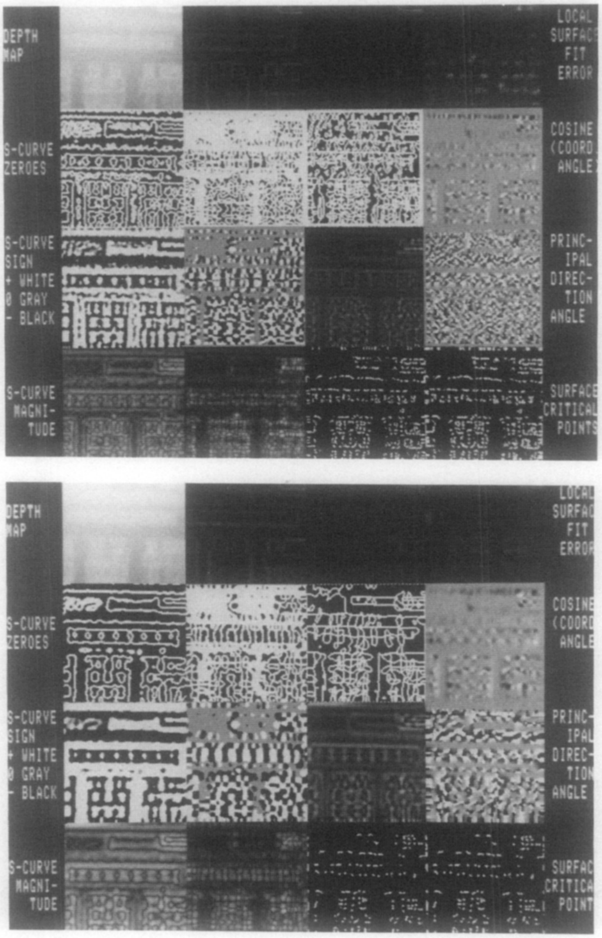

FIG. 16. S-curvature characterizations of keyboard (3 × 3 window operator results—top; 5 × 5 window results—bottom).

- Figure 26. 9 × 9 operator applied to the $\sigma = 16.0$ noisy image.
- Figure 27. 11 × 11 operator applied to the $\sigma = 22.9$ noisy image.
- Figure 28. 13 × 13 operator applied to the $\sigma = 22.9$ noisy image.

Note how well the sign of the mean curvature represents the important surface variations of the cube even in the presence of significant noise. The mean curvature images are surprisingly consistent and qualitatively meaningful in all five figures even though second derivatives are involved in its computation. The Gaussian curvature images seem to be more susceptible to noise, but the closest vertex of the cube is consistently marked as a high curvature spot. The degradation in the zeros of the cosine-of-the-coordinate-angle is quite interesting. The critical point images are quite consistent and demonstrate the necessity of a large window size to suppress spurious critical points in the presence of noise. Note that fewer spurious critical points result in the 11 × 11 $\sigma = 22.9$ results shown in Fig. 27 than in the 5 × 5

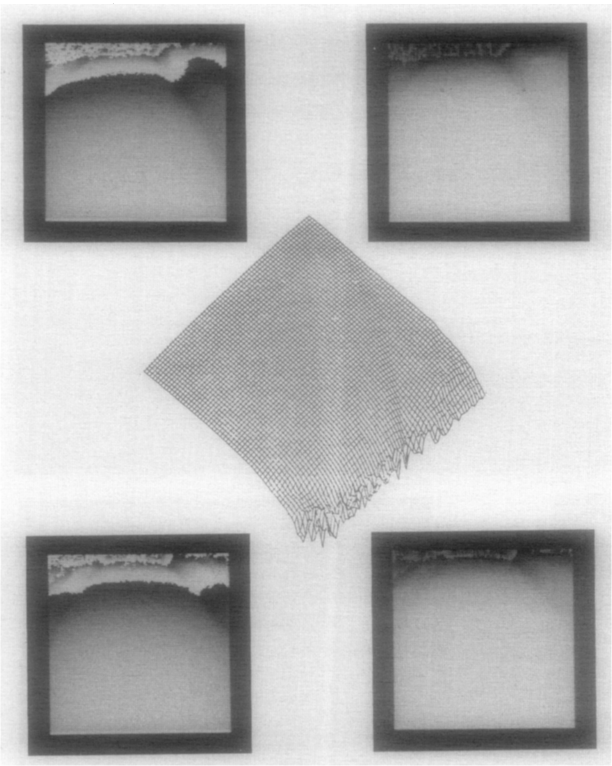

FIG. 17. Original and unwrapped range images of road scene (128 × 128 ERIM range images).

$\sigma = 2.3$ results shown in Fig. 24. Our conclusion is that even though we are using second derivative information, our surface descriptors are still useful in the presence of noise. Moreover, the surface characteristics appear to degrade slowly as the noise level increases.

9. FUTURE RESEARCH DIRECTIONS

An object recognition problem has been defined and surface characteristics for range image segmentation and surface/object matching have been identified. Our experimental results indicate the efficacy of our surface characterization approach. We summarize the topics that need to be addressed in future research:

(1) *Noise*. Sensor noise and quantization noise are important problems in range data processing. Fundamental trade-off decisions between smoothing and spatial localization of surface region edges must be addressed. While our current results are good, we believe even better results can be obtained by providing more elaborate control over smoothing operations and other additional processing. For example, Brady *et al.* [9] show results where smoothing is inhibited around ramp edges. Techniques like this will help to improve these results.

(2) ***High-level data-driven description***. Each basic isolated surface type is structurally simple. We have had some preliminary success building a higher-level data-driven geometric description of range images using low order polynomial

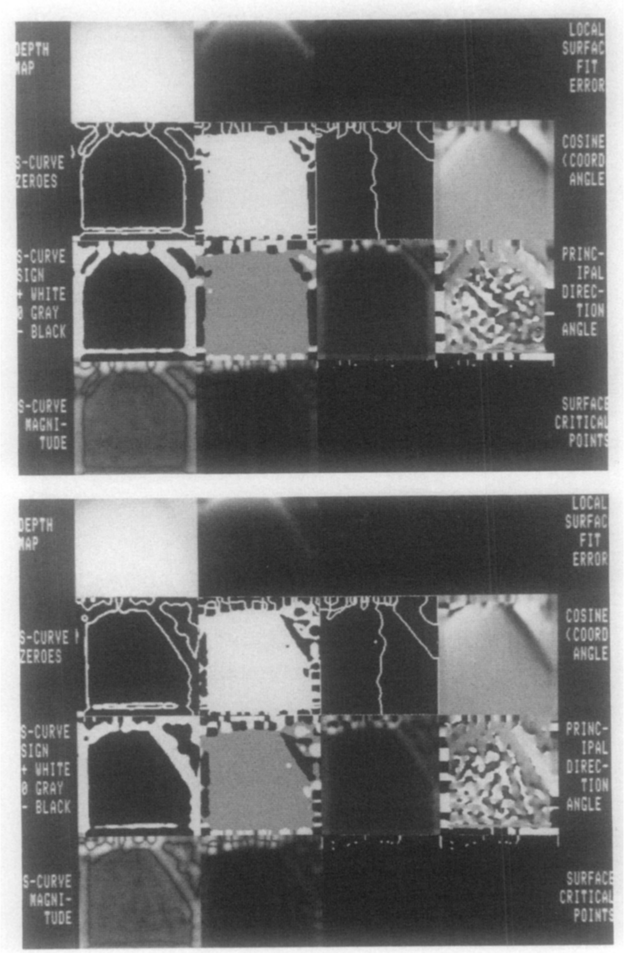

FIG. 18. *S*-curvature characterizations of road scenes (9 × 9) derivative window operator, zero threshold = 2%).

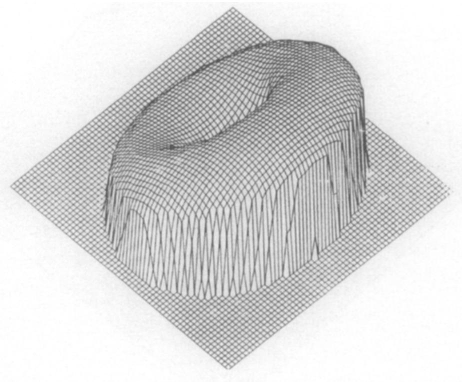

FIG. 19. Surface plot of torus depth map (128 × 128 synthetic depth map).

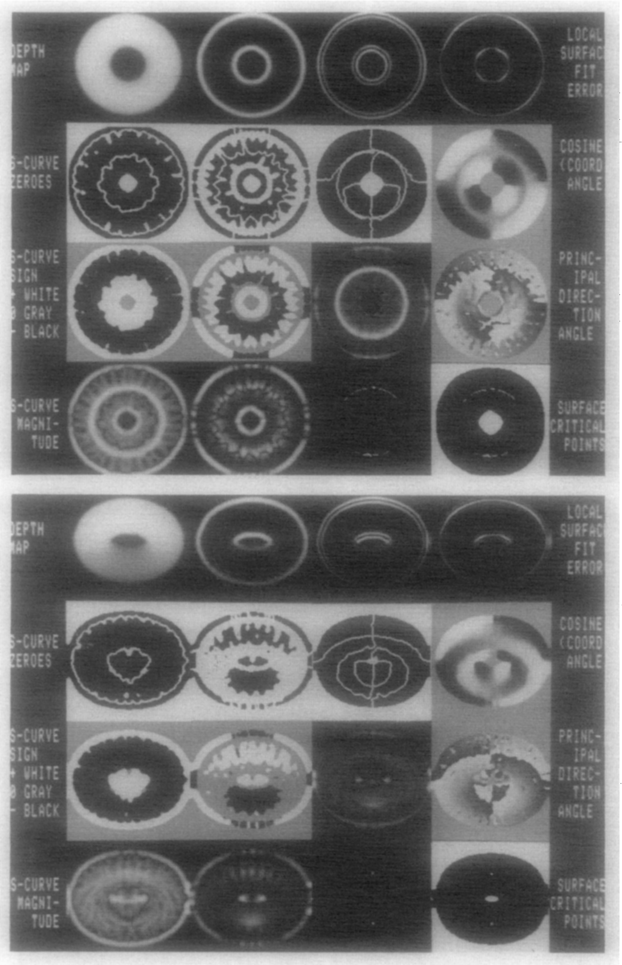

FIG. 20. S-curvature characterizations of two views of torus (5 × 5 derivative window operator, zero threshold = 1%).

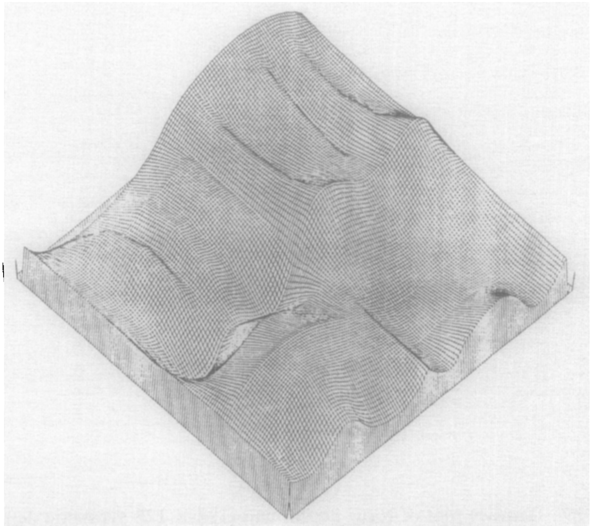

FIG. 21. Surface plot of undulating surface depth map (128 × 128 synthetic depth map).

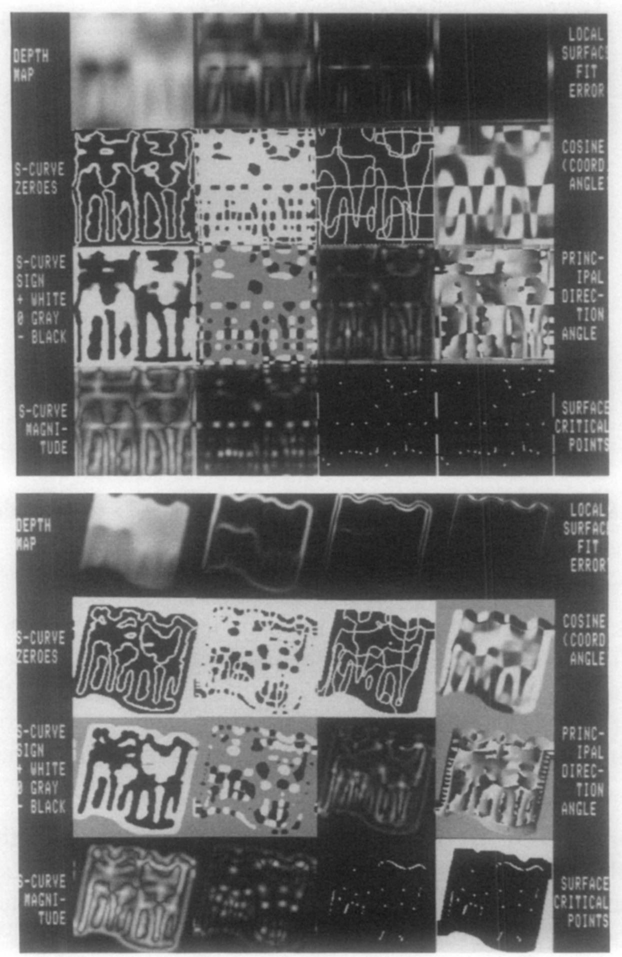

FIG. 22. *S*-curvature characterizations of two views of undulating surface (5 × 5 derivative window operator, zero threshold = 1%).

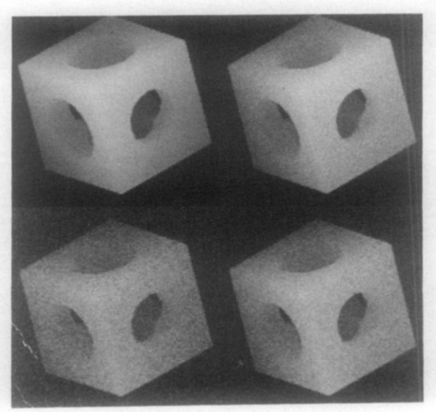

FIG. 23. Block with different noise levels (2.3, 9.2, 16.0, 22.9) (128 × 128 synthetic depth maps).

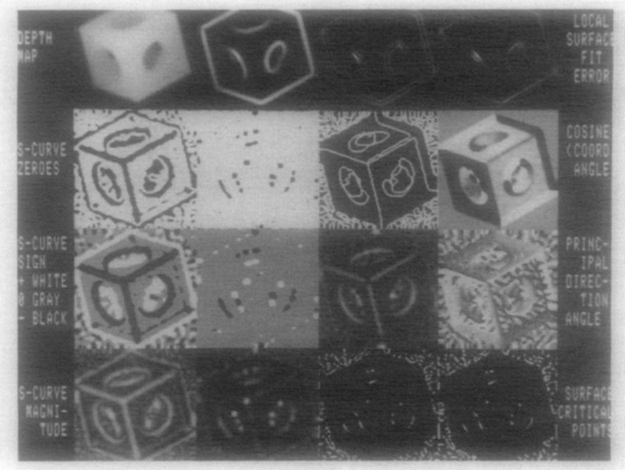

FIG. 24. S-curvature results for 5 × 5 operator with sigma = 2.3 (zero threshold = 6%).

surface patches. The RMS fit errors are typically less than 1.2 depth levels for real range images. The dimensionality of this higher-level description is extremely small compared to the range image itself whenever surfaces occupy more than a hundred pixels in an image. This method is not limited to planes, sphere, cones, cylinders, and other quadrics often treated in the literature. More research is needed to develop this concept.

(3) *Matching.* There are many matching algorithm problems that need to be solved before these characteristics are used for recognition purposes. A view-independent matching representation is needed. We would like to compute that representation automatically from given object models of arbitrary shape.

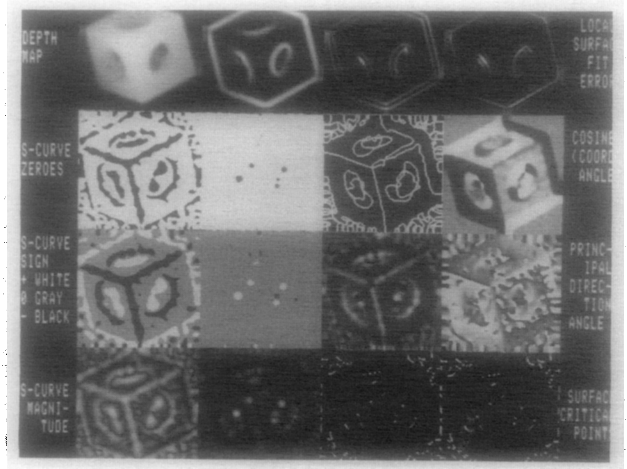

FIG. 25. S-curvature results for 7 × 7 operator with sigma = 9.2 (zero threshold = 12%).

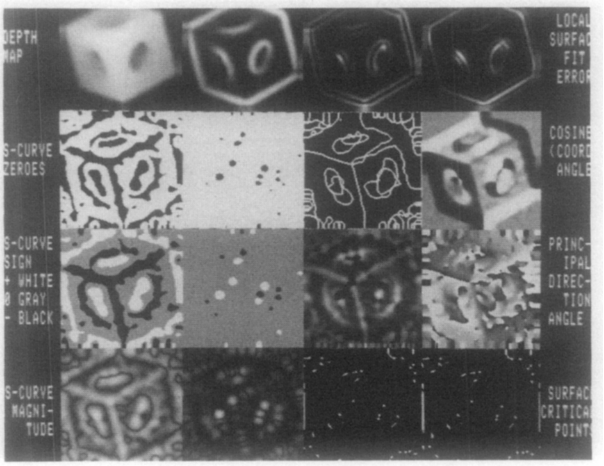

FIG. 26. S-curvature results for 9 × 9 operator with sigma = 16.0 (zero threshold = 14%).

(4) *Feedback.* We mentioned verification feedback earlier. An interaction of data-driven and model-driven processes is required for robust recognition. The use of feedback within vision systems must be analyzed.

(5) *Occlusion.* We hope to include minor occlusion handling as part of the general purpose matching algorithm. The surface characteristics are suitable, but the higher level processes require research.

(6) *Learning.* Eventually, vision systems will have to learn about their environment from what they sense. Research is needed to determine how a system should handle data with surface characteristics that do not match any known models in the object list.

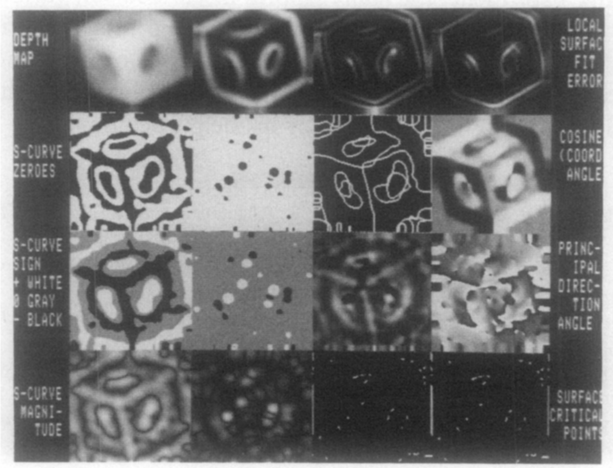

FIG. 27. S-curvature results for 11 × 11 operator with sigma = 22.9 (zero threshold = 14%).

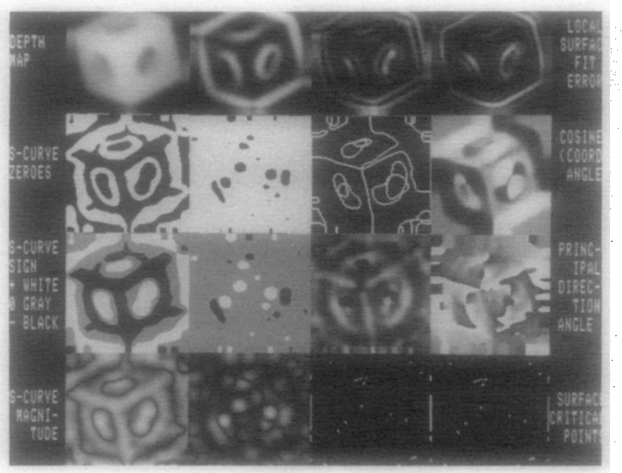

FIG. 28. *S*-Curvature results ofr 13 × 13 operator with sigma = 22.9 (zero threshold = 14%).

One applied goal of this research is to develop a system that will correctly recognize objects from any view selected at random from an object list that contains 20 or 30 objects with curved and flat surfaces. These objects will span a range of object complexities and include objects of similar appearance to challenge the system. We believe that visible-invariant surface characteristics can be used to achieve this goal.

ACKNOWLEDGMENTS

We wish to acknowledge (1) the Environmental Research Institute of Michigan (ERIM) for allowing us to use range images obtained by their laser depth sensor, and (2) the Structural Dynamics Research Corporation (SDRC) and General Electric-CAE International for allowing us to use the SDRC/I-DEAS GEOMOD solid modeling software on our Apollo engineering workstations to create solid object models. We thank Paul Eichel for many helpful discussions on various topics during the course of this research. We are indebted to Professor Mario Micallef for interesting and informative discussions on the equations of prescribed mean curvature and prescribed Gaussian curvature and for several pointers to the mathematical literature. Lastly, we thank the reviewer for his helpful comments.

REFERENCES

1. M. D. Altschuler, J. L. Posdamer, G. Frieder, B. R. Altschuler, and J. Taboada, The numerical stereo camera, in *Proc. SPIE 3-D Machine Perception Conf.*, Vol. 283, 1981, pp. 15–24.
2. H. Asada and M. Brady, The curvature primal sketch, in *Proc. IEEE Workshop on Computer Vision: Repres. and Control*, Annapolis, Md., May 1984, pp. 8–17.
3. P. R. Beaudet, Rotationally Invariant Image Operators, in *Proc. 4th Intl. Joint Conf. Pattern Recognition*, Kyoto, Japan, November 1978, pp. 579–583.
4. P. J. Besl and R. C. Jain, Three-dimensional object recognition, *ACM Comput. Surveys* **17**, No. (1), 1985, pp. 75–145.
5. P. J. Besl and R. C. Jain, *Surface Characterization for Three-Dimensional Object Recognition in Depth Maps*, RSD-TR-20-84, Univ. of Michigan, Ann Arbor, Mich., December 1984.
6. B. Bhanu, Representation and shape matching of 3-D objects, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 3, 1984, 340–350.

7. R. M. Bolle and D. B. Cooper, Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 4, 1984, 418–429.
8. R. C. Bolles, P. Horaud, and M. Jo Hannah, 3 DPO: A three dimensional part orientation system, in *Proc. 8th Intl. Joint Conf. on Artif. Intell.*, Karlsruhe, West Germany, August 1983, pp. 1116–1120.
9. M. Brady, J. Ponce, A. Yuille, and H. Asada, *Describing Surfaces*, in *Proc. 2nd Intl. Sympos. on Robotics Research*, MIT Press, Cambridge, Mass., 1985.
10. I. Chakravarty and H. Freeman, Characteristic views as a basis for three-dimensional object recognition, in *Proc. SPIE Robot Vision Conf.*, Vol. 336, Arlington, Va., May 1982, pp. 37–45.
11. S. Chern, A proof of the uniqueness of Minkowski's problem for convex surfaces, *Amer. J. Math.* **79**, 1957, 949–950.
12. E. N. Coleman and R. Jain, Obtaining shape of textured and specular surface using four-source photometry, *Comput. Graphics Image Process.* **18**, No. 4, 1982, 309–328.
13. L. Dreschler and H. H. Nagel, Volumetric model and 3D-Trajectory of a moving car derived from monocular TV-frame sequences of a street scene, in *Proc. 7th Intl. Joint Conf. on Artificial Intell.*, Vancouver, B.C., Canada. August 1981, pp. 692–697.
14. O. D. Faugeras, New steps toward a flexible 3-D vision system for robotics, in *Proc. 7th Intl. Joint Conf. Pattern Recognition*, Montreal, Quebec, Canada, August 1984, pp. 796–805.
15. I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis–Horwood, Chichester, U.K., 1979.
16. *GEOMOD 2.5 User Manual and Reference Manual*, Structural Dynamics Research Corporation, Milford, Ohio, 1984.
17. D. Gilbarg and N. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer-Verlag, New York, 1977.
18. E. Guisti, On the equation of surfaces of prescribed mean curvature: Existence and uniqueness without boundary conditions, *Invent. Math.* **46**, 1978, 111–137.
19. R. M. Haralick, Digital step edges from zero-crossings of second directional derivatives, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 1, 1984, 58–68.
20. R. M. Haralick, L. T. Watson, and T. J. Laffey, The topographic primal sketch, *Int. J. Robot. Res.* **2**, No. 1, 1983, 50–72.
21. R. M. Haralick, A Bayesian approach to robust local facet estimation, in preparation, 1985.
22. B. K. P. Horn, Extended Gaussian images, *Proc. IEEE* **72**, No. 12, 1984, 1671–1686.
23. C.-C. Hsiung, *A First Course in Differential Geometry*, Wiley–Interscience, New York, 1981.
24. K. Ikeuchi and B. K. P. Horn, Numerical shape from shading and occluding boundaries, *Artif. Intell.* **17**, 1981, 141–184.
25. K. Ikeuchi, B. K. P. Horn, S. Nagata, T. Callahan, and O. Feingold, *Picking Up an Object from a Pile of Objects*, MIT AI Lab Memo 726, May 1983.
26. S. Inokuchi, K. Sato, and F. Matsuda, Range Imaging System for 3-D Object Recognition in *Proc. 7th Intl. Conf. on Pattern Recognition*, Montreal, Quebec, Canada, August 1984, pp. 806–808.
27. D. J. Ittner and A. K. Jain, 3-D surface discrimination from local curvature measures, in *Proc. Computer Vision and Pattern Recognition Conf.*, San Francisco, Calif., June 1985, pp. 119–123.
28. R. A. Jarvis, A perspective on range finding techniques for computer vision, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 2, 1983, 122–139.
29. R. A. Jarvis, A laser time-of-flight range scanner for robotic vision, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 5, 1983, 505–512.
30. J. J. Koenderink and A. J. van Doorn, Internal representation of solid shape with respect to vision, *Biol. Cybernet.* **32**, No. 4, 1979, 211–216.
31. J. J. Koenderink and A. J. van Doorn, The singularities of the visual mapping *Biol. Cybernet.* **24**, No. 1, 1976, 51–59,
32. H. S. Kim, R. Jain, and R. A. Volz, Object recognition using multiple views, in *Proc. of Intl. Conf. on Robotics and Automation*, St. Louis, Mo., March 1985, pp. 28–33.
33. D. T. Kuan and R. J. Drazovich, Model-based interpretation of range imagery, in *Proc. Nat'l Conf. on Artificial Intelligence*, Austin, Texas, August 1984, AAAI, pp. 210–215.
34. D. J. Langridge, Detection of discontinuities in the first derivatives of surfaces, *Comput. Vision Graphics Image Process.* **27**, No. 3, 1984, 291–308.
35. R. A. Lewis and A. R. Johnston, A scanning laser rangefinder for a robotic vehicle, in *Proc. 5th Intl. Joint Conf. on Artificial Intell.*, Cambridge, Mass., August 1977, pp. 762–768.

36. C. Lin and M. J. Perry, Shape description using surface triangularization, in *Proc. IEEE Workshop on Computer Vision: Repres. and Control*, Rindge, N. H., August 1982, pp. 38–43.

37. M. M. Lipschutz, *Differential Geometry*, McGraw–Hill, New York, 1969.

38. J. J. Little, An iterative method for reconstructing convex polyhedra from extended Gaussian images, in *Proc. Natl. Conf. on Artificial Intelligence*, Washington, D. C., August 1983, AAAI, pp. 247–250.

39. D. H. Marimont, A representation for image curves, in *Proc. Natl. Conf. on Artificial Intelligence*, Austin, Texas, August 1984, AAAI, pp. 237–242.

40. G. Medioni and R. Nevatia, Description of 3-D surfaces using curvature properties, in *Proc. Image Understanding Workshop*, New Orleans, La., October 1984, DARPA, pp. 291–299.

41. H. Minkowski, *Allgemeine Lehrsatze über die konvexen Polyeder*, Nachrichten von der Königlichen Gesellschaft der Wissenschaften, Mathematisch-Physikalische Klasse. Göttingen. pp. 198–219, 1897.

42. C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation*, Freeman, San Francisco, 1973.

43. L. R. Nackman, Two-dimensional critical point configuration graphs, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 4, 1984, 442–449.

44. R. Nevatia and T. O. Binford, Description and recognition of curved objects, *Artif. Intell.* **8**, 1977. 77–98.

45. B. O'Neill, *Elementary Differential Geometry*, Academic Press, New York, 1966.

46. M. Oshima and Y. Shirai, Object recognition using three-dimensional information, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**, No. 4, 1983, 353–361.

47. S. Parthasarathy, J. Birk, and J. Dessimoz, Laser rangefinder for robot control and inspection, in *Proc. Soc. Photo-Opt. Instrum. Eng. Conf. Robot Vision*, Vol. 336, Arlington, Va., May 1982, pp. 2–11.

48. M. Potmesil, Generating models of solid objects by matching 3-D surface segments, in *Proc. 8th Intl. Joint Conf. on Artif. Intell.*, Karlsruhe, West Germany, August 1983, pp. 1089–1093.

49. Y. Sato, H. Kitagawa, and H. Fujita, Shape measurement of curved objects using multiple slit ray projections, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-4**, No. 6, November 1982, 641–646.

50. I. K. Sethi and S. N. Jayaramamurthy, Surface classification using characteristic contours, in *Proc. 7th Intl. Conf. on Pattern Recognition*, Montreal, Quebec, Canada, August 1984, pp. 438–440.

51. D. R. Smith and T. Kanade, Autonomous scene description with range imagery, in *Proc. Image Understanding Workshop*, New Orleans, La., October 1984, DARPA, pp. 282–290.

52. K. Sugihara, Range-data analysis guided by junction dictionary, *Artif. Intell.* **12**, 1979, 41–69.

53. D. J. Svetkoff, P. F. Leonard, R. E. Sampson, and R. Jain, Techniques for real-time 3-D feature extraction using range information, in *Proc. Soc. Photo-Opt. Instrum. Eng. Conf. Intelligent Robotics and Computer Vision*, Cambridge, Mass., November 1984, pp. 302–309.

54. J. B. K. Tio, C. A. McPherson, and E. L. Hall, Curved surface measurement for robot vision, in *Proc. Pattern Recognition and Image Processing Conf.*, 1982, pp. 370–378.

55. A. P. Witkin, Recovering surface shape and orientation from texture, *Artif. Intell.* **17**, 1981, 17–45.

56. R. J. Woodham, Analyzing images of curved surfaces, *Artif. Intell.* **17**, 1981, 117–140.