Segmentation Through Variable-Order Surface Fitting

PAUL J. BESL, MEMBER, IEEE, AND RAMESH C. JAIN, SENIOR MEMBER, IEEE

Abstract—Computer vision systems attempt to recover useful information about the three-dimensional world from huge image arrays of sensed values. Since direct interpretation of large amounts of raw data by computer is difficult, it is often convenient to partition (segment) image arrays into low-level entities (groups of pixels with similar properties) that can be compared to higher-level entities derived from representations of world knowledge. Solving the segmentation problem requires a mechanism for partitioning the image array into low-level entities based on a model of the underlying image structure. Using a piecewise-smooth surface model for image data that possesses surface coherence properties, we have developed an algorithm that simultaneously segments a large class of images into regions of arbitrary shape and approximates image data with bivariate functions so that it is possible to compute a complete, noiseless image reconstruction based on the extracted functions and regions. Surface curvature sign labeling provides an initial coarse image segmentation, which is refined by an iterative region growing method based on variable-order surface fitting. Experimental results show the algorithm's performance on six range images and three intensity images.

Index Terms—Image segmentation, range images, surface fitting.

I. INTRODUCTION

COMPUTER vision systems attempt to recover useful information about the three-dimensional (3-D) world from huge image arrays of sensed values. Since direct interpretation of large amounts of raw data by computer is difficult, it is often convenient to partition (segment) image arrays into low-level entities (groups of pixels with particular properties) that can be compared to higher-level entities derived from representations of world knowledge. Solving the segmentation problem requires a mechanism for partitioning the image array into useful entities based on a model of the underlying image structure.

In most easily interpretable images, almost all pixel values are statistically and geometrically correlated with neighboring pixel values. This pixel-to-pixel correlation, or *spatial coherence*, in images arises from the spatial coherence of the physical surfaces being imaged. In range images, where each sensed value measures the distance to physical surfaces from a known reference surface, the pixel values collectively exhibit the same spatial coher-ence properties as the actual physical surfaces they represent. This has motivated us to explore the possibilities of a surface-based image segmentation algorithm that uses the spatial coherence (*surface coherence*) of the data to organize pixels into meaningful groups for later visual processes.

Many computer vision algorithms are based on inflexible, unnecessarily restricting assumptions about the world and the underlying structure of the sensed image data. The following assumptions are common: 1) all physical objects of interest are polyhedral, quadric, swept (as in generalized cylinders), convex, or combinations thereof; 2) all physical surfaces are planar, quadric, swept, or convex; 3) all image regions are rectangular or regularly shaped and are approximately constant in brightness; and 4) all image edges are linear or circular. The extensive research based on these assumptions solves many important application problems, but these assumptions are very limiting when analyzing scenes containing real-world objects with free-form, sculptured surfaces. Therefore, we have developed an image segmentation algorithm based only on the assumption that the image data exhibits *surface coherence* in the sense that the image data may be interpreted as noisy samples of a piecewise-smooth surface function. A preliminary grouping of pixels is based on the sign of mean and Gaussian surface curvature. This initial, coarse segmentation is refined by an iterative region growing procedure based on variable-order bivariate surface fitting. The order of the surface shape hypotheses is automatically controlled by *fitting surfaces* to the image data and *testing the surface fits* by 1) checking the spatial distribution of the signs of residual fitting errors (the regions test) and 2) comparing the mean square residual error of the fit to a threshold proportional to an estimate of the image noise variance. In this iterative process, images are not only segmented into regions of arbitrary shape, but the image data in those regions is also approximated with flexible bivariate functions such that it is possible to compute a complete, noiseless image reconstruction based on the extracted functions and regions. We believe that an explicit image description based on flexibly shaped approximating functions defined over arbitrary connected image regions can be useful in many computer vision applications, but will be critically important to object reconstruction and object recognition algorithms based on range imaging sensors when object volumes are bounded by free-form, smooth surfaces.

Although this segmentation algorithm may be most useful for range images, it is capable of segmenting any type

of image that can be adequately represented as a noisy, sampled version of a piecewise-smooth graph surface. Therefore, we first include a brief discussion of the relationship between general computer vision and range image understanding. Since other segmentation methods in the literature involve several ideas closely related to those presented here, we also include a brief discussion of previous work in intensity image segmentation, followed by a survey of more recent work in range image analysis, in order to clarify the differences of the surface-based segmentation algorithm. Mathematical preliminaries are then presented to precisely define the problem we are attempting to solve, followed by a qualitative description of a method for general smooth surface decomposition. Several key ideas behind the algorithm philosophy are described next. Then the entire algorithm is outlined to introduce the role of the individual algorithm elements, followed by a detailed explanation of each element. Experimental results show the algorithm's excellent performance on a variety of six range and three intensity images from a database of successful test results on over forty images. We conclude with comments on future improvements and applications to other types of multidimensional image data.

A. Vision and Range

Most past computer vision research has been concerned with extracting useful information from one or more intensity images of a scene. The desired "useful" information has often been depth or range information. Indeed, the dominant images-to-surfaces vision paradigm [73], [44] dictates that various visual cues can be used to infer the distance of many scene points from sensed light intensity values as the human visual system does. Several methods for obtaining range (shape) from intensity images based on various visual cues are summarized below.

When the sampled values in an image array represent light intensity at each point, knowledge of the intensity image formation process and an appropriate set of constraints can be used to recover the shape of the physical 3-D surfaces represented by the data. Vision researchers have developed many techniques (see survey [18]) for obtaining 2.5-D descriptions (registered range images) of intensity images that indicate the sensor-to-physical-surface distance at many points in a scene: shape from shading [58], shape from texture [103], shape from contour [66], shape from binocular stereo [43], shape from photometric stereo [105], [25], shape from motion [101], [63], shape from shadows [67].

The above are predominantly *passive* approaches for obtaining range information in the sense that energy is not projected into the environment. Many *active* approaches for obtaining range images have also been developed [64] including amplitude-modulated laser radar [108]; frequency-modulated laser radar [7]; time-of-flight laser radar [71]; structured light with lines [93], grids [46], and coded binary patterns [61]; intensity ratio [21]; moire interferometry [86]; and focussing methods [68]. But once

a range image has been acquired for a given scene by any of the above methods, the extraction of useful information still requires processing a huge array of values where each value represents the distance to a physical surface from a known reference surface. Hence, the ability to obtain range at each pixel in an image does not in itself solve computer vision problems. Range images provide sampled geometric information in an *explicit form* rather than in an *implicit form* dependent on surface reflectance and illumination. The data must still be organized into a more structured form for interpretation purposes.

Witkin and Tenenbaum [104] have argued that perceptual organization mechanisms exist in the early stages of human visual processing that are independent of the high-level knowledge necessary for correct image interpretation and are independent of the image formation process. That is, people can visually segment image regions into meaningful entities even when they know nothing about the entities or the image formation process. Consider the fact that people with no knowledge of or experience with the formation of images from electron microscopes, X-ray imagers, ultrasonic sensors, and imaging radars can often partition images into important regions that are meaningful to experts in the respective fields. Therefore, it should be possible to group pixels in many types of images using only relatively low-level information. However, it is not at all clear how these general-purpose low-level grouping mechanisms operate.

We believe that perception of surfaces is a low-level grouping operation that plays a fundamental role in many image understanding tasks. Therefore, a segmentation algorithm that groups pixels based on a surface interpretation should be valuable to many applications. For example, explicit surface approximations over range image regions is directly useful for surface inspection, assembly verification, automatic shape acquisition, and autonomous navigation. If early vision processes focus on segmenting range images (however they are acquired) into surfaces defined over image regions, we believe that it will eventually be possible to achieve robust recognition of arbitrary 3-D objects by matching *perceived surface descriptions* with known object models. Although many matching approaches are based on lower dimensional features, such as points (i.e., object vertices) and edges (i.e., occluding edges and separating boundaries between surfaces), we believe that matching based on surface shape holds the most promise for general-purpose vision because surface matching would not be hindered by occlusion of individual point or edge features. Moreover, our experimental results show that surface-based segmentation is also promising for other types of images, such as intensity images, whenever the image data exhibits surface coherence properties.

B. Intensity Image Segmentation

A problem with many computer vision techniques is the assumption that there is only one physical surface or object represented in an image. When many surfaces of many

objects are present, it is often necessary to organize pixels into connected groups or image regions that correspond to individual objects or surfaces, and then apply higher-level algorithms to the isolated image regions. The fundamental, complementary issues in organizing image pixels into regions are similarity (uniformity) and difference (contrast). Given the sensed values at two image pixels and their neighbors, the computer must answer the question: "does this pixel possess enough of the same properties as that pixel to say that these two pixels are similar?"

Segmentation of digital images has been an active area of research for many years (see surveys [47], [31], [37], [65], [90], [88]). Many popular segmentation techniques use histogram-based thresholding or template matching, but these methods provide little information when the image data does not conform to the restrictive image model assumptions. Edge detection techniques (see survey [29]) attempt to define regions by locating pixels that lie on the boundaries between regions using difference measures on neighboring pixels (e.g., image gradient magnitude). Region growing techniques (see survey [107]) attempt to group pixels into connected regions based on similarity measures, such as approximate equality (e.g., [20]). Edge-detection and region-growing can be data-driven operations based on generic notions of difference and similarity that make no commitment to the set of possible image interpretations, or they can be model-driven operations based on application-specific object models and domain knowledge. Model-driven techniques can reduce computational requirements by incorporating high-level knowledge about the scenes represented in images to restrict the search for valid interpretations [41].

A commonly used definition of image segmentation [57] states that if I is the set of all image pixels and $P(\ )$ is a *uniformity precidate* defined on groups of connected pixels, a segmentation of I is a partitioning set of connected subsets or image regions $\{R_1, \cdots, R_N\}$ such that

$$\bigcup_{l=1}^{N} R_l = I \quad \text{where} \quad R_l \cap R_m = \phi \quad \forall l \neq m, \quad (1)$$

the uniformity predicate $P(R_l)$ = True for all regions, and

$$P(R_l \cup R_m) = \text{False} \quad (2)$$

whenever R_l is adjacent to R_m. Different segmentation algorithms may be viewed as implementations of different uniformity predicates. Uniformity predicates may be classified according to knowledge requirements [65]: signal-level methods are based purely on the numbers in a digital image, physical-level methods include knowledge about image formation, and semantic-level methods include even more knowledge about the type of scenes being viewed. The surface-based segmentation algorithm in this paper is a signal-level method where the uniformity predicate on pixels is true if almost all the pixel data in a region can be represented well by an approximating (surface) function.

Functional approximation ideas have been used on intensity images in the past to define uniformity measures for region-growing segmentation at the signal-level. Pavlidis [80] developed a region-growing segmentation approach based on a piecewise-linear scanline function approximation. Scanline intervals with similar slopes were merged to define regions. The uniformity predicate requires pixels in the same region to be approximated by straight lines with similar slopes. Haralick and Watson [48] proved the convergence of the facet iteration algorithm for flat (constant), sloped (planar), and quadratic polynomial facets (local surfaces) defined over preselected image window sizes. This algorithm was intended more for noise removal than segmentation, but may be considered as a segmentation algorithm where the image segments are the resulting small facets. A physical surface in an image is typically represented by many image facets. The uniformity predicate in this case requires pixels are well approximated by the facet surfaces. The window operator size for the surface fits, which limits the facet size, and the surface type are preselected parameters independent of the data. Pong et al. [82] have obtained good results with a similar algorithm based on property vectors of facets rather than the facet surface fits.

Functional approximation ideas are also used to derive window coefficients [85] for edge detection approaches to segmentation. In most edge-based techniques, pixels are simply labeled as edge or non-edge, and an edge-linking step is required to create refined region descriptions. A good example of a more complete pixel labeling scheme based on local surface function (facet) approximations is the topographic primal sketch [49]. In this approach, the output consists of 1) step edge, ridge, and valley lines, 2) peak, pit, saddle, and flat points, and 3) planar slopes, convex, concave, and saddle-shaped regions. The uniformity predicate in this case groups pixels with the same topographic label. This method is purely local however and does not prescribe the integration of global similarity information. The surface type labeling used in the surface-based segmentation algorithm also suffers from the same problem, but global information is effectively integrated by the iterative region growing algorithm.

C. Range Image Segmentation

Region growing based on function approximation ideas are used commonly in range image analysis (see survey [9]). The uniformity predicate in the work listed below requires that region pixels are well approximated by planar or quadric surfaces. Shirai and Suwa [94], Milgrim and Bjorklund [75], Henderson and Bhanu [53], Henderson [52], Bhanu [12], and Boyter [17] segment range images into fitted planar surfaces extracted via region growing. Other work has been geared toward detecting cylinders in range data [1], [77], [83], [15], [69]. Hebert and Ponce [51] segmented planes, cylinders, and cones from range data. Sethi and Jayaramamurthy [92] handled spheres and ellipsoids in addition to planes, cylinders, and cones. Oshima and Shirai [79] used planes, spheres, cylinders,

and cones. Dane [27] and Faugeras *et al.* [34] at INRIA allow for region growing based on planar or general quadric surface primitives. The above do not directly address other types of surfaces except that the INRIA [33] and Henderson/Bhanu approaches have worked with arbitrary curved surfaces represented by many-faceted polyhedral approximations. Many of these methods obtain an initial segmentation of small primitive regions and then iteratively merge the small primitives until all merges (allowed by a smoothness or approximation error constraint) have taken place. The RANSAC method of Bolles and Fischler [15] has used iterative model fitting directly on the data based on randomly selected initial points (seeds). Our approach also works directly on the data, but seed regions are extracted deterministically and the model itself may change as required by the data.

Concepts and techniques from differential geometry have been useful in describing the shape of arbitrary smooth surfaces arising in range images [19], [74], [96], [62], [36], [10], [102], [32]. (This approach has also been applied to intensity image description [72], [81], [11].) For segmentation based on differential geometric quantities, such as lines of curvature or surface curvature, the uniformity predicate requires region pixels to possess similar geometric properties. As the name implies, the *differential* geometry of surfaces analyzes the *local differences* of surface points. Although *global similarities* in surface structure are also analyzed, most theorems in differential geometry address only global topological similarities, such as the one-hole equivalence of a doughnut and a coffee cup with a handle. Global shape similarity theorems do exist for the surfaces of convex objects, and they have been successfully utilized in extended Gaussian image (EGI) convex surface shape matching schemes [55]. Difficulties arise when local descriptors are used to identify the shape and global similarities of arbitrary nonconvex object surfaces from arbitrary-viewpoint range-image projections. The mathematics of differential geometry gives little guidance for an integrated global shape description or for computational matching methods in this general case. Brady *et al.* [19] extract and analyze a dense set of integrated 3-D lines of curvature across entire surface patches to describe the global surface properties. This method can take hours to describe simple objects. Our approach integrates global information into parametric surface descriptions and runs in minutes on similar objects with similar computing power.

Many researchers have favored the extraction of lower-dimensional features, such as edges, to describe range images instead of surfaces [97], [59], [60], [39], [76], [100], [95], [54], [50], [16], [13]. The uniformity predicate in these approaches requires that range and the range gradient are continuous for all pixels in region interiors where only the region boundaries are computed explicitly. By detecting and linking range and range gradient discontinuities, the space curves that bound arbitrary smooth surfaces are isolated creating an image segmentation. However, most of the above edge-based work has focussed on

straight and circular edges for matching with polyhedra and cylinders and their combinations. Although edge-based approaches offer important computational advantages for today's computer vision systems, we believe that such systems cannot provide the detailed surface information that will be required from future general-purpose range-image vision systems. Only more research experience will determine the advantages and disadvantages of these approaches for different applications, but the generality of an arbitrary surface segmentation and surface description approach is necessary today for automated free-form, sculptured surface reconstruction and shape acquisition tasks as in [84].

II. PROBLEM DEFINITION

In the surface-based approach to segmentation, the relevant structure of an image is viewed as a piecewise-smooth graph surface contaminated by noise as defined below. We emphasize the geometric shape of the image data in this approach, *not* the noise process as in random field image models [106], [26], [30], [24]. Several terms are introduced to give a reasonably precise description of the problem we are attempting to solve.

A 3-D *smooth graph surface* is a twice-differentiable function of two variables:

$$z = f(x, y). \tag{3}$$

A *piecewise-smooth graph surface* $g(x, y)$ can be partitioned into smooth surface primitives $f_l(x, y)$ over support regions R_l:

$$z = g(x, y) = \sum_{l=1}^{N} f_l(x, y) \, \chi(x, y, R_l) \tag{4}$$

where $\chi(x, y, R_l)$ is the characteristic function of the region R_l, which is unity if $(x, y) \in R_l$ and zero otherwise. For each piecewise-smooth surface $g(x, y)$, it is convenient to associate a *region label function* $l_g(x, y)$ defined as

$$l_g(x, y) = \sum_{l=1}^{N} l \, \chi(x, y, R_l). \tag{5}$$

If $\vec{a}_l$ is the vector of all parameters needed to precisely specify the smooth function $f_l(x, y)$, then any piecewise-smooth surface may be represented as the piecewise-constant function $l_g(x, y)$ (with minimum value 1 and maximum value N), which contains all segmentation information, and the list of N parameter vectors $\{\vec{a}_l\}$, which contains all shape information.

A *digital surface* is a noisy, quantized, discretely sampled version of piecewise-smooth graph surface:

$$z_{ij} = \bar{g}(i, j) = \lfloor a\big(g\big(x(i), y(j)\big) \\ + n\big(x(i), y(j)\big)\big) + b \rfloor \tag{6}$$

where a and b are the quantizer's scale factor and offset respectively, the floor function indicates truncation (quantization) to an integer, and the additive noise process $n(x, y)$ is nominally zero-mean with finite variance

$\sigma^2(x, y)$ at each point. The discrete image location (i, j) need not be linearly related to the Euclidean (x, y) location allowing for the nonlinear relationships involved in some range sensors (see Appendix). A *range image* is a particular type of a digital surface where the z_{ij} values represent the distance to a physical surface from a reference surface. An *intensity image* is another type of digital surface where the z_{ij} values represent the number of visible photons incident at the (i, j) location in the focal plane of a camera. Other image types are defined based on the meaning of the sensed z_{ij} values. This underlying model is quite general and can be used to represent many types of images unless multiplicative noise or some other type of nonadditive noise is present. Many textured surfaces may also be considered as an approximating smooth surface plus random sensor noise along with structured noise to represent the given texture.

The *segmentation/reconstruction* problem that we are attempting to solve is a generalization of the segmentation problem and may be stated as follows. Given only a digital surface, denoted $\tilde{g}(i, j)$ and specified by the z_{ij} values, find $\hat{N}$ approximating functions $\hat{f}_l(x, y)$ and $\hat{N}$ image regions $\hat{R}_l$ over which those functions are evaluated such that the total image representation error

$$\epsilon_{\text{tot}} = \left\| \tilde{g}(i, j) - \hat{g}(x(i), y(j)) \right\|_I \quad (7)$$

between the reconstructed image function

$$\hat{g}(x, y) = \sum_{l=1}^{\hat{N}} \hat{f}_l(x, y) \chi(x, y, \hat{R}_l) \quad (8)$$

evaluated at the points $(x(i), y(j))$ and the data $\tilde{g}(i, j)$ is small and the total number of functions and regions $\hat{N}$ is small. The function norm is left unspecified, but may be the max norm, the (Euclidean) root-mean-square error norm, or the mean absolute error norm. The implicit logical segmentation predicate in the above problem statement may be written as the *surface coherence* predicate:

$$P_\epsilon(R_l) = \begin{cases} \text{TRUE} & \text{if } \left\| \tilde{g} - \hat{f}_l \right\|_{R_l} < \epsilon \\ \text{FALSE} & \text{otherwise} \end{cases} \quad (9)$$

where the value of ϵ depends on the mean variance of the noise process $n(x, y)$ in the image region.

The two trivial solutions may be discarded immediately. The "one pixel per region" solution minimizes the approximation error (zero error), maximizes the number of regions, and requires no work, but is of course also useless. The "one function per image" solution minimizes the number of regions (one region), maximizes the approximation error, requires work, may be useful for some purposes, but does not solve the real problem. We seek an algorithm that tends to segment images into regions that can be directly associated with meaningful high-level entities. In the case of range images, the surface functions defined over the image regions should mathe-

matically represent the 3-D shape of visible physical surfaces in the scene.

The problem statement places no constraints on the functions except that they are smooth, no constraints on the image regions except that they are connected, no constraints on the form of the additive noise term except that it is zero-mean. We want the total approximation error and the number of regions to be small, but we have not attempted to weight the relative importance of each. Without such weights, it is difficult to form an objective function and apply existing optimization methods.

It is not at all clear from the above statement that such a "chicken-and-egg" segmentation problem can be solved at the signal level. It is straightforward to fit functions to pixel data over regions if the regions have been determined, but how are the regions to be determined? Similarly, it is possible to determine the image regions if the set of functions are known, but how are the functions extracted? But even the number of functions/regions present in the data is not known. We seek a signal-level, data-driven segmentation procedure based only on knowledge of piecewise-smooth surfaces.

III. SMOOTH SURFACE DECOMPOSITION

The problem statement says that the smooth component functions $f_l(x, y)$ of the underlying model $g(x, y)$ are allowed to be arbitrary smooth surfaces, which can be arbitrarily complicated. However, arbitrary smooth surfaces can be subdivided into simpler regions of constant surface curvature sign based on the signs of the mean and Gaussian curvature at each point [10]. As shall be discussed in more detail later, there are only eight possible surface types surrounding any point on a smooth surface based on surface curvature sign: peak, pit, ridge, valley, saddle ridge, saddle valley, flat (planar), and minimal. These fundamental surface shapes, shown in Fig. 1, are very simple and do not contain inflection points (compare to codons for planar curve description as in [87]. Our hypothesis is that these simple surface types are well approximated for image segmentation (i.e., perceptual organization) purposes by bivariate polynomials of order M or less where M is small. The experimental results included here and in [8] attempt to show that this assumption is reasonable for a large class of images when $M = 4$ (biquartic surfaces). Even the range image surfaces of quadric primitives can be approximated well enough for segmentation purposes with such polynomial surfaces. This assumption is only limiting in the context of the segmentation algorithm when a large smooth surface bends much faster than x^4. If a particular application encounters a significant number of such surfaces, the limit of $M = 4$ can be raised. If a range imaging application can guarantee that only planar and quadric surfaces will appear, they can use only those types of functions for fitting purposes. In fact, any ordered set of bivariate approximating functions can be used if they satisfy the set of requirements defined below. In summary, *arbitrary smooth surfaces may be decomposed into a union of simple surface-*
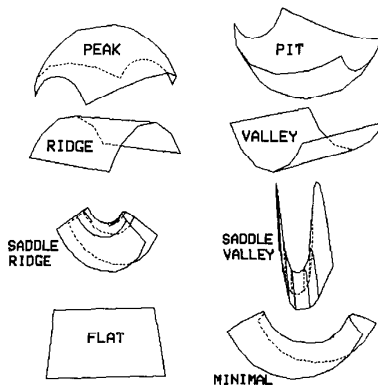
Fig. 1. Eight fundamental surface types from surface curvature sign.

curvature-sign primitives that are well approximated by low-order bivariate polynomials.

The intermediate goal of a segmentation algorithm then is to isolate all underlying *simple surfaces* (surface-curvature-sign primitives) in the image data and fit those simple surfaces with simple bivariate approximating functions. This creates an image segmentation in terms of the support regions of the simple approximating functions and an image reconstruction in terms of those simple approximating functions evaluated over the support regions. If the boundaries of the *smooth surfaces* of the underlying piecewise-smooth surface model are desired, then smoothly joining, adjacent simple surface regions can be merged to create the smooth surface support regions. This noniterative postprocessing step is covered in [8] and is not discussed here. We currently leave the function description over the smooth surface regions in the form of a collection of simple polynomial surfaces. The final collection of smooth surfaces and their support regions is the underlying piecewise-smooth image description that we wished to recover in the problem definition. In applications, it may be desirable to go back and fit the segmented smooth surface with application surfaces, such as quadrics, composite Bezier patches [35], or rational *B*-splines [99], rather than leaving it as a set of polynomial surfaces.

A. Approximating Function Requirements

Arbitrarily complicated smooth surfaces can be decomposed into a disjoint union of surface-curvature-sign surface primitives. If these surface primitives can be approximated well by a small set of approximating functions, a composite surface description for arbitrary smooth surfaces can be obtained. There are several constraints that the set of approximating functions must satisfy. Of course, the approximating functions must be able to approximate the fundamental surface-curvature-sign surface-primitives well. For *dimensionality reduction* reasons, the approximating functions should be representable by a small amount of data. For *generality*, the approximating surfaces must be well-defined over any arbitrary connected region in the image plane. The approximating functions

of the digital surface segmentation must be useful for *extrapolation* into neighboring areas of a surface in order for the region growing method to be successful. *Interpolation* capabilities are also useful for evaluating points between pixels if surface intersections are required. The approximating functions should also be easily *differentiable* so that differential geometric shape descriptors can be recomputed from them and so that other processes may compare surface normals and other differential quantities. Finally, the complete set of approximating functions should be *totally ordered* so that each approximant is capable of describing lower order approximants exactly, but cannot adequately approximate higher order functions. This provides the basis for a set of increasingly complicated hypotheses about the form of the underlying data. Note that general 3-D surface representation capability is not needed because digital surfaces are discrete representations of graph surface functions.

Low-order bivariate polynomials satisfy all of the above requirements, and the surface fitting procedure requires only a linear least-squares solver for the $p \times q$ ($p > q$) equation $[A] \vec{x} = \vec{b}$ [70], [5]. We have found that the set of planar, biquadratic, bicubic, and biquartic polynomials performed well in our experiments without significant computational requirements (a few seconds per fit on a VAX 11-780). *However, any set of approximating functions that satisfy the above constraints may be used instead.* To maintain generality in the algorithm description, it is only assumed that there is a set of approximating functions, denoted as F, that contains $|F|$ discrete types of functions that can be ordered in terms of the "shape potential" of each surface function relative to the set of fundamental surface-curvature-sign surface primitives.

In our case $|F| = 4$ and the set of approximating functions F can be written in the form of a single equation:

$$
\begin{aligned}
\hat{f}(m, &\vec{a}; x, y) \\
&= \sum_{i+j \leq m} a_{ij} x^i y^j \qquad\qquad\qquad (10) \\
&= a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 \\
&\quad + a_{02}y^2 + a_{21}x^2y + a_{12}xy^2 + a_{30}x^3 \\
&\quad + a_{03}y^3 + a_{31}x^3y + a_{22}x^2y^2 \\
&\quad + a_{13}xy^3 + a_{40}x^4 + a_{04}y^4. \qquad (m = 4) \quad (11)
\end{aligned}
$$

Planar surfaces are obtained by restricting the parameter vector space $\mathcal{R}^{15}$ to three-dimensional subspace where only a_{00}, a_{10}, a_{01} may be nonzero. Biquadratic surfaces are restricted to a six-dimensional subspace, and bicubic surfaces to a ten-dimensional subspace. A least-squares solver computes the parameter vector $\vec{a}$ and the RMS fit error ϵ from the digital surface data over a region quickly and efficiently. Moreover, a QR least-squares solution approach allows surface region fits to be updated recursively during the region growing process as new data points are added [42], [22] for better computational efficiency.

B. Simple to Complex Hypothesis Testing

The key idea behind the algorithm, which is independent of the set of approximating functions actually chosen, is that one should start with the simplest hypothesis about the form of the data and then gradually increase the complexity of the hypothesized form as needed. This is the *variable-order* concept, which has not been used in previous segmentation algorithms. In our case, surface type labels at each pixel allow us to find large groups of identically labeled pixels. Then, a small subset of those pixels, known as a *seed region*, is chosen using a simple shrinking method that attempts to ensure that every pixel in the seed region is correctly labeled. The simplest hypothesis for any surface fitting approach is that the data points represented in the seed region lie in a plane. The hypothesis is then tested to see if it is true. If true, the seed region is grown based on the planar surface fit. If the simple hypothesis is false, the algorithm responds by testing the next more complicated hypothesis (e.g., a biquadratic surface). If that hypothesis is true, the region is grown based on that form. If false, the next hypothesis is tested. This process continues until either 1) all preselected hypotheses have been shown to be false or 2) the region growing based on the surface fitting has converged in the sense that the same image region is obtained twice. Since all smooth surfaces can be partitioned into simple surfaces based on surface curvature sign, false-hypotheses may occur only when the isolated seed region surface-type labels are incorrect (due to noise) or when the underlying surface bends faster than the highest order approximating surface. During execution of the algorithm, bad seed regions are rejected immediately when the surface fit error is poor and large quickly bending surfaces are broken into two or more surface regions.

IV. ALGORITHM PHILOSOPHY

This section includes qualitative comments about the system structure of the surface-based segmentation algorithm. The success of the algorithm is based on the effective combination of simple component algorithms, not on the capabilities of any single processing step.

A. Initial Guess Plus Iteration

Like many region growing schemes, the basic approach of this algorithm might be summarized as "make an initial guess and then iteratively refine the solution." This idea is at least as old as Newton's method for finding the zeros of a complicated function. Unlike other region growing schemes, the initial guess at the underlying surface segmentation is based on invariant differential geometric principles and is quantified in terms of surface curvature sign labels, or surface type labels [10]. The iterative refinement process is based on function approximation and region growing. Once a surface has been fitted to the kth group of connected pixels, the $(k + 1)$th group of pixels is obtained by finding all new connected pixels that are compatible with the fitted surface of the previous group. When the same group of pixels is ob-

tained twice, the iteration terminates yielding an extracted region. This process is described in more detail later.

Although we shall not prove it in this paper, there is the usual relationship between the quality of the initial guess and the number of iterations required. If the initial guess is very good, only a few iterations are required. Many iterations may be required if the initial guess is not good. For bad initial guesses, no number of iterations will yield the proper convergence to a solution. In our algorithm, the quality of the initial guess is related to the quality of the image data, and the performance of the segmentation algorithm degrades gracefully with increasing noise levels.

B. Stimulus Bound Image Analysis

The variable-order surface fitting approach may be thought of as a hypothesize and test (hypothesize and verify) algorithm where the hypotheses can be automatically changed by the input data and each surface fit is bound by (must conform to) the input data. Therefore, we suggest the use of the adjective *stimulus bound* [89] for the type of hypothesis testing done by the surface-based segmentation algorithm, where the *stimulus* is the original sensed data values. In a stimulus bound process, all interpretive processing of the data is *bound* to or constrained by the original data or stimulus in each stage of processing to reduce the probability of interpretation errors. In our case, each simple surface function hypothesis is tested against the original data via surface fitting followed by two tests: 1) an RMS fit error test (related to the chi-square test), and 2) a regions test (related to the nonparametric statistics runs test). Hence, each iteration and the final interpretation are bound by the original stimulus.

It is generally acknowledged that vision algorithms should function at several different levels using associated vision modules to process the signal and symbol information at different levels. It often occurs that each level's vision module accepts input only from the previous, lower level and provides output only to the subsequent, higher level. Fig. 2(a) shows a typical example of such a process. This assumption may be rooted in human visual models where retinal information is not directly available to the high level cerebral processes. However, human vision is a fundamentally dynamic perceptual process in which subsequent, highly correlated "video frames" are always immediately available to the visual system after any given instant in time. Therefore, it may be inappropriate to apply dynamic human visual model principles to static computational vision problems. The *stimulus bound* philosophy states that the output from all lower level vision modules should be available to high-level vision modules. In particular, the original image from the sensor must be available to every vision module in a static vision system as shown in Fig. 2(b). In the surface-based segmentation algorithm, *every pixel in every region* is constantly checked to see how close the sensed value at a given pixel is to the approximating surface function for the given region. The global grouping of pixels relies on
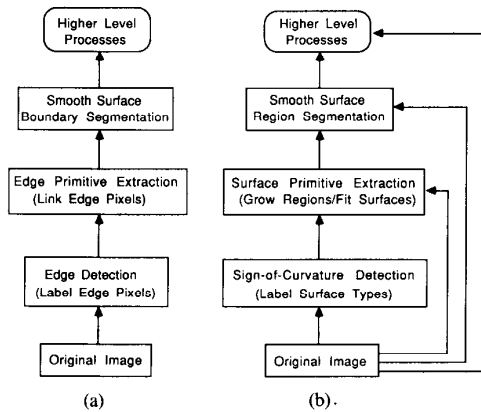
Fig. 2. Conventional edge approach versus stimulus bound approach.

simple differencing between pixel values and the interpreted surface primitives. These concepts are expressed in more detail in later sections. Without constantly checking symbolic interpretations against the original data, a vertical chain of interpretive vision modules is only as robust as the weakest module. Many edge-based intensity image vision schemes have failed in practice because precise, correctly linked edges could not be extracted from real images.

We believe that surface-based range image vision systems have an advantage over edge-based systems in that it is possible to quickly check final object or surface interpretations against the original data (via simple image differencing) because synthetic range images can be generated from models with a depth-buffer algorithm [78] using only the object or surface geometry; light and surface reflectance are not involved. The final interpretation in the form of a reconstructed range image can be subtracted from the original range image to create an *interpretation error image*, which can then be used to evaluate the quality of the image interpretation globally and locally. This is only possible when an image interpretation includes segmentation and reconstruction information as described in the problem definition.

C. Emerging Commitment

The surface-based segmentation algorithm is primarily data-driven in that only generic knowledge of surfaces, curvature, noise, and approximation are used. Of course, data-driven and model-driven elements must cooperate in any algorithm that attempts to interpret a digital image in terms of specific model information. An important feature of any image interpretation approach is the *process of commitment* to the final interpretation. A special-purpose model-driven program can make a commitment to its set of possible interpretations when the program is written or compiled [41], thus avoiding certain computations that might otherwise be required. A data-driven program may postpone making a commitment to a final interpretation in order to be more generally applicable, but it should reduce the amount of information that must be manipu-

lated by later higher-level processes that use specific world model information by generating intermediate symbolic primitives. We believe that our approach follows a *principle of emerging commitment* that is *gradual* and *locally reversible*, but not random. One must make steps toward image interpretation, yet it is impossible to always avoid errors that necessitate steps or labels being undone. An algorithm should make a series of small steps towards the goal, where each step need not produce perfect results, can easily be undone, but still produces useful information for the next step. Simulated annealing algorithms [4], [38] might also be said to follow a principle of emerging commitment, but the surface-based segmentation algorithm described here is very directed in its search process and provides a much more structured output.

V. ALGORITHM DESCRIPTION

The algorithm presented in this paper uses a general piecewise-smooth surface model to do pixel grouping assuming the image data exhibits *surface coherence* properties. If all pixel values are viewed as noisy samples of an underlying *piecewise-smooth surface* function defined over the entire image, the segmentation process should not only provide detailed definitions of the segmented regions, but should also provide the component surface functions of the underlying piecewise-smooth surface. Surface-based segmentation includes surface (image) reconstruction.

In the first stage of the segmentation algorithm described below, each pixel in an image is given a label based on its value and the values of its neighboring pixels. This label can only take on eight possible values based on two surface-curvature signs and indicates the qualitative shape of an approximating surface that best-fits the image data surrounding that point. This surface-type label image can be analyzed for connected regions using a standard connected component analysis algorithm [91], [3]. Any individual pixel label can be wrong, but it is likely to be correct if it lies in the interior of a large region of identically labeled pixels. Moreover, due to the constrained nature of the surface types represented by the eight labels, it is also likely that a simple surface function will approximate a group of correctly labeled pixels. The surface type label image is used to provide seed regions to the region-growing algorithm. A pixel's similarity with a group of other pixels is measured by comparing 1) the difference between the pixel's value and the other pixels' approximating surface value (at the given pixel location) and 2) a parameter that measures the goodness of the fit of the surface to the other pixels. This similarity measure allows for a pixel to enter and leave a group of pixels depending on the other pixels currently in that group. Hence, a mistake in grouping can be undone. Regions are grown until convergence criterion are met, and a concise, parametrically defined surface description is recorded along with a definition of the image region. It is common for images reconstructed from the segmentation description to be almost indistinguishable from the original image.

This algorithm can be viewed as a two-stage process. The first stage computes an "initial-guess" coarse segmentation in terms of regions of identical surface type labels. The second stage iteratively refines the coarse image segmentation and simultaneously reconstructs image surfaces. The entire algorithm is outlined below.

The first stage creates a surface type label image $T(i, j)$ from the original image $\bar{g}(i, j)$ in the following manner:

• Compute partial derivative images $\bar{g}_u(i, j)$, $\bar{g}_v(i, j)$, $\bar{g}_{uu}(i, j)$, $\bar{g}_{vv}(i, j)$, $\bar{g}_{uv}(i, j)$ from the original image $\bar{g}(i, j)$ using local fixed-window surface fits that are accomplished via convolution operators.

• Using the partial derivative images, compute the mean curvature image $H(\bar{g}_u, \bar{g}_v, \bar{g}_{uu}, \bar{g}_{vv}, \bar{g}_{uv})$ and the Gaussian curvature image $K(\bar{g}_u, \bar{g}_v, \bar{g}_{uu}, \bar{g}_{vv}, \bar{g}_{uv})$.

• Compute the sign $(+, -, 0)$ of mean curvature, denoted sgn (H), and the sign of Gaussian curvature, denoted sgn (K). The signum function sgn (x) maps negative numbers to -1, positive numbers to $+1$, and zero maps to zero.

• Use surface curvature sign to determine a surface type label $T(i, j)$ for each pixel (i, j).

The second stage performs iterative region growing using variable-order surface fitting as described below. Its input consists of the original image and the surface type label image. In order to determine the next (first) seed region to use, a connected component algorithm isolates the largest connected region of any surface type in the $T(i, j)$ image, and then a 3×3 binary image erosion operator shrinks the region until a small seed region of appropriate size is obtained. The output of the second stage consists of a region label image $\hat{l}_g(i, j)$, which contains all region definitions in one image (the segmentation information), and a list of coefficient vectors $\{\vec{a}_l\}$, one for each region (the shape reconstruction information).

---

1) **Declarations:**
   Surface-Order: $m \in F = \{1$ (Planar), 2 (Biquadratic), 3 (Bicubic), 4 (Biquartic)$\}$;
   Max-Surface-Order: $|F| = 4$ (Biquartic);
   Surface-Fit: $\{\vec{a} =$ Coefficient Vector (3, 6, 10, or 15 numbers), $\sigma =$ RMS Fit Error$\}$;
   Surface-Type-Image: $T(i, j)$ where $T \in \{1, 2, 3, 5, 6, 7, 8, 9\}$;
   Region-Label-Image: $\hat{l}_g(i, j)$ where $\hat{l} \in \{1, \cdots, \hat{N}\}$;
   Surface-Fit-List: $\{\vec{a}_l\}$ where $l \in \{1, \cdots, \hat{N}\}$;
   Reconstruction-Image: $\hat{g}(i, j)$;
   Error-Image: $e(i, j) = |\hat{g}(i, j) - \bar{g}(i, j)|$;
   Current-Region, New-Region, Seed-Region: Four-Connected Subsets of Image I
2) **Initialization:**
   Set *Error-Image* = Big Error Value;
   Set *Reconstruction-Image* = No Value;
   Set *Region-Label-Image* = No Label;
3) **Start-Iteration:**
   Set *Surface-Order* = Planar $(z = a + bx + cy)$;
   Set *Seed-Region* = Next-Seed-Region (*Surface-Type-Image*);
   IF *Seed-Region* Smaller Than Threshold Size (e.g., 30 pixels),
   THEN GoTo **All-Done;**
   ELSE Set *Current-Region* = *Seed-Region*;
4) **Surface-Fitting:**
   Perform *Surface-Order* Fit to z_{ij} Values in *Current-Region* to obtain *Surface-Fit*;
5) **Surface-Fit-Testing:**
   IF *Surface-Fit* OK using RMS Error Test and Regions Test,
   THEN GoTo **Region-Growing;**
   ELSE Increment *Surface-Order*;
   (Example: Planes become Biquadratics: $z = a + bx + cy + dxy + ex^2 + fy^2$)
   IF *Surface-Order* > *Max-Surface-Order*,
   THEN GoTo **Accept-Reject;**
   ELSE GoTo **Surface-Fitting;**
6) **Region-Growing:**
   Find *New-Region* Consisting of Compatible Connected Neighboring Pixels where Compatibility means Pixel Values must be Close to Surface and Residual Error must be Smaller Than Current Value in Error Image and Derivative Estimates from Pixel Values must be Close to Surface Derivatives;
   IF *Current-Region* $\approx$ *New-Region*,
   THEN GoTo **Accept-Reject;**
   ELSE Set *Current-Region* = *New-Region*; GoTo **Surface-Fitting;**

7) **Accept-Reject:**
   IF *Surface-Fit* OK using RMS Error Test,
   THEN GoTo **Accept-Surface-Region;**
   ELSE Zero Out *Seed-Region* Pixels in *Surface-Type-Image*; GoTo **Start-Iteration;**
8) **Accept-Surface-Region:**
   Zero Out *Current-Region* Pixels in *Surface-Type-Image*;
   Label *Current-Region* Pixels in *Region-Label-Image*;
   Evaluate *Current-Region* Pixels in *Reconstruction-Image* using *Surface-Fit*;
   Update *Current-Region* Pixels in *Error-Image* with Absolute Residual Errors;
   Add *Surface-Fit* to *Surface-Fit-List*;
   GoTo **Start-Iteration;**
9) **All-Done:**
   Surface-Fit-List Contains All Function Definitions for Image Reconstruction
   Region-Label-Image Contains All Region Definitions for Image Segmentation
   Reconstruction-Image Contains Noiseless, Smooth Surface Version of Original Image;
   Error-Image Contains Approximation Error at Each Pixel to Evaluate Reconstruction Quality;

It is not necessary to maintain a separate version of the reconstructed image as this can always be recomputed from the surface fit list and the region label image. However, displaying this image during program execution is an excellent way to monitor the progress of the algorithm. The error image can also be recomputed from the surface fit list, the region label image, and the original image, but it is maintained throughout the iteration process to counteract the tendency of surfaces without sharp boundaries to grow slightly beyond their actual boundaries. The error image is updated at each pixel with the absolute error between the approximating surface and the original data when a surface/region is accepted. During the region growing procedure, the error image is consulted to see if the current approximating function represents a given pixel better than it has been represented before. If so, a pixel that was labeled as a member of a previously determined region is free to be labeled with a better fitting region as long as the pixel is connected to the better fitting region. Thus, *labeling decisions are reversible*. Later surfaces in this sequential algorithm can relabel a pixel even though it was already labeled as part of another surface.

The algorithm above terminates when the next seed region extracted from the surface type label image is too small (e.g., less than 30 pixels). However, some pixels may still be unlabeled at this point. These pixels are coalesced into a binary surface type image in which all pixels that have already been labeled are turned off (black) leaving all unlabeled pixels on (white). This new "left-overs" surface type image is then processed by extracting and fitting the next seed region as usual except that the region growing constraints are relaxed (e.g., the allowable RMS fit error limit is doubled). When the next seed region from the left-overs surface type image is too small, the algorithm finally terminates.

The outline above provides a high-level description of all the main elements of the segmentation algorithm. We have omitted several details that are covered in subsequent sections. The algorithm as stated here does not always yield clean high-quality edges between regions, and

it is still possible that some pixels may be left unlabeled (ungrouped with a surface). Hence, a local region refinement operator capable of cleaning up pixel-size irregularities was used to create the final segmentations shown in the experimental results section. Also, as mentioned above, surface curvature sign primitive regions must be merged at polynomial surface primitive boundaries that lie within the boundaries of a smooth surface. The details on a region refinement operation and a one-step region merging method for smoothly joining surface primitive boundaries are available in [8], and further enhancements are currently being developed. These fine points are not at all related to the performance of the segmentation algorithm as described here since the necessary procedures are performed after the termination of the iterative region growing.

VI. Noise Estimation for Threshold Selection

Digital surfaces exhibit the property of surface coherence when sets of neighboring pixels are spatially consistent with each other in the sense that those pixels can be interpreted as noisy, quantized, sampled points of some relatively smooth surface. In order for the surface-based segmentation algorithm to group pixels based on underlying smooth surfaces, it needs to know how well the approximating functions should fit the image data. This information should be derived from the image data in a data-driven algorithm. If the noise in the image is approximately stationary ($\sigma^2(x, y) \approx \sigma_{img}^2 = $ constant), we can compute a single estimate of the noise variance σ_{img}^2 (that should be applicable at almost all image pixels) by averaging estimates of the noise variance at each pixel. To compute an estimate of the noise variance at each pixel, we perform a equally-weighted least-squares planar fit in the 3×3 neighborhood W_3 surrounding the pixel. If the pixel lies in the interior portion of a smooth surface region and if the radius of the mean surface curvature is larger than a few pixels, the error in the planar surface fit will be primarily due to noise. In contrast, steeply sloped image regions typically have large mean curvatures and

bad planar fits. To get a good estimate of the magnitude of the additive noise and the quantization noise in the image, it is necessary to exclude these pixels where the gradient magnitude is large. Therefore, we only include pixels in the mean noise variance calculation if the gradient magnitude is below a preset threshold (8 levels/pixel was used in our experiments). A more detailed discussion of this idea is given in [8]. The equation for the mean image noise variance σ_{img}^2 may be expressed as

$$\sigma_{img}^2 = E(\sigma_{W_3}^2) = \frac{1}{N_{int}} \sum_{l=1}^{N_{int}} \left(\sum_{p \in (R_l - \partial R_l)} \sigma_{W_3}^2(p) \right) \quad (12)$$

where ∂R represents the boundary of the region R, N_{int} is the total number of surface interior pixels contributing to the sum, and where $\sigma_{W_3}(p)$ is the root-mean-square-error (RMSE) of the least-squares planar surface fit (a_{00}, a_{10}, a_{01}) in the 3×3 window W_3 around the pixel p:

$$\sigma_{W_3}^2(p) = \frac{1}{9} \sum_{(i,j) \in W_3} \left(z_{ij} - (a_{00} + a_{10}i + a_{01}j) \right)^2 \quad (13)$$

where i and j are interpreted as integer row and column coordinates. Although the regions themselves are not known at the time the noise variance is estimated, we get a good approximation to σ_{img} by not averaging pixels with high slopes using the preset threshold.

The noise variance estimate allows us to automatically set the ϵ parameter of the surface coherence predicate (the maximum allowable RMS surface fit error) and two other thresholds to an appropriate value as described later. Note that we are attempting to estimate noise variance for *continuous smooth surface detection* purposes, *not for discontinuity detection* as in [45]. Although we do not claim to have solved the automatic threshold selection problem, the three relevant thresholds are directly tied to the geometric and statistical properties of the data via empirical relationships providing good performance for many images. Other noise variance estimation techniques, such as computing the mean square difference between a median filtered version of an image and the original image, are currently being evaluated.

VII. SURFACE TYPE LABELING

Differential geometry states that local surface shape is *uniquely determined* by the first and second fundamental forms. Gaussian and mean curvature combine these first and second fundamental forms in two different ways to obtain scalar surface features that are *invariant to rotations, translations, and changes in parameterization* [8]. Therefore, visible surfaces in range images have the same mean and Gaussian curvature from any viewpoint under orthographic projection. Also, *mean curvature uniquely determines the shape of graph surfaces* if a boundary curve is also specified [40] while *Gaussian curvature uniquely determines the shape of convex surfaces and convex regions of nonconvex surfaces* [23], [55]. There are eight fundamental viewpoint independent surface types that can be characterized using only the sign of the mean curvature (H) and Gaussian curvature (K) as shown in Fig.

	$K > 0$	$K = 0$	$K < 0$
$H < 0$	Peak T=1	Ridge T=2	Saddle Ridge T=3
$H = 0$	(none) T=4	Flat T=5	Minimal Surface T=6
$H > 0$	Pit T=7	Valley T=8	Saddle Valley T=9

Fig. 3. Surface type labels from surface curvature sign.

3. Gaussian and mean curvature can be computed directly from a range image using window operators that yield least squares estimates of first and second partial derivatives as in [2], [6], [48]. The key point is that every pixel in an image can be given a surface type label based on the values of the pixels in a small neighborhood about that pixel.

Surface curvature estimates are extremely sensitive to noise because they require the estimation of second derivatives, in which high frequency noise is amplified. In fact, 8-bit quantization noise alone can seriously degrade the quality of surface curvature estimates unless large window sizes are used (at least 9×9). Yet reliable estimates of surface curvature sign can still be computed in the presence of additive noise and quantization noise [10]. Since we need to compute five different derivative estimates to compute surface curvature, we could use large $N \times N$ derivative estimation window operators (N odd), or we can smooth the image with a small $L \times L$ window operator (L odd), store the smoothed values at higher precision, and operate on the smoothed image with smaller $M \times M$ derivative estimation window operators (M odd) where $L + M = N + 1$. Assuming window separability and therefore linear time requirements, the former requires time proportional to $5N$ whereas the latter requires time proportional to $N + 4M + 1$. The relative weighting factors used in determining the derivative and smoothing window coefficients have an important influence on the quality of the derivative estimates. In our experiments with 8-bit images, we obtained good consistent results using one 7×7 binomial weight (approximately Gaussian) smoother and five 7×7 equally weighted least squares derivative estimation operators with over 30 percent fewer computations than the equivalent 13×13 windows. For reference purposes, we list the specific numbers needed for this particular computation.

Since all our operators are separable, window masks can be computed as the outer product of two column vectors. The binomial smoothing window may be written as $[S] = \vec{s}\vec{s}^T$ where the column vector $\vec{s}$ is given by

$$\vec{s} = \frac{1}{64}[1 \ 6 \ 15 \ 20 \ 15 \ 6 \ 1]^T. \quad (14)$$

For 7×7 binomial smoothing window, it is clear that we should try to maintain an extra 12 bits ($12 = 2\log_2(64)$) of fractional information in the intermediate image smoothed by $[S]$. For an $L \times L$ binomial smoother, $2L - 2$ bits of fractional information should be maintained. The equally weighted least-squares derivative estimation window operators are given by

$$[D_u] = \vec{d}_0 \vec{d}_1^T \quad [D_v] = \vec{d}_1 \vec{d}_0^T \quad [D_{uu}] = \vec{d}_0 \vec{d}_2^T$$

$$[D_{vv}] = \vec{d}_2 \vec{d}_0^T \quad [D_{uv}] = \vec{d}_1 \vec{d}_1^T \tag{15}$$

where the column vectors $\vec{d}_0$, $\vec{d}_1$, $\vec{d}_2$ for a 7×7 window are given by

$$\vec{d}_0 = \tfrac{1}{7} [1 \; 1 \; 1 \; 1 \; 1 \; 1 \; 1]^T \tag{16}$$

$$\vec{d}_1 = \tfrac{1}{28} [-3 \; -2 \; -1 \; 0 \; 1 \; 2 \; 3]^T \tag{17}$$

$$\vec{d}_2 = \tfrac{1}{84} [5 \; 0 \; -3 \; -4 \; -3 \; 0 \; 5]^T \tag{18}$$

The partial derivative estimate images are computed via the appropriate 2-D image convolutions (denoted $*$):

$$\tilde{g}_u(i,j) = D_u * S * \tilde{g}(i,j)$$

$$\tilde{g}_v(i,j) = D_v * S * \tilde{g}(i,j) \tag{19}$$

$$\tilde{g}_{uu}(i,j) = D_{uu} * S * \tilde{g}(i,j)$$

$$\tilde{g}_{vv}(i,j) = D_{vv} * S * \tilde{g}(i,j)$$

$$\tilde{g}_{uv}(i,j) = D_{uv} * S * \tilde{g}(i,j) \tag{20}$$

Mean (H) and Gaussian (K) curvature images are computed using the partial derivative estimate images:

$$H(i,j) = \frac{\left(1 + \tilde{g}_v^2(i,j)\right) \tilde{g}_{uu}(i,j) + \left(1 + \tilde{g}_u^2(i,j)\right) \tilde{g}_{vv}(i,j) - 2\tilde{g}_u(i,j)\, \tilde{g}_v(i,j)\, \tilde{g}_{uv}(i,j)}{2\left(\sqrt{1 + \tilde{g}_u^2(i,j) + \tilde{g}_v^2(i,j)}\right)^3} \tag{21}$$

$$K(i,j) = \frac{\tilde{g}_{uu}(i,j)\, \tilde{g}_{vv}(i,j) - \tilde{g}_{uv}^2(i,j)}{\left(1 + \tilde{g}_u^2(i,j) + \tilde{g}_v^2(i,j)\right)^2}. \tag{22}$$

A toleranced signum function

$$\text{sgn}_\epsilon(x) = \begin{cases} +1 & \text{if } x > \epsilon \\ 0 & \text{if } |x| \le \epsilon \\ -1 & \text{if } x < \epsilon \end{cases} \tag{23}$$

is used to compute the individual surface curvature sign images $\text{sgn}_\epsilon(H(t,j)$ and $\text{sgn}_\epsilon(K(i,j))$ using a preselected zero threshold ϵ. For our experimental results, we used $\epsilon_H = 0.03$ and $\epsilon_K = 0.015$ for 7×7 windows. Ideally, these thresholds should depend on the noise variance estimate, but the algorithm performance is not very sensitive to these numbers for reasonable quality images.

The surface curvature sign images are then used to determine the surface type image:

$$T(i,j) = 1 + 3(1 + \text{sgn}_{\epsilon_H}(H(i,j))$$

$$+ \left(1 - \text{sgn}_{\epsilon_K}(K(i,j))\right). \tag{24}$$

This equation is shown in table form in Fig. 3. Fig. 1 displayed the eight fundamental shapes. Depending on the number of digitized bits and the amount of noise in the original image and the window sizes used in derivative estimation, regions of a given surface type label tend to connect (in the sense of four-connectedness) with distinct,

but adjacent regions with the same label. Therefore, it is not always possible, in general, to simply isolate a four-connected region of pixels of a particular surface type label and identify that region as a single surface of the appropriate type for surface fitting purposes. Hence, there is a need for a general purpose method to isolate useful interior seed regions from the larger regions of identical surface type labels extracted from the surface type label image $T(i,j)$.

VIII. SEED REGION EXTRACTION

We adopted the following strategy that breaks the unwanted connections with other adjacent regions and attempts to provide small, maximally interior regions that are good for surface fitting. The largest connected region of any fundamental surface type in the surface type label image is isolated (denoted R_T^0) and is then eroded (contracted) repetitively (using a 3×3 binary region erosion operator) until the region disappears. After the kth contraction (erosion), there exists a largest four-connected subregion R_T^k in the pixels remaining after the k contractions of the original region. If we record $|R_T^k|$, the number of pixels in the largest connected subregion, as a function

of the number of contractions k, a *contraction profile* for the original region is created. Contraction profiles for five regions of a surface type label image (for the coffee cup range image) are shown in Fig. 4. A seed region size threshold t_{seed} for the minimum number of pixels required to be in a seed region (e.g., 10) is a preselected parameter. If we examine the contraction profile, there will always be an contraction number k such that $|R_T^k| \ge t_{\text{seed}}$ and $|R_T^{k+1}| < t_{\text{seed}}$. The region R_T^k is selected as the *seed region* (or kernel region) for subsequent surface fitting and region growing. The circles in Fig. 4 indicate the size of the selected seed region. The threshold t_{seed} must always be greater than or equal to the minimum number of points required for the simplest surface fit (i.e., 3 points for a plane).

The fundamental purpose of the contraction profile computation for seed region extraction is to find a small enough isolated region that 1) is not inadvertently connected to any separate, but adjacent surface regions, and 2) is far enough inside the boundaries of the actual surface primitive to provide good surface fitting. The 3×3 erosion operation (i.e., zero out pixels that have zero-valued neighbors and leave other pixels alone) is a simple, common image processing operation that can be accomplished in less than a video frame time on existing image processing hardware. Other methods for obtaining seed re-
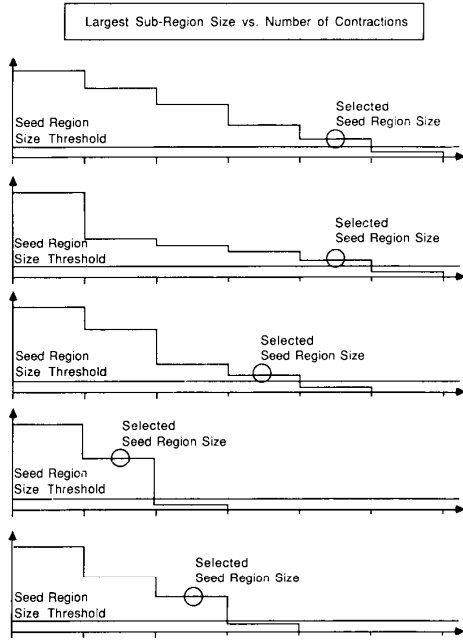
Fig. 4. Contraction profiles for five surface type regions.

gions are possible, but this method is simple and potentially very fast.

IX. ITERATIVE VARIABLE ORDER SURFACE FITTING

A plane is always fitted first to the small seed region using equally weighted least squares. If the seed region belongs to a surface that is not extremely curved, a plane will fit quite well to the digital surface defined by the original image data. If the plane fits the seed region within the maximum allowable RMS error threshold $\epsilon_{max} = w_1\sigma_{img}$, then the seed is allowed to grow. The value of w_1 was empirically set to 2.5 in our experiments to allow for variations in the image noise process. If not, the seed is fitted with the next higher-order surface (e.g., biquadratic), and the algorithm proceeds similarly. When the seed is allowed to grow, the functional description of the surface over the seed region is tested over the entire image to determine what pixels are compatible with the seed region as described in the next section.

This surface fitting process may be stated more precisely as follows. Let I be the rectangular image region over which a hypothetical piecewise smooth function $z = g(x, y)$ is defined. Let $\hat{R}_l^{k=0}$ denote the seed region provided by the seed extraction algorithm that is assumed to be contained in the unknown actual region R_l in the image: $\hat{R}_l^{k=0} \subseteq R_l \subseteq I$. The seed region $\hat{R}_l^{k=0}$ must be converted to a full region description $\bar{R}_l$ that approximates the desired region description R_l.

Now, let $\vec{a}_l^{\,k}$ be the parameter vector associated with the functional fit to the pixel values in the given region $\hat{R}_l^k$ of the kth iterative surface fit. Let $\mathfrak{R}^n$ denote the set of all parameter vectors for the set of approximating functions F, and let $|F|$ be the number of different types of surface

functions to be used. A particular function type (or fit order) is referred to as m^k where $1 \le m^k \le |F|$. The general fitting function of type m^k is denoted $z = \hat{f}(m^k, \vec{a}_l^{\,k}; x, y)$. The general surface fitting process, denoted $L_{\hat{f}}$, maps the original image data $\tilde{g}(x, y)$, a connected region definition $\hat{R}_l^k$, and the current fit order m^k into the range space $\mathfrak{R}^n \times \mathfrak{R}^+$ where $\mathfrak{R}^+$ is the set of possible errors (nonnegative real numbers):

$$(\vec{a}_l^{\,k}, \epsilon_l^k) = L_{\hat{f}}(m^k, \hat{R}_l^k, \tilde{g}) \qquad (25)$$

and has the property that the error metric

$$\epsilon_l^k = \left\| \hat{f}(m^k, \vec{a}_l^{\,k}; x, y) - \tilde{g}(x, y) \right\|_{\hat{R}_l^k} \qquad (26)$$

is the minimum value attainable for all functions of the form specified by m^k. Equally weighted least-squares surface fitting minimizes the error metric

$$(\epsilon_l^k)^2 = \frac{1}{|\hat{R}_l^k|} \sum_{(x,y)\in\hat{R}_l^k} \left(\hat{f}(m^k, \vec{a}_l^{\,k}; x, y) - \tilde{g}(x, y) \right)^2 \qquad (27)$$

where $|\hat{R}_l^k|$ is the number of pixels in the region $\hat{R}_l^k$ (the area of the region). The parameter vector $\vec{a}_l^{\,k}$ and the surface fit order m^k are passed onto the region growing procedure if the RMS fit error test and the regions test are passed. Otherwise, m^k is incremented and the higher order surface is fitted. If all four fit orders were tried and the error was never less than the threshold ϵ_{max}, the seed region is rejected by marking off the pixels in the surface type label image, and then continuing by looking for the next largest connected region of any surface type.

A. RMS Fit Error Test

The RMS fit error test tests the surface fit error, which measures the variance of the error of the fit due to the noise in the data, against the maximum allowable fit error as determined from the noise variance estimate for the image: $\epsilon_l^k < \epsilon_{max} = w_1\sigma_{img}$. If the error is small enough, the surface fit passes the test; otherwise, it fails. The coefficient $w_1 = 2.5$ is an empirically determined parameter.

B. Regions Test

The regions test is required because it is possible for a lower order function to fit a higher order function over a finite region within the maximum allowable fit error threshold even though the lower order fit is not appropriate. It is possible to detect the presence of a higher order function in the data (without letting the fit error increase all the way up to the error threshold) by analyzing the distribution of the sign of the fit errors (residual errors) at each individual pixel of the fit. We have generalized the runs test of nonparametric statistics [28] to assist in the detection of higher order behavior. This test is discussed in detail in [8], and is summarized here.

Consider that three long residual-sign intervals occur when fitting a line to a slowly bending curve as in Fig. 5. Fitting a plane to a small portion of a sphere is very similar except that two large residual-sign regions occur as is
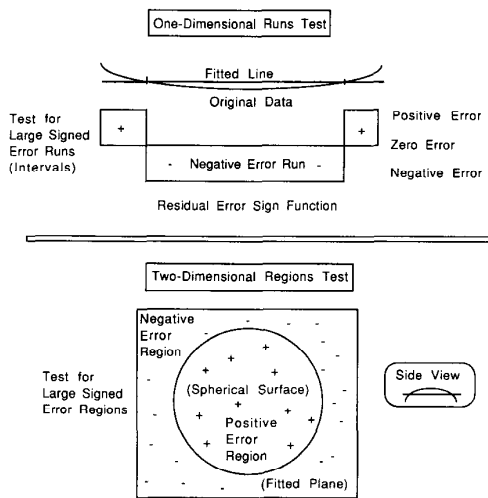
Fig. 5. Runs test and regions test ideas for noiseless data examples.

also shown in Fig. 5. The regions test is performed as follows: 1) For each original pixel value in the region $\hat{R}_l^k$ lying *above* the fitted surface $\hat{f}(m^k, a_l^k; x, y)$, turn a pixel on in a *positive residual error* image; for each pixel below the surface, turn a pixel on in a negative error image (all pixels initially off in both images). 2) Perform one 3×3 erosion on each error image and count the pixels left in the largest connected region in each image. 3) If either count is greater than r_t percent of the size of the current region $|R_l^k|$, then increase the fit order m^k. The experimental results used a regions test threshold of $r_t = 0.9 + 0.2\sigma_{\text{img}}$ percent, which was determined empirically.

X. REGION GROWING

After a surface of order m^k is fitted to the region $\hat{R}_l^k$ in the kth iteration, the surface description is used to grow the region into a larger region where all pixels in the larger region are connected to the original region and compatible with the approximating surface function for the original region. The parallel region growing algorithm accepts as input the original digital surface $\tilde{g}(x, y)$, the approximating function $\hat{f}(m^k, \vec{a}_l^k; x, y)$ and the surface fit error ϵ_l^k from the surface fitting algorithm. It does not use the region definition until later. To determine the zeroth-order "surface continuity" compatibility of each pixel $p \in I$ with the approximating surface description, the polynomial based prediction for the pixel value and the actual pixel value

$$\hat{z}(p) = \hat{f}(m^k, \vec{a}_l^k; x(p), y(p)) \quad \text{and}$$

$$z(p) = \tilde{g}(x(p), y(p)) \qquad (28)$$

are compared to see if the pixel p is compatible with the approximating surface function. If the magnitude of the difference between the function value and the digital surface value is less than the allowed tolerance value, denoted $w_0\epsilon_l^k$, then the pixel p is added to the set of compatible pixels, denoted $C(m^k, \vec{a}_l^k, \epsilon_l^k)$, which are com-

patible with the surface fit to the region $\hat{R}_l^k$. Otherwise, the pixel is incompatible and discarded. The result of this process is the compatible pixel list:

$$C(m^k, \vec{a}_l^k, \epsilon_l^k) = \left\{ p \in I \colon \left| \hat{z}(p) - z(p) \right| \leq w_0\epsilon_l^k \right\}.$$
$$(29)$$

This set of compatible pixels $C(\cdot)$ is essentially a thresholded absolute value image of the difference between the original image data and the image created by evaluating the function $\hat{f}$ at each pixel. For our experimental results, the factor $w_0 = 2.8$ was used. This ensures that approximately 99.5 percent of all samples of a smooth surface corrupted by normally distributed measurement noise will lie within this error tolerance. This factor has been found to work well in the presence of other types of noise also.

The compatible pixel list is then post processed to remove any pixels that do not possess "surface normal continuity" compatibility with the approximating surface. Let $\tilde{g}_u(p)$ and $\tilde{g}_v(p)$ denote the first partial derivative estimates of the local surface as computed from the image data at the pixel p via convolutions as mentioned earlier. Let $\hat{g}_u(p)$ and $\hat{g}_v(p)$ denote the first partial derivatives of the approximating surface as computed from the polynomial coefficients at the pixel p. Let $\tilde{n}$ be the unit normal vector as determined by the data, and let $\hat{n}$ be the unit normal vector as determined by the approximating surface:

$$\tilde{n} = \frac{[-\tilde{g}_u \ -\tilde{g}_v \ 1]^T}{\sqrt{1 + \tilde{g}_u^2 + \tilde{g}_v^2}} \qquad \hat{n} = \frac{[-\hat{g}_u \ -\hat{g}_v \ 1]^T}{\sqrt{1 + \hat{g}_u^2 + \hat{g}_v^2}}.$$
$$(30)$$

A pixel is compatible in the sense of surface normal continuity if the angle between the two unit normals is less than some threshold angle θ_t:

$$\cos^{-1}(\hat{n} \cdot \tilde{n}) \leq \theta_t. \qquad (31)$$

For our experimental results, the threshold angle is given by $\theta_t = 12 + 16\sigma_{\text{img}}$ degrees, where the coefficients were determined empirically. The test may be rewritten in the following form to avoid square roots and to incorporate the derivative values directly:

$$\frac{(\tilde{g}_u - \hat{g}_u)^2 + (\tilde{g}_v - \hat{g}_v)^2 + (\tilde{g}_u\hat{g}_v - \tilde{g}_v\hat{g}_u)^2}{(1 + \tilde{g}_u^2 + \tilde{g}_v^2)(1 + \hat{g}_u^2 + \hat{g}_v^2)}$$
$$\leq \sin^2(\theta_t). \qquad (32)$$

Since the compatibility test for surface normal continuity involves many computations per pixel, it is only applied to those pixels that have passed the compatibility test for surface continuity. Excellent segmentation results have been obtained without the surface normal continuity test on many images that lack small orientation discontinuities. However, a data-driven smooth-surface segmentation algorithm must always perform the test to ensure that growing regions do not inadvertently grow over small or noisy orientation discontinuities.

A. Region Iteration

When the parallel region growing computation has operated on every pixel, the compatible pixel list $C(m^k, \vec{a}_l^k, \epsilon_l^k) \subseteq I$ is complete. The largest connected region in this set of pixels that overlaps the seed region $\hat{R}_l^k$ must then be extracted to create the next region $\hat{R}_l^{k+1}$. This process is denoted $\Lambda(\cdot)$. The output region $\hat{R}_l^{k+1}$ must have the property that it is the largest connected region in the list of compatible pixels satisfying

$$\hat{R}_l^k \cap \hat{R}_l^{k+1} \neq \phi = \text{Null Set} \qquad (33)$$

because it is possible to get larger connected regions in the compatible pixel list than the connected region corresponding to the seed region. The iterative process of region definition via largest, overlapping, connected region extraction may be expressed as follows:

$$\hat{R}_l^{k+1} = \Lambda\big(C(m^k, \vec{a}_l^k, \epsilon_l^k), \hat{R}_l^k\big) = \Phi(\hat{R}_l^k) \qquad (34)$$

where $\Phi(\cdot)$ represents all operations required to compute the region $\hat{R}_l^{k+1}$ from the region $\hat{R}_l^k$. It is interesting to note that since the regions of an image form a metric space [8], the desired solution region is a fixed point $R = \Phi(R)$ of the mapping $\Phi(\cdot)$.

The new region is then considered as a seed region and processed by the surface fitting algorithm

$$(\vec{a}_l^{k+1}, \epsilon_l^{k+1}) = L_f(m^{k+1}, \hat{R}_l^{k+1}, \tilde{g}) \qquad (35)$$

to obtain a new parameter vector and a new surface fit error. If this region is allowed to grow again $\epsilon_l^{k+1} < \epsilon_{\max}$, then the compatible pixel list is recomputed $C(m^{k+1}, \vec{a}_l^{k+1}, \epsilon_l^{k+1})$, the largest connected overlapping region of $C(\cdot)$ is extracted, and so on until the termination criteria are met.

B. Sequential Versus Parallel Region Growing

The region growing process is formulated above for a parallel implementation where bivariate polynomials are evaluated over images and regions. It must be noted that this simple, parallel region growing formulation is equivalent to more complicated, sequential, spiraling region growing approaches *until the last iteration*. At the last iteration, the processing of the compatible pixel list becomes an important feature of the segmentation algorithm. After the growing region has been accepted, any other sufficiently large, reasonably shaped regions in the compatible pixel list are also accepted as part of the same surface. For example, in the coffee cup range image shown in the experimental results section, the flat background visible through the handle of the cup is correctly assigned to the larger background surface without high-level knowledge, only the surface compatibility concepts. Thus, nonadjacent compatible regions can be labeled as such during the surface acceptance stage without further postprocessing operations because of the parallel region growing process during the last iteration prior to acceptance. On a sequential machine, sequential region growing methods can offer faster performance for the other it-

erations, but the parallel formulation only takes a few seconds on a VAX 11/780 for the 128 × 128 images shown in the experimental results section.

XI. TERMINATION RULES

The termination criteria are expressed as the following set of rules:

1) IF $|\hat{R}_l^k| \approx |\hat{R}_l^j|$ for any $j < k$, THEN stop! Basically, this rule states the condition that we are looking for a fixed point of the mapping Φ in the metric space of image regions. Note that only the size of the region needs to be checked from iteration to iteration, not the detailed region description.

2) IF $\epsilon_l^k > \epsilon_{\max}$ AND $m^k \geq |F|$, THEN stop! The image data is varying in a way that the highest order function cannot approximate.

3) At least two iterations are required for a given surface fit order m^k before the algorithm is allowed to stop.

These rules state the essential concepts involved in terminating the surface fitting iteration. There is also a maximum limit on the number of possible iterations to prevent extremely long iterations. In all tests done to this point, the maximum limit of 30 iterations has never been reached and the average number of iterations is approximately eight.

XII. SURFACE ACCEPTANCE AND REJECTION DECISIONS

After the surface growing iterations have terminated, we are left with the set of compatible pixels and the connected surface region itself along with the function parameters and the fit error. For growth surface regions that exceed the error threshold $\epsilon_{\max}$, but not by much, an acceptance zone is defined above the error threshold such that surface regions within the acceptance zone are accepted. The acceptance threshold used for our experiments is 50 percent greater than $\epsilon_{\max} = w_1 \sigma_{\text{img}}$ where $w_1 = 2.5$. Surface regions with fit errors beyond the acceptance zone are rejected.

When a surface region is rejected for any reason, the seed region responsible for the surface region is marked off in the surface type label image as having been processed, which prohibits the use of the same original seed region again. When a surface region is accepted, all pixels in that region are similarly marked off in the surface type label image so that they are not considered for future seed regions. In this respect, surface rejection and surface acceptance are similar. However, the surface acceptance process also updates the region label image, the reconstruction image, and the error image. In addition, the acceptance process dilates the accepted region description and checks if there are any connected groups of pixels in that dilated region that are surface-continuity compatible with the accepted surface and connected with the accepted region. Surface-normal compatibility is not required when adding these pixels because of the difficulty in getting accurate surface normal estimates near surface region boundaries.

XIII. Experimental Results

The surface-based segmentation algorithm has been applied successfully to more than 40 test images. In this section, the segmentation algorithm's performance on six range images and three intensity images is discussed. The following set of images is displayed for each input image:

1) original gray scale image (upper left).
2) surface type label image (lower left).
3) region label image segmentation plot (lower right).
4) reconstructed gray scale image (upper right).

The surface type label image shows the coarse "initial guess" segmentation provided by labeling each pixel with one of eight labels according to the sign of the mean and Gaussian curvature. Each region in this image is an isolated set of connected pixels that all have the same surface type label. The region label image shows the final refined segmentation obtained from the iterative region-growing algorithm. Each region in this image is the support region over which a particular polynomial surface function is evaluated. The reconstructed image is computed from the region label image and the list of surface parameters, and it shows the visual quality of the approximate surface representation. For each image, we also include the noise variance estimate computed from the orginal image and an error statistic computed from the original-reconstruction difference image.

When a user runs the program on an image, the name of the image is typically the only input required by the program. All internal parameters are either fixed or automatically varying based on the noise variance estimate. The user does have the option to change five of the fixed internal parameters and to override the three automatically set thresholds: 1) the maximum allowed RMS fit error $\epsilon_{\max}$, 2) the surface normal compatibility angle threshold θ_t, and 3) the regions test threshold r_t. Eight of the nine images shown here were obtained without any adjustments whatsoever, but more interesting results were obtained by overriding the automatically set thresholds for the computer keyboard range image, which has nonstationary noise. This was necessary because of the stationarity assumption of the current noise variance estimation algorithm, which allows us to describe the image noise with a single number.

A. Interpretation of Intensity Image Results

The entire segmentation algorithm is based only on the knowledge of piecewise-smooth surfaces and digital surfaces. Since intensity images are also digital surfaces, the algorithm can be applied to intensity images for segmentation purposes. It is important to understand that the dimensionality of a digital surface is the same regardless of the meaning of the sensed values at each pixel. And since the difference between range images and intensity images is the interpretation of the sensed values (depth versus light intensity), the difference in the algorithm output lies in how the surface segmentation is interpreted. Intensity image surface primitives are only surface function ap-

proximations to the intensity image data and nothing more. The segmentation results will be useful when intensity image surfaces correspond to physical surfaces in a scene. This is of course equivalent to an implicit assumption that edge detection approaches use: the boundaries of intensity image surfaces correspond to the boundaries of physical surfaces in a scene. However, our image description is much richer than most edge-based image descriptions because not only are guaranteed closed-curve edges of regions detected, but the *approximate value of every single image pixel* is encoded in the polynomial coefficients. If intensity image surface primitives can be reliably extracted, it is possible to apply shape from shading ideas [58] to intensity surface primitives [14].

B. Coffee Cup Range Image (ERIM)

The coffee cup range image is a 128 × 128 8-bit image from an ERIM phase-differencing range sensor [108]. The segmentation results are shown in Fig. 6. The measured noise variance is $\sigma_{\text{img}} = 1.02$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 1.46$. The final segmentation clearly delineates the outside cylindrical surface of the cup, the foot of the cup, the inside cylindrical surface of the cup, the background table surface (which was recognized as a single surface with three subregions despite the nonadjacency of the region visible through the handle and the small hole in the side of the cup), and the cup handle surface (which is represented as two surfaces due to the twisting of the surface from this view). Although this image is easy to segment by many other methods, the subtle difference in surface variations between the foot and the main body of the cup is difficult to detect with an edge detector. Two small meaningless surfaces did arise on the steeply sloped sides of the cup because the laser range sensor has difficulty obtaining good results when most of the laser energy is reflected away from the sensor. Note that although this algorithm knows nothing about *cylinders*, the cylindrical surface of the cup is adequately segmented.

Fig. 7 shows discrete contour lines for the original image (left) and the reconstructed image (right). These contour lines bound regions of constant range. This presentation is needed to adequately appreciate the shape information in the noiseless reconstructed image as compared to the noisy original image. The background appears to be curved due to image parameterization distortions caused by the range sensor's two orthogonal axis mirrors and equal angle increment sampling as discussed in the Appendix. Fig. 8 shows the variations in RMS fit error, region size, and surface fit order as a function of the region growing iteration number for the background surface. Fig. 9 shows the actual polynomial coefficients used in the image reconstruction for the six primary regions: 1) background, 2) cup body, 3) cup interior, 4) top of handle, 5) bottom of handle, and 6) foot of cup. The mean absolute error (e1), the standard deviation (e2), and
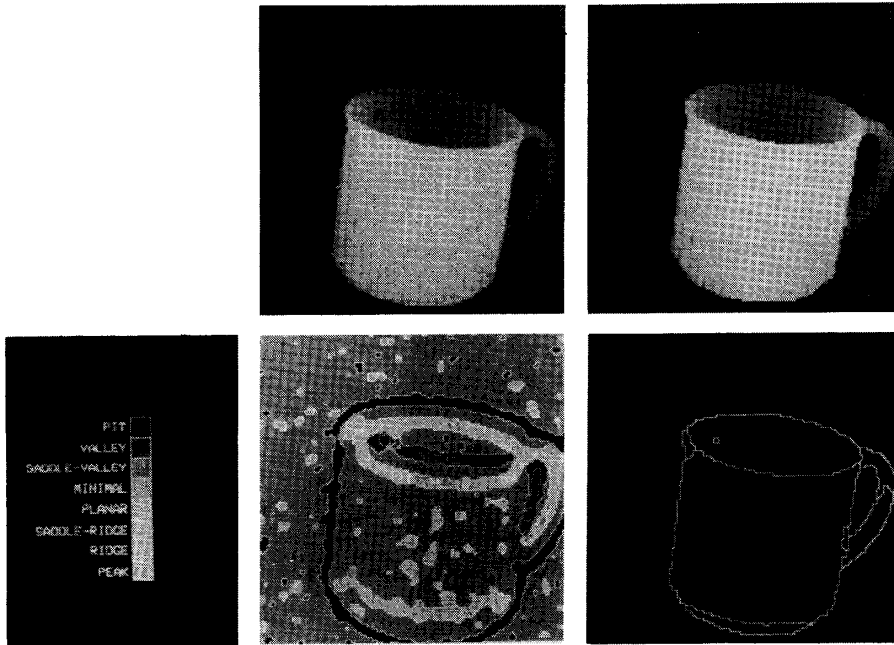
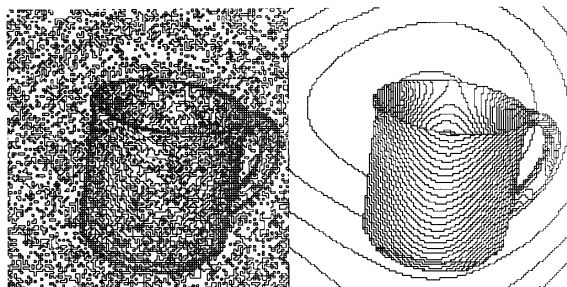Fig. 6. Segmentation results for coffee cup range image.

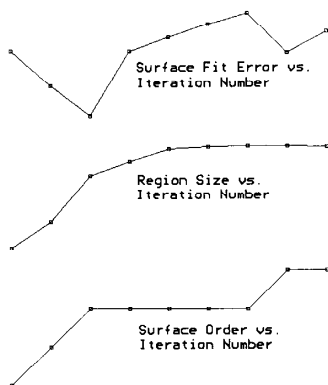Fig. 7. Range contour lines for original and reconstructed images.

Fig. 8. Fit error, region size, and surface order versus iteration number.

Polynomial Graph Surfaces for '/img/range/cofcup2' (128 x 128 Image)
          2   2   2   2   3   3   3   2 2   3   4   4
z=a+bx+cy+dxy+ex +fy +gx y+hxy +ix +jy +kx y+lx y +mxy +nx +oy

Surf# 1 Biquartic Surface
a= 41.1084853 b= 0.141577803 c= 0.2713745965
d= -0.002495014122 e= -0.0007755753671 f= -0.003601368877
g= 1.481712253e-05 h= 2.726318379e-05 i= -2.309573363e-06
j= 8.758485509e-06 k= -1.803061683e-08 l= -8.351990453e-08
m= -8.546141676e-08 n= -5.871664019e-10 o= 1.594833765e-08
0.733465 0.890638 2.81471 (e1,e2,emax) 9321 Pxls in Rgn

Surf# 2 Bicubic Surface
a= 64.72905408 b= 4.004588601 c= -0.6523437561
d= -0.008861504524 e= -0.02260194283 f= 0.008512778086
g= -4.525700906e-05 h= 4.449644712e-05 i= 7.129137939e-06
j=-5.132800061e-05
0.894967 1.12435 4.17196 (e1,e2,emax) 3301 Pxls in Rgn

Surf# 3 Biquartic Surface
a= -163.4152524 b= -5.291701634 c= 35.73599936
d= -0.822060872 e= 0.303405367 f= -0.4267766669
g= 0.000534934843 h= 0.01567119329 i= -0.002583054026
j= -0.003845895577 k= 8.273502856e-06 l= -1.84330636e-05
m= -0.0001033402269 n= 7.022812258e-06 o= 6.765861748e-05
0.862173 1.04452 2.94707 (e1,e2,emax) 846 Pxls in Rgn

Surf# 4 Bicubic Surface
a= -56553.34566 b= 1085.103359 c= -550.9673791
d= -6.935330046 e= -7.005112872 f= -1.772838004
g= 0.02117382282 h= 0.01207493125 i= 0.01546967941 j=0.001442356458
0.693086 0.870873 1.94343 (e1,e2,emax) 85 Pxls in Rgn

Surf# 5 Biquartic Surface
a= 58478.67644 b= -4177.347605 c= 5553.461039
d= -62.08406574 e= 61.15735537 f= -78.3777114
g= 0.02145723219 h= 0.980323446 i= -0.3041497425
j= 0.2363115629 k= 0.001884292974 l= -0.005593080286
m= 0.001788868803 n= 0.0003022405187 o= -0.001835806151
1.57884 1.93 5.13262 (e1,e2,emax) 107 Pxls in Rgn

Surf# 6 Biquadratic Surface
a= 16.5160474 b= 4.368161058 c= 0.9102155505
d= 0.0009597935563 e= -0.03381975184 f= -0.01009001358
1.30028 1.61381 3.97869 (e1,e2,emax) 146 Pxls in Rgn

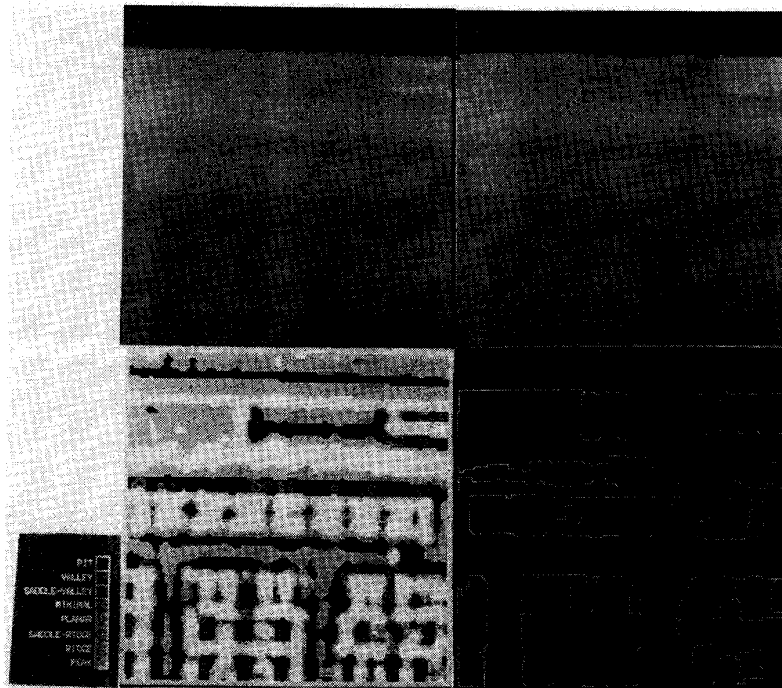Fig. 9. Bivariate polynomial coefficients describing coffee cup surfaces.

Fig. 10. Segmentation results for computer keyboard range image.

the maximum error (emax) are listed for the region as well as the number of pixels in the accepted region (statistical outliers not included).

C. Computer Keyboard Range Image (ERIM)

The computer keyboard range image is a 128 × 128 6-bit image from the ERIM range sensor. The segmentation results are shown in Fig. 10. The measured noise variance is $\sigma_{img} = 1.68$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 1.96$. The surface type label image shows the uneven distribution in the additive noise field. The smooth surface of the keyboard body has very little noise in comparison with the noise on the keys themselves. This results from the specularity of the key surfaces and diffuseness of the keyboard body. This nonstationarity of the noise disobeyed the stationarity assumption of the noise estimation program and the automatically set thresholds did not provide the segmentation quality of other images. Therefore, manually set thresholds were used for the results shown here. Some individual keys have been segmented whereas other keys were grouped together. However, the center of each key is available from the surface type label image if needed. Although it cannot be seen in this presentation of results, each key center is represented as a small isolated pit or valley region surrounded by ridge and peak regions representing the surrounding parts of the key.

D. Ring on Steps (ERIM)

The ring on steps range image is a noisy 128 × 128 8-bit image from the ERIM range sensor. The segmenta-

tion results are shown in Fig. 11. The ring has a rectangular cross section and the step lower part of the steps is cut off at an oblique angle. The measured noise variance is $\sigma_{img} = 2.05$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 3.31$. This image is the noisiest range image in results documented here. The steeply sloped surfaces are much noisier than the other surfaces as occurred in the coffee cup image. Fig. 12 shows the contour lines for the original image (left) and the reconstructed image (right). The noiseless quality of the reconstructed image is quite apparent in this presentation.

E. Auto Part (INRIA)

The original data for the auto part was acquired from the INRIA range sensor (made available courtesy of Prof. T. Henderson of Univ. of Utah and INRIA) and was formatted as a long list of (x, y, z) points. Although the data was easily divided into scan lines, a different number of points occurred on each scan line, and the points were not regularly spaced. This data was converted to 128 × 128 8-bit range image by a separate processing step not documented here. The segmentation results for this auto part range image are shown in Fig. 13. The measured noise variance is $\sigma_{img} = 0.60$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i,j)|) = 1.48$. This 2.5-D segmentation is similar to 3-D segmentations published in [33], [12], [52].

F. Cube with Three Holes

The cube with three holes drilled through it provides an interesting nonconvex combination of flat and cylindrical
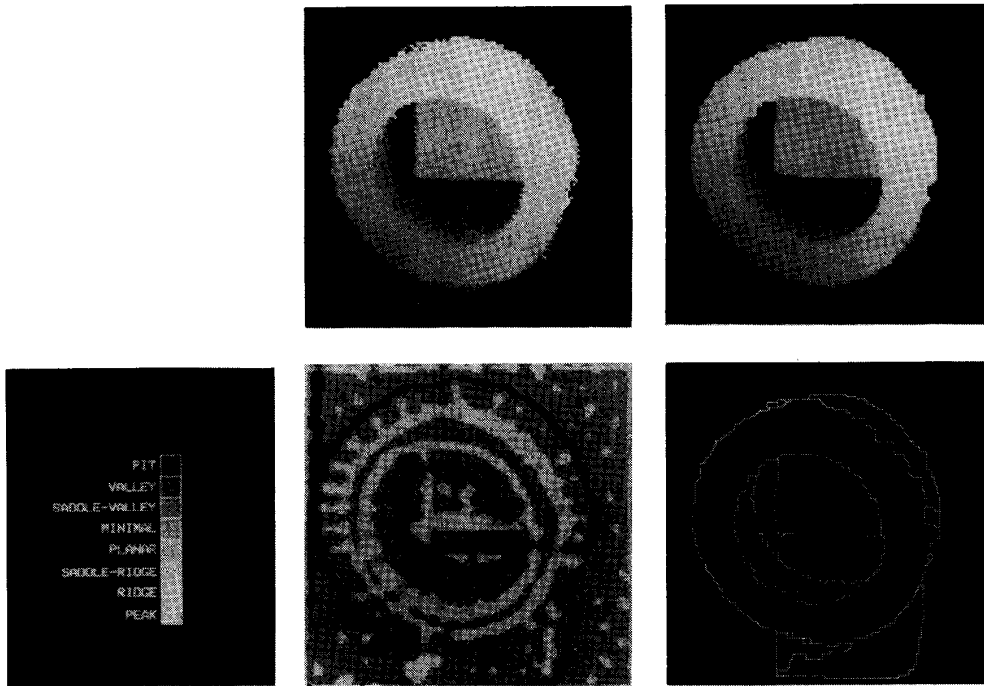
Fig. 11. Segmentation results for ring on steps range image.
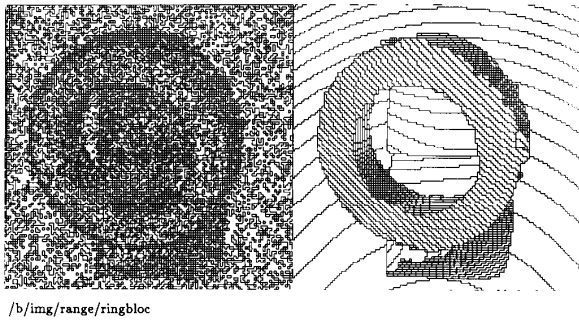

/b/img/range/ringbloc

Fig. 12. Range contour lines for original and reconstructed images.

surfaces. This range image was created using a depth-buffer algorithm on a 3-D solid model created using SDRC/GEOMOD and adding pseudo-Gaussian noise. The segmentation results are shown in Fig. 14. The measured noise variance is $\sigma_{img} = 1.89$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 2.94$. The three linear dihedral edges of the cube have been determined to subpixel precision by intersecting the planar descriptions for the three planes. The results here show the raw segmentation in the region label image.

G. Road Scene Range Image (ERIM)

The road scene range image is a 128 × 128 range image from the ERIM sensor. The segmentation results are shown in Fig. 15. The measured noise variance is $\sigma_{img} =$

1.82, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 0.96$. The edges of the road are clearly delineated in the segmentation results. The false edge crossing the road results from the limited bending capability of the biquartic polynomial within the tolerances specified by the automatic threshold setting mechanisms. This edge can be removed in several ways: 1) the error tolerances can be increased manually, 2) higher order surfaces can be used, or 3) the range data can be precorrected (resampled) to eliminate the geometric distortions produced by equal angle increment sampling in scanning laser radars that use two mirrors rotating around orthogonal axes as discussed in the Appendix.

H. Road Scene Intensity Image

A different road scene is represented in the 128 × 128 8-bit intensity image. The intensity image segmentation results are shown in Fig. 16. The measured noise variance is $\sigma_{img} = 2.27$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 5.48$. The edges of the road are clearly delineated, and the quality of the image reconstruction is quite good. A faster version of the segmentation algorithm might be used for navigation by growing fixed image regions directly in front of the vehicle in both registered range and intensity images. The polynomial surface primitives will grow only over the image regions corresponding to the road. The complementary information in the range and intensity images can be combined to avoid obstacles and plan paths over smooth surfaces.
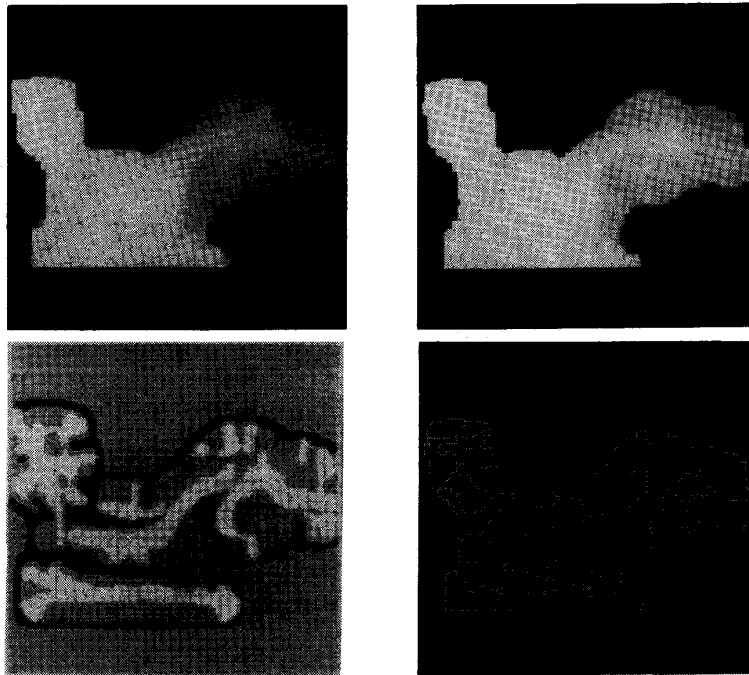
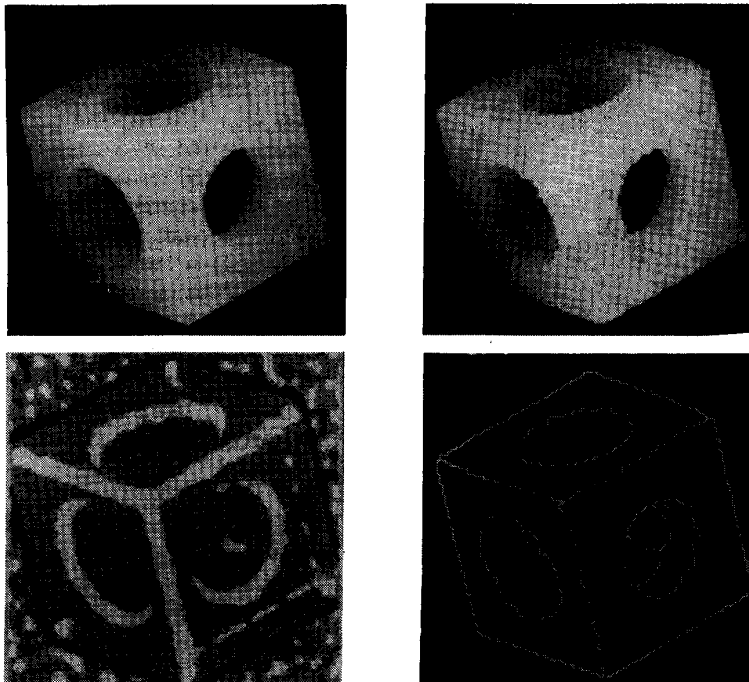Fig. 13. Segmentation results for auto part range image.

Fig. 14. Segmentation results for cube with three holes range image.

I. Space Shuttle Intensity Image

The segmentation results for an image of a space shuttle launch are shown in Fig. 17. The measured noise variance is $\sigma_{img} = 2.71$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 4.32$. The reconstructed image lacks detail whenever the detail in the original image consists of only a few pixels (10 or less) or is only one pixel wide. For example, a small piece of the gantry tower is missing in the reconstructed image. The surface type label image segmentation appears completely incoherent when com-
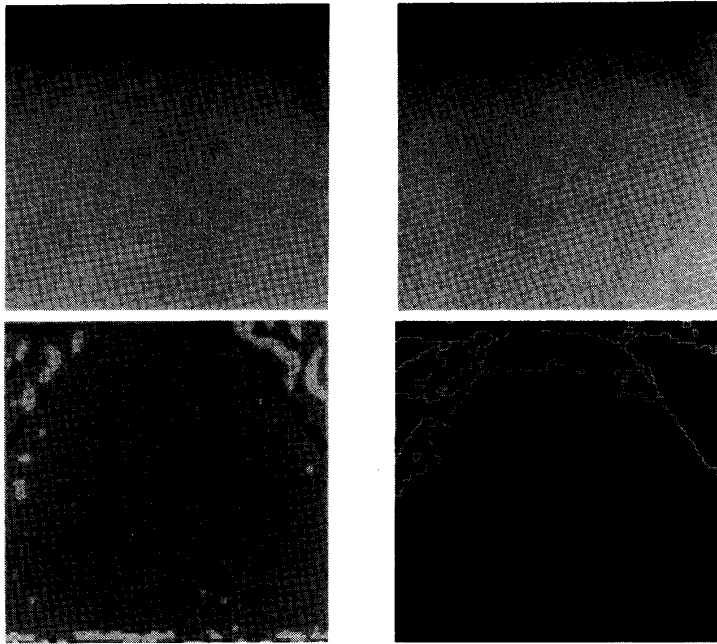
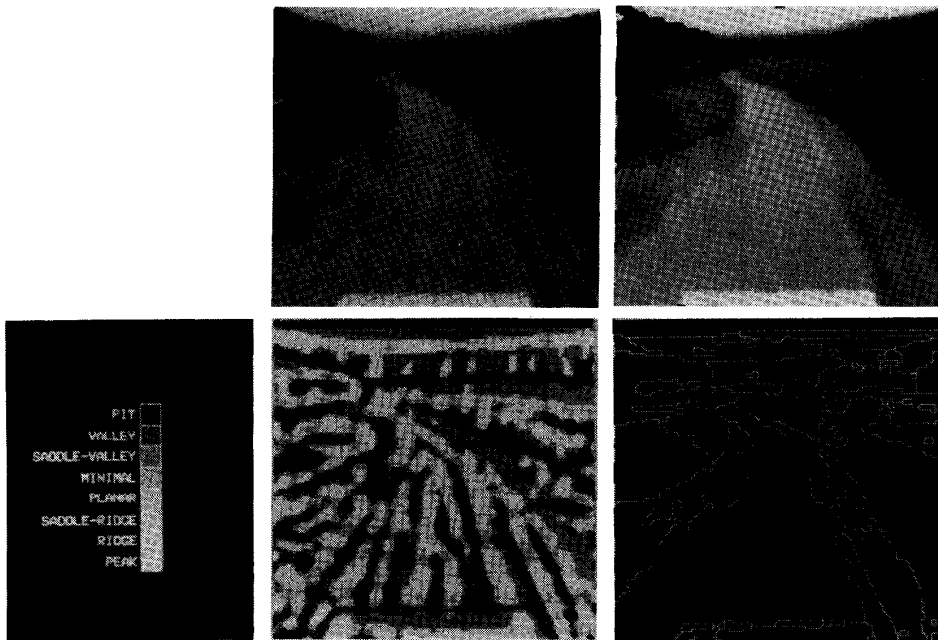Fig. 15. Segmentation results for road scene range image.

Fig. 16. Segmentation results for road scene intensity image.

pared to the original shuttle image. This is unlike most range images where some structure is usually perceivable. However, it still provided enough grouping information to the region growing algorithm to produce the final segmentation. The sky, the smoke clouds, the main tank, and the bright flames are isolated as intensity-image surface primitives.

J. House Scene Intensity Image (Univ. of Mass.)

A house scene, which has been segmented by many other techniques in the literature, was used to test the performance of the surface-based segmentation algorithm. The segmentation results for the 256 × 256 8-bit house scene image are shown in Fig. 18. Owing to the sequen-
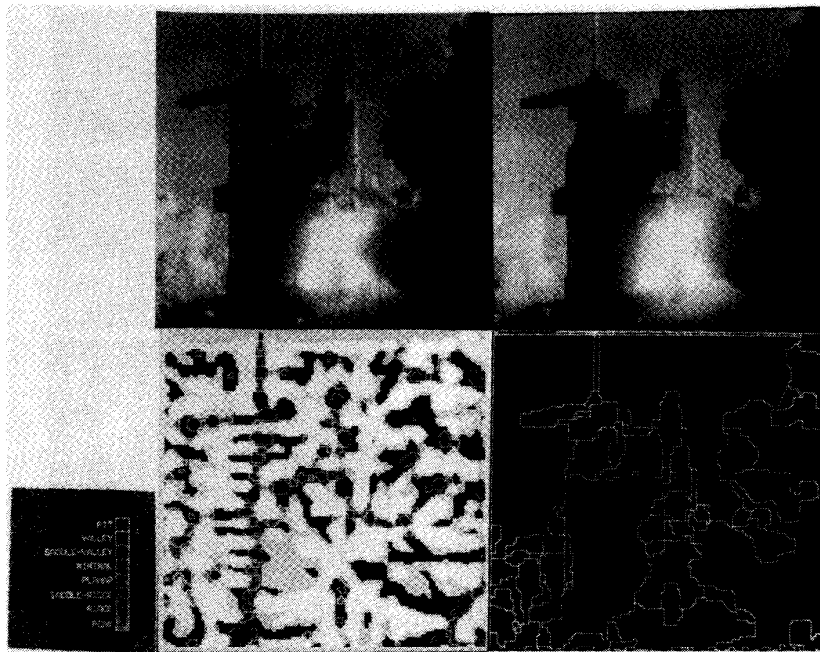
Fig. 17. Segmentation results for space shuttle intensity image.

Fig. 18. Segmentation results for house scene intensity image.

tial nature of the algorithm, segmentation processing time is related to image complexity, and several CPU hours on a VAX 11-780 were needed to compute these results consisting of 136 intensity surface primitives. The measured noise variance is $\sigma_{img} = 3.93$, and the mean absolute deviation between the final reconstructed image and the original image is $E(|e(i, j)|) = 9.44$. The sky, the roof of the house, the lawn, the garage door, the man's pants,

the shutters, the side of the house, the chimney, and the trees are all well segmented. The tree regions are very textured, but are still adequately segmented because of the final processing of coalesced unexplained pixels. Note the smoothness of the tree region in the reconstructed image. The quality of the image reconstruction and the segmentation were obtained with the exact same set of internal parameters used for five of the range images. We know

of no other algorithm that can claim this type of segmentation performance on such a wide variety of images.

XIV. CONCLUSIONS AND FUTURE DIRECTIONS

The experimental results obtained by applying the surface-based segmentation algorithm with a fixed set of input parameters to a large test database of over 40 images, including range and intensity images, indicates that data-driven segmentation of digital surfaces based on a piecewise-smooth surface model, surface curvature sign, and polynomial surface approximations is feasible and provides excellent results. Nine sets of image results are included here to document these claims.

This surface-based approach is very general in many respects. Flat surfaces are described explicitly as being flat, and arbitrary curved surfaces are described as being curved within the context of the same variable-order surface fitting algorithm. Most techniques in the literature need to handle flat and curved (quadric) surfaces as separate special cases. No *a priori* assumptions about surface convexity, surface symmetry, or object shape are used. The final segmentation/reconstruction description is driven by the content of the data, not by expected high-level models as is done in many other approaches. Moreover, the exact same algorithm with the exact same set of parameters is shown to be capable of segmenting range images and intensity images. We believe that any image that can be represented by a piecewise-smooth surface over sufficiently large regions (more than 10–30 pixels) can be segmented well by this algorithm.

The basic sign-of-curvature/iterative variable-order fitting approach is applicable to the segmentation of signals in any number of dimensions, not just scalar functions of two variables. The method is shown to be successful for edge interval segmentation in [8]. If one-pixel wide edges are available, x and y can be parameterized as a function of arc length yielding a 2-vector function of a single variable. Only three sign-of-curvature labels are needed for each 1-D function: concave up, concave down, and flat. In the future, we hope to be able to apply the algorithm to signals representing scalar functions of three variables, such as dynamic scenes and 3-D images from CAT scanners. In that case, 27 sign-of-curvature labels are needed, and approximating functions require many coefficients (20 for a tricubic).

Perception of surfaces plays a key role in image understanding. We have shown experimentally that the segmentation of range images into scene surfaces can be data-driven and need not involve higher level knowledge of objects. The perceptual organization capabilities of the surface-based image segmentation algorithm appear to also be worthwhile capabilities for intensity image segmentation as is shown via experimental results. More research is needed to determine how higher level knowledge should be used in relating intensity-image surface primitives to the real scene surfaces.

Better methods of noise estimation are needed to improve the automatic threshold selection process. For non-stationary noise, it may be necessary to store an estimate of the noise variance for each pixel or region in the image. The noise variance estimates must then be consulted during each iteration. We are also looking into various types of adaptive smoothing, such as in [98], to improve the quality of the partial derivative estimates used to compute surface curvature and check surface normal compatibility. Also, various applications will require different types of surface models. The shape description needs for NC machining may be quite different than those for surface shape matching in 3-D object recognition, and neither application may be able to use the extracted polynomial surface primitives in the current form described here. Thus, the conversion of the shape information into more useful forms for given applications is another key issue that must be addressed.

APPENDIX
EQUAL ANGLE INCREMENT SAMPLING

Formulas for calculating the geometric distortion introduced by equal angle increment sampling for range sensors with two orthogonal axis rotating mirrors and with spherical azimuth/elevation scanning mechanisms are included here. Let z, x, y be 3-D Cartesian coordinates with z representing depth from the x, y focal plane. Let r, θ, ϕ be 3-D orthogonal axis angular coordinates used for range sensors with two orthogonal axis mirrors. Let r, θ, ψ be 3-D spherical coordinates used for azimuth-elevation range sensors.

The transformations from orthogonal axis angular coordinates to Cartesian coordinates are given by the following:

$$x(r, \theta, \phi) = \frac{r \tan \theta}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \tag{36}$$

$$y(r, \theta, \phi) = \frac{r \tan \phi}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \tag{37}$$

$$z(r, \theta, \phi) = \frac{r}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}}. \tag{38}$$

Note the symmetry between the horizontal and vertical angles. The inverse transformations are given by

$$r(x, y, z) = \sqrt{x^2 + y^2 + z^2} \tag{39}$$

$$\theta(x, z) = \tan^{-1}\left(\frac{x}{z}\right) \tag{40}$$

$$\phi(y, z) = \tan^{-1}\left(\frac{y}{z}\right). \tag{41}$$

Note that horizontal angle θ is not a function of the vertical Cartesian coordinate y and that the vertical angle ϕ is not a function of the horizontal Cartesian coordinate x.

The spherical coordinate transformation from (x, y, z) to (r, θ, ϕ) coordinates is given by the following equations where ψ is the elevation angle and θ is the azimuth angle:

$$x(r, \theta, \psi) = r \cos \psi \sin \theta \tag{42}$$

$$y(r, \psi) = r \sin \psi \tag{43}$$

$$z(r, \theta, \psi) = r \cos \psi \cos \theta. \tag{44}$$

The inverse transformations for r and θ are identical to the orthogonal axis case, but the expression for the elevation angle is given by

$$\psi(x, y, z) = \tan^{-1} \left(\frac{y}{\sqrt{x^2 + z^2}} \right). \tag{45}$$

Note that ψ depends also on x in addition to y and z. Hence, the only difference between orthogonal axis angular coordinates and spherical coordinates is in the vertical angles ϕ and ψ.

The "warping" of surfaces in image coordinates by equal angle increment sampling in θ and ϕ or ψ, which was mentioned in the text, can be understood by comparing the depth expression for Cartesian coordinates to the depth expressions for orthogonal axis angular coordinates and spherical coordinates:

$$z(x, y) = \sqrt{\left(r(x, y) \right)^2 - (x^2 + y^2)} \tag{46}$$

$$z(\theta, \phi) = \frac{r(\theta, \phi)}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \tag{47}$$

$$z(\theta, \psi) = r(\theta, \psi) \cos \theta \cos \psi. \tag{48}$$

Flat surfaces in $z(x, y)$ data will appear curved in $z(\theta, \phi)$ data or $z(\theta, \psi)$ data because of the differences in surface parameterization. If given range images from an orthogonal axis coordinate $z(\theta, \phi)$ range sensor or spherical coordinate $z(\theta, \psi)$ range sensor, the Cartesian x and y coordinates can be computed for each angle pair $((\theta, \phi)$ for orthogonal axis angular coordinates or (θ, ψ) for spherical coordinates):

$$x_{\text{ortho}}(\theta, \phi) = z(\theta, \phi) \tan \theta$$

$$x_{\text{spher}}(\theta, \psi) = z(\theta, \psi) \tan \theta \tag{49}$$

$$y_{\text{ortho}}(\theta, \phi) = z(\theta, \phi) \tan \phi$$

$$y_{\text{spher}}(\theta, \psi) = \sqrt{z^2(\theta, \psi) + x_{\text{spher}}^2(\theta, \psi)} \tan \psi. \tag{50}$$

The "difficulty" with these x, y coordinates, from an image processing viewpoint, is that they do not lie on an equally spaced grid of image pixels. If desired, interpolation can be used to resample the surface data to obtain an equally spaced sampled Cartesian orthographic projection $z(x, y)$ range image, but this is not necessary in many cases. Since most of the range images in this paper use a relatively small field of view, the range images can be segmented and approximate surface shape can be reconstructed directly without resampling. Once the appropriate image regions have been segmented, accurate physical surface shape in Cartesian coordinates can be computed (if all range sensor parameters are known) by computing the Cartesian x, y, z coordinates from the angular coordinates at each pixel in the segmented image regions and then fitting new graph surfaces to the Cartesian data.

REFERENCES

[1] G. J. Agin and T. O. Binford, "Computer description of curved objects," in *Proc. 3rd Int. Joint Conf. Artificial Intelligence*, Stanford, CA, Aug. 20-23, 1973, pp. 629-640.

[2] R. L. Anderson and E. E. Houseman, *Tables of Orthogonal Polynomial Values Extended to N = 104*, Iowa State College Agricultural and Mechanic Arts, Ames, IA, Res. Bull. 297, Apr. 1942.

[3] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[4] S. Barnard, "A stochastic approach to stereo vision," in *Proc. 5th Nat. Conf. Artificial Intelligence*, AAAI, Philadelphia, PA, August 11-15, 1986, pp. 676-680.

[5] R. H. Bartels and J. J. Jezioranski, "Least-squares fitting using orthogonal multinomials," *ACM Trans. Math. Software*, vol. 11, no. 3, pp. 201-217, Sept. 1985.

[6] P. R. Beaudet, "Rotationally invariant image operators," in *Proc. 4th Int. Conf. Pattern Recognition*, Kyoto, Japan, Nov. 7-10, 1978, pp. 579-583.

[7] G. Beheim and K. Fritsch, "Range finding using frequency-modulated laser diode," *Appl. Opt.*, vol. 25, no. 9, pp. 1439-1442, May 1986.

[8] P. J. Besl, "Surfaces in early range image understanding," Ph.D. dissertation, Dep. Elec. Eng. Comput. Sci., Univ. Michigan, Ann Arbor, Rep. RSD-TR-10-86, Mar. 1986; see also *Surfaces in Range Image Understanding*. New York: Springer-Verlag, 1988.

[9] P. J. Besl and R. C. Jain, "Three dimensional object recognition," *ACM Comput. Surveys*, vol. 17, no. 1, pp. 75-145, Mar. 1985.

[10] ——, "Invariant surface characteristics for three dimensional object recognition in range images," *Comput. Vision, Graphics, Image Processing*, vol. 33, no. 1, pp. 33-80, Jan. 1986.

[11] P. J. Besl, E. J. Delp, and R. C. Jain, "Automatic visual solder joint inspection," *IEEE J. Robotics Automation*, vol. RA-1, no. 1, pp. 42-56, May 1985.

[12] B. Bhanu, "Representation and shape matching of 3-D objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 3, pp. 340-350, May 1984.

[13] B. Bhanu, S. Lee, C. C. Ho, and T. Henderson, "Range data processing: Representation of surfaces by edges," in *Proc. Int. Pattern Recognition Conf.*, IAPR-IEEE, Oct. 1986, pp. 236-238.

[14] R. M. Bolle and D. B. Cooper, "Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 4, pp. 418-429, July 1984.

[15] R. C. Bolles and M. A. Fischler, "A RANSAC-based approach to model fitting and its application to finding cylinders in range data," in *Proc. 7th Int. Joint Conf. Artificial Intelligence*, Vancouver, B.C., Canada, Aug. 24-28, 1981, pp. 637-643.

[16] R. C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 3-26, Fall 1986.

[17] B. A. Boyter, "Three-dimensional matching using range data," in *Proc. 1st Conf. Artificial Intelligence Applications*, IEEE Comput. Soc., 1984, pp. 211-216.

[18] M. Brady, "Computational approaches to image understanding," *ACM Comput. Surveys*, vol. 14, no. 1, pp. 3-71, Mar. 1982.

[19] M. Brady, J. Ponce, A. Yuille, and H. Asada, "Describing surfaces," *Comput. Vision, Graphics, Image Processing*, vol. 32, pp. 1-28, 1985.

[20] C. Brice and C. Fennema, "Scene analysis using regions," *Artificial Intell.*, vol. 1, pp. 205-226, 1970.

[21] B. Carrihill and R. Hummel, "Experiments with the intensity ratio depth sensor," *Comput. Vision, Graphics, Image Processing*, vol. 32, pp. 337-358, 1985.

[22] D. Chen, "A regression updating approach for detecting multiple curves," in *Proc. 2nd World Conf. Robotics Research*, Scottsdale,

AZ, Aug. 18–21, 1986, Paper RI/SME, MS86-764; also *IEEE Trans. Pattern Anal. Machine Intell.*, to be published.

[23] S. S. Chern, "A proof of the uniqueness of Minkowski's problem for convex surfaces," *Amer. J. Math.*, vol. 79, pp. 949–950, 1957.

[24] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal markovian random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 195–219, Mar. 1987.

[25] E. N. Coleman and R. Jain, "Obtaining shape of textured and specular surfaces using four-source photometry," *Comput. Graphics Image Processing*, vol. 18, no. 4, pp. 309–323, Apr. 1982.

[26] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 25–39, 1983.

[27] C. Dane, "An object-centered three-dimensional model builder," Ph.D. dissertation, Dep. Comput. Inform. Sci., Moore School Elec. Eng., Univ. Pennsylvania, Philadelphia, 1982.

[28] W. W. Daniel, *Applied Nonparametric Statistics.* Boston, MA: Houghton-Mifflin, 1978.

[29] L. S. Davis, "A survey of edge detection techniques," *Comput. Graphics Image Processing*, vol. 4, pp. 248–270, 1975.

[30] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 1, pp. 39–55, Jan. 1987.

[31] S. Dizenzo, "Advances in image segmentation," *Image and Vision Comput.*, vol. 1, no. 4, pp. 196–210, Nov. 1983.

[32] T. G. Fan, G. Medioni, and R. Nevatia, "Description of surfaces from range data using curvature properties," in *Proc. Computer Vision and Pattern Recognition Conf.*, IEEE Comput. Soc., Miami, FL, June 22–26, 1986, pp. 86–91.

[33] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, Fall 1986.

[34] O. D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of range data into planar and quadric patches," in *Proc. 3rd Computer Vision and Pattern Recognition Conf.*, Arlington, VA, 1983, pp. 8–13.

[35] I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture.* UK: Ellis Horwood, Chichester, 1979.

[36] F. P. Ferrie and M. D. Levine, "Piecing together 3D shape of moving objects: An overview," in *Proc. Computer Vision and Pattern Recognition Conf.*, IEEE Comput. Soc., San Francisco, CA, June 9–13, 1985, pp. 574–584.

[37] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, pp. 3–16, 1981.

[38] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[39] B. Gil, A. Mitiche, and J. K. Aggarwal, "Experiments in combining intensity and range edge maps," *Comput. Vision, Graphics, Image Processing*, vol. 21, pp. 395–411, Mar. 1983.

[40] D. Gilbarg and N. Trudinger, *Elliptic Partial Differential Equations of Second Order.* Berlin: Springer-Verlag, 1983.

[41] C. Goad, "Special purpose automatic programming for 3D model-based vision," in *Proc. Image Understanding Workshop*, DARPA, Arlington, VA, June 23, 1983, pp. 94–104.

[42] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins University Press, 1983.

[43] W. E. L. Grimson, "A computer implementation of a theory of human stereo vision," M.I.T. Artificial Intelligence Lab., Cambridge, MA, Memo. 565, 1980.

[44] ——, *From Images to Surfaces.* Cambridge, MA: M.I.T. Press, 1981.

[45] W. E. L. Grimson and T. Pavlidis, "Discontinuity detection for visual surface reconstruction," *Comput. Vision, Graphics, Image Processing*, vol. 30, pp. 316–330, 1985.

[46] E. L. Hall, J. B. K. Tio, C. A. McPherson, and F. A. Sadjadi, "Measuring curved surfaces for robot vision," *Computer*, vol. 15, no. 12, pp. 42–54, Dec. 1982.

[47] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vision, Graphics, Image Processing*, vol. 29, pp. 100–132, 1985.

[48] R. M. Haralick and L. Watson, "A facet model for image data," *Comput. Graphics Image Processing*, vol. 15, pp. 113–129, 1981.

[49] R. M. Haralick, L. T. Watson, and T. J. Laffey, "The topographic primal sketch," *Int. J. Robotics Res.*, vol. 2, no. 1, pp. 50–72, Spring 1983.

[50] M. Hebert and T. Kanade, "The 3-D profile method for object recognition," in *Proc. Computer Vision and Pattern Recognition Conf.*, IEEE Comput. Soc., San Francisco, CA, June 9–13, 1985, pp. 458–463.

[51] M. Hebert and J. Ponce, "A new method for segmenting 3-D scenes into primitives," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982, pp. 836–838.

[52] T. C. Henderson, "Efficient 3-D object representations for industrial vision systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 6, pp. 609–617, Nov. 1983.

[53] T. C. Henderson and B. Bhanu, "Three-port seed method for the extraction of planar faces from range data," in *Proc. Workshop Industrial Applications of Machine Vision*, Research Triangle Park, NC, May 1982, pp. 181–186.

[54] M. Herman, "Generating detailed scene descriptions from range images," in *Proc. Int. Conf. Robotics and Automation*, St. Louis, MO, Mar. 25–28, 1985, pp. 426–431.

[55] B. K. P. Horn, "Extended Gaussian images," *Proc. IEEE*, vol. 72, no. 12, pp. 1656–1678, Dec. 1984.

[56] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intell.*, vol. 17, pp. 141–184, Aug. 1981.

[57] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a directed split-and-merge procedure," in *Proc. 2nd Int. Joint Conf. Pattern Recognition*, 1974, pp. 424–433.

[58] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intell.*, vol. 17, pp. 141–184, Aug. 1981.

[59] S. Inokuchi and R. Nevatia, "Boundary detection in range pictures," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami, FL, Dec. 1–4, 1980, pp. 1031–1035.

[60] S. Inokuchi, T. Nita, F. Matsuday, and Y. Sakurai, "A three-dimensional edge-region operator for range pictures," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, West Germany, Oct. 19–22, 1982, pp. 918–920.

[61] S. Inokuchi, K. Sato, and F. Matsuda, "Range imaging system for 3-D object recognition," in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30–Aug. 2, 1984, pp. 806–808.

[62] R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 5, pp. 608–620, Sept. 1987.

[63] R. Jain, "Dynamic scene analysis," in *Progress in Pattern Recognition*, vol. 2, A. Rosenfeld and L. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1983.

[64] R. A. Jarvis, "A perspective on range finding techniques for computer vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 2, pp. 122–139, Mar. 1983.

[65] T. Kanade, "Survey: Region segmentation: Signal vs. semantics," *Comput. Graphics Image Processing*, vol. 13, pp. 279–297, 1980.

[66] ——, "Recovery of the three-dimensional shape of an object from a single view," *Artificial Intell.*, vol. 17, pp. 409–460, Aug. 1981.

[67] J. R. Kender and E. M. Smith, "Shape from darkness: Deriving surface information from dynamic shadows," in *Proc. 5th Nat. Conf. Artifical Intelligence*, AAAI, Philadelphia, PA, Aug. 11–15, 1986, pp. 664–669.

[68] G. Kinoshita, M. Idesawa, and S. Naomi, "Robotic range sensor with projection of bright ring pattern," *J. Robotic Syst.*, vol. 3, no. 3, pp. 249–257, 1986.

[69] D. T. Kuan and R. J. Drazovich, "Model-based interpretation of range imagery," in *Proc. Nat. Conf. Artificial Intelligence*, Austin, TX, Aug. 6–10, 1984, pp. 210–215.

[70] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[71] R. A. Lewis and A. R. Johnston, "A scanning laser rangefinder for a robotic vehicle," in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, Cambridge, MA, Aug. 22–25, 1977, pp. 762–768.

[72] C. Lin and M. J. Perry, "Shape description using surface triangularization," in *Proc. Workshop Computer Vision: Representation and Control*, IEEE Comput. Soc., Rindge, NH, Aug. 23–25, 1982, pp. 38–43.

[73] D. Marr, *Vision.* New York: Freeman, 1982.

[74] G. Medioni and R. Nevatia, "Description of 3-D surfaces using curvature properties," in *Proc. Image Understanding Workshop*, DARPA, New Orleans, LA, Oct. 3–4, 1984, pp. 291–299.

[75] D. L. Milgrim and C. M. Bjorklund, "Range image processing: Planar surface extraction," in *Proc. 5th Int. Conf. Pattern Recognition*, Miami, FL, Dec. 1–4, 1980, pp. 912–919.

[76] B. Gil, A. Mitiche, and J. K. Aggarwal, "Experiments in combining intensity and range edge maps," *Comput. Vision, Graphics, Image Processing*, vol. 21, pp. 395–411, Mar. 1983.

[77] R. Nevatia and T. O. Binford, "Structured descriptions of complex objects," in *Proc. 3rd Int. Joint Conf. Artificial Intelligence*, Stanford, CA, Aug. 20–23, 1973, pp. 641–647.

[78] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*, 2nd ed. New York: McGraw-Hill, 1979.

[79] M. Oshima and Y. Shirai, "Object recognition using three-dimensional information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 4, pp. 353–361, July 1983.

[80] T. Pavlidis, "Segmentation of pictures and maps through functional approximation," *Comput. Graphics Image Processing*, vol. 1, pp. 360–372, 1972.

[81] F. G. Peet and T. S. Sahota, "Surface curvature as a measure of image texture," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 6, pp. 734–738, Nov. 1985.

[82] T. C. Pong, L. G. Shapiro, L. T. Watson, and R. M. Haralick, "Experiments in segmentation using a facet model region grower," *Comput. Vision, Graphics, Image Processing*, vol. 25, pp. 1–23, 1984.

[83] R. J. Popplestone, C. M. Brown, A. P. Ambler, and G. F. Crawford, "Forming models of plane-and-cylinder faceted bodies from light stripes," in *Proc. 4th Int. Joint Conf. Artificial Intelligence*, Tbilisi, Georgia, USSR, Sept. 1975, pp. 664–668.

[84] M. Potmesil, "Generating models of solid objects by matching 3D surface segments," in *Proc. 8th Int. Joint Conf. Artificial Intelligence*, Karlsruhe, West Germany, Aug. 8–12, 1983, pp. 1089–1093.

[85] J. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1979, pp. 75–149.

[86] G. T. Reid, "Automatic fringe pattern analysis: a review," *Opt. Lasers Eng.*, vol. 7, pp. 37–68, 1986.

[87] W. Richards and D. D. Hoffman, "Codon constraints on closed 2D shapes," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 265–281, 1985.

[88] E. M. Riseman and M. A. Arbib, "Computational techniques in the visual segmentation of static scenes," *Comput. Graphics Image Processing*, vol. 6, pp. 221–276, 1977.

[89] I. Rock, *The Logic of Perception*. Cambridge, MA: M.I.T. Press, 1983.

[90] A. Rosenfeld and L. S. Davis, "Image segmentation and image models," *Proc. IEEE*, vol. 67, no. 5, pp. 764–772, May 1979.

[91] A. Rosenfeld and A. Kak, *Digital Picture Processing*, vols. 1 and 2. New York: Academic, 1982.

[92] I. K. Sethi and S. N. Jayaramamurthy, "Surface classification using characteristic contours," in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30–Aug. 2, 1984, pp. 438–440.

[93] Y. Shirai, "Recognition of polyhedrons with a range finder," *Pattern Recognition*, vol. 4, pp. 243–250, 1972.

[94] Y. Shirai and M. Suwa, "Recognition of polyhedra with a range finder," in *Proc. 2nd Int. Joint Conf. Artificial Intelligence*, London, UK, Aug. 1971, pp. 80–87.

[95] D. R. Smith and T. Kanade, "Autonomous scene description with range imagery," *Comput. Vision, Graphics, Image Processing*, vol. 31, pp. 322–334, 1985.

[96] W. Snyder and G. Bilbro, "Segmentation of three-dimensional images," in *Proc. Int. Conf. Robotics and Automation*, IEEE Comput. Soc., St. Louis, MO, Mar. 25–28, 1985, pp. 396–403.

[97] K. Sugihara, "Range-data analysis guided by junction dictionary," *Artificial Intell.*, vol. 12, pp. 41–69, 1979.

[98] D. Terzopoulos, "Computing visible surface representations," Artificial Intell. Lab., M.I.T., Cambridge, MA, AI Memo 800, Mar. 1985.

[99] W. Tiller, "Rational B-splines for curve and surface representation," *IEEE Comput. Graphics Applications*, vol. 3, no. 6, pp. 61–69, 1983.

[100] F. Tomita and T. Kanade, "A 3D vision system: Generating and matching shape descriptions in range images," in *Proc. Int. Conf. Robotics*, IEEE Comput. Soc., Atlanta, GA, Mar. 13–15, 1984, pp. 186–191.

[101] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: M.I.T. Press, 1979.

[102] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal, "Curvature-based representation of objects from range data," *Image and Vision Comput.*, vol. 4, no. 2, pp. 107–114, May 1986.

[103] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artificial Intell.*, vol. 17, pp. 17–45, Aug. 1981.

[104] A. P. Witkin and J. Tenenbaum, "The role of structure in vision," in *Human and Machine Vision*, Beck et al., Eds. New York: Academic, 1983, pp. 481–543.

[105] R. J. Woodham, "Analysing images of curved surfaces," *Artificial Intell.*, vol. 17, pp. 117–140, Aug. 1981.

[106] J. W. Woods, "Two-dimensional discrete Markov fields," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 232–240, 1972.

[107] S. W. Zucker, "Region growing: Childhood and adolescence," *Comput. Graphics Image Processing*, vol. 5, pp. 382–399, 1976.

[108] D. M. Zuk and M. L. Delleva, "Three-dimensional vision system for the adaptive suspension vehicle," Defense Supply Service, Washington, Final Rep. 170400-3-F, ERIM, DARPA 4468, 1983.
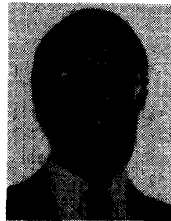
Paul J. Besl (M'81–S'84–M'87) graduated *summa cum laude* in physics from Princeton University, Princeton, NJ, in 1978 and received the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of Michigan, Ann Arbor, in 1981 and 1986, respectively. In 1987, he received a Rackham Distinguished Dissertation Award for his thesis on range image understanding from the University of Michigan.

From 1979 to 1981, he did computer simulations for Bendix Aerospace Systems in Ann Arbor, MI, and from 1981 to 1983, worked on the GEOMOD solid modeling system at Structural Dynamics Research Corp. in Cincinnati, OH. Currently, he is a Research Scientist at General Motors Research Laboratories in Warren, MI, where his primary research interest is computer vision, especially range image analysis and geometric modeling for image understanding.

Dr. Besl is a member of the Association for Computing Machinery, the American Association for Artificial Intelligence, and the Machine Vision Association of the Society of Manufacturing Engineers.

Ramesh C. Jain (M'79–SM'83) received the B.E. degree from Nagpur University in 1969 and the Ph.D. degree from the Indian Institute of Technology, Kharagpur, India, in 1975.

He is a Professor of Electrical Engineering and Computer Science, and Director of the Computer Vision Research Laboratory at the University of Michigan, Ann Arbor. Formerly he worked at General Motors Research Labs, Wayne State University, University of Texas at Austin, University of Hamburg, West Germany, and Indian Institute of Technology, Kharagpur, India. His current research interests are in computer vision, and artificial intelligence. He has been active in dynamic scene analysis, range image understanding, industrial inspection, object recognition, knowledge-based systems, and related areas. He has published research papers addressing several aspects of the above areas. He is a consultant to many companies in the areas of computer vision, artificial intelligence, and computer graphics.

Dr. Jain is a member of the Association for Computing Machinery, the American Association for Artificial Intelligence, the Pattern Recognition Society, the Cognitive Science Society, the Optical Society of America, the Society of Photo-Optical Instrumentation Engineers, and Society of Manufacturing Engineers. He has been involved in organization of several professional conferences and workshops. Currently, he is on the Editorial Boards of *IEEE Expert*, *Machine Vision and Applications*, *Computer Vision Graphics and Image Processing*, the *Bulletin of Approximate Reasoning*, and *Image and Vision Computing*.