

# Saliency Sequential Surface Organization for Free-Form Object Recognition

Kim L. Boyer and Ravi Srikantiah

*Signal Analysis and Machine Perception Laboratory, Department of Electrical Engineering,  
The Ohio State University, Columbus, Ohio 43210*

E-mail: kim@ee.eng.ohio-state.edu

and

Patrick J. Flynn

*Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, Indiana*

E-mail: flynn@nd.edu

Received September 14, 2001; accepted July 30, 2002

---

We introduce an efficient, robust means of obtaining reliable surface descriptions, suitable for object recognition, at multiple scales from range data. Mean and Gaussian curvatures are used to segment the surface into regions of four saliency classes, each based on curvature consistency. We evaluate curvature consistency in a robust multivoting scheme. Contiguous regions consistent in both mean and Gaussian curvature are identified as the most homogeneous, and therefore (probably) the most salient, followed by those consistent in mean curvature only, followed by those consistent in Gaussian curvature only. Segments at each level of the hierarchy are extracted in the order of size, large to small, such that the most salient features of the surface are recovered first. To demonstrate an application of the work, we present an effective recognition system for free form objects based on attributed graphs constructed from the surface segmentation. © 2002 Elsevier Science (USA)

*Key Words:* surface organization; segmentation; free-form objects; object recognition; range data.

---

## 1. INTRODUCTION

In general, it is unreasonable to expect a vision system, human or machine, to recognize objects by operating at a single level of detail. There is ample evidence that such matching occurs over different scales in the human visual system, and sound engineering arguments for doing so in machine vision systems. In this contribution, we work with range data

of the type produced by a variety of commodity sensors to construct object descriptions based on surface curvatures. From these, we build such a multiscale recognition system that describes objects at successively higher resolutions until a suitable degree of discrimination is obtained.<sup>1</sup>

Mokhtarian *et al.* [1–3] proposed a multiscale description technique in which the surface is smoothed by a Gaussian filter at varying scales, and the curvature values are recomputed for each scale. Their method uses semigeodesic parameterizations of the local surface about each mesh point, smoothed by small 2D Gaussian weighting windows. Their segmentation of the surface is defined by zero-crossing contours of the mean and Gaussian curvatures, as well as the curvature extrema. The problem with such a scheme is its computational burden; smoothing and curvature estimation are among the most computationally intensive of operations. Our approach requires no establishment of semigeodesic coordinates, which vary from point to point, and we filter the data only once with an isotropic 3D Gaussian filter. The filtering is a true convolution and is not dependent on a local coordinate system. All of the advantages claimed (justifiably) in [3] hold for our approach, as well: combines multiscale description and curvature estimation; applicable to noisy, free-form, incomplete meshes; and efficient and robust to mesh geometry.

Zhang and Hebert [4] proposed a multiscale classification based on a smoothing parameter applied repeatedly to smooth the curvature estimates (not the actual surface). While eliminating the need to recompute curvature, this is still very computationally demanding. Here, we propose a multiscale classification based on the parameters of a multivoting technique. This approach avoids resmoothing and recomputing surface curvatures, thereby minimizing the computational burden, while offering the advantages of a traditional multiscale technique.

The notion of curvature consistency has been explored by Sander and Zucker [5], and later enhanced by Ferrie *et al.* [6] to deal with discontinuities better. Our approach to consistency differs from those efforts in that we are less concerned with regularizing the differential geometry to achieve better surface estimation than we are in simply grouping contiguous patches of the surface having consistent curvature characteristics. Ours is a segmentation algorithm based on sequential extraction of consistent regions, not a surface estimation algorithm. This lends our approach a measure of robustness not possible with methods that impose a set of surface class definitions (cap, cup, ridge, saddle, . . .).

## 2. AN OVERVIEW OF THE RECOGNITION SYSTEM

The primary contribution of this paper in its novel approach to surface organization. However, we implemented a recognition system for free form objects to demonstrate the utility of our approach. The input to the system consists of a polygonal mesh representing either a single range map, or a registered version of two or more such maps. A brief description of each of the major portions of the system follows.

*1. Estimation of local surface curvatures.* We first smooth the data *slightly* using a 3D isotropic Gaussian filter (operating directly on the mesh) to attenuate noise. We then compute local surface properties in the form of mean and Gaussian curvatures at each mesh point. This is by far the most computationally intensive step, *but it is done only once.*

<sup>1</sup> The resolution of the range image is fixed; it is the resolution of our *description* of the surface that varies.

2. *Surface organization.* The mean and Gaussian curvatures define homogeneity measures that form the basis for surface organization. Segmentation is achieved by grouping portions of the surface that exhibit consistent curvature, within a tolerance. The tolerance window, softened by a multivoting technique, acts as a reliable scale parameter that produces segmentations at varying resolutions without actually degrading the surface. The order of segment extraction provides a partial ordering on the basis of region saliency. Because we group together contiguous points meeting a set of curvature homogeneity criteria, we achieve considerable robustness and a surface description especially effective for free form objects.

3. *Object representation.* The system next builds a symbolic description of the object (or scene) based on the properties of individual segments and their spatial relationships. The segment properties and relations are represented in the form of an attributed graph whose nodes represent the segments and whose links encode the geometric relationships between segment pairs.

4. *Matching.* The matching process searches to find a sufficiently large subgraph in the modelbase that matches a scene subgraph to a maximal extent. The strength of the match is computed as a weighted error measure between the scene and model parameters of the maximally matched subgraphs. If the match obtained is good enough, the object is deemed to have been recognized, and the search terminates. If no suitable match is obtained, the scale of the segmentation is refined and the process repeats until either a match is obtained or the range of scales is exhausted.

### 3. SURFACE CURVATURE

While this section offers some (minimal) background on surface curvature computation, a thorough development is beyond the scope of this paper. Flynn and Jain [7] offer a review of some existing methods of surface curvature estimation. They classify estimation techniques into two broad categories: analytic and numerical. Analytic methods, such as we use here, estimate the curvature of a point on the surface based on a local surface fit called a *Monge patch*. A Monge patch is a surface of the form

$$\mathbf{x}(u, v) = (u, v, f(u, v)). \quad (1)$$

With this formalism, the Gaussian and mean curvatures are expressed as, respectively,

$$K = \frac{f_{uu}f_{vv} - f_{uv}^2}{(1 + f_u^2 + f_v^2)^2} \quad (2)$$

$$H = \frac{(1 + f_v^2)f_{uu} - 2f_u f_v f_{uv} + (1 + f_u^2)f_{vv}}{2(1 + f_u^2 + f_v^2)^{\frac{3}{2}}}, \quad (3)$$

where the subscripts indicate partial differentiation with respect to the corresponding variable. We remind the reader that the Gaussian curvature  $K$  is the product of the principal curvatures (normal curvature extrema) and the mean curvature  $H$  is the mean value of the normal curvature over all directions, equal to the average of the principal curvatures.<sup>2</sup>

<sup>2</sup> The normal curvature is that of the curve formed by the intersection of the surface with a plane containing the local surface normal, as a function of the orientation (direction) of that plane as it rotates about the normal.

Besl and Jain [8] use a set of discrete orthogonal polynomials to provide a quadratic surface fit over a given  $N \times N$  window, and the coefficients that minimize the mean squared error to estimate the first and second partial derivatives.

Flynn and Jain [9] use a bicubic polynomial,

$$f(u, v) = c_1u^3 + c_2u^2v + c_3uv^2 + c_4v^3 + c_5u^2 + c_6uv + c_7v^2 + c_8u + c_9v + c_{10}, \quad (4)$$

to compute a local surface fit using linear regression.

Numerical methods offer an alternative to analytical methods in computing surface curvature. These use an ensemble of directional curvature estimates. Fan *et al.* [10] use normal curvature estimates along four directions at  $45^\circ$  intervals to calculate the principal curvatures and their directions.

Other numerical methods [11–13] estimate curvature by computing the angular change in normal direction over a neighborhood of points over the surface. Hoffman and Jain [12] estimate  $\kappa(p, q)$ , the curvature in moving from point  $p$  to point  $q$  on the surface, as

$$\kappa(p, q) = \frac{\|\bar{n}_p - \bar{n}_q\|}{\|\bar{p} - \bar{q}\|} s(p, q), \quad (5)$$

where  $s(p, q)$  is a “sign” factor that indicates whether the curvature is positive (convex) or negative (concave). If the two normals  $\bar{n}_p$  and  $\bar{n}_q$  at pixels  $p$  and  $q$ , respectively, initially approach each other, the curvature is classified as negative (concave); otherwise the curvature is positive (convex).

Smith [13] observes that the mean curvature of a discretized surface made from triangular facets is zero everywhere except at the discontinuities along the edges of the facets. A working definition of the mean curvature is then computed by summing the angle changes of the normal along the curve at the discontinuities and normalizing by the arc length over which the summation is performed. The Gaussian curvature at a vertex is computed as a ratio of the sum of the angles at a vertex to the number of facets meeting at the vertex.

The experiments by Flynn and Jain [7] make it apparent that analytical methods have a distinct edge over numerical methods in overall performance. As Besl and Jain [8] point out, “In general, direct numerical differentiation is discouraged if a problem can be addressed using other means. . . .”

### 3.1. Computing Mean and Gaussian Curvatures

We now describe the analytic methods we use to compute Gaussian and mean curvatures. We choose this approach for its solid theoretical foundation and because its performance is among the best in its class.

The first step is to establish a local coordinate system at each point on the surface such that the point is at the origin and a reference plane is established tangential to the surface at that point. What we are looking for is a means to identify the principal (orthogonal) directions of the spread of data (surface points) in the neighborhood along which we can set the axes of our local coordinate system. The direction of least spread gives the normal to our surface patch while the other two maximal directions define the tangent plane. Principal components analysis provides the solution.

### 3.1.1. Principal Components Analysis

Given a set of  $N$  points  $\mathbf{x}_i = [x_i, y_i, z_i]^T$  in  $\mathcal{R}^3$  we define a symmetric covariance matrix  $\mathbf{S}$  as

$$\mathbf{S} = \sum_{i=1}^N [\mathbf{x}_i - \hat{\mathbf{x}}][\mathbf{x}_i - \hat{\mathbf{x}}]^T, \quad (6)$$

where  $\hat{\mathbf{x}}$  is the mean of the data set. The data set is represented by  $N$  contiguous points on the surface that project to within a circle of radius  $R$  on the tangent plane, with its center being the point whose curvature is required. These  $N$  points may also be smoothed with a lowpass Gaussian filter having impulse response

$$g(r, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{r^2}{2\sigma^2}}, \quad (7)$$

where  $r$  is the Euclidean distance of a point from the center of the filter in  $\mathcal{R}^3$ , and  $\sigma$  represents a scale factor that determines the width of the Gaussian filter.

Let  $\lambda_{max}$ ,  $\lambda$ , and  $\lambda_{min}$  be the three eigenvalues of the covariance matrix  $\mathbf{S}$ , and let  $\mathbf{v}_{max}$ ,  $\mathbf{v}$ , and  $\mathbf{v}_{min}$  denote the corresponding normalized eigenvectors. The vector  $\mathbf{v}_{min}$  that is oriented along the direction of least spread denotes the normal, while  $\mathbf{v}$  and  $\mathbf{v}_{max}$  are tangent to the surface. Our coordinate system is now represented by the three eigenvectors in a  $3 \times 3$  matrix as  $\mathbf{V} = [\mathbf{v}_{max} \ \mathbf{v} \ \mathbf{v}_{min}]$ . We invert the orientation of the basis vectors if necessary to ensure consistent normals (outward or inward) and that all local coordinate systems are right-handed.

### 3.1.2. Least-Squares Surface Fit

Once the local coordinate system obtained from principal components analysis and tessellation has been fixed, the points from the surface patch are transformed into this coordinate system and we fit a biquadratic surface of the form

$$w = f(u, v) = a_1u^2 + a_2uv + a_3v^2 + a_4u + a_5v + a_6 \quad (8)$$

using least squares. Because we intend to collect points of consistent curvature, and therefore having no inflections, there is nothing to be gained through the additional complexity of bicubic or higher order polynomial fits. Recall that the  $(u, v)$  coordinates are in the local tangent plane while the  $w$  coordinate is the local normal. Under normal error conditions, the best estimate of  $\mathbf{a} = [a_1 \ \dots \ a_6]$  is one that minimizes the squared error and is given by

$$\hat{\mathbf{a}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{w}, \quad (9)$$

where  $\mathbf{w}$  is the vector of actual height observations and each row of  $\mathbf{A}$  is the corresponding set of explanatory variables ( $u^2$ ,  $uv$ , and so on). Having found the coefficients of the biquadratic in Eq. (8) we can now compute the partial derivatives as

$$f_u = \frac{\partial f}{\partial u} = 2a_1u + a_2v + a_4 \quad f_u|_{(u,v)=0} = a_4 \quad (10)$$

$$f_v = \frac{\partial f}{\partial v} = a_2u + 2a_3v + a_5 \quad f_v|_{(u,v)=0} = a_5 \quad (11)$$

$$f_{uu} = \frac{\partial^2 f}{\partial u^2} = 2a_1 \quad (12)$$

$$f_{uv} = \frac{\partial^2 f}{\partial u \partial v} = a_2 \quad (13)$$

$$f_{vv} = \frac{\partial^2 f}{\partial v^2} = 2a_3. \quad (14)$$

Substituting these values into Eqs. (2) and (3) gives the Gaussian and mean curvatures at that point.

## 4. SURFACE ORGANIZATION

Our next objective is to obtain an object representation just sufficiently detailed to distinguish it uniquely. Local curvatures at each point provide an overrich, inefficient description of the surface; we need an abstraction, or simplified representation. To this end, we shall group points on the surface that exhibit some degree of consistency with respect to position and shape. These groupings, in defining a partition of the surface, should be small enough to offer adequate detail, while being few enough to provide a suitably compact description. Unfortunately, and as mentioned above, choosing a single level of detail that works well for all surfaces is effectively impossible. Therefore, in this section we present a method that extracts surface patches at varying levels of detail, enabling us to construct and use a description of the object that is just sufficiently rich for the task of recognition.

### 4.1. Prior Work in Range Image Segmentation

Range image segmentation techniques can primarily be classified into two distinct categories—edge-based [14, 15], and region- (or surface-) based [9, 12, 16–18]. There are also hybrid techniques [19–23]. Ours is region-based.

Most edge-based techniques detect two common types of range edges—steps and creases. The first correspond to range discontinuities, the second to orientation discontinuities. After such points of discontinuity have been detected, the edges are “grown” to ensure topological consistency. For a sampling of work in edge-based range image segmentation see [14, 15, 22, 24].

Region-based techniques typically apply two criteria: (i) the homogeneity of the surface and (ii) pixel connectivity. Milgrim and Bjorklund [25] used a least-squares technique to fit a plane to a neighborhood of pixels. Information about the normal vectors and the planar fit error are used to form connected components of pixels satisfying planarity constraints. Henderson [26] used a sequential region-growing algorithm (via a planarity test) that creates convex planar surfaces from initial sets of three close noncollinear points. Hebert and Ponce [27] segmented depth maps into planar, cylindrical, and conical primitives based on the distribution of surface normals. A similar segmentation technique was described by Han *et al.* [28], in which planar regions are first extracted by histogram analysis of normal values, the cylindrical regions are extracted next by the computation of axis information and projection (to extract the resulting circles), and finally spherical regions are extracted by estimating possible center points for each pair of surface points and voting to decide the majority value of the center. Sethi and Jayamurthy [29] handled spheres and ellipsoids in

**TABLE 1**  
**Surface Description from Gaussian and Mean Curvatures (from [8])**

	$H < 0$	$H = 0$	$H > 0$
$K < 0$	Saddle ridge	Minimal surface	Saddle valley
$K = 0$	Ridge	Plane	Valley
$K > 0$	Peak	—	Pit

addition to planes, cylinders, and cones by classifying surfaces using characteristic contours. Oshima and Shirai [30] grouped points into small planar surface elements and then merged these to form elementary planar or curved (spherical, cylindrical and conical) regions. These elementary regions were then merged again into consistent global regions that were fitted with biquadratic surfaces.

Besl and Jain [16] used the signs of the Gaussian and mean curvatures computed at each point (HK sign map) to classify each point as one of eight qualitative surface types described in Table 1. This lower level segmented description is then passed through a higher level surface approximation stage that uses an iterative region growing method based on variable order surface fitting. Table 2 presents a categorization based on principal curvatures. We do not use either of these classifications directly.

Hoffman and Jain [12] obtained a six-dimensional feature vector at every point on the surface composed of the position of the range pixel  $(x, y, z)$  and the components of the normal vector  $(n_x, n_y, n_z)$  obtained using principal component analysis. A squared error clustering technique was then used to obtain preliminary over-segmented patches that were then merged to give a final segmentation. Nonparametric statistical tests for curvature values and eigenanalysis are then used to classify the underlying surface as convex, concave, or planar, and the boundaries are classified as crease and non-crease edges.

Vemuri and Aggarwal [17] classify each point based on principal curvatures into one of five qualitative surface types: planar, parabolic, elliptic, hyperbolic, and umbilic. Haralick *et al.* [18] classify each point as one of six qualitative surface types: peak, pit, ridge, ravine, saddle, flat and hillside.

Abdelmalek [31] proposed a three-sign curvature labeling scheme based on the sign of the mean curvature only. Taylor *et al.* [32] used a split-and-merge technique where the homogeneity measure uses three surface parameters—the two spherical angles indicating the directions of surface normals and the range values at the point.

Flynn and Jain [9] describe a surface classification scheme based on a quadric surface model. A sample of surface points is classified as planar or nonplanar through two hypothesis tests. If the sample is nonplanar, curvature features are evaluated at each point to classify

**TABLE 2**  
**Surface Description from Principal Curvatures (from [8])**

	$\kappa_1 < 0$	$\kappa_1 = 0$	$\kappa_1 > 0$
$\kappa_2 < 0$	Peak	Ridge	Saddle
$\kappa_2 = 0$	Ridge	Flat	Valley
$\kappa_2 > 0$	Saddle	Valley	Pit

the sample as spherical, cylindrical, or conical. A nonlinear optimization technique is used to refine the parameters of the surface type.

Three-dimensional shape descriptors are beginning to see use in new contexts. In such applications, shape similarity or dissimilarity is of interest, but shape orientation is not. Hence, rotation-invariant descriptors are preferred. Suzuki *et al.* [33] employ a rotation-invariant distance-indexed histogram of surface point data as a discriminant for different 3D shapes described as polygonal meshes. The proposed application was shape retrieval from a 3D model database, an area of interest in virtual reality and synthetic environments. Ankerst *et al.* [34] also employed histogram features in multimedia data retrieval and proposed a modified nearest-neighbor search procedure for shape retrieval employing the histogram features. When comparing and contrasting such techniques with the work described here, we note that the problem being addressed here is object modeling and recognition rather than database indexing. Moreover, connected surface data is employed rather than pure point data. In addition, our differential features are local in nature and rotation-invariant by design. We believe that their estimation is tolerant of the noise levels encountered in modern range sensors.

Among the hybrid techniques, Yokoya and Levine [20] combine a region segmentation based on *HK*-sign maps with jump and roof edge maps to obtain a final segmentation. Jain and Nadabar [21] describe a Markov Random Field (MRF) approach that integrates both region and edge based segmentations. Bhandarkar and Siebert [22] exploit the synergy between the processes of detecting surface discontinuities and surface regions by using the geometrical properties of the detected surface regions. Homogeneous surface regions are extracted using a combination of clustering in parameter space and region growing. Liou *et al.* [35] treated the range image segmentation problem as one of signal level perceptual organization, to our knowledge the first use of this term. Their work was notable in that it permitted a parallel implementation, assumed no image noise model, and could handle range or grayscale images in the same framework. However, it used elementary edge models (perhaps too elementary), included no notion of saliency, and has not been shown to work with free-form objects. More recently, Jiang [36] has proposed a system for range image edge linking based on the assumption that closed edge contours bound surfaces described by a set of low-order functions. It begins from a binary edge map. Each region (contiguous set of nonedge pixels) is tested to see if it fits one of the surface models (plane or biquadratic). If not, its boundary is dilated, thus eroding the region, and the eroded region is retested until it is sufficiently simple to pass the test. Although this system works well in the given domain, it is not appropriate for free form objects.

Wuescher and Boyer [37] describe a novel scheme for simultaneous surface parameterization and organization in which initial edge estimates are used to choose “seed” points for a robust sequential estimator (RSE) that estimates the parameters of the underlying surface. An iterative reweighted least squares (IRLS) technique is used to obtain robust parameter estimates of each of the patches represented by the seed points. An asymptotically consistent Akaike information criterion (CAIC) is used to compute a goodness measure for the underlying approximating model as

$$CAIC = -2L_p + p(\log(n) + 1) \quad (15)$$

where  $L_p$  denotes the log-likelihood of the model with  $p$  parameters,  $\log(n) + 1$  is the cost of fitting an additional parameter, and  $n$  is the number of observations. Points are

added sequentially by expanding the initial window in four directions and updating the RLS estimates. For each model obtained by an expansion of the initial window and halted by outliers, a normalized measure is found:

$$MCAIC = K \frac{\text{initial CAIC}}{N} \quad (16)$$

where  $K$  is an arbitrary constant and  $N$  is the total number of points in the final maximal window. The model which minimizes  $MCAIC$  is chosen as the underlying parameterization for that seed. This method performs well with planar and biquadratic models in the presence of noise that is not necessarily Gaussian distributed and handles well the presence of statistical outliers.

## 4.2. Proposed Segmentation Scheme

As described above, most methods of finding continuous and homogenous surface regions use clustering or region growing. An important factor for segmentation in object recognition is the need to obtain a consistent segmentation of the surface. The somewhat random nature of the foregoing approaches can confound most object recognition systems. Other techniques require that the surface regions have an underlying parametric representation, which while true of surface types encountered when dealing with industrial parts, may not be valid for free form surfaces. In this section, we will describe a segmentation scheme based on a voting technique that will enable us to extract robust, reliable segments with a high degree of consistency in free form surfaces.

### 4.2.1. Algorithm Description

Our algorithm extends the voting technique developed by Wuescher and Boyer [38] for extracting constant curvature segments in 2D edge maps. The original algorithm partitions a contour into segments of (approximately) constant curvature and segments of rapidly changing curvature ( $\lambda$ -segments). Here, we partition a surface into four types of primitives, in descending order of constraint and, therefore, saliency:

1. Segments having (nearly) constant mean and Gaussian curvatures ( $\mu\gamma$ -segments). These are the most homogeneous and, therefore, most salient of the segment classes, for a given size.
2. Segments of (nearly) constant mean curvature only ( $\mu$ -segments). These are the next most salient segments.
3. Segments of (nearly) constant Gaussian curvature only ( $\gamma$ -segments). The saliency of these segments is essentially identical to that of those of constant mean curvature, except that the calculation of mean curvature is less noise-sensitive and, therefore, more reliable.
4. Segments for which both curvatures change rapidly ( $\lambda$ -segments). These segments are not homogeneous and, of themselves, are not particularly salient. These are just the "leftovers" after all segments of the preceding three types have been extracted, down to a detail (scale) limit.

For a given hierarchical level, significance is ranked in decreasing order of segment size, i.e., largest segment first. The algorithm consists of the following major stages. Throughout the following, "consistent" means "approximately constant" as determined by a tolerance interval.

1. *Curvature computation.* Calculate the mean and Gaussian curvature at each point on the surface, as described above.

2.  *$\mu\gamma$ -segment extraction.* Obtain groupings of regions that are simultaneously consistent in both mean and Gaussian curvature using an overlapped two-dimensional voting technique. Identify reliable contiguous regions from the points associated with these bins by iteratively extracting the largest segments. These are the  $\mu\gamma$ -segments.

3.  *$\mu$ -segment extraction.* Obtain groupings of regions that are consistent in mean curvature using an overlapped one-dimensional voting technique. Identify reliable contiguous regions from the points associated with these bins by iteratively extracting the largest segments. These are the  $\mu$ -segments.

4.  *$\gamma$ -segment extraction.* Obtain groupings of regions that are consistent in Gaussian curvature using an overlapped one-dimensional voting technique. Identify reliable contiguous regions from the points associated with these bins by iteratively extracting the largest segments. These are the  $\gamma$ -segments.

5.  *$\lambda$ -segment recovery.* The segments that remain unmapped either are merged with the surrounding regions if they are insignificantly small, or are given labels of their own and are called  $\lambda$ -segments.

#### 4.2.2. Curvature Voting: $\mu\gamma$ -Consistency

With the mean and Gaussian curvatures ( $\kappa_\mu$  and  $\kappa_\gamma$ , respectively) calculated at each surface point, we next turn our attention to extracting contiguous regions that are consistent in both these measures. We begin by defining tolerance windows  $\tau_\mu$  and  $\tau_\gamma$  for the two curvatures such that, if

$$|\kappa_\mu(p_1) - \kappa_\mu(p_2)| \leq \tau_\mu \quad \text{and} \quad |\kappa_\gamma(p_1) - \kappa_\gamma(p_2)| \leq \tau_\gamma, \quad (17)$$

then the points  $p_1$  and  $p_2$  on the surface *may* be grouped into the same constant curvature bin (a  $\mu\gamma$ -bin). Note that this statement does not imply that all such points *will* be grouped together (even if contiguous), only that they *can* be. That decision is deferred to a later stage in the algorithm.

The grouping process begins with a two-dimensional multivoting technique. Each point on the surface casts votes for *all* bins  $b_{ij}$  in the quantized  $2D$  space  $\mathcal{B}^2$  defined by

$$\mathcal{B}^2 = \{(\kappa_{\mu_i}, \kappa_{\gamma_j}) \mid \kappa_{\mu_i} = \beta_\mu i, \kappa_{\gamma_j} = \beta_\gamma j, \beta_\mu \in \mathcal{R}, \beta_\gamma \in \mathcal{R}, i \in \mathcal{Z}, j \in \mathcal{Z}\} \quad (18)$$

and satisfying the constraints

$$|\kappa_\mu(p) - \kappa_{\mu_i}| \leq \tau_\mu \quad \text{and} \quad |\kappa_\gamma(p) - \kappa_{\gamma_j}| \leq \tau_\gamma, \quad (19)$$

where  $\beta_\mu$  and  $\beta_\gamma$  are the quantization bin sizes of the mean and Gaussian curvatures, respectively. Each vote is tagged with the spatial location of the corresponding point.

When all the votes are tabulated, the result is a form of two-dimensional histogram, built with overlapping votes. The peaks in this histogram represent the combined mean and Gaussian curvature values most likely to form the largest segments under the combined

consistency constraint, although many small segments of similar mean and Gaussian curvatures could generate more votes in one bin than a single large segment in another. Therefore, we sequentially extract the segments according to size.

#### 4.2.3. Sequential Segment Extraction

Working with the 2D histogram, we extract contiguous segments, beginning with the largest and working our way down, that satisfy the constraints imposed by Eq. (17). We start with the bin receiving the most votes,  $b_{ij} = (\kappa_{\mu_i}, \kappa_{\gamma_j})$ , covering the range  $(\kappa_{\mu_i} \pm \tau_{\mu})$  and  $(\kappa_{\gamma_j} \pm \tau_{\gamma})$ . We form groupings of contiguous sets of points casting votes into this bin and identify the largest such set. We examine the segments sequentially, always retaining the current largest segment.

Once the point count of the segments remaining to be considered in the bin is less than the size of its current largest segment, we have recovered the largest segment in this bin, although not necessarily in the data set. There may yet be a larger segment waiting to be found in a bin receiving fewer total votes. Therefore, after finding the largest segment in the most populous bin, we extract the largest segment in the next most populous bin, retaining it if it is larger than the largest segment from the first bin. This procedure continues until the size of the largest recovered segment over all examined bins exceeds the number of votes in the largest remaining bin. At this point, it will be impossible to find a segment in that or any subsequent bin larger than the current largest segment. The points on the surface corresponding to this segment are labeled and deleted from the surface, their votes are removed from the histogram, and the process repeats from the beginning with the now-modified histogram.

We continue until there are no longer any contiguous sets of points that satisfy the consistency constraint of Eq. (17) and which are larger than a prespecified minimum area,  $A_{\min}$ . This minimum area determines the smallest constant curvature segment we will extract, and is chosen according to the nature of the problem domain. It is very likely that segments larger than  $A_{\min}$  remain, but the mean and/or Gaussian curvatures change so rapidly over these regions that it is impossible to fit a constant curvature region of area  $A_{\min}$  or greater. Not surprisingly, our experiments on objects such as soft rubber toys show that, for reasonable tolerance windows, the simultaneous constraints on mean and Gaussian curvatures may leave significant portions of the surface unmapped. This, of course, is the reason for our introduction of the curvature-based saliency hierarchy.

#### 4.2.4. Extracting $\mu$ and $\gamma$ Segments

The next step is to extract those regions consistent in either mean or Gaussian curvature alone. As we have described, these regions are less salient than the  $\mu\gamma$  segments, but are still highly important to the recognition process. As we also mentioned above, the  $\mu$  segments, having consistent mean curvature and more noise immunity, are extracted first, followed by the  $\gamma$  segments (consistent Gaussian curvature). The algorithm for each is the same and is simply a one-dimensional version of the two-dimensional voting and extraction scheme described for the  $\mu\gamma$  segments.

The procedure begins with a projection of the overlapped 2D histogram described in Section 4.2.2 onto the corresponding ( $\mu$  or  $\gamma$ ) axis of the voting space. Then, the tolerance window for the curvature,  $\tau$ , is

$$|\kappa(p_1) - \kappa(p_2)| \leq \tau. \quad (20)$$

The value of  $\tau$  in each dimension is the same as that used in the corresponding dimension in the 2D  $\mu\gamma$  case. Pairs of points satisfying this constraint may be grouped into the same constant curvature segment. Once such a histogram is obtained, the procedure for successive extraction of the largest segments is identical to that described above, with the consistency criterion of Eq. (20) replacing that of Eq. (17).

#### 4.2.5. Completing the Process: $\lambda$ -Segments

At this stage, there will remain portions of the surface not assigned to any segment. Those of area greater than  $A_{\min}$ , the minimum area parameter, are labeled as  $\lambda$  segments. This simply means that both curvatures are changing rapidly over this region. We can still use these segments in the structural surface description for recognition, since such a segment may still be described by attributes such as area and shape, as well as its location and orientation relative to other segments.

Segments smaller than  $A_{\min}$  are not useful descriptors of the surface; they are simply merged with the surrounding extracted segments by assigning each point to the closest segment already extracted.

#### 4.2.6. Algorithm Parameters

We have four classes of parameters in the algorithm to set: the Gaussian filter width ( $\sigma$ ), the mean and Gaussian curvature tolerances ( $\tau_\mu$  and  $\tau_\gamma$ ), the mean and Gaussian curvature quantization bin sizes ( $\beta_\mu$  and  $\beta_\gamma$ ), and the minimum segment size ( $A_{\min}$ ). Here we shall describe the qualitative effect each of these parameters has on the final organization. Experimental results that demonstrate these effects are presented below. In these experiments we fix the radius  $R$  over which we establish the coordinate system at 6 mm. This parameter is not especially critical, although a radius much larger than the inherent scale of the surface undulations would be ill-advised.

1. *Scale of the Gaussian smoothing filter ( $\sigma$ ).* Large values of  $\sigma$  lead to smoother surfaces with less detail and produce surface decompositions into fewer, larger segments. Smaller values of  $\sigma$  preserve more detail and yield more, smaller segments. But the impact of this parameter on the final segmentation is not as pronounced as one might expect because we compute the curvatures analytically by fitting a biquadratic polynomial to Monge patches, which also smooths the surface.

2. *Curvature tolerances ( $\tau_\mu$  and  $\tau_\gamma$ ).* These parameters control the strictness of the two consistency criteria applied to the surface patches and effect a scale selection. Note that these tolerances are also related to the Gaussian filter width  $\sigma$ ; a higher value of  $\sigma$  produces a smoother surface that will permit a smaller tolerance window to achieve a similar segmentation, although (again) that interaction is lessened by the smoothing built into the surface fitting operation.

3. *Curvature quantization bin sizes ( $\beta_\mu$  and  $\beta_\gamma$ ).* These values are not especially critical, so long as the bin size is smaller than the tolerance. Our experiments indicate that  $\beta = \tau/5$  is a good choice for both curvatures; this is consistent with the observations of Wuescher and Boyer in their contour decomposition algorithm.

4. *Minimum segment size ( $A_{\min}$ ).* This parameter also allows us to control the amount of detail that we wish to preserve by defining the lower limit on the size of any constant curvature segment of interest. Smaller  $A_{\min}$  yields fewer  $\lambda$  segments; larger, more.

### 4.3. Experimental Organization Results

We first analyze the effect of each of the algorithm parameters on the surface organization. Next, we present segmentation results for several objects that demonstrate the stability of the algorithm with respect to various surface types. We have conducted a large number of experiments; our results here are but a small sample of typical results. All object dimensions given below are approximate.

All range images were collected using a Minolta Vivid 700 laser range scanner. The mean range image density for our sensor and scene geometries is approximately 0.5 mm. The sensor range accuracy is better than 0.2 mm, even for reasonably oblique geometries, based on controlled tests. *We do no resampling or other manipulation of the data anywhere in the process.* We simply group mesh vertices; this eliminates a source of possible error in other methods as we also avoid resampling in the smoothing and curvature calculations.

#### 4.3.1. Stability with Respect to Parameter Changes

We begin the experiments by choosing a set of *standard* parameters (Table 3). In the absence of ground truth (more on this below), these parameters were chosen empirically to provide a “good” segmentation over a fairly large set of surfaces. The effect of each parameter is analyzed by perturbing it about the standard value while holding the others constant. We will use a range model of a molded-foam toy cow (125 mm nose to tail) to illustrate. In recognition we compute the organization for a range of (effective) scales; the standard values simply provide a reference point from which to study the algorithm’s behavior.

To preserve visual contrast in these images, color is used only to distinguish regions; there is no other information—such as saliency category or mean curvature value(s)—encoded in the color scheme. Below, in the object-by-object segmentation results, the regions are color-coded by type.

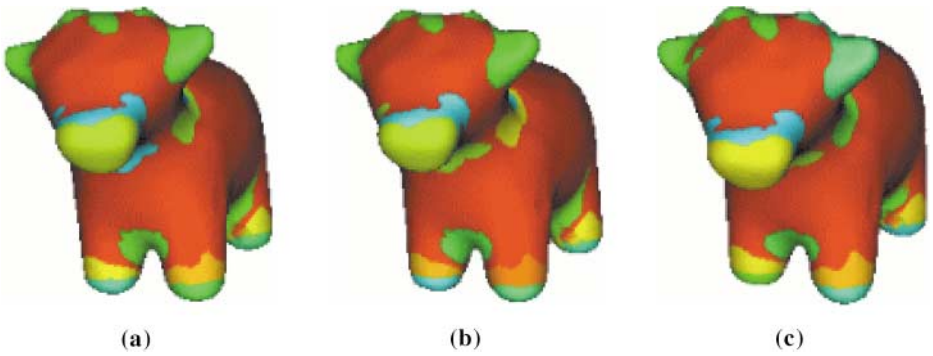
Figure 1 illustrates the effect of changing the quantization bin size by  $\pm 50\%$  with respect to the standard value. The surface organization is quite stable over this range; so long as the quantization is reasonably finer than the tolerance window, the segmentation is relatively unaffected.

Figure 2 illustrates the effects of changing the tolerance window sizes by (nearly)  $\pm 50\%$ . A gradual change in detail is observed as the window sizes are changed. Smaller window sizes impose tighter homogeneity constraints, producing smaller segments with more detail. As the tolerance levels are increased, details gradually disappear and segments begin to merge to yield larger, more salient segments. Unlike fixed-threshold techniques, this parameter permits a gradual and graceful change in surface organization, a (pseudo) multiresolution segmentation, that can be used to advantage in recognition.

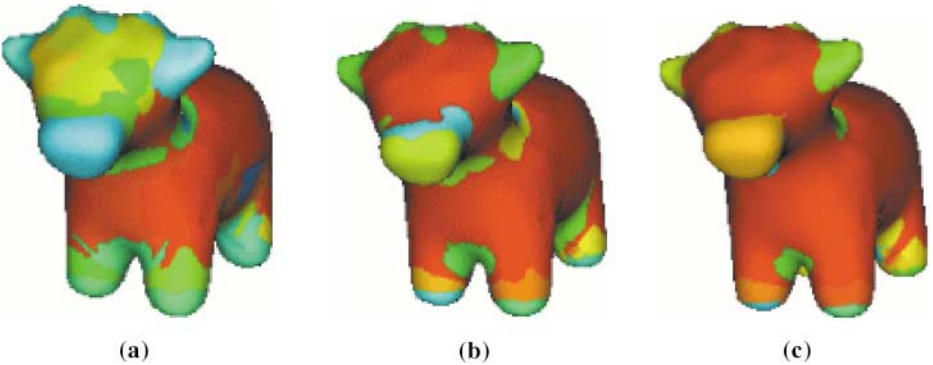
**TABLE 3**  
Standard Values for the Six Parameters

Parameter	$\beta_\mu$	$\beta_\gamma$	$\tau_\mu$	$\tau_\gamma$	$\sigma$	$A_{\min}$
Standard Value	$5 \times 10^{-3}$	$5 \times 10^{-4}$	$25 \times 10^{-3}$	$25 \times 10^{-4}$	2	500
Units	$\text{mm}^{-1}$	$\text{mm}^{-2}$	$\text{mm}^{-1}$	$\text{mm}^{-2}$	mm	mesh pts.

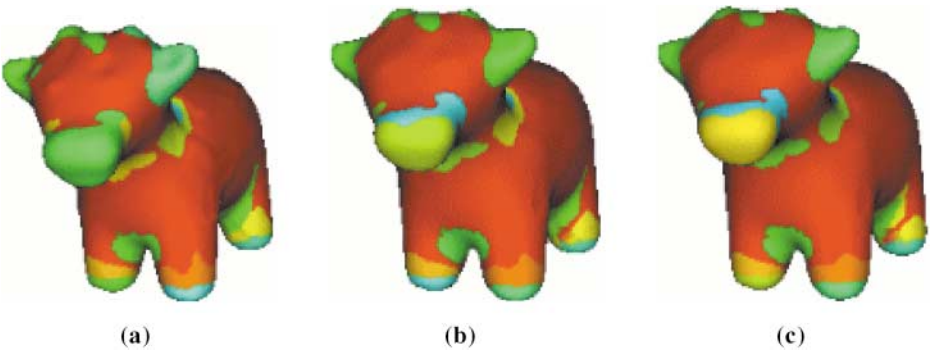
*Note.* Units assume that the original range measurements are expressed in millimeters, as is the case for our Minolta unit. 500 mesh points ( $A_{\min}$ ) correspond to a patch of about  $125 \text{ mm}^2$  for a flat surface with normal facing the sensor.



**FIG. 1.** Variation of segmentation with bin size: front view of cow segmented with (a)  $\beta_\mu = 2.5 \times 10^{-3}$ ,  $\beta_\gamma = 2.5 \times 10^{-4}$ ; (b)  $\beta_\mu = 5 \times 10^{-3}$ ,  $\beta_\gamma = 5 \times 10^{-4}$ ; (c)  $\beta_\mu = 7.5 \times 10^{-3}$ ,  $\beta_\gamma = 7.5 \times 10^{-4}$ .



**FIG. 2.** Variation of segmentation with tolerance window: front view of cow segmented with (a)  $\tau_\mu = 15 \times 10^{-3}$ ,  $\tau_\gamma = 15 \times 10^{-4}$ ; (b)  $\tau_\mu = 25 \times 10^{-3}$ ,  $\tau_\gamma = 25 \times 10^{-4}$ ; (c)  $\tau_\mu = 35 \times 10^{-3}$ ,  $\tau_\gamma = 35 \times 10^{-4}$ .



**FIG. 3.** Variation of segmentation with smoothing parameter: front view of cow segmented with (a)  $\sigma = 1$ , (b)  $\sigma = 2$ , (c)  $\sigma = 3$ .

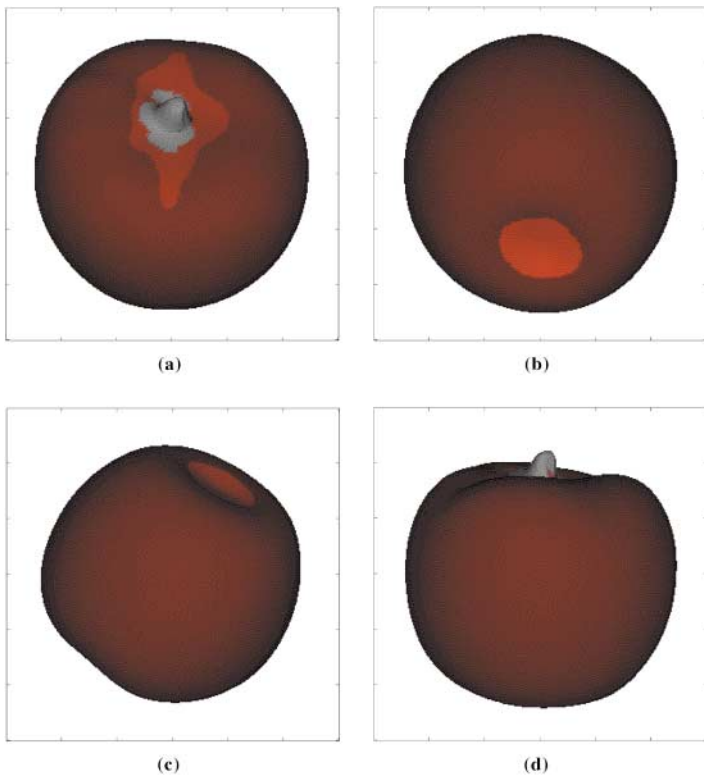


FIG. 4. Four views showing segmentation results for the apple, segmented using standard parameters.

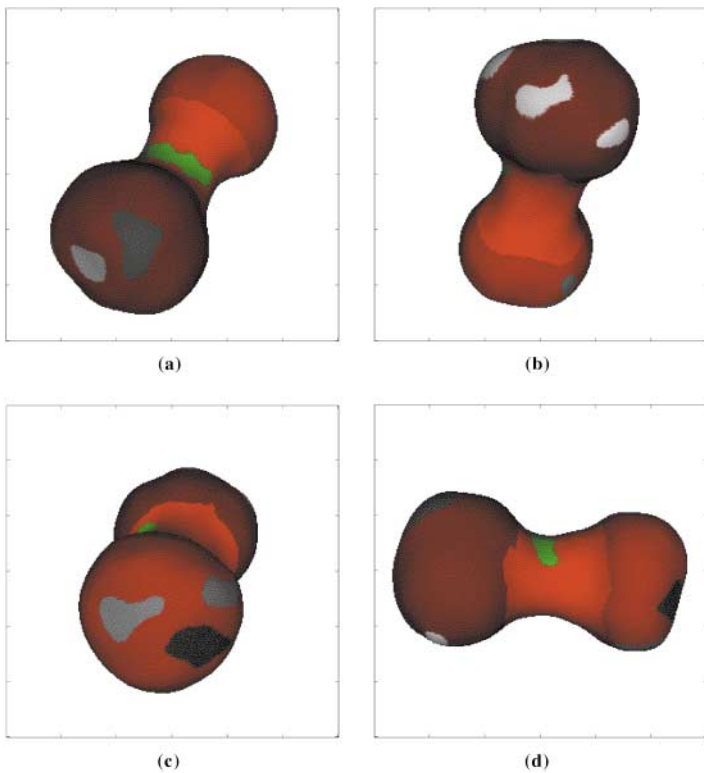


FIG. 5. Four views showing segmentation results for the dumbbell, segmented using standard parameters.

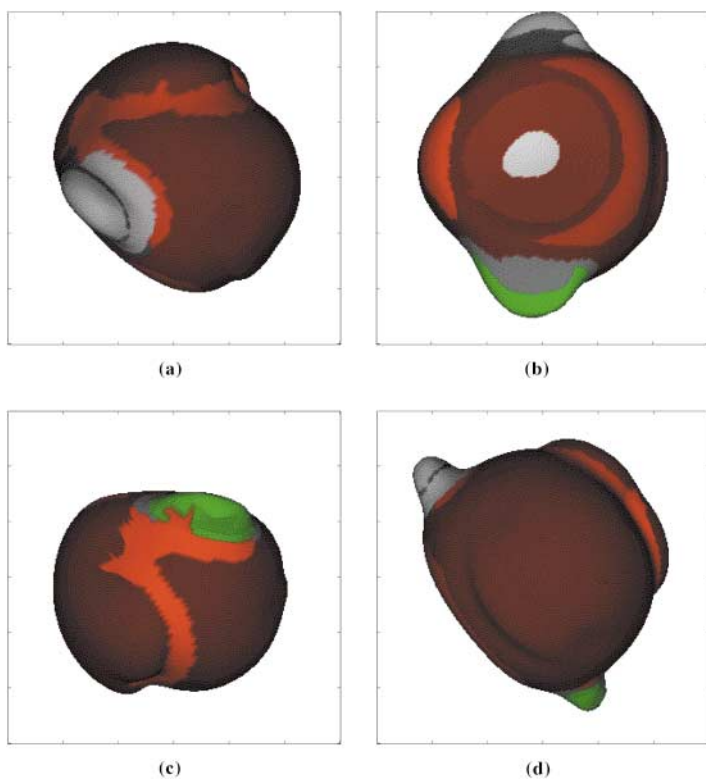


FIG. 6. Four views showing segmentation results for the whale, segmented using standard parameters.

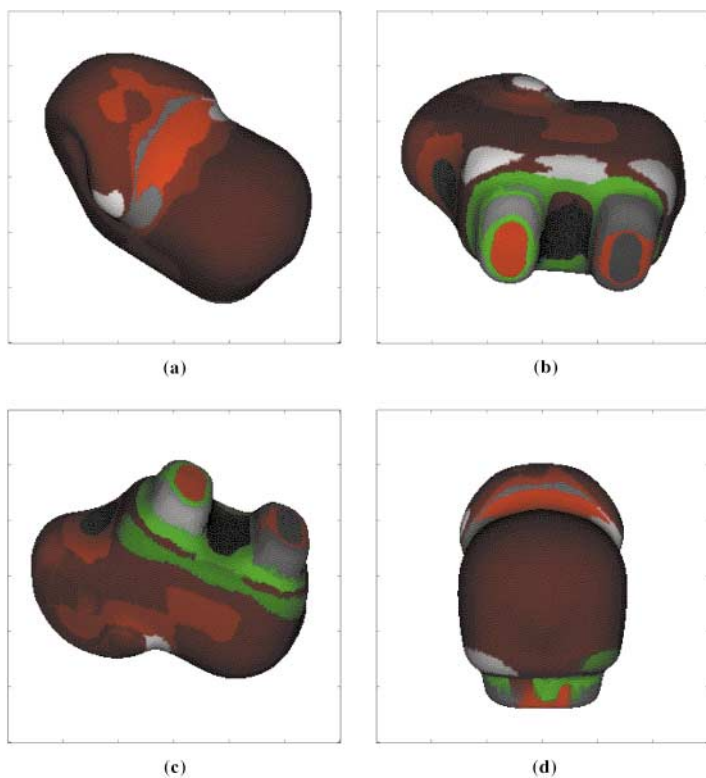


FIG. 7. Four views showing segmentation results for the lamb, segmented using standard parameters.

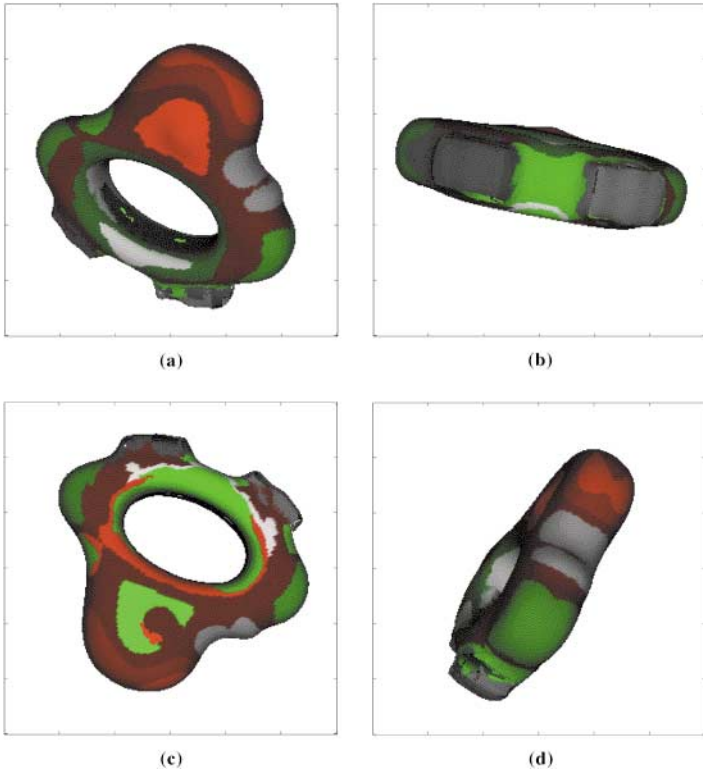


FIG. 8. Four views showing segmentation results for the duckrattle, segmented using standard parameters.

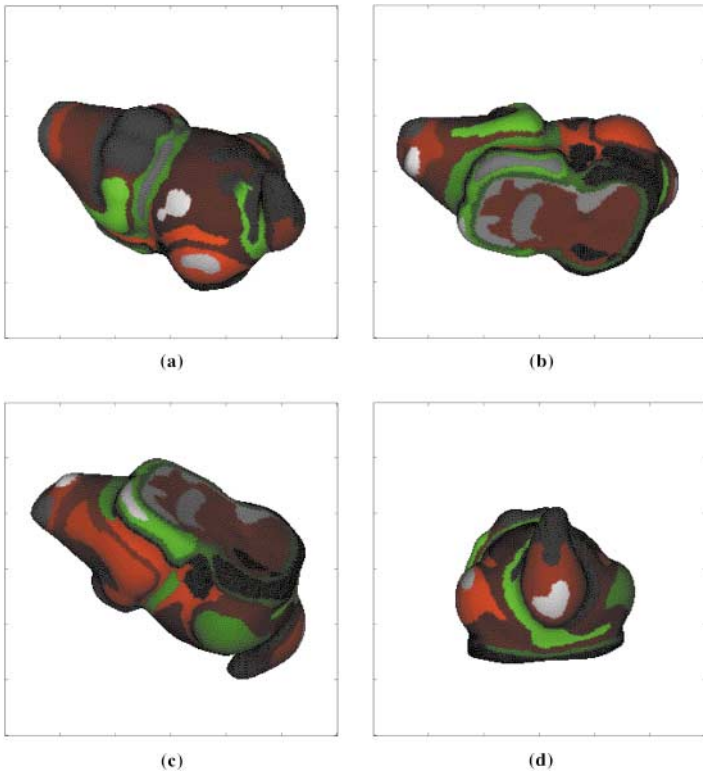


FIG. 9. Four views showing segmentation results for the crocodile, segmented using standard parameters.

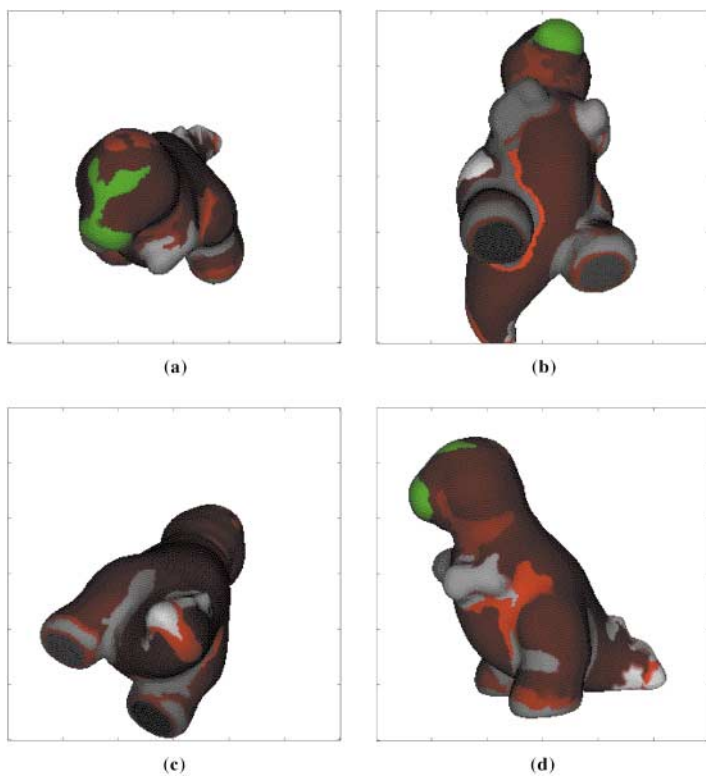


FIG. 10. Four views showing segmentation results for dinosaur 1, segmented using standard parameters.

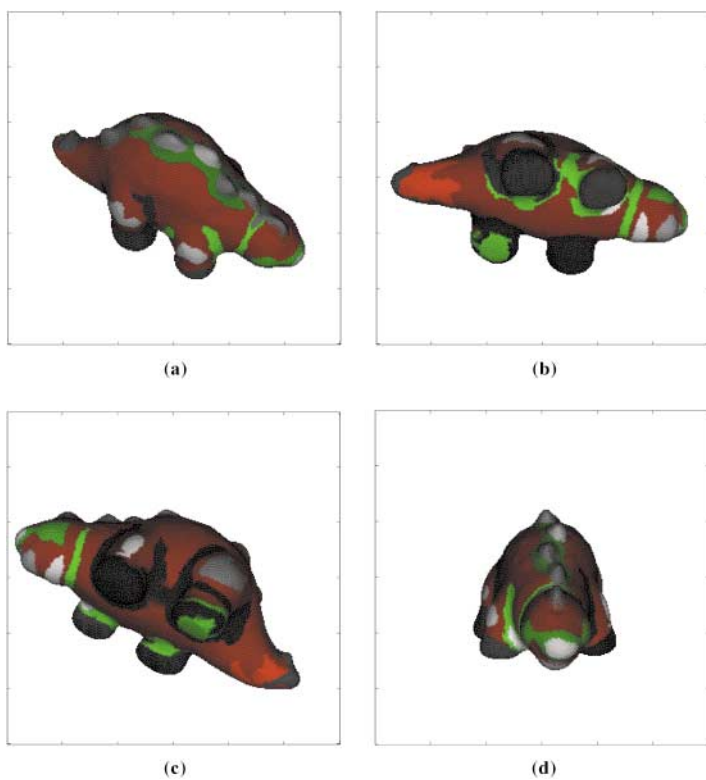


FIG. 11. Four views showing segmentation results for dinosaur 2, segmented using standard parameters.

Figure 3 illustrates the effects of changing  $\sigma$  by  $\pm 50\%$ . Although a gradual change in detail is discernible, the effect of this parameter is suppressed by the polynomial fitting used in computing the mean and Gaussian curvatures, as we intend. We simply use the Gaussian filter at a single scale for modest noise mitigation.

Changing the minimum permissible segment size produces the expected result. As the size is reduced, more (small) consistent pieces of the surface are accepted into one of the  $\mu\gamma$ ,  $\mu$ , or  $\gamma$  categories, leaving fewer  $\lambda$  segments. As the minimum size is increased, more  $\lambda$  segments result, some very large.

#### 4.3.2. Organization Results Using Standard Parameters

In this section we present segmentation results for various objects using the standard parameters. We begin with simple cases for which the expected segmentations are relatively straightforward and for which the ground truths, to the extent that the term is meaningful, are somewhat intuitive. We then move on to more complex surfaces. In these tests, surfaces are organized at a fixed scale. For some of the detail-rich objects, a finer scale organization would be superior. These observations support the multiscale approach for object recognition presented later in the paper.

In these images, the regions are color-coded as follows:  $\mu\gamma$  segments are shades of red,  $\mu$  segments are green,  $\gamma$  segments are blue, and  $\lambda$  segments are gray. The assignment of colors for each region type proceeds from dark (earlier extraction, larger region) to light (later extraction, smaller region). In some cases, regions of the same type and nearly the same size are adjacent, resulting in a rather low-contrast border, so the reader may need to look carefully for some boundaries.

*1. Apple.* Figure 4 shows four views of one of the simplest test cases used in this series of experiments, a foam toy apple with a radius of about 35 mm. The surface is almost completely spherical, except for the “stem” on top and a depression of the surface at the bottom. Due to its simplicity, this case also serves to provide a reality check for the algorithm. A sphere of radius  $R$  is expected to have a constant mean curvature of  $1/R$  and a constant Gaussian curvature of  $1/R^2$ , and this is what we see. Of the four segments, the stem is  $\lambda$ , the rest are all  $\mu\gamma$ .

*2. Dumbbell.* Figure 5 depicts the segmentation of a dumbbell (130 mm in length) made by amateur hands from children’s modeling clay. The dumbbell is far from ideal; the “spurious”  $\lambda$  segments in the bulbs at either end correspond to actual shallow depressions in the body. So, again, this result is as close to ideal as one can reasonably expect. This object consists of three  $\mu\gamma$  segments (the bulbs and the main part of the neck), one  $\mu$  segment (in the neck), and eight  $\lambda$  segments.

*3. Whale.* Our next test surface is that of a toy rubber whale (70 mm fin–fin) shown in Fig. 6; (a) is a side view. This object is more interesting due to the existence of a few characteristic features that form the fins and the tail. The fins form regions of extreme curvature and are easily segmented, although at this scale one fin tip emerges as a  $\mu$  segment, the other,  $\lambda$ . The tail is evident only as a jump edge that gradually merges with the main body and head of the whale. As a result, the regions that characterize the boundary between the tail and the body appear as thin  $\mu\gamma$  strips in the segmentation. The belly of the whale forms a concave region and is seen as a set of roughly concentric segments (b). Other small segments that characterize the mouth of the whale are also evident. This object consists of six  $\mu\gamma$ , one  $\mu$ , two (small)  $\gamma$ , and four  $\lambda$  segments.

4. *Lamb*. Figure 7 shows four views of a plastic lamb (76 mm long). The body of the lamb is seen as a large  $\mu\gamma$  region and forms the most salient segment of the surface. The base of the lamb, with the “legs,” forms the next most significant region. The region between the back and the neck and the region under the chin form negative curvature regions and are discriminated. What remains is mostly unstable (transition) regions that appear as  $\lambda$ -segments. The flat region at the bottom of the “feet” in front is  $\mu\gamma$  surrounded by  $\mu$ ; at the rear it is  $\lambda$  surrounded by  $\mu\gamma$ . At a finer scale, these features stabilize and are more consistent. At this scale, the lamb has 8  $\mu\gamma$ , 3 each  $\mu$  and  $\gamma$ , and 12  $\lambda$  segments.

5. *Duck Rattle*. The duck rattle (75 mm outside diameter) shown in Fig. 8 is small and rich in features. The large amount of detail results in the object being segmented into a larger number of smaller segments. In fact, the greater detail would require a finer resolution for a faithful representation of its features. Nevertheless, given the object’s complexity and its (perhaps not obvious) asymmetry, the segmentation is quite reasonable. The protrusion to the top in (a) consists of several  $\mu\gamma$  segments. The bumps show up as  $\lambda$  segments, and most of the outer part of the torus is of type  $\mu$ . This object is atypical among our set in that it shows more  $\mu$  segments (eight) than  $\mu\gamma$  (six); it also shows one  $\gamma$  segment (the interior surface of the torus) and seven  $\lambda$  segments.

6. *Crocodile*. This is another feature-rich object (100 mm long), but one which also contains relatively large smooth regions. The segmentation in Fig. 9 shows some stable regions and quite a few  $\lambda$  regions, especially the eyes and in the flat region at the bottom—where we would expect to have a single large plane segment (although a significant part of the bottom is correctly identified as  $\mu\gamma$ ). The problem is that planar regions yield linearly dependent rows in matrix  $\mathbf{A}$  (Eq. (9)) that lead to  $\mathbf{A}^T \mathbf{A}$  being nearly singular. Such near-singularity can produce unreliable curvature estimates that in turn lead to unstable segmentation. However, this problem is only exacerbated in traditional segmentation approaches that detect, for example, curvature sign changes. The snout and back are mostly made up of  $\mu\gamma$  segments; the neck, ears, and tail transitions are  $\mu$  segments. The crocodile presents 9  $\mu\gamma$ , 6  $\mu$ , 0  $\gamma$ , and 14  $\lambda$  segments.

7. *Dinosaurs*. We conclude these results with another set of objects that provide a rather equitable mix of features and segments. Four views of each dinosaur are represented in Figs. 10 and 11. Dinosaur 1 (135 mm tall) has a relatively large featureless region ( $\mu\gamma$ ) that is broken by a few  $\lambda$  regions near the tail and hips. Transitions from the body to the appendages also produce  $\lambda$  segments, as expected. The snout and eye area form a  $\gamma$  segment. In this object we find eight  $\mu\gamma$ , one  $\mu$ , two  $\gamma$ , and eight  $\lambda$  segments. Dinosaur 2 (165 mm long) is characterized by a series of spikes, realized as  $\lambda$  segments, but very stable across multiple images, that are present along the length of its back and head, projecting from a long  $\mu$  ridge area. Such features can be important for recognition from single views of the object. The neck is of type  $\mu$ , as are the joints where the legs attach. The main body and tail and most of the head consist of  $\mu\gamma$  segments. The eyes and the bridge of the nose are a  $\mu$  segment. Dinosaur 2 yields two  $\mu\gamma$ , four  $\mu$ , zero  $\gamma$ , and 16  $\lambda$  segments.

Over this set of objects, the algorithm recovered 45  $\mu\gamma$ , 24  $\mu$ , 8  $\gamma$ , and 70  $\lambda$  segments using standard parameters. It is interesting that relatively few regions of these surfaces were consistent in Gaussian curvature alone. The  $\gamma$  segments that were recovered were all quite small. We are not prepared to declare these segments superfluous, however. Although they are small and somewhat rare for these objects at this scale, it is possible that for other

objects and/or at other scales, they may be more significant. Also, they may still provide useful object discrimination, when present. This merits more investigation.

The reader may wonder whether the extraction sequence ( $\mu$  segments precede  $\gamma$  segments) has an impact on the segmentation. In our investigations, we were unable to detect any significant impact; the segmentations were nearly identical. As noted above, we extract  $\mu$  segments first because we expect them to be more stable in the presence of noise.

## 5. SURFACE DESCRIPTION AND OBJECT RECOGNITION

Because we do not impose surface class definitions, and because we do not mark local events such as inflections directly, there is no such thing as ground truth for comparison. Indeed, regardless of segmentation strategy, the concept of ground truth hardly applies to free-form objects. Therefore, to justify our approach to surface organization we demonstrate its effectiveness in a simple object recognition system.

We begin by building a graph theoretical description of the surface (whole object, for modeling; one view, for recognition) using the organization described above. In the simplest model-based systems, objects are recognized by finding one or more distinguishing features and the relations between (or among) them [39, 40]. More complex systems however, require the use of sophisticated descriptions of the scene [41]. In this chapter we will describe a technique to build reliable structural descriptions that can be used in a variety of recognition systems, before moving on to describe one such system that uses an interpretation tree based on inexact graph matching.

### 5.1. Prior Work in Object Representation and Matching

Chin and Dyer [42] broadly classify model-based recognition systems as either graph theoretic or non-graph-theoretic, based upon how the objects in the scene and in the library are described. In graph theoretic systems [43–47], an object is described as an assembly of several parts, each represented as a node in a graph carrying information characterizing the part. The arcs in this graph encode the relationships among the parts. Recognition involves finding a subgraph isomorphism between the scene graph and each of the model graphs. In non-graph-theoretic methods [40, 48, 49], local features such as coordinate values of point features and surface normals are used to describe the object. Recognition entails matching such features in scenes to those in the model base using a constrained tree search or a more sophisticated hashing or indexing strategy that is less sensitive to modelbase size.

Sengupta and Boyer [50] observe that although non-graph-theoretic methods traditionally offer superior performance for large libraries, graph theoretic methods provide more complete descriptions of the models and are not necessarily optimized for recognition alone. Systems employing graph theoretic models can reason more naturally about missing or occluded parts and can allow for partial recognition of an object as an instance of a broader appearance class. As such, graph theoretic systems offer significant potential for (more) generic object recognition, an option not obviously available with appearance based methods.

Wong *et al.* [43] use a representation scheme based on an attributed hypergraph representation. Initially, an elementary area attributed graph is constructed from the geometric primitives of a distinct surface (such as surface boundaries). Then, a primitive block-attributed graph (that represents blocks bounded by surfaces with no concave angular relations between their surfaces) is constructed with geometric primitives such as shapes and areas as

the vertex attributes and angular orientation, etc., forming the edge attributes. Finally, a hypergraph that uses primitive blocks such as polyhedra or blocks with complete or partial cylindrical, conical, or planar surfaces is constructed that can potentially represent more complicated objects. The matching process is then a morphism-finding operation that can be restricted to the comparison of subgraphs depicted by hyperedges.

Oshima and Shirai [44] begin by grouping points on the surface into small planar surface elements. The planar elements are then merged to obtain larger elementary regions that are classified as planar or curved. Adjacent curved regions are then merged to produce larger regions called global regions to which quadric surfaces can be fitted. A scene description is then built in terms of the properties of the regions and their relationships.

Flynn and Jain [45] describe a geometric inference engine which produces a relational graph representation of a 3-D object from a CAD model specified using an industry design standard (IGES). The nodes of the graph primarily consist of two types of attributes: lines and curves, and surfaces. The former consist of line segments, circular arcs, parametric spline curves, and composite curves, while the latter can be planes or surfaces of revolution such as cylinders, cones, spheres, and toroids. Estimated node attributes depend upon the node type, while binary link attributes include orientation, proximity, and other miscellaneous relations. The matching problem was addressed in a later paper [51]. Descent in an interpretation tree implies an increasing level of commitment to a particular identity hypothesis for a single model object. Unary predicates such as area, type (plane, cylinder, or sphere), and intrinsic parameter (for curved surfaces), together with binary predicates such as rotation validity, orientation, visibility, and parallel plane, are used to prune the search tree. Once object pose has been estimated, a synthetic range image and the corresponding surface segmentation of the hypothesized model in the hypothesized position is generated and a matching score is calculated by a pixel-to-pixel comparison between the real and synthetic images.

Flynn and Jain [52] also propose a matching technique that uses invariant features to index into interpretation tables. Unlike the technique in [51] where associations between scene and model surface patches are proposed without any prior knowledge (other than weak constraints such as surface type) and subsequently tested using predicates specific to the models, in this scheme invariant scene features are used to propose associations (via table look-up) that provide an initial set of pre-constrained hypotheses which can be pruned further by applying the same predicates used in the interpretation tree search.

Kim and Kak [47] propose a computational procedure that uses the notion of maximal matching in bipartite graphs (a problem for which a low-order polynomial time solution is known [53]) to prune large segments of the search space. Since bipartite matching by itself is incapable of imposing relational constraints, a bipartite matching scheme embedded in discrete relaxation is used to severely prune the set of possible hypotheses.

Sengupta and Boyer [50] address the problem of hierarchically organizing structural modelbases using an information theoretic criterion. In such an organization, the objects in the modelbase are modeled in the form of random parametric structural descriptions (RPSDs), while objects in the scenes are modeled as parametric structural descriptions (PSDs). The organization process is driven by pairwise dissimilarity values between RPSDs that are most useful when the descriptions being compared are sufficiently similar to make the measurement of their differences meaningful. For a modelbase of structurally homogeneous objects, this method achieves a significant speedup in recognition compared to a linear search of the modelbase.

Sengupta and Boyer [54] also propose a two-tiered organization of the modelbase using an eigenvalue or spectral-based representation for CAD models. The eigenvalues provide a gross description of the objects and help to divide a large modelbase into structurally homogeneous partitions. An intermodel dissimilarity measure based on these gross features is defined to partition the modelbase. Object models in each partition are organized hierarchically as described in [50]. In recognition, gross features computed from a surface cluster are used to prune the modelbase by selecting a few “favorable” partitions in which the correct object model is likely to lie. Such an organization provides a significant increase in recognition speed even for relatively small modelbases.

Techniques for the representation of free-form 3D objects for computer vision applications have been studied for over a decade; [55] is a recent survey of representation and matching strategies. Besl [56] identified the key elements of free-form matching and also proposed the triangular mesh as well-matched to vision problems [57]. The utility of meshes in this context has been documented in [58, 59]. Parametric surfaces of various types have also been employed in vision applications (e.g., [60]) and offer less verbose representations at the expense of complex fitting and control problems.

Parvin and Medioni [61] proposed a system for object description and matching from multiple range views. Notable in this work is the labeling and grouping of 3D edges (creases and limbs) and vertex description for model-building. A dynamic system formulation is used and the system’s convergent state represents a consistent labeling and is the basis for a graph representation of the object. A graph-matching technique with various correspondence costs for graph edges and nodes is used for object recognition, with a constraint satisfaction network providing overall control.

## 5.2. Representation and Matching

We use the surface regions obtained from organization as nodes in an attributed graph whose arcs express the geometric relationships between them. Each node  $n_i$  contains unary information about the surface segment, such as shape, orientation, and location. For each pair of nodes  $n_i$  and  $n_j$ , we create an arc (link)  $l_{ij}$  that encodes the binary relations relative size, position, and orientation. A more detailed description of these attributes follows.

### *Node Attributes*

1. *Segment type*:  $T(i) \in \{\mu\gamma, \mu, \gamma, \lambda\}$ .

2. *Segment area*:  $A_{abs}(i)$ . We approximate this with the number of mesh points covered by the segment; this is valid for fairly dense meshes at the same sampling density, such as ours.

3. *Average Gaussian curvature*:  $K_\gamma(i)$ .

4. *Average mean curvature*:  $K_\mu(i)$ .

5. *Orientation*:  $N(i)$ . The principal component whose eigenvalue is at maximal variance with the remaining two represents the orientation of the segment. The orientation of a cylindrical segment, for instance, would be along its axis, while that of a planar segment would be its normal. If  $\lambda_1(i)$ ,  $\lambda_2(i)$ , and  $\lambda_3(i)$  represent the eigenvalues of the data set along the principal directions  $\Lambda_1(i)$ ,  $\Lambda_2(i)$ , and  $\Lambda_3(i)$ , where  $\lambda_1(i)$  and  $\lambda_2(i)$  represent the

minimum and maximum eigenvalues, then the orientation of the segment  $\mathbf{N}(i)$  is given by

$$\mathbf{N}(i) = \mathbf{A}_{\arg \max(|\lambda_1 - \lambda_3|, |\lambda_2 - \lambda_3|)}. \quad (21)$$

6. *Centroid:  $C(i)$ .*

### Arc Attributes

1. *Relative orientation:  $\Theta(i, j)$ .* The relative orientation between two nodes  $n_i$  and  $n_j$ , defined as the smallest angle between the oriented axes of the two segments, where their respective orientations are as defined in Eq. (21), and measured as

$$\Theta(i, j) = \cos^{-1} |\mathbf{N}(i)\mathbf{N}(j)|. \quad (22)$$

2. *Intercentroidal distance:  $D(i, j)$ .* The Euclidean distance between the centroids of the two surface segments.

The arc attributes are all derivatives of the node attributes; they do not need to be pre-computed.

## 5.3. Graph Matching

As is standard in graph theoretical formulations, recognition is accomplished by constructing an inexact subgraph isomorphism between the graph describing the object model and that describing the arrangement of surface primitives in the scene. Given a scene  $\mathcal{S} = \{\mathbf{R}^1, \mathbf{R}^2, \dots, \mathbf{R}^p\}$  containing the set of  $p$  objects, we need to find the set of models,  $\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_m\} \in \mathcal{M}$ , that provide the best set of matches, where  $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_m\}$  is a model description database (modelbase, for short) containing  $m$  models. In this simple demonstration, we will restrict our attention to the case  $p = 1$ , i.e., single object scenes. Since we work with a single viewpoint, most of the object will not be visible due to self-occlusion and the problem is demonstrative of the general case, apart from the range map version of the perceptual organization issue, which is a separate problem.

Stated from the viewpoint of a graph matching problem, our intention is to find the largest subgraph in the model description for which every node maps onto a node in the scene graph according to a set of constraints. To avoid exhaustive search, we use an incremental matching scheme that prunes the tree at each successive stage.

### Algorithm Description

1. *Significance reordering.* To maximize the efficiency of the matching process, we consider the most salient segments first. Because our surface organization algorithm produces segments in a natural (partial) ordering by saliency (partial), this step is almost automatic. Although segment extraction can be embedded within the matching process to take advantage of this natural ordering, the demonstration system here begins from a completely segmented scene. Since we have all the segments at hand, we merge the  $\mu$  and  $\gamma$  segment lists by size. If segmentation were embedded within matching, then the small (er)  $\mu$  segments would be considered before the large(r)  $\gamma$  segments; we believe this would be an effective tradeoff, permitting faster recognition in most domains.

2. *Applying unary constraints.* We next find the set of nodes  $\mathbf{M}_i \subset \mathcal{M}$  in the modelbase compatible with node  $\mathbf{S}_i$ , the  $i$ th most salient node in the scene. A unary constraint that gives a measure of compatibility is used to obtain the subset  $\tilde{\mathbf{M}}_i = \{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_m\}$  (a *match pool*) such that the following compatibility constraints are satisfied:

- (a) **Area constraint:**  $d_{\mathbf{S}_i, \mathbf{M}_j}(1) = d(A_{\mathbf{S}_i}, A_{\mathbf{M}_j})$ , where  $A_{\mathbf{S}_i}$  and  $A_{\mathbf{M}_j}$  represent the surface areas of  $\mathbf{S}_i$  and  $\mathbf{M}_j$ , respectively.
- (b) **Gaussian curvature constraint:**  $d_{\mathbf{S}_i, \mathbf{M}_j}(2) = d(K_{\mathbf{S}_i}, K_{\mathbf{M}_j})$ , where  $K_{\mathbf{S}_i}$  and  $K_{\mathbf{M}_j}$  represent the average Gaussian curvature,  $K_\gamma$ .
- (c) **Mean curvature constraint:**  $d_{\mathbf{S}_i, \mathbf{M}_j}(3) = d(k_{\mathbf{S}_i}, k_{\mathbf{M}_j})$ , where  $k_{\mathbf{S}_i}$  and  $k_{\mathbf{M}_j}$  represent the average Gaussian curvature,  $K_\mu$ .

In each case, the distance measure  $d(a, b)$  is defined as

$$d(a, b) = \frac{|a - b|}{\max(|a|, |b|)}. \quad (23)$$

The compatibility  $\xi_0$  between nodes  $\mathbf{S}_i$  and  $\mathbf{M}_i$  is the weighted average

$$\xi_0 = \sum_{k=1}^3 w_k d_{\mathbf{S}_i, \mathbf{M}_j}(k), \quad (24)$$

where  $w_k$  represents the weight for the  $k$ th constraint and the weights sum to unity; these experiments use uniform weighting. The two nodes  $\mathbf{S}_i$  and  $\mathbf{M}_i$  are considered  $\xi_0$  compatible if  $\xi_0 < D_0$ , a threshold.

In this combination of distances, and in the total match quality measures below, we use the (possibly) weighted average of measures made in different units. Our intent here is simply to demonstrate that our segmentations can serve as the basis of a free form object recognition system. As the reader will see, at least in these experiments, this simple approach to combining the error measures works quite well. Nevertheless, a more carefully developed match measure would be strongly advised in a production system.

3. *Applying binary (arc) constraints.* The set of node pairs  $\{P_1^i, P_2^i, \dots, P_m^i | P_k^i \in \langle \mathbf{S}_i, \tilde{\mathbf{M}}_i \rangle, \tilde{\mathbf{M}}_i \subset \mathcal{M}\}$  are those eligible to be placed in the interpretation tree at depth  $i$ . Our next step is to obtain a measure that will decide into which paths of the tree these pairs of nodes can be placed at this depth such that the compatibility constraint for each of those paths is satisfied. Let  $\mathbf{T}_1^{i-1}, \mathbf{T}_2^{i-1}, \dots, \mathbf{T}_n^{i-1}$  be the  $n$  possible paths along the tree at the  $(i-1)$ th level where each path  $\mathbf{T}_k^{i-1}$  contains  $t_k$  pairs of nodes. Then we need a compatibility relation between the candidate pair  $P_l^i$  and the paths  $\{\mathbf{T}_{1..t_k}^{i-1}, \mathbf{T}_{1..t_k}^{i-2}, \dots, \mathbf{T}_{1..t_k}^1\}$ . To get this, we first define a set of binary constraint measures between two generic pairs of nodes  $\langle \mathbf{S}_i, \mathbf{M}_j \rangle$  and  $\langle \mathbf{S}_k, \mathbf{M}_l \rangle$  as follows:

- (a) **Uniqueness:**  $\langle \mathbf{S}_i, \mathbf{M}_j \rangle$  and  $\langle \mathbf{S}_k, \mathbf{M}_l \rangle$  are said to be  $\xi_1$ -compatible if and only if  $\mathbf{S}_i \neq \mathbf{S}_k$  and  $\mathbf{M}_j \neq \mathbf{M}_l$ .
- (b) **Direction consistency:** If  $\theta_1$  and  $\theta_2$  denote the angles between the orientation of  $\langle \mathbf{S}_i, \mathbf{S}_k \rangle$  and  $\langle \mathbf{M}_j, \mathbf{M}_l \rangle$ , respectively, then a measure of the directional consistency is given by

$$\xi_2 = \frac{|\theta_1 - \theta_2|}{\frac{\pi}{2}}. \quad (25)$$

(c) **Distance consistency:** If  $L_1$  and  $L_2$  denote the distance between the centroids of  $\mathbf{S}_i$  and  $\mathbf{S}_k$ , and  $\mathbf{M}_j$  and  $\mathbf{M}_l$ , respectively, then a measure of centroidal distance consistency is given by

$$\xi_3 = \frac{|L_1 - L_2|}{\max(L_1, L_2)}. \quad (26)$$

The corresponding path consistencies  $\theta_{err}$  and  $D_{err}$  for  $\mathbf{T}_k^i$  are evaluated as the mean values of each of  $\xi_2$  and  $\xi_3$  over all possible combinations of pairs in  $\mathbf{T}_k^i$ , where each  $\mathbf{T}_k^i$  is obtained by appending  $P_l^i$  to each existing  $\mathbf{T}_k^{i-1}$ . If the combined constraint  $\theta_{err} < \Theta_{thresh}$ ,  $D_{err} < D_{thresh}$ , where  $\Theta_{thresh}$  and  $D_{thresh}$  are appropriate thresholds, is satisfied in addition to the  $\xi_1$  constraint for every combination of pairs in  $\mathbf{T}_k^n$ , then the path is retained, otherwise it is discarded.

*4. Total match quality.* For those paths passing the screening process above, we compute a total match error measure that forms the basis of the final recognition decision. In this work, we explore two basic formulations of this error measure.

The first error measure simply forms a weighted sum of the respective error components as follows:

$$M1_{err} = \omega_1 U_{err} + \omega_2 \theta_{err} + \omega_3 D_{err} + \omega_4 A_{map}. \quad (27)$$

$U_{err}$  is the mean of the unary consistency measure  $\xi_0$  over all pairs in  $\mathbf{T}_k^i$ , and  $A_{map}$  is the fraction of the scene area not contained in the path. This measure rewards high-quality matches explaining substantial portions of the surface.

The  $\omega_i$  are appropriate weights, summing to unity; these experiments again use uniform weighting. It might be more appropriate to adjust the weights such that the unary and binary weights are balanced (the relative orientation and pairwise distance measures received weight equal to that of all three unary attributes taken together), but recognition performance has not proven very sensitive to this.

In the second error measure, we normalize the accumulated error in the unary and binary attributes by the amount of surface explained

$$M2_{err} = \frac{\alpha_1 U_{err} + \alpha_2 \theta_{err} + \alpha_3 D_{err}}{1 - A_{map}}. \quad (28)$$

This measure differs from the first by rewarding matches that explain the matched portions of the surface well, even if relatively little of the surface is matched. As explained below, we can still impose a lower limit on the amount of surface matched to control spurious matches to small fragments. We shall also be making the decision in the context of other potential matches passing the screening process.

*5. Termination.* The search terminates if one of the following conditions occurs:

(a) The unmatched fraction of the viewed area covered by the matched pairs is sufficiently low ( $A_{map} < A_{thresh}$ ) and the strength of the best path for that depth is good enough ( $M_{err_{best}} < M_{thresh}$ ), where  $A_{thresh}$  and  $M_{thresh}$  are appropriate thresholds of match. Here,  $M_{err}$  refers to either  $M1_{err}$  or  $M2_{err}$ , whichever is in use; similarly for  $M_{thresh}$ . Once such an acceptable match has been found, the identification of the object is considered to be beyond reasonable doubt and we have a criterion for an early exit before the nodes exhaust. We have also conducted some experiments in which we require the decision to be sufficiently unambiguous; we discuss this further in the experimental results.

While these simple approaches have proven effective for this early demonstration system, an acceptance criterion couched in more rigorous decision theory would be preferred. This requires the statistical characterization of the match errors to be expected over the given modelbase for both correct and erroneous matches and the range of viewing and clutter conditions to be anticipated.

(b) No more nodes remain.

#### 5.4. Multiresolution Matching and Object Recognition

The matching algorithm described above produces the best match of the scene for a fixed set of segmentation parameters. However, for some objects, especially under occlusion, fixed parameter segmentation may yield a description inadequate for unique identification. In this case, organization at another scale may permit the object to be recognized. In our approach, the mean and Gaussian tolerances  $\tau_\mu$  and  $\tau_\gamma$  control the homogeneity constraint; a smaller tolerance window implies smaller segments and a higher level of detail in the segmentation, while a larger window leads to larger segments with less detail. A change in the tolerance window must be accompanied by a corresponding change in the minimum allowable area parameter,  $A_{\min}$ , to allow smaller segments at tighter tolerances while forcing the process to produce larger segments at wider tolerances. The modelbase of features for recognition is constructed over multiple scales and contains an overcomplete description of each object to be used for sequential matching at successively finer scales.

One might argue that since the finest scale description contains the most detail, we could perform all matching at this single scale. However, while segments at finer scales are smaller and are more likely to match something, smaller segments are less reliable descriptors of the object structure. A fine-scale search should be attempted only when a coarse scale fails to produce a suitable match. A coarse-scale match is inherently stronger because the respective matched segments represent larger portions of the object.

#### 5.5. Experimental Results in Recognition

Our modelbase consists of 10 objects, including those discussed above and others similar. While the modelbase is not large, these experiments are simply intended to illustrate the feasibility of the proposed framework for hierarchical surface description with application to free form object recognition. We recognize objects from single views. Objects in the modelbase are represented at multiple scales; the scene is then described at a sequence of scales in a coarse-to-fine strategy that proceeds only as far as necessary to achieve a suitably unambiguous match. The six scales are defined in Table 4. All parameters not specified take on the standard values indicated in Table 3.

We present detailed results for five objects (cow, dinosaurs 1 and 2, duckrattle, crocodile) from the modelbase. These were the most interesting; the others were less challenging and are less informative to study. We tabulate results for 14 experiments over these five objects and highly detailed results for two experiments showing the match errors (numerical scores) obtained *without* imposing the area  $A_{map}$  and  $M1_{err}$  (or  $M2_{err}$ ) match error thresholds. Thresholds for the average angular deviation  $\theta_{err}$ , average centroidal distance error  $D_{err}$ , and average unary match error  $U_{err}$  have all been set at 0.3, restricting the individual percentage error between the scene and the model to within 30%. We describe potential matches passing this constraint on the respective error components as *qualifiers*. To provide

**TABLE 4**  
**Table of Parameters for Varying Effective Scales**

Scale	$\tau_\mu$ (mm)	$\tau_\gamma$ (mm <sup>-2</sup> )	$A_{\min}$ (meshpts.)
0	$40 \times 10^{-3}$	$40 \times 10^{-4}$	800
1	$35 \times 10^{-3}$	$35 \times 10^{-4}$	700
2	$30 \times 10^{-3}$	$30 \times 10^{-4}$	600
3	$25 \times 10^{-3}$	$25 \times 10^{-4}$	500
4	$20 \times 10^{-3}$	$20 \times 10^{-4}$	400
5	$15 \times 10^{-3}$	$15 \times 10^{-4}$	300

a broader range of results, the tables include as qualifiers any potential matches violating the 30% error constraint in at most one of the components.

One of the more striking observations is that the unary error provides a very strong indication of match correctness; this suggests that the surface organization is stable and reliable, even for free form objects such as these. It also suggests that significant pruning of the model search space could be achieved with little computational cost by first enforcing the unary error constraint.

Tables 5 and 6 indicate the match errors obtained for two views of dinosaur 1 over a range of effective scales. The net match errors are indicated in the last two columns of each table, and errors corresponding to the correct object are indicated in bold. (We will explain the stars and daggers below.) The matches for successively decreasing scales are separated by horizontal lines.

There were 84 combinations of object–view–scale using these five objects. Of these, 61 produced at least one qualifier; 23 failed to pass two or more of the respective error thresholds on the components for any object in the modelbase. Of the 61 qualifiers, and neglecting the  $A_{map}$  threshold,  $M1_{err}$  produced its lowest value for the correct answer 55 times, as did  $M2_{err}$  (two errors in common). Thus, in most cases (more than 90%), the correct answer receives the best score among the qualifiers, regardless of scale. But, as we have already indicated, we do not simply rely on this for our recognition decision; we follow a scale-sequential strategy to achieve error-free recognition (for these tests). We will enlarge on this below.

We generally observe more qualifying matches as the scale is refined; in the case of view 3 of dinosaur 1 (and some others), no qualifier exists at the coarsest scale. Although the number of matches increases at finer scales, the match strengths generally do not. A closer look at the values of  $A_{map}$ , the fraction of the surface *not* matched, tells us that a significant fraction of the qualifying matches explain only small amounts of the surface. The fine-scale segmentations are less stable and less reliable; small changes in the surface at these scales can cause the segmentation to vary considerably, resulting in weaker matches. It is for this reason that we resort to finer scales and rely on the corresponding matches *only* for those cases that do not provide suitable matches at coarser resolutions.

We will now investigate a few approaches to constructing decision rules based on these error measures. None of these should be taken as the final word on the matter; we are simply exploring options to show that information suitable for recognition is captured in the surface organizations we compute.

**TABLE 5**  
**Recognition Results for View 2 of Object Dino1**

Scale	Match	$\theta_{err}$	$D_{err}$	$U_{err}$	$A_{map}$	$M1_{err}$	$M2_{err}$
0	cow	0.004461	0.028443	0.491685	0.901626	0.356554	1.777536
0	dino1	0.045288	0.039902	0.063246	0.901626	0.262516	0.502972
0	dino2	0.069266	0.052631	0.410916	0.901626	0.358610	1.805403
0	dkrtl	0.031306	0.268947	0.369799	0.901626	0.392919	2.270417
0	lamb	0.063849	0.131126	0.549890	0.901626	0.411623	2.523926
0	croc	0.082807	0.047234	0.547391	0.901626	0.394765	2.295437
1	cow	0.013159	0.010975	0.463572	0.912922	0.350157	1.866932
1	dino1	0.053760	0.068520	0.061469	0.912922	0.274168	0.703392
1	dino2	0.151887	0.247920	0.356081	0.930826	0.421678	3.642438
1	lamb	0.097099	0.095761	0.442575	0.912922	0.387089	2.432432
1	croc	0.060859	0.034392	0.306068	0.912922	0.328560	1.536240
2	cow	0.047875	0.153626	0.234605	0.936033	0.343035	2.272562
2	dino1	0.103230	0.071278	0.192548	0.220580	0.146909*	0.156978†
2	dino2	0.227900	0.007757	0.269593	0.925939	0.357797	2.274022
2	dumbbell	0.228854	0.015511	0.584513	0.925939	0.438704	3.730605
2	lamb	0.001157	0.016480	0.293779	0.936033	0.311862	1.622790
2	duck	0.000917	0.017962	0.415682	0.936033	0.342648	2.264496
2	croc	0.010033	0.018374	0.364406	0.936033	0.332211	2.046946
3	cow	0.048420	0.025243	0.279861	0.878715	0.308060	0.971610
3	dino1	0.062355	0.130955	0.189321	0.226508	0.152285	0.164894
3	dino2	0.059621	0.000129	0.296468	0.963190	0.329852	3.225736
3	dumbbell	0.147668	0.054661	0.483269	0.957903	0.410875	5.428708
3	dkrtl	0.012601	0.196111	0.513679	0.926780	0.412293	3.288683
3	lamb	0.010924	0.099585	0.479057	0.947288	0.384214	3.728234
3	duck	0.069143	0.003989	0.185942	0.947569	0.301660	1.647060
3	whale	0.086560	0.245029	0.432363	0.947288	0.427810	4.830981
3	croc	0.062516	0.072734	0.259697	0.942001	0.334237	2.269849

### 5.5.1. Decisions Based on $M1_{err}$

If we were to set a threshold of say,  $A_{map} < 0.6$ , requiring that we match 40% of the object in the scene, many of the qualifying, but incorrect, matches would be eliminated. Using a simple decision rule of  $A_{map} < 0.6$  and  $M1_{err} < 0.3$  results in the recognition decisions indicated by the  $\star$  in the penultimate column of Tables 5 and 6. Notice that the recognition attempts required a range of resolutions for success. *Over our range of experiments, this simple decision rule produced no errors.* Although we are hardly naive enough to tout this approach as error-free, given the simplicity of the experiment, it is clear that the surface organization and description formalism we have outlined shows considerable promise in the difficult domain of free form object recognition.

An examination of the  $A_{map}$  and  $M1_{err}$  values reveals that the multiresolution approach identifies a “natural” scale at which the object’s match error drops and a substantial fraction of its visible surface is explained. There may be more than one such scale for some objects, and they may be view-dependent. We can exploit this observation to design an alternative decision rule based on the (lack of) ambiguity in the decision. To this end, we also consider a revised decision rule using the *relative* strengths of each of the matches at varying degrees of resolution, with respect to the next best match in its class.

**TABLE 6**  
**Recognition Results for View 3 of Object Dino1**

Scale	Match	$\theta_{err}$	$D_{err}$	$U_{err}$	$A_{map}$	$M1_{err}$	$M2_{err}$
0	—	—	—	—	—	—	—
1	dino1	0.037559	0.015024	0.107350	0.015265	0.043799*	0.0541367 <sup>†</sup>
1	dumbbell	0.285602	0.102935	0.461910	0.937897	0.447086	4.564712
1	lamb	0.248014	0.114480	0.322501	0.937897	0.405723	3.676661
1	croc	0.085438	0.098340	0.439602	0.941879	0.391315	3.575191
2	cow	0.075700	0.018257	0.373191	0.926804	0.348488	2.127384
2	dino1	0.037043	0.015838	0.118490	0.046838	0.054552	0.059930
2	dino2	0.031518	0.069419	0.504877	0.934105	0.384980	3.064547
2	lamb	0.089678	0.181398	0.496194	0.926804	0.423518	3.494125
2	croc	0.245985	0.140699	0.465289	0.926804	0.444694	3.879866
3	cow	0.026720	0.068319	0.383664	0.943444	0.355537	2.821416
3	dino1	0.121143	0.044266	0.195583	0.110126	0.117779	0.135221
3	dino2	0.073159	0.053258	0.349848	0.941500	0.354441	2.713755
3	lamb	0.285892	0.067405	0.283789	0.914383	0.387867	2.480368
3	duck	0.026628	0.112533	0.320655	0.937470	0.349321	2.451165
3	croc	0.033069	0.000443	0.471734	0.960226	0.366368	4.234307
4	cow	0.032239	0.230450	0.292255	0.972030	0.381744	6.613586
4	dino1	0.025122	0.078170	0.138592	0.246042	0.121981	0.106939
4	dino2	0.178310	0.071172	0.297019	0.856974	0.350869	1.273666
4	lamb	0.296976	0.167419	0.466206	0.898692	0.457323	3.061950
4	duck	0.159775	0.220843	0.313207	0.847587	0.385353	1.517423
4	croc	0.091191	0.121522	0.248851	0.889163	0.337682	1.388119

Table 7 provides a summary of the relative strengths of the matches at each resolution. The relative match strength is defined as the ratio of the second-best match error to the best. Retaining our  $A_{map}$  threshold of 0.6, we have a simple hypothesis testing problem with the “likelihood ratios” in Table 7; we set the threshold for recognizing an object at

**TABLE 7**  
**Variation of Relative  $M1_{err}$  Match Strength with Scale**

View	Scale 0	Scale 1	Scale 2	Scale 3	Scale 4	Scale 5
cow-1	2.345645	1.631319	1.498845	1.092815	1.029699	1.254331
cow-2	8.755001	3.528829	1.168854	3.841046	n/a	1.177489
cow-3	1.582068	1.473070	1.794896	1.193082	n/a	1.294097
cow-4	4.697705	2.813201	1.267138	1.202800	n/a	1.388854
dino1-1	2.226016	1.265129	1.314889	1.158758	n/a	n/a
dino1-2	1.358218	1.198389	2.122824	1.980891	n/a	n/a
dino1-3	n/a	8.934336	6.388180	2.965902	2.768316	n/a
dino2-1	1.117691	1.989754	1.031256	1.096964	n/a	n/a
dino2-2	1.464530	3.885735	1.769712	1.062514	n/a	n/a
dino2-3	n/a	2.522330	1.023218	1.430262	n/a	n/a
dkrtl	1.317156	1.655223	1.420961	1.972873	n/a	n/a
croc-1	1.095976	1.083372	1.181369	1.227600	1.128735	n/a
croc-2	1.558302	1.349457	1.642006	1.434179	n/a	n/a
croc-3	1.030076	1.937453	1.892136	2.592301	n/a	n/a

*Note.* These numbers do not reflect the  $A_{map} < 0.6$  constraint. Where “n/a” appears, no qualifiers exist.

1.6. The lowest resolution at which  $A_{map} < 0.6$  and  $M1_{rel} > 1.6$  occurs is accepted as an appropriately strong match. Locations at which such decisions are taken appear in bold in Table 7. It can be seen that with this decision rule, all scenes except view 1 of crocodile find an appropriate match. Note that, in this test, we only required that the *best* match meet the  $A_{map} < 0.6$  constraint. The second-best match used in the relative match strength calculation could have any value for  $A_{map}$ . While the wisdom of this approach is certainly debatable, it only came into play for view 1 of the crocodile. In that case, the only model to meet the  $A_{map}$  constraint was the crocodile at scale 2. If we were to impose the  $A_{map}$  constraint prior to computing relative match strength, then the crocodile would be correctly recognized at scale 2, just as it was with the first decision rule.

Another indication of the confidence in a recognition decision is its *entropy*. The entropy measures the amount of probabilistic uncertainty remaining in the experiment after making the observation(s) and measurements and can be thought of as the average amount of information remaining to be learned upon being told the correct answer. We first convert the match strengths into probability estimates by normalizing their reciprocals to sum to unity. Then, we compute the entropy as

$$H = - \sum_j \hat{p}_j \log \hat{p}_j, \quad (29)$$

where  $\hat{p}_j$  represents the estimate of the probability of the observation being object  $j$  based on the matching scores. The units of the entropy depend on the base of the logarithm. In these calculations, we used the base-2 logarithm and the corresponding unit of information is the *bit*. The lower the entropy value, the more confidence we have in the decision; less uncertainty remains. Table 8 summarizes the entropy values for the object-view-scale combinations tested.

While the relative match strength compares the top two candidates for a given test, the entropy measures the overall uncertainty in the decision. In some sense, then, the relative

**TABLE 8**  
**Variation of  $M1_{err}$  Entropy (in Bits) with Scale**

View	Scale 0	Scale 1	Scale 2	Scale 3	Scale 4	Scale 5
cow-1	2.16	2.14	2.78	3.13	3.16	3.15
cow-2	1.46	1.98	2.52	2.47	n/a	3.29
cow-3	2.27	2.29	2.68	2.77	n/a	3.30
cow-4	2.11	2.32	2.89	2.93	n/a	3.30
dino1-1	2.46	2.78	2.78	3.15	n/a	n/a
dino1-2	2.57	2.31	2.71	3.09	n/a	n/a
dino1-3	n/a	1.18	1.64	2.40	2.40	n/a
dino2-1	2.80	2.50	2.52	3.15	n/a	n/a
dino2-2	1.56	1.66	2.52	2.97	n/a	n/a
dino2-3	n/a	1.82	2.01	2.78	n/a	n/a
dkrtl	2.31	2.51	2.78	3.10	n/a	n/a
croc-1	2.78	2.78	2.99	2.99	3.16	n/a
croc-2	2.54	2.78	2.76	3.13	n/a	n/a
croc-3	2.78	2.74	2.74	2.65	n/a	n/a

*Note.* These numbers do not reflect the  $A_{map} < 0.6$  constraint. Where “n/a” appears, no qualifiers exist.

match strength is a more targeted assessment of the (lack of) ambiguity in the decision. Although we would expect the two quantities to be somewhat (negatively) correlated, they do measure slightly different things.

In the current testing scenario (and using  $M1_{err}$ ), we present the system with a single object at a time, so we would opt for the relative match strength as the better indicator. However, in a multiobject scene, a measure based on entropy would provide some insight into how well the initial organization into plausible objects (e.g., as provided by Fisher’s surface clusters [62]) has been accomplished. Moreover, the different models receiving qualifying match results are generally going to explain *different* regions of the observed surface(s). The range of potential matches, as quantified in part by the entropy, can provide insight into the collection of objects in view, perhaps even without surface clustering. This promising idea merits further study as part of a sequential identification–extraction approach in which those surface segments explained by the current-strongest match are identified and removed from further consideration until as much of the observation as possible is recognized.

### 5.5.2. Decisions Based on $M2_{err}$

Recall that this measure introduces the mapped surface area in the denominator of a ratio that, essentially, computes the error on a “per-matched-area” basis. This has the effect of supporting matches that, while not necessarily explaining a great deal of the model, do so very well. In some objects there exist small, yet highly descriptive features. Recognizing features such as these can point decisively to the correct object identification. The plates along the spine of Dinosaur 2 are a good example. Of course, in this sense, saliency can be a matter of modelbase context; if many objects were to have such plates, then they would become far less useful for identification.

If we apply a decision rule similar to the first one described for  $M1_{err}$  above, namely a threshold on the matching score and on the amount of surface mapped, we can again achieve error-free matching over our set of tests, as denoted by the daggers (†) in the last column of

**TABLE 9**  
**Variation of Relative  $M2_{err}$  Match Strength with Scale**

View	Scale 0	Scale 1	Scale 2	Scale 3	Scale 4	Scale 5
cow-1	5.392846	1.904813	3.702750	2.640425	2.787917	6.716165
cow-2	30.985020	11.441009	1.261774	20.415650	n/a	1.809979
cow-3	17.176075	6.634198	1.975630	10.412856	n/a	2.054655
cow-4	19.975684	3.969480	1.301541	1.169311	n/a	1.843591
dino1-1	8.161105	1.919571	20.848565	1.512015	n/a	n/a
dino1-2	3.534068	2.184044	10.337672	5.892337	n/a	n/a
dino1-3	n/a	66.040021	35.497606	18.127049	11.910236	n/a
dino2-1	1.927831	4.257715	1.021740	1.183371	n/a	n/a
dino2-2	4.224263	7.630564	9.559729	7624.71	n/a	n/a
dino2-3	n/a	7.413723	1.071202	564996.1	n/a	n/a
dkrtl	2.147838	2.619997	2.792530	4.993006	n/a	n/a
croc-1	1.842175	1.424806	1.745791	2.931066	1.074804	n/a
croc-2	4.292633	2.855044	2.528574	2.223830	n/a	n/a
croc-3	1.369101	7.417557	6.168213	9.660936	n/a	n/a

*Note.* These numbers do not reflect the  $A_{map} < 0.6$  constraint. Where “n/a” appears, no qualifiers exist.

**TABLE 10**  
**Variation of  $M2_{err}$  Entropy (in Bits) with Scale**

View	Scale 0	Scale 1	Scale 2	Scale 3	Scale 4	Scale 5
cow-1	1.53	1.60	2.28	2.12	2.95	1.63
cow-2	0.56	0.98	1.89	0.97	n/a	2.54
cow-3	0.81	1.56	2.02	1.07	n/a	2.83
cow-4	0.88	1.54	2.32	2.16	n/a	2.83
dino1-1	1.40	2.29	1.08	2.52	n/a	n/a
dino1-2	2.27	2.08	1.63	1.92	n/a	n/a
dino1-3	n/a	0.31	0.53	1.15	1.23	n/a
dino2-1	2.70	2.03	2.06	2.54	n/a	n/a
dino2-2	1.18	1.01	1.37	0.004	n/a	n/a
dino2-3	n/a	1.05	1.83	0.0001	n/a	n/a
dkrtl	2.08	2.25	2.47	2.40	n/a	n/a
croc-1	2.57	2.57	2.80	2.70	2.83	n/a
croc-2	1.93	2.40	2.52	2.70	n/a	n/a
croc-3	2.48	1.92	2.05	1.54	n/a	n/a

*Note.* These numbers do not reflect the  $A_{map} < 0.6$  constraint. Where “n/a” appears, no qualifiers exist.

Tables 5 and 6. In this case, we require  $M2_{err} < 0.35$  and, again,  $A_{map} < 0.6$ , and we make identical decisions to those reached in the  $M1_{err}$  case. Of course, these thresholds are (still) just empirical; more careful study would be needed to set proper decision points for a given application.

Even though the decision rules based on match scores and  $A_{map}$  achieved identical results in these tests, there is nonetheless a difference in the two matching scores. The decisions reached using  $M2_{err}$  are generally far less ambiguous than those using  $M1_{err}$ . The ratio between the best and second best scores is always greater, by a factor of more than three on average using  $M2_{err}$ . Also, the entropy in the decision using  $M2_{err}$  is lower than that using  $M1_{err}$ , by a factor of more than 1.6 on average. This can be seen in Tables 9 and 10 as compared to Tables 7 and 8, respectively.

A suitable decision rule based on the relative  $M2_{err}$  matching scores from the two best candidates would be easy to construct, indeed. The numbers in bold in Table 9 are based on a 2 : 1 ratio to accept the decision. This is a stronger requirement than the 1.6 : 1 used with  $M1_{err}$ , but is easily met, usually at the first scale.

## 6. FINAL COMMENTS

This paper has introduced a novel, stable means of obtaining reliable surface descriptions at multiple (effective) scales from range data. We demonstrated the use of this surface organization scheme in the modeling and recognition of free form objects; it is useful for simpler things such as polyhedra, as well.

The surface organization scheme is robust to parameter changes, yet tunable for multiscale use. One of the most attractive features of the approach is that it extracts surface segments sequentially in the order of their saliency by finding the largest segments first in a given consistency class, and by working through the consistency categories in the order of their inherent saliency. This suggests the possibility of embedding surface description (segmentation and attribution) within the object recognition loop, allowing the

process to terminate once a suitably reliable match has been found, perhaps by entropy analysis.

With respect to timing, on a Pentium II, 450 MHz, 512 MB RAM, the system takes about 60 s to segment a fairly complex mesh (ours contain as many as 10–20 K vertices) using the standard parameters. This is for the complete object and covers everything. Matching requires about 11 s, over multiple scales, for our (small) modelbase on the same machine. More advanced machines offer the obvious improvements.

Because the unary match error is a very strong indicator of the correct object's identity (at least for our tests), it is likely that only the unary error calculation could be executed in a pre-screening step that would avoid doing a true graph match in many cases. One hopes that the graph matching would only be necessary to resolve ambiguities among close decisions based on the unary error; we could certainly avoid the procedure for those objects whose unary errors are large. We have not pursued this notion in detail here because the performance of such an approach will depend on the content of the modelbase. Although we believe it to be true more often than not, the fact that unary errors alone are a powerful cue for our small set of objects does not (necessarily) guarantee similar behavior for another object set. In particular, objects with similar features arranged in different spatial configurations would require relational information—and therefore a graph matching operation—to distinguish among them.

Besides object recognition, we envision applications of this surface organization scheme in three-dimensional perceptual organization and other intermediate level vision tasks.

There is certainly room for improvement. Our curvature estimation method is more computationally intensive than we would like and could be subject to significant error in the presence of statistical outliers. A more (statistically) robust method, and one which is quicker (if we can achieve both goals simultaneously) would be very desirable. Statistical robustness might be achieved via a weighted analysis in which points lying farther from the point of interest are derated in their contribution to the principal components analysis and/or the polynomial fit.

Another area in need of attention is the decision rule for recognition. For demonstration purposes, we presented a range of options that are, admittedly, rather *ad hoc*. Formal statistical analysis, including formal models of scene clutter, and extensive experimentation are needed to complete the development. Moreover, in different application contexts, one may wish to consider maximum *a posteriori*, maximum likelihood, Neyman–Pearson, or constant false alarm rate approaches to decision rule design; we have only explored an approximation to maximum likelihood.

## REFERENCES

1. F. Mokhtarian, N. Khalili, and P. Yuen, Multi-scale free-form surface description, in *Proc. Indian Conference on Computer Vision, Graphics, and Image Processing, 1998*, pp. 70–75.
2. F. Mokhtarian, N. Khalili, and P. Yuen, Multi-scale 3-d free-form surface smoothing, in *Proc. British Machine Vision Conference, 1998*, pp. 730–739.
3. F. Mokhtarian, N. Khalili, and P. Yuen, Curvature computation on free-form 3-d meshes at multiple scales, *Comput. Vision Image Understanding* **83**, 2001, 118–139.
4. D. Zhang and M. Hebert, Multi-scale classification of 3d objects, in *IEEE Conf. Computer Vision and Pattern Recog., 1997*, pp. 864–869.
5. P. Sander and S. Zucker, Inferring differential structure from 3-D images: Smooth cross sections of fiber bundles, *IEEE Trans. Pattern Anal. Mech. Intell.* **12**, 1990, 833–854.

6. F. P. Ferrie, S. Mathur, and G. Soucy, Feature extraction for 3-D model building and object recognition, in *Three-Dimensional Object Recognition Systems* (A. K. Jain and P. J. Flynn, eds.), pp. 57–88, Elsevier, Amsterdam, 1993.
7. P. J. Flynn and A. K. Jain, On reliable curvature estimation, in *Proc. IEEE Conf. Comput. Vision and Patt. Recog.*, June 1989, pp. 110–116.
8. P. J. Besl and R. C. Jain, Invariant surface characteristics for 3d object recognition in range images, *Comput. Vision Graphics Image Process.* **33**, 1986, 33–80.
9. P. J. Flynn and A. K. Jain, Surface classification: Hypothesis testing and parameter estimation, in *Proc. IEEE Conf. Comput. Vision and Patt. Recog.*, June 1988, pp. 261–267.
10. T. Fan, G. Medioni, and R. Nevatia, Surface segmentation and description from curvature features, in *Proc. 1987 DARPA Image Understanding Workshop, 1987*, pp. 351–359.
11. D. J. Ittner and A. K. Jain, 3-d surface discrimination from local curvature measures, in *Proc. IEE 1985 Conf. Computer Vision and Pattern Recognition, May 1985*, pp. 119–123.
12. R. L. Hoffman and A. K. Jain, Segmentation and classification of range images, *IEEE Trans. Pattern Anal. Mach. Intell.* **9**, 1987, 608–620.
13. A. D. Smith, The folding of the human brain: From shape to function, in Ph.D. Thesis, Kings College, London, 1999.
14. A. Mitiche and J. K. Aggarwal, “Detection of edges using range information,” *IEEE Trans. Pattern Anal. Mach. Intell.* **5**, 1983, 174–178.
15. T. J. Fan, G. Medioni, and R. Nevatia, Segmented descriptions of 3d surfaces, *IEEE J. Robot. Automat.* **3**(6), 1987, 527–538.
16. P. Besl and R. Jain, Segmentation through variable order surface fitting, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 1988, 167–192.
17. B. C. Vemuri and J. K. Aggarwal, Curvature-based representation of objects from range data, *Image Vision Comput.* **4**(2), 1986, 107–114.
18. R. M. Haralick, L. T. Watson, and T. J. Laffey, The topographic primal sketch, *Int. J. Robot. Res.* **2**(1), 1983, 50–72.
19. F. Tomita and T. Kanade, A 3d vision system generating and matching shape descriptions in range images, in *Proc. IEEE 1st CAIA, 1984*, pp. 186–191.
20. N. Yokoya and M. D. Levine, Range image segmentation based on differential geometry—A hybrid approach, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1989, 643–649.
21. A. K. Jain and S. G. Nadabar, Mrf model-based segmentation of range images, in *Proc. Int. Conf. Computer Vision, December 1990*, pp. 667–671.
22. S. M. Bhandarkar and A. Siebert, Integrating edge and surface information for range image segmentation, *Pattern Recognition* **25**, 1992, 947–962.
23. M. A. Wani and B. B. Batchelor, Edge region-based segmentation of range images, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 1994, 287–303.
24. S. Inokuchi, T. Nita, F. Matsuda, and Y. Sakurai, A three-dimensional edge-region operator for range images, in *Proc. Sixth Int. Conf. Pattern Recognition, 1982*, pp. 918–920.
25. D. L. Milgrim and C. M. Bjorklund, Range image processing planar surface extraction, in *Proc. Fifth Int. Conf. Pattern Recognition, December 1980*, pp. 912–919.
26. T. C. Henderson, Efficient 3-d object representation for industrial vision systems, *IEEE Trans. Pattern Anal. Mach. Intell.* 1983, 609–617.
27. M. Hebert and J. Ponce, A new method for segmenting 3-d scenes into primitives, in *Proc. Int. Conf. Pattern Recognition, October 1982*, pp. 836–838.
28. J. H. Han, R. A. Volz, and T. M. Mudge, Range image segmentation and surface parameter extraction for 3-d object recognition of industrial parts, in *IEEE Int. Conf. Robotics Automation, 1987*, Vol. 1, pp. 380–386.
29. I. K. Sethi and S. N. Jayamurthy, Surface classification using characteristic contours, in *Proc. Seventh Int. Conf. Pattern Recognition, July-August 1984*, pp. 438–440.
30. M. Oshima and Y. Shirai, Object recognition using three-dimensional information, *IEEE Trans. Pattern Anal. Mach. Intell.* **5**, 1983, 353–361.

31. N. N. Abdelmalek, Algebraic error analysis for surface curvature and segmentation of 3-d range images, *Pattern Recognition* **23**(8), 1990, 807–817.
32. R. W. Taylor, M. Savini, and A. P. Reeves, Fast segmentation of range imagery into planar regions, *Comput. Vision Graphics Image Process.* **45**, 1989, 42–60.
33. M. T. Suzuki, T. Kato, and N. Otsu, A similarity retrieval of 3d polygonal models using rotation invariant shape descriptors, in *Proceedings IEEE Int'l. Conf. Systems, Man, and Cybernetics, October 2000*, pp. 2946–2952.
34. M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl, 3d shape histograms for similarity search and classification in spatial databases, in *Proceedings 6th Int'l. Symposium on Spatial Databases, July 1999*, pp. 207–226.
35. S.-P. Liou, A. H. Chiu, and R. C. Jain, A parallel technique for signal-level perceptual organization, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 317–325.
36. X. Jiang, An adaptive contour closure algorithm and its experimental evaluation, *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 2000, 1252–1265.
37. K. L. Boyer, M. J. Mirza, and G. Ganguly, The robust sequential estimator: A general approach and its application to surface organization in range data, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 1994, 987–1001.
38. D. M. Wuescher and K. L. Boyer, Robust contour decomposition using a constant curvature criterion, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 41–51.
39. R. C. Bolles and P. Horaud, 3DPO: A three-dimensional part orientation system, *Int. J. Robot. Res.* **5**, 1986, 3–26.
40. W. E. L. Grimson and T. Lozano-Perez, Model-based recognition and localization from sparse range or tactile data, *Int. J. Robot. Res.* **3**, 1984, 3–35.
41. T. J. Fan, G. Medioni, and R. Nevatia, Recognizing 3-D objects using surface descriptions, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1989, 1140–1157.
42. R. T. Chin and C. R. Dyer, Model-based recognition in robot vision, *ACM Comput. Surv.* **18**, 1986, 67–108.
43. A. Wong, S. Lu, and M. Rioux, Recognition and shape synthesis of 3d objects based on attributed hypergraphs, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 1989, 279–290.
44. M. Oshima and Y. Shirai, Object recognition using 3d information, *IEEE Trans. Pattern Anal. Mach. Intell.* **5**, 1983, 353–361.
45. P. J. Flynn and A. K. Jain, CAD-based computer vision: From CAD models to relational graphs, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 114–132.
46. C. H. Chen and A. C. Kak, A robot vision system for recognizing 3d objects in low-order polynomial time, *IEEE Trans. Systems Man Cybernet.* **19**, 1989, 1535–1563.
47. W. Kim and A. Kak, 3d object recognition using bipartite matching embedded in discrete relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 224–251.
48. Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, Affine invariant model based object recognition, *IEEE Trans. Robot. Automat.* **6**, 1990, 578–589.
49. D. Forsyth, J. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, Invariant descriptors for 3d object recognition and pose, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 971–991.
50. K. Sengupta and K. L. Boyer, Organizing large structural modelbases, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 1995, 321–332.
51. P. J. Flynn and A. K. Jain, Bonsai: 3-d object recognition using constrained search, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 1991, 1066–1074.
52. P. J. Flynn and A. K. Jain, 3d object recognition using invariant feature indexing of interpretation tables, *Comput. Vision Graphics Image Process.* **55**, 1992, 119–129.
53. J. E. Hopcroft and R. M. Karp, A  $n^{5/2}$  algorithm for maximum matching in bipartite graphs, *J. SIAM Comput.* **2**, 1973, 225–231.
54. K. Sengupta and K. L. Boyer, Modelbase partitioning using property matrix spectra, *Comput. Vision Image Understanding* **70**, 1998, 177–196.
55. R. Campbell and P. Flynn, A survey of free-form object representation and recognition techniques, *Comput. Vision Image Understanding* **81**, 2001, 166–210.

56. P. J. Besl, The free-form surface matching problem, in *Machine Vision for Three-Dimensional Scenes* (H. Freeman, Ed.), pp. 25–71, Academic Press, San Diego, 1990.
57. P. J. Besl, Triangles as a primary representation, in *Object Representation for Computer Vision* (M. Hebert, J. Ponce, T. Boult, and A. Zisserman, Eds.), pp. 191–206, Springer-Verlag, Berlin/New York, 1996.
58. J. P. Thirion, The extremal mesh and the understanding of 3D surfaces, *Int. J. Comput. Vision* **19**, 1996, 115–128.
59. R. Campbell, *Recognition of Free-Form 3D Objects in Range Data Using Global and Local Features*, Ph.D. thesis, The Ohio State University, 2001.
60. G. Taubin, F. Cukierman, J. Ponce, and D. Kriegman, Parametrized families of polynomials for bounded algebraic curve and surface fitting, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 1994, 287–303.
61. B. Parvin and G. Medioni, B-rep object description from multiple range views, *Int. J. Comput. Vision* **20**, 1/2, 1996, 81–112.
62. R. B. Fisher, *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis*, Wiley, New York, 1989.