

# SEGMENTATION THROUGH SYMBOLIC SURFACE DESCRIPTIONS

*Paul Besl and Ramesh Jain*

**Computer Vision Laboratory  
Electrical Engineering and Computer Science Dept.  
The University of Michigan  
Ann Arbor, MI 48109-1109**

**February 1986**

**CENTER FOR RESEARCH ON INTEGRATED MANUFACTURING**

**Robot Systems Division**

**COLLEGE OF ENGINEERING  
THE UNIVERSITY OF MICHIGAN  
ANN ARBOR, MICHIGAN 48109-1109**



## TABLE OF CONTENTS

1. INTRODUCTION .....	1
2. SURFACE CHARACTERIZATION .....	3
3. PROBLEMS WITH SURFACE-CURVATURE-SIGN .....	5
4. ALGORITHM PHILOSOPHY .....	6
5. SEGMENTATION ALGORITHM DESCRIPTION .....	8
6. APPROXIMATING FUNCTION SELECTION .....	10
7. NOISE ESTIMATION AND ERROR TOLERANCE SPECIFICATION .....	13
8. SEED REGION EXTRACTION .....	14
9. ITERATIVE VARIABLE ORDER SURFACE FITTING .....	15
10. PARALLEL REGION GROWING .....	18
11. REGIONS TEST .....	26
12. TERMINATION RULES .....	26
13. SURFACE ACCEPTANCE AND REJECTION DECISION .....	27
14. ALGORITHM SUMMARY .....	28
15. EXPERIMENTAL RESULTS .....	29
15.1 Output Format .....	29
15.2 Real Range Image Results .....	31
15.2.1 Coffee Cup Range Images (ERIM) .....	31

15.2.2 Computer Keyboard Range Image (ERIM) .....	34
15.2.3 Renault Auto Part (INRIA) .....	34
15.3 Synthetic Range Image Results .....	35
15.3.1 Block with Three Holes .....	36
15.4 Intensity Image Results .....	37
15.4.1 Space Shuttle Intensity Image .....	37
15. CONCLUDING REMARKS .....	38
REFERENCES .....	41

## ABSTRACT

Perception of surfaces plays a key role in three-dimensional object recognition. Recognition of three-dimensional objects from their images is performed by matching *perceived surface descriptions* with stored models of objects. The first step in object recognition is the *segmentation* of an image. If the pixels of a digital surface can be correctly grouped into symbolic surface descriptions that have meaningful interpretations based only on general knowledge of surfaces, this grouping process would provide a fundamental service to higher level matching (recognition) processes. In this paper, an approach for image segmentation in terms of symbolic surface descriptions is discussed. Our approach may be considered *stimulus bound* in that it considers that the ultimate truth is the sensed image data, and hence the output of all processes should conform to the sensed values. Range images offer a simpler domain to study grouping processes for segmentation than intensity images because depth values depend only on geometry, not on reflectance and illumination. Since the problems of range image segmentation are also encountered in the more difficult intensity image segmentation, an effective solution to those problems will be extremely useful to both areas. Our experimental results indicate excellent segmentation performance on range images and support the statement that techniques for range image segmentation may be successfully applied to intensity images.

## 1. INTRODUCTION

Perception of surfaces plays a key role in three-dimensional (3-D) object recognition. Recognition of 3-D objects from images is performed by matching *perceived surface descriptions* with stored models of objects. Recognizing the importance of surfaces in object recognition, computer vision researchers have developed several techniques for obtaining 2.5-dimensional descriptions of images that indicate relative depth. A plethora of *shape from* techniques have been developed. Most of these techniques assume that there is only one surface of interest in an image. Efforts have been made to develop early vision techniques that handle discontinuities in multiple surface images [Grimson and Pavlidis 1985][Terzopolous 1985]. In most realistic applications, images contain several objects. The first step to object recognition in such cases is the segmentation of an image into its *meaningful* parts for matching purposes.

Segmentation of images has been an active area of research for many years (see e.g. surveys by [Haralick and Shapiro 1985] [Riseman and Arbib 1977]). Techniques for edge detection continue to be developed in an effort to find discontinuities in images. Region based approaches try to group points based on the similarity of their properties. It is implicitly assumed that a closed contour or a region represents a surface in an image. Despite efforts of numerous researchers, segmentation remains a difficult unsolved problem.

In the last few years, range images have been attracting increasing attention from vision researchers. Range images (or depth maps) are obtained using a range finder or a *shape from* approach. Since depth information is explicitly

available in range images, it is generally believed that the segmentation and object recognition problems will be easier in range images. If a range image contains several objects or complex surfaces, the segmentation problem, and hence the object recognition task, will still be very complex. We believe that range images offer a good domain to study grouping processes for segmentation. This is due to the fact that most aspects of segmentation in range images are the same as in intensity images, except that in intensity images the intensity at a point is the result of several interrelated factors. It appears (and our results support this) that the techniques for segmentation of range images may be applied to intensity images obtained under uniform illumination.

Concepts and techniques from differential geometry are useful in understanding range images [Brady et al. 1985][Medioni and Nevatia 1984][Besl and Jain 1985a]. 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, many 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. There are global shape similarity theorems for the surfaces of convex objects, and the appropriate mathematics have already been successfully utilized in Extended Gaussian Image (EGI) shape matching schemes [Horn 1984]. Difficulties arise when local descriptors are used to identify the shape of arbitrary non-convex objects from arbitrary-viewpoint range-image projections. Classical mathematics gives little guidance for computational matching methods. Although special feature recognition approaches offer important

advantages for applied computer vision systems (e.g. [Horaud and Bolles 1984]), a successful surface-matching approach based on arbitrary surfaces (e.g. [Potmesil 1983]) could provide considerably more general object recognition capabilities. Our goal is to describe global similarities in surface points (range image pixels) for arbitrary surfaces without making domain-specific assumptions.

If the pixels of a digital surface can be correctly grouped into smooth surface regions that directly correspond to the surfaces of objects in a scene, this grouping process would provide a fundamental service to higher level matching (recognition) processes. This is the basic motivation for our range image segmentation approach.

In this paper, our image segmentation approach of obtaining symbolic descriptions of surfaces in an image is described. Our approach may be considered *stimulus bound* in that it considers that the ultimate truth is the sensed data, and hence the output of all processes should conform to the sensed values. The first step in our algorithm is to classify each point into one of the eight fundamental types based on signs of the Gaussian and mean curvatures. The theory behind this classification is discussed in details elsewhere [Besl and Jain 1986]; only a very brief review is given here.

## 2. SURFACE CHARACTERIZATION

Besl and Jain [1985a] have implemented a surface characterization algorithm that computes surface curvature, critical points, and depth-discontinuity edges as output. Differential geometry states that local surface shape is *uniquely deter-*



*mined* 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. Therefore, visible surfaces in range images have the same mean and Gaussian curvature from any viewpoint under orthographic projection. Also, mean curvature *uniquely determines* graph surfaces if a boundary curve is specified while Gaussian curvature uniquely determines convex regions of a surface. 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 Figure 1. Gaussian and mean curvature can be computed directly from a smoothed range image using window operators that yield least squares estimates of first and second partial derivatives [Beaudet 1978]. 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.

		K		
H	+	0	-	
-	peak	ridge	saddle ridge	
0	(none)	flat	minimal surface	
+	pit	valley	saddle valley	

**Figure 1.**  
Surface Shapes from Mean (H) and Gaussian (K) Curvature Sign

### 3. PROBLEMS WITH SURFACE-CURVATURE-SIGN

It is possible to decide the nature of the surface shape at each point based on the sign of mean and Gaussian curvature (HK-Sign). A simple segmentation approach is to consider 4-connected points of the same surface type as belonging to the same surface in the image. Based on our experiments with this approach, the following three problems were observed:

- (1) **Smoothing:** Preliminary smoothing appears to be required to obtain reasonable differential-geometric quantities from digital surface data. The HK-sign surface labels reflect the geometry of the smoothed surface data and not the original surface data. A final data-driven description, however, should be as precise as possible about the original image and should not contain misleading information about the original digital surface shape. Hence, the raw visible-invariant pixel labeling results obtained via smoothing, derivative estimation, and surface curvature computation must be refined to be useful.
- (2) **Unwanted Connections Caused By Noise:** In the presence of noise, HK-sign surface labels of one surface region tend to connect (in the sense of four-connectedness) with equivalent labels of neighboring, but distinct, surface regions. Therefore, it is not possible, in general, to simply isolate a four-connected region of pixels of a particular HK-sign type and identify that region as a single surface of the appropriate type.
- (3) **Global Symbolic Surface Properties Lacking:** In order to perform surface matching between a range image description and a world model, it is

advantageous to have an explicit, symbolic representation of a surface that possesses the global surface shape properties. A parametric equation for a surface is an example of an explicit symbolic representation.

The seemingly paradoxical motivation to compute a digital surface description for object recognition purposes without knowing anything about specific objects (or even object classes) inhibits us from bringing in model-based techniques to attempt to counteract the above difficulties. If suitable surfaces can be fitted to any possible HK-sign region, such surface fits would provide the desired general global surface information in symbolic form. As shall be shown, it is also possible to remove the effects of smoothing and unwanted connections using a post-surface-characterization, variable-order surface fitting approach to region growing.

#### **4. ALGORITHM PHILOSOPHY**

A *stimulus bound* philosophy is proposed for computer vision algorithms in general and for range image segmentation and object recognition algorithms in particular. It is generally recognized that vision algorithms should function at several different levels using different associated vision modules to process the signal and symbol information at different levels. However, it is often implicitly assumed that each level's vision module accepts input only from the previous, lower level and provides output only to the subsequent, higher level (see Figure 2). This assumption appears to be rooted in human visual models where retinal information is not directly available to the high level cerebral processes. How-

ever, 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, including the sensor itself, should be available to any high-level vision module. In particular, the original image must be available to every vision module in a static vision system. This is shown in Figure 2(a).

It is not that other researchers have serious reservations about using the original image data at higher levels, and many algorithms do just that. But just as many approaches seem to be locked into the linear, vertical processing paradigm (shown in Figure 2(b)). Consider the ACRONYM system [Brooks 1983] for example, which involved several man-years of effort. Only the one lowest level of that multilevel system has any direct interaction with the original data. Moreover, the weakest components of the system all occurred at the lowest levels resulting in a non-robust system despite the major effort devoted to the highest system levels. Without feedback, a chain of interpretive vision modules is only as robust as the weakest module.

Another possible reason for the general lack of feedback in vision systems may be related to the dominance of the edge-detection paradigm in computer vision. In a surface-based approach, a hypothesized range image surface may be verified by computing an error measure over the individual errors made at each pixel in the image. This is easy because the sensed level at a pixel can be directly

compared to the predicted level of the depth at that point. In edge-based approaches, it is difficult to measure edge error directly against the original data because an explicit edge description is not present in the original data. This certainly does not imply that edge-based feedback is not possible, but only that it is less direct than surface-based feedback.

## **5. SEGMENTATION ALGORITHM DESCRIPTION**

It is assumed that (1) an original range image and (2) the eight level HK-sign map are given as input to the surface-based segmentation algorithm. The HK-sign map has a reduced number of levels for each pixel compared to the original image, but retains the same spatial resolution. A connected-component algorithm [Rosenfeld and Kak 1982] may be used to isolate four-connected regions of pixels of a fundamental surface type. Naive attempts to fit surfaces to these connected regions may work occasionally, but fail miserably most of the time, especially in the presence of noise. A more elaborate strategy is called for.

In the surface-based segmentation algorithm, every connected region of pixels of a given type is eroded (contracted) until a sufficiently small number of pixels are obtained in the largest connected subregion. This region serves as the seed region to an iterative region growing algorithm based on fitting variable-order surfaces to the original image data in the seed region and subsequent growth regions. The iterative region growing process is controlled directly by (1) the surface fit error obtained at each iteration, (2) a pre-specified error tolerance, and (3) a regions test, which is a generalization of the one-dimensional runs test

of nonparametric statistics for regression analysis. The iteration continues until the termination criteria are met at which point the computed surface region description is rejected or accepted. The region-growing stage of the segmentation algorithm outputs (A) three different region descriptions for each surface primitive in the form of the best-fit region label buffer (BF-RL-buffer), the first-come, first-served, within-tolerance region label buffer (FW-RL-buffer), and the best-fit-so-far surface list (BFS-list), and (B) a list of graph surface equations and fit errors for each symbolic smooth surface primitive. Each of the three region descriptions may be considered as a single segmentation output, but better results are possible if these three descriptions are combined. These outputs are combined via the region-merging stage of the segmentation algorithm that computes a single region description for each surface primitive (from the three separate descriptions), and then merges surfaces that join smoothly at their common boundary. A best-fit reconstructed image is computed from the best-fit region label buffer and the list of surface equations. This reconstructed image is very similar to the original image. The sequential control flow of the surface-based segmentation algorithm is shown in Figure 3. The data flow is indicated in Figure 4. The central topic of this paper is the iterative region-growing algorithm, which is the most critical element in the surface-based segmentation approach. All other aspects of the segmentation algorithm, including the region description combination algorithm and the region merging algorithm, are discussed in [Besl 1986].

**6. APPROXIMATING FUNCTION SELECTION**

The eight fundamental surface shapes of the HK-sign surface primitives (listed in Figure 1) are relatively simple and exhibit even-order behavior. Arbitrarily complicated smooth surfaces can be decomposed into a disjoint union of such surface primitives. Hence, if these surface primitives can be approximated well, a composite surface description for arbitrary surfaces can be obtained. For dimensionality reduction, the approximating functions should be representable by a relatively small amount of data. For generality, the approximating surfaces must be well-defined over arbitrary connected regions. 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. Interpolating capabilities are also useful. The approximating functions should also be easily differentiable so that differential geometric shape descriptors can be recomputed from them and so that higher level processes may compare surface normals and other differential quantities. Finally, the set of approximating functions should be totally ordered so that each approximant is capable of describing lower order approximants exactly, but cannot approximate higher order functions.

These observations can be summarized in the following list of requirements. Note that general 3-D surface representation capability is not needed because digital surfaces are discrete representations of graph surfaces. A small finite ordered set of approximating surface functions is needed for the region-growing algorithm that

- (1) Can approximate any smooth surface of constant HK-sign well over any type of connected domain,
- (2) Can extrapolate accurately to arbitrary locations outside the data set used for fitting (this capability permits region growing),
- (3) Allows a quick, computationally efficient surface fitting algorithm to be used,
- (4) Can be represented by a small amount of data,
- (5) Allows computation of partial derivatives for each member of the set,
- (6) Can interpolate to locations within the region for fitting.

Basic least-squares-fitted bivariate polynomials satisfy the above requirements. We have found that they perform quite well with a comparatively miniscule amount of mathematical machinery. To maintain generality in the algorithm statement, it is only assumed that there is a set of approximating functions, denoted as  $\mathbf{F}$ , that contains  $|F|$  discrete types of functions that can be ordered in terms of the "shape potential" of each type of surface function relative to the set of fundamental HK-Sign surface primitives.

First-order (planar), second-order (biquadratic), and fourth-order (biquartic) bivariate polynomials are proposed as the set of approximating functions for HK-sign surface primitives. Third-order (bicubic) and fifth-order (biquintic) polynomials add little surface representation capability for this particular problem because of the even-order (roughly symmetric) nature of the HK-sign surface primitives, and are therefore omitted from the set of approximating functions. Bivariate polynomials of order six and higher are avoided for data-fitting purposes owing to potential oscillatory and numerical difficulties. Therefore,  $|F| = 3$  and the set of approximating functions  $\mathbf{F}$  can be written in the form of a single equation:



$$\hat{g}(x,y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 + a_6x^2y + a_7xy^2 + a_8x^3 + a_9y^3 + a_{10}x^3y + a_{11}x^2y^2 + a_{12}xy^3 + a_{13}x^4 + a_{14}y^4.$$

Planar surfaces are obtained by restricting the parameter vector space  $\mathbf{R}^{15}$  to three-dimensional subspace where only  $a_0, a_1, a_2$  may be non-zero; biquadratic surfaces are restricted to a six-dimensional subspace including terms up through  $a_5$ . A least-squares surface fitting algorithm can compute the parameter vector  $\mathbf{a}$  and the RMS fit error  $\epsilon$  directly from the digital surface data in a region quickly and efficiently. Moreover, a least-squares approach allows surface region fits to be updated incrementally during the region growing process as new data points become available.

There are problems with these three bivariate polynomial surfaces. Many HK-sign surface primitives exist that cannot be exactly represented by bivariate polynomials. Perfect hemispheres and cylinder halves provide two good examples of common real-world shapes that fall into this category. Our main concern is that the fit is good enough for *perceptual organization tasks*. Every application will have different types of relevant surfaces. For example, if we are given strong evidence that a peak surface primitive exists in a digital surface, the proposed bivariate polynomials can be used to group all the pixels of that peak surface into a coherent surface region, even if the underlying surface is a hemisphere. The symbolic surface description consisting of a region  $R_i$  and a function  $\hat{g}_i(x,y)$  should be useful information for high-level processes. For example, if a domain-specific algorithm were looking for a sphere, the existence of a round region that is well-approximated by a polynomial with approximately constant surface

curvature would provide extremely strong evidence that a sphere is present. The surface matching algorithm is expected to use the surface region description *and the original data* in that region to verify that the surface is indeed a hemisphere.

## 7. NOISE ESTIMATION AND ERROR TOLERANCE

### SPECIFICATION

Digital surfaces exhibit the property of "surface coherence" when sets of neighboring pixels are spatially consistent with each other (i.e., when a logical connection exists between neighboring pixels in that those pixels are sample points of some relatively smooth surface). The following approach has been implemented, tested, and found to be a useful indication of image quality and surface coherence. A detailed discussion of this approach is given in [Besl 1986]. The goal of this approach is to determine the amount of noise present on the interior of smooth surface regions by attempting to measure the surface "fitability" of the image data. The proposed image quality measure algorithm is stated as follows: First, perform a least-squares planar fit to every 3x3 neighborhood in an image using convolution window operators. Second, if the slope of the planar surface at a pixel is greater than a steep slope threshold, discard the pixel since it appears to be on or very near to a depth discontinuity. Similarly, if the slope of the planar surface is exactly zero, the neighborhood of the pixel is likely to be flat-synthetic, saturated, or completely dark and should be discarded because it is not representative of actual smooth surfaces in an image. Third, if the pixel has not been discarded, compute the planar fit error and add to average error.

Finally, the average error after the entire image has been processed is the quality measure  $\rho$ . This assumes that there are relatively few orientation discontinuity pixels compared to the number of visible surface interior pixels in the image. The equation for the quality measure  $\rho$  may be expressed as

$$\rho \approx \frac{1}{N_p} \sum_{\substack{\text{surface} \\ p \in \text{interior} \\ \text{pixels}}} \epsilon_{1,3}(p)$$

where  $\epsilon_{1,3}(p)$  is the root-mean-square-error (RMSE) of the least-squares (1st order) planar surface in the 3x3 neighborhood of the pixel  $p$  and  $N_p$  is the number of pixels contributing to the sum. Tests were done with 5x5 windows and higher-order surfaces, but 3x3 planar fit measure produced the most useful results. The quality measure  $\rho$  allows us to tie the thresholds involved in the surface characterization and surface segmentation algorithms to the amount of noise in the image in an empirical manner. This technique, though not perfect, provides at least some basis for automated threshold selection for the overall segmentation algorithm.

## 8. SEED REGION EXTRACTION

Most of the problems with the HK-sign map mentioned earlier are avoided by adopting the following strategy. First, the largest connected region of any fundamental HK-sign surface type in the image is isolated. If the region is contracted (eroded) repetitively using a 3x3 region contraction operator, the region will eventually disappear as long as the entire image is not of a single surface type. If the image is single surface type, that case presents no problems. For

each contraction (erosion), there is a largest four-connected sub-region of the original region. If we record the number of pixels in the largest connected sub-region as a function of the number of contractions, a *contraction profile* for the original region is created. If a lower threshold is set for the number of pixels required to be in a seed region, there will be a minimum number of pixels in the contraction profile that is greater than or equal to the lower threshold. The largest connected sub-region with the minimum number of pixels greater than or equal to the lower threshold is assigned to be a *seed region* (or *kernel region*). The lower threshold must be greater than or equal to the minimum number of points required for the simplest surface fit.

Since the fundamental purpose of the contraction profile computation is to find a small enough isolated region that (1) is not inadvertently connected to any adjacent regions, and (2) is far enough inside the boundaries of the actual surface primitive to have escaped the undesired side effects of smoothing and differentiation at surface boundaries, there is an upper limit on the number of necessary contractions. This limit is based on the window size of the smoothing and derivative operators used in the computation of surface curvature. For 11x11 smoothing and 9x9 derivative window operators, a limit of ten contraction iterations has served perfectly thus far in experiments.

## **9. ITERATIVE VARIABLE ORDER SURFACE FITTING**

Each isolated seed region is given as the input to the iterative region growing algorithm based on variable order surface fitting. The basic concepts of this

algorithm are the following. First, it must be decided how well smooth surfaces should fit the data. The general-purpose noise estimation procedure discussed above provides an indication of the RMS error to be expected for smooth surface fitting. This number is used to guide the choice of the maximum RMS error threshold  $\epsilon_{\max}$  for the iterative surface fitting algorithm. We use this same number for the overall average image error threshold used to terminate the entire algorithm.

A plane is always fitted first to the small seed region using an equally-weighted least squares approach. If the seed region belongs to a surface that is not too highly curved, a plane will fit quite well to the original digital surface. If the plane fits the seed region to within the maximum fit error threshold for the RMS error  $\epsilon_{\max}$ , then the seed is allowed to grow. If not, the seed is fitted with the next higher-order surface (the biquadratic in our implementation) 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.

This process may be stated mathematically as follows. Let  $I$  be the rectangular image region over which a hypothetical piecewise smooth function  $z = f(x,y)$  is defined. Let  $\hat{R}_i^{(0)}$  be the seed region provided by the seed extraction algorithm that corresponds to the actual region  $R_i$  in the image. This seed region is assumed to be a subset of the actual HK-sign surface primitive region  $R_i$  that is determined as the seed region is grown:

$$\hat{R}_i^{(0)} \subseteq R_i \subseteq I.$$

The HK-sign map provides subsets of the desired segmentation regions (HK-sign surface primitive regions) through the seed region extraction process. The task at hand now is to convert  $\hat{R}_i^{(0)}$  to a full region description  $\hat{R}_i$  that approximates the desired region description  $R_i$ .

Now, let  $\mathbf{a}_i^{(k)}$  be the parameter vector associated with the functional fit to the depth values in a given region  $\hat{R}_i^{(k)}$  from the set of all connected surface regions in the  $k$ -th iteration. Let  $\mathbf{P}$  be the set of all parameter vectors for all forms of functions of interest. For our selection of the three bivariate polynomials,  $\mathbf{P} = \mathbf{R}^{15}$ . Let  $|F|$  be the number of different types of surface functions to be used (in our case,  $|F| = 3$ ). A particular function type (or fit order) is referred to as  $m_F$  where

$$m_F \in \{1, 2, \dots, |F|\}.$$

The actual fitting function of form  $m_F$  is denoted  $z = \hat{g}(m_F, \mathbf{a}, x, y)$ . The surface fitting process, denoted  $L_{\hat{g}}$ , maps the original image  $g(x, y)$ , a connected region definition  $\hat{R}_i^{(k)}$ , and the fit order  $m_F$  into the range space  $\mathbf{P} \times \mathbf{R}^+$  where  $\mathbf{R}^+$  is the set of possible errors (the set of non-negative real numbers):

$$(\mathbf{a}_i^{(k)}, \epsilon_i^{(k)}) = \mathbf{L}_{\hat{g}}(m_F, \hat{R}_i^{(k)}, g)$$

that has the property that the error metric (function norm)

$$\epsilon_i^{(k)} = \int_{\hat{R}_i^{(k)}} |\hat{g}(m_F, \mathbf{a}_i^{(k)}, x, y) - g(x, y)|$$

is the minimum value attainable for all functions of the form specified by  $m_F$ .

Equally-weighted least-squares surface fitting minimizes the error metric

$$(\epsilon_i^{(k)})^2 = \frac{1}{|\hat{R}_i^{(k)}|} \sum_{(x,y) \in \hat{R}_i^{(k)}} (\hat{g}(m_F, \mathbf{a}_i^{(k)}, x, y) - g(x,y))^2$$

where  $|\hat{R}_i^{(k)}|$  is the number of pixels in the region  $\hat{R}_i^{(k)}$ , or the area of the region. Least-squares polynomial surface fitting is fast and direct and "shape potential" of polynomial surfaces is adequate for HK-sign surface primitives.

Thus, given (1) a seed region  $\hat{R}_i^{(k)}$  (where  $k = 0$  for the original seed region), (2) the original image  $g(x,y)$ , and (3) the simplest interesting function type  $m_F$  (such that  $\epsilon_i^{(k)} < \epsilon_{\max}$ ), the parameter vector  $\mathbf{a}_i^{(k)}$  and the error (RMS)  $\epsilon_i^{(k)}$  are computed for the  $k$ -th iteration on the  $i$ -th HK-surface primitive region from the image. When the error is less than the predetermined maximum fit error threshold  $\epsilon_{\max}$ , then the seed region is allowed to grow. If all three fit orders were tried and the error was never less than the threshold, the seed region is rejected by marking off the pixels in the HK-sign map, and then continuing by looking for the next largest connected region of any surface type. At this point, it is also noted that even though a successful surface fit and region growth occur, the fit order for the next iteration may also be increased depending on the results of the regions test.

## 10. PARALLEL REGION GROWING

After a surface is fitted to a region, the surface description is used to grow the region into a (hopefully) 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. On the  $k$ -th iteration for the seed region corresponding to the actual HK-surface primitive region  $R_i$ , the parallel region growing algorithm accepts the original digital surface  $g(x, y)$ , the approximating function  $\hat{g}(m_F, \mathbf{a}_i^{(k)}, x, y)$  from the surface fitting algorithm, and the surface fit error  $\epsilon_i^{(k)}$  from the fit to the seed region. *It does not use the seed region directly, only the function derived from the seed region at this phase in the growth.* For each pixel  $p \in I$  (the entire image), the two values

$$z_{\hat{g}}(p) = \hat{g}(m_F, \mathbf{a}_i^{(k)}, x(p), y(p)) \quad \text{and} \quad z_g(p) = g(x(p), y(p))$$

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  $w(\epsilon_i^{(k)})$ , then the pixel  $p$  is added to the set of pixels  $C(\hat{R}_i^{(k)}, \hat{g}, \epsilon_i^{(k)})$  that are compatible with the region  $\hat{R}_i^{(k)}$ ; otherwise, it is incompatible and discarded. The result of this process is

$$C(\hat{R}_i^{(k)}, \hat{g}, \epsilon_i^{(k)}) = \left\{ p \in I: |z_{\hat{g}}(p) - z_g(p)| \leq w(\epsilon_i^{(k)}) \right\}.$$

In order to stress all the implied dependencies, we specifically note that the set of compatible pixels  $C(\cdot)$  is dependent on (1) the region  $\hat{R}_i^{(k)}$  and therefore also on the seed region extraction algorithm that created the first region  $\hat{R}_i^{(0)}$ , (2) the function  $\hat{g}$  and therefore on the function type  $m_F$  and all the properties of that function type, (3) the parameter vector  $\mathbf{a}_i^{(k)}$  and error  $\epsilon_i^{(k)}$ , which are in turn



dependent on the type of surface fitting algorithm and again on the original digital surface  $g$  and the seed region, and (4) the error tolerance increase function  $w(\epsilon) > \epsilon$ .

The compatible set of pixels also depends indirectly on all the surface regions previously discovered in the image. This indirect dependence plays a key role in the region growing algorithm so we introduce part of this material now.

For simple surfaces with very precisely defined surface boundaries, the influence of other already determined surface regions is negligible or non-existent, but for noisy digital surfaces, the influence may be more prominent. When a surface region iteration terminates and the surface is accepted by successfully passing the test rules discussed subsequently, the error at each pixel of the grown surface region is stored in the image error buffer to explicitly note the spatial distribution of the approximation errors, and the surface region label for each pixel of the accepted surface region is stored in a region label buffer (RL-buffer) to explicitly note the pixels that the approximating surface fits to within the specified threshold. During the region growing process that forms the compatible pixel list, each pixel is also checked to insure that either the pixel error is less than the error in the error buffer (the best fit error so far) or that the pixel has not yet been officially claimed by another surface (using the FW-RL-buffer). If neither of these conditions are satisfied, then the pixel is not considered to be compatible with the growing surface and is not marked as such. The error buffer approach is used to provide a "soft inhibition" capability for pixels already associated with a given surface primitive as opposed to strictly forbidding the reassignment of

pixels to other surfaces once they have been assigned to one surface. This is done in an attempt to alleviate problems with the order dependent properties of this sequential algorithm. Also, the results of this approach are extremely useful for determining smooth surface merges between surface primitives.

Other growth-inhibiting geometric structures could be used during the construction of the compatible pixel list. For instance, surfaces should not grow over depth discontinuities or orientation discontinuities by the definition of a smooth surface. Thus, if reliable estimates of step edges (depth discontinuities) and roof edges (orientation discontinuities) can be found, the resulting edge pixels could be used as a mask of pixels that are always incompatible for region growth.

The error tolerance increase function  $w(\cdot)$  is a function that increases the value of  $\epsilon_i^{(k)}$  so that, if a pixel really lies on the smooth surface  $\hat{g}(m_F, \mathbf{a}_i^{(k)}, x, y)$  and if  $\epsilon_i^{(k)}$  is an estimate of the standard deviation of the measurement noise, then we want to be reasonably sure that all (or almost all) the pixels that belong to the smooth surface are correctly grouped with that surface. For example, a reasonable form of the error increase function  $w(\cdot)$  is

$$w(\epsilon) = w_0 \epsilon \quad \text{where } w_0 = 3.$$

because approximately 99% all samples of the smooth surface corrupted by normally-distributed measurement noise will lie within this error tolerance. In this simple error increase function, the factor  $w_0$  then controls the "aggressiveness" of each region growing process and therefore, partially controls the speed and accuracy of the iterative surface fitting process. If  $w_0$  is too large, a surface

primitive can grow right over an orientation discontinuity if it is not sharp enough. If  $w_0$  is too small, each iteration may only include a few more pixels making the iteration process very slow and possibly cause actual compatible regions of the digital surface to be missed and given separate labels. Hence, there is a trade-off decision to be made between the two undesirable tendencies. We have found that a factor of 2.8 achieves excellent performance on good-quality images.

Regardless of the factor  $w_0$ , the function  $w(\cdot)$  must take on the following sort of hard limiting modification or no progress can be made when the fit is very good:

$$w(\epsilon) = \max(1.1, w_0\epsilon)$$

This is *the standard error increase function* used for the experimental results. Since the levels of the digital surface  $g(x, y)$  are quantized to one depth level, attempts to find compatible pixels with depth levels that are less than one depth level from the approximating surface are not very successful on average. The threshold 1.1 was chosen arbitrarily to satisfy the need for a threshold slightly greater than one; it has worked quite well in practice. The digital algorithm often requires attention to details not present in continuous domain mathematical analysis.

Several tests were performed with a more aggressive variant of the error increase function listed above. This function is referred to as  $w_A(\cdot)$  and it involves the max-norm (sup-norm) of the functional fit. Let the error  $\epsilon_\infty$

represent the max-norm error between an approximant  $\hat{g}(x,y)$  and the data  $g(x,y)$ :

$$\epsilon_{\infty} = \max_{(x,y) \in R} | \hat{g}(x,y) - g(x,y) |$$

Since the  $\epsilon$  in the other definitions above is a Euclidean error norm, it is written as  $\epsilon_2$  in the expression for the aggressive error increase function:

$$w_A(\epsilon_2, \epsilon_{\infty}) = \max(1.1, w_0\epsilon_2, \epsilon_{\infty})$$

This error increase function provided faster algorithm performance in general because region growth was more aggressive. Moreover, this error increase function is able to guarantee convergence of the iterative algorithm because regions can only get bigger. Slightly better algorithm results were obtained for specific cases, but much worse results were obtained in other cases. Linear, convex combinations of cautious and aggressive error increase functions were also tried so that seeds experience fast growth when the error is small, but established regions slowed their growth as their surface fit errors increased. The use of other types of functions was also explored. Experimental tests showed that the final segmentation results were often sensitive to the choice of the error increase function. The standard function mentioned above is the best choice for most images in our test database.

When the parallel region growing computation is complete, the pixel list  $C(\hat{R}_i^{(k)}, \hat{g}, \epsilon_i^{(k)}) \subseteq I$  is complete. The largest connected region of this list that overlaps the seed region  $\hat{R}_i^{(k)}$  must then be extracted to create the next region  $\hat{R}_i^{(k+1)}$ . This iterative process of region definition via largest, overlapping,

connected region extraction is expressed using the function  $\Lambda(\cdot)$ :

$$\hat{R}_i^{(k+1)} = \Lambda( C(\hat{R}_i^{(k)}, \hat{g}, \epsilon_i^{(k)}), \hat{R}_i^{(k)} ) = \Phi(\hat{R}_i^{(k)})$$

where  $\Phi(\cdot)$  represents all operations required to compute the region  $\hat{R}_i^{(k+1)}$  from the region  $\hat{R}_i^{(k)}$ . The output region  $\hat{R}_i^{(k+1)}$  must have the property that it is the largest connected region in the list of compatible pixels satisfying

$$\hat{R}_i^{(k)} \cap \hat{R}_i^{(k+1)} \neq \phi = \text{Null Set} .$$

This constraint is required because it is possible to get larger connected regions in the compatible pixel list than the connected region corresponding to the seed region. This next region is then considered as a seed region and processed by the surface fitting algorithm

$$(\mathbf{a}_i^{(k+1)}, \epsilon_i^{(k+1)}) = L_{\hat{g}}(m_F, \hat{R}_i^{(k+1)}, g)$$

to obtain a new parameter vector and a new surface fit error. If this region is allowed to grow again  $\epsilon_i^{(k+1)} < \epsilon_{\max}$ , then the compatible pixel list is recomputed

$$C(\hat{R}_i^{(k+1)}, \hat{g}(m_F, \mathbf{a}_i^{(k+1)}, x, y), \epsilon_i^{(k+1)}),$$

the largest connected overlapping region of  $C(\cdot)$  is extracted and so on until the termination criteria are met.

We have not mentioned the details of what happens when the surface fitting process has yielded a fit error that is not acceptable (i.e., not below the maximum error threshold  $\epsilon_i^{(k)} \geq \epsilon_{\max}$ ). When this occurs, the algorithm immediately increments the surface type  $m_F$  to the next most general, flexible surface in the ordered, finite approximating function set and reattempts a surface fit of the

higher order surface type if there are enough pixels in the connected region being fitted. If  $m_F$  reaches the maximum value  $|F|$ , the number of approximating functions in the set, and the surface still does not fit the data to within the maximum error threshold, or if there are not enough pixels in the region to attempt a higher order fit, then the iterative surface fitting algorithm attempts to accept the surface even though the ideal termination criteria have not been met. As an example, if  $\epsilon_{\max} < \epsilon_i^{(k)} < 1.5\epsilon_{\max}$ , then the surface region is accepted because it is close enough to be acceptable. It has been found experimentally that this extra 50% margin of acceptance allows the surface fitting iteration to yield usable results that might not otherwise be obtained.

It must be noted that this parallel region growing approach is entirely equivalent to a sequential spiraling region growing approach *until the last iteration*. At the last iteration, the processing of the compatible pixel list becomes an important feature of the segmentation algorithm as described in the section on surface acceptance and rejection decisions. In the coffee cup image, for example, the flat background visible through the handle of the cup is correctly assigned to larger background surface without high-level knowledge, only a surface consistency criterion. This is an excellent example of the data-driven pixel grouping capabilities of this segmentation algorithm.

By performing the parallel region growth at each iteration, it is possible to observe the compatibility of other image regions during each iteration. On a sequential machine, the parallel approach requires more computation, but the simplicity of the algorithm also has advantages. On a parallel machine, the time

required to compute the compatible pixel list depends on the number of pixels that can be processed by each independent processor. A separate connected component processing step is required by the parallel approach, but it is assumed that this can be accomplished on special purpose hardware very quickly.

## **11. REGIONS TEST**

A fit error test is used to decide if region growth for a particular fit order is allowed. However, it is possible for a lower order function to fit a higher order function well (within the error threshold) over a region even though the lower order fit is not appropriate. If it were possible to detect that a higher order function is present in the data before the error tolerance exceeded the error threshold, the region growth process could proceed more quickly and accurately than otherwise. Indeed, it is possible to detect the presence of a higher order function in the data (without having to allow the fit error to increase up to the given error threshold) by analyzing the distribution of the sign of the fit errors (residuals) at each individual pixel of the fit. We used a generalized version of nonparametric statistics' runs test to improve the rate of convergence. This test is discussed in detail in [Besl 1986].

## **12. TERMINATION RULES**

The termination criteria for the iterative surface fitting algorithm are formulated in terms of the following quantities:

- (1) The maximum error threshold:  $\epsilon_{\max}$

- (2) The total number of pixels in the compatible pixel list:  $|C(\hat{R}_i^{(k)}, \hat{g}, \epsilon_i^{(k)})|$
- (3) The total number of pixels in the four-connected region of interest:  $|\hat{R}_i^{(k)}|$
- (4) The current mode of fitting:  $m_F$

These quantities are used at each iteration to decide whether to continue fitting and growing or to stop.

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

- (1) RULE 1: IF  $\epsilon_i^{(k)} > \epsilon_{\max}$  AND  $m_F \geq |F|$ , THEN Stop!
- (2) RULE 2: IF  $|\hat{R}_i^{(k)}| \approx |\hat{R}_i^{(j)}|$  for any  $j < k$ , THEN Stop!
- (3) RULE 3: IF  $|\hat{R}_i^{(k)}| = |C(\hat{R}_i^{(k)}, \hat{g}, \epsilon_i^{(k)})|$ , THEN Stop!
- (4) RULE 4: AT LEAST TWO (2) Iterations are required for a given surface fit order  $m_F$  before the algorithm is allowed to stop due to Rules 2 or 3. (This is a metarule, a rule about other rules.)

These rules state all the essential concepts involved in terminating the surface fitting iteration. Of course, there is a maximum limit on the number of possible iterations to prevent extremely long iterations and guard against the possibility of infinite loops. 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 seven.

### 13. 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% greater than  $\epsilon_{\max}$ . 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 HK-sign map 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 HK-sign map 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 must update several other data structures besides the writable copy of the HK-sign map. The interaction between different processes and data structures are shown in Figures 3 and 4, and is described in detail in [Besl 1986].

#### **14. ALGORITHM SUMMARY**

In summary, the iterative region-growing algorithm based on a variable-order approximating surface function accepts (a) the HK-sign map and (b) the original image as input and produces (1) three separate surface region descriptions as well as (2) a list of surface equations and surface fit errors as output. Two reconstruction images are also created that allow the user to visually evaluate the quality of the surface approximations. Segmentation into meaningful surface primitives is not yet complete because it is necessary to integrate the three

separate region descriptions into one consistent region description for each surface primitive. After the region description combination process, surface primitives that join smoothly at their shared boundaries are merged together to create the final data-driven surface region description. These operations are described in more detail in [Besl 1986].

## 15. EXPERIMENTAL RESULTS

We applied our algorithm to more than thirty test images. In this section, the segmentation algorithm's performance on real and synthetic range images and an intensity image is discussed. All input images are 128 x 128 pixels, 8 bits per pixel. All real range images were acquired by the Environmental Research Institute of Michigan's (ERIM) phase-difference laser range finder except for the Renault Auto Part Image, which was created by converting a range data xyz-file from INRIA (courtesy of Prof. T. Henderson) into a range image using a special purpose program written by us.

### 15.1. Output Format

For each input image, the following set of images are displayed in the following order:

- (1) Original Image (Gray Scale)
- (2) Best-Fit Reconstructed Image (Gray Scale)
- (3) HK-Sign Map (Gray Scale)
- (4) Final Region Label Buffer (Gray Scale)
- (5) Final Segmentation Plot Overlaid on Original Image (Gray Scale)
- (6) HK-Sign Map Segmentation Plot (Binary)
- (7) Best-Fit Segmentation Plot (Binary)
- (8) Final Labeled Segmentation Plot (Binary)

(9) Edge Description Plot (Binary)

The original image is the only variable input to the standard algorithm. All input thresholds were fixed for the main body of results, as discussed previously. Such results are referred to as *standard results*. Occasionally, significantly better results were obtainable using different thresholds. These exceptional cases are mentioned explicitly and are referred to as *modified threshold* results.

The HK-Sign Map results demonstrate the intermediate surface characterization output and the input to the iterative region-growing algorithm. The best-fit reconstructed image shows the visual quality of the approximate surface representation. The best-fit segmentation plot shows the underlying segmentation in the best-fit region label buffer of the best-fit reconstructed image. The final segmentation plots show the quality of the final segmentation output alone and overlaid on the original image. The edge description plots show the edge interpretations obtained by applying the simplified sign of curvature method to edges. This set of images graphically describes the information available from the algorithm.

We emphasize that the experimental results shown below were obtained using a *fixed set of input thresholds* (except for those results noted explicitly). Several of the fixed numerical inputs required by the algorithm are noted.

- (1) Pre-Smoothing Operator Window Size for Surface Characterization: (11 x 11)
- (2) Derivative Operator Window Size for Surface Characterization: (9 x 9)
- (3) Curvature Smoothing Operator Window Size: (7 x 7)
- (4) Zero Curvature Thresholds: 0.015 (Mean) and 0.006 (Gaussian).
- (5) Regions Test Threshold: 2.0%

- (6) Error Threshold  $\epsilon_{\max}$  : 4.2 levels on 8-bit scale
- (7) Error Increase Factor: 2.8
- (8) Acceptability Zone: 50%

This set of inputs summarizes several potentially variable algorithm inputs. Any other inputs, thresholds, or constants required by the algorithm that are not mentioned were assigned to their default values mentioned previously in the text.

## 15.2. Real Range Image Results

For all images, two figures are presented. The first figure contains the original and best-fit reconstructed images, the HK-sign (surface curvature sign) map, and the final region label buffer in one photograph, and the final segmentation plot overlaid on the original image in the second photograph. The second figure consists of a set of four plots: the initial segmentation in the HK-sign map, the best-fit region label buffer segmentation, the final region segmentation, and the boundary edge descriptions.

### 15.2.1. Coffee Cup Range Images (ERIM)

The coffee cup range image from the ERIM range sensor is a very high quality image. The original range image, the best-fit reconstruction image, and the final segmentation overlay (using a modified threshold) are shown in Figure 5. Note the excellent likeness of the reconstructed image showing that the surface primitives interpreted by the algorithm do accurately represent the image data. In Figure 6, four separate images are represented. Figure 6(a) shows the initial segmentation obtained by computing the signs of the mean and Gaussian curva-

ture at every pixel and drawing boundaries around connected regions of surface curvature sign. Figure 6(b) represents the segmentation in the best-fit region label buffer (BF-RL-buffer). After the region combination and region merging processes have operated on the region growing algorithm output, the final (labeled) region segmentations are obtained, which is shown in Figure 6(c). Figures 6(b),(c),(d) show a slight problem in that the cup handle is divided into two separate surfaces. The fixed set of thresholds used for all test images represent a compromise. The better result shown in the final overlaid segmentation plot of Figure 5 is a modified threshold result. Figure 6(d) display the functional edge descriptions of the edge intervals of the boundaries of the regions in the final region segmentations. Note in Figure 6(d) that straight edges are identified as straight and curved edges are identified as curved.

The overlaid segmentation plot in Figure 5 was obtained with a slightly different set of thresholds showing that almost perfect segmentation results are possible if one is allowed to adjust the thresholds of the algorithm. In this case, the fixed maximum fit-error threshold of 4.2 was too high for the high-quality coffee cup range image, and better results were obtained by reducing the threshold to 1.7.

The final segmentations clearly delineate the outside cylindrical surface of the cup, the background table surface (recognized in two parts despite the topological separation of the small patch visible through the handle in Figure 6(c)), the inside cylindrical surface of the cup, and the cup handle surface, which is represented either as one symbolic primitive or two depending upon the input

thresholds (compare segmentation plots of Figure 5 and Figure 6(c)). Although the hole in the cup in the first image is not big enough to merit its own symbolic surface region description, it is clearly visible in the error image. Figure 7 shows the thresholded error image overlaid on the original range image. This output image type is included only for this view of the coffee cup. Note how clearly the hole in the coffee cup body stands out. Figure 8 displays a reconstruction image obtained by applying the surface equations to the final region segmentation. Hence, this is slightly different than the best-fit reconstruction and the FW reconstruction.

Remember that this algorithm knows nothing about *cylinders*, and even less about coffee cups, yet all surfaces are meaningful to one who understands the semantic content of the image. Except for the small hole in the side of the cup, this range image is the epitome of a coherent digital surface. As expected, excellent results are obtained with the exception of a region merge error on the cup handle for the fixed set of thresholds.

Note the straightness of the edges bounding the outside cylindrical surface region in the final segmentation output in Figure 6(d) as compared to the straightness in the original image. The edges in the original image are not very straight owing to noise effects from the range finder at steeply sloped surfaces. The net straightness of the edge description is due to the sign-of-curvature method combined with region refinement techniques that use 3x3 window mappings.

The quality measure  $\rho$  is 0.84 for the coffee cup image where the handle is clearly visible. This value is higher than those for synthetic images without noise and lower than those for the synthetic images with  $\sigma = 2.3$  pseudo-Gaussian additive white noise.

### **15.2.2. Computer Keyboard Range Image (ERIM)**

A similar set of experimental results are presented for two range images of a computer keyboard in Figures 9 and 10. The gray scale images (original and reconstructed) in Figure 9 may appear washed out due to the low contrast of this range image. Note the intermixing to the keyboard surfaces in the best-fit region label buffer as shown in Figure 10(b). This intermingling (or overlapping) of the surface primitive descriptions is strong evidence for a smooth surface merge. Figure 10(c) shows that the surfaces are correctly merged in the final region segmentation. Despite the low contrast of the original image, excellent results have been obtained using the fixed set of thresholds. Hence, local differences do not need to be large for differential geometry based features to provide good segmentation.

The quality measure for the keyboard image is 1.35. The actual key surfaces are combined together into single regions due to the large standard fit error tolerance (4.2). Lower thresholds can be used to detect individual key surfaces, but the segmentation becomes more ragged.

### **15.2.3. Renault Auto Part (INRIA)**

The original data for the auto part is formatted as a list of (x,y,z) triplets. Although the data is easily divided into scan lines, a different number of pixels

occurred on each scan line, and the pixels were not always regularly spaced. This data was converted to a 64x64 range image, which was then expanded to a 128x128 range image, and smoothed to create the input image used here. Although the best-fit reconstruction image in Figure 11 and the best-fit region label segmentation image in Figure 12(b) are quite good, the final region segmentation is not very meaningful due to the high maximum fit-error threshold of 4.2. Note that the image quality measure is 0.43, which means that the image data is not very noisy. By reducing the fit-error threshold to 1.7, the modified threshold results shown in the overlaid plot image of Figure 11 were obtained.

### **15.3. Synthetic Range Image Results**

Although synthetic images usually lack the realism of actual data and do not provide test results that represent algorithm performance on real data, we have found that synthetic range images with added noise are quite realistic and do adequately test many algorithm capabilities. The range image synthesis problem is much simpler than the intensity image synthesis problem because it depends only upon geometry. The SDRC/GEOMOD solid modeler was used to create several different (non-convex) object models on Apollo workstations. The object models were stored in ASCII files and ported over to the VAX/UNIX system where a depth-buffer (z-buffer) display program was used to convert the object models to range images. The use of synthetic range images was very useful for this research and will be even more critical to object recognition research since arbitrary views of 3-D objects can be obtained much more easily using the



depth-buffer algorithm than could be obtained otherwise.

### 15.3.1. Block with Three Holes

The block with three holes drilled through it provides an interesting non-convex combination of flat and cylindrical surfaces. Figures 13 and 15 show the block with two different levels of added noise: 2.3 and 9.0 respectively. The corresponding image quality measures are 1.82 and 5.93; the values are smaller than the added noise due to the fact that the noisy images were rescaled to fit into 8 bits. If we use the empirically determined, fitted equation  $\epsilon_{\max} = 1.1 + 1.1\rho$  to guide our selection of a maximum fit-error threshold, we would use the  $\epsilon_{\max}$  thresholds of 3.1 and 7.6. It is usually not a serious problem to use a threshold larger than that predicted by the simple equation. The problems with the coffee cup handle and auto part are representative of what can happen, and are not regarded as serious errors. Because 3.1 is less than the standard fixed 4.2 threshold, we expect the results shown in Figure 13 to be fairly good, and they are (see Figures 13 and 14). However, 7.6 is significantly greater than the 4.2 threshold. When the maximum fit-error threshold is not large enough, higher order surfaces are often invoked when not needed. The shape potential of the higher order surfaces is able to respond to the *noise in the data* rather than the *structure* in the data. Figure 16 shows the failure of the algorithm on the noisier block. However, by increasing the  $\epsilon_{\max}$  threshold to 8.0, the results shown in the overlaid segmentation plot image in Figure 15 are obtained. These results are quite good considering the noise level in the image. A key point

here is that the image quality measure can be used to predict the success or failure of the algorithm for a fixed threshold and suggest a more appropriate threshold.

#### **15.4. INTENSITY IMAGE RESULTS**

The entire segmentation algorithm is based only on the knowledge of surfaces. Since intensity images are also digital surfaces (just as range images are digital surfaces), the algorithm can be applied to intensity images for symbolic-surface-primitive segmentation purposes. And just as the only difference between range images and intensity images is in the meaning of the values at each pixel (depth vs. light intensity), the difference in the algorithm output lies in how the surface segmentation is interpreted. That is, an intensity smooth-surface-primitive does not necessarily correspond to the 3-D surface primitives of objects in a scene although it may; intensity image surface primitives have an entirely different meaning than range image primitives. However, it appears that the algorithm may be very useful for intensity image segmentation purposes as long as this fundamental difference is kept in mind.

##### **15.4.1. Space Shuttle Intensity Image**

The original shuttle image, the reconstructed image, and the overlaid segmentation plot image are shown in Figure 17. The reconstructed image lacks detail whenever the detail in the original image consists of only a few pixels (10 or less). For example, the windows on the cockpit of the shuttle only occupy about three pixels and are therefore missed in the image reconstruction. The

surface-curvature sign segmentation in Figure 18(a) appears to be completely meaningless when compared to the shuttle image. This is quite unlike most range images where some semblance of structure is usually perceivable. However, it provided enough significant grouping information to the region growing algorithm to create the best-fit region label buffer segmentation shown in Figure 18(b). The final segmentation is displayed in Figures 18(c) and 17. Considering that the algorithm was developed for range image analysis based on surface concepts, the segmentation for the shuttle image is quite good. This is expected because the quality measure is only 2.29. The sky and many smoke clouds are isolated as intensity-image smooth-surface-symbolic primitives. We emphasize that the exact same fixed set of thresholds were used here as were used for the range images. This is an extremely strong testimonial to the soundness of the digital surface approach when the same algorithm with the same set of thresholds can segment the coffee cup image, the keyboard image, the block with holes image (low-noise), and the space shuttle intensity image without a single modification.

## **16. CONCLUDING REMARKS**

The experimental results obtained by applying the surface-based segmentation algorithm with a fixed set of input thresholds to a large test database of over thirty images, including real and synthetic range images and intensity images, indicates that the surface-based approach to segmentation and early image understanding is very promising. Six sets of image results are included here.

The approach is very general in that flat surfaces are described explicitly as being flat and arbitrary curved surfaces are described as being curved. No *a priori* assumptions about surface convexity, surface symmetry, or object shape are used. The symbolic 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 thresholds is shown to be useful for range images and intensity images.

Future research is needed to determine an effective matching algorithm and modeling scheme that will enable object recognition given the output from the segmentation algorithm and the original image. Future research is also necessary to refine the segmentation algorithm. The current algorithm is sequential and, in certain cases, the order dependency of the region processing (large regions to small regions) adversely affects final region descriptions despite attempts to avoid it by the use of the image error buffer and region label buffers. If a proper parallel processing approach can be developed, there would be no such order dependency. In addition, the region merging stage may require more research. For intensity image segmentation, it will be especially important to develop clean and flexible ways to incorporate specific domain knowledge so that the final region descriptions are representative of physical scene surfaces.

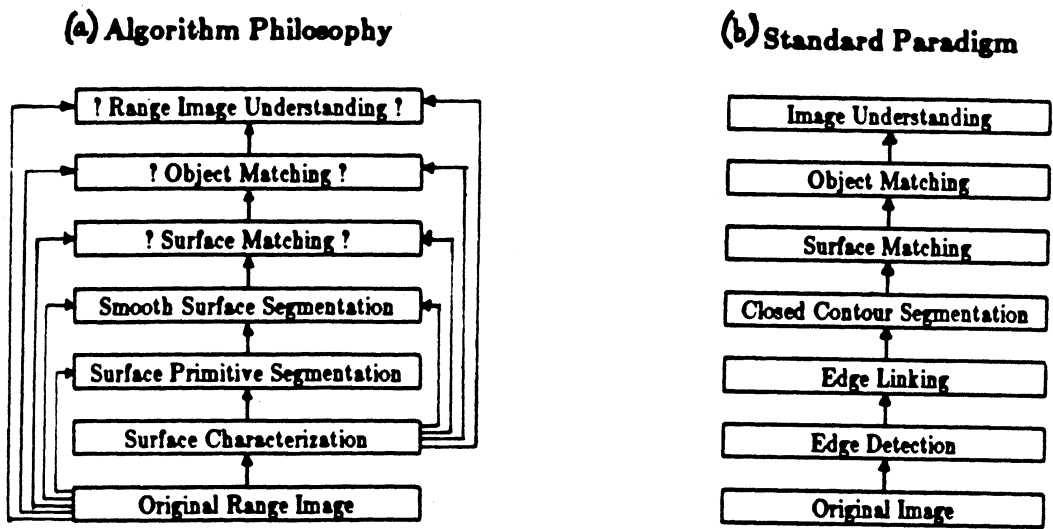
Perception of surfaces plays a key role in image understanding. We have shown that the segmentation of range images into scene surfaces can be data-driven and need not involve higher level knowledge of objects. The perceptual organization capabilities of the surface-based range image segmentation algorithm

may 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 smooth-surface primitives to the real scene surfaces.

## REFERENCES

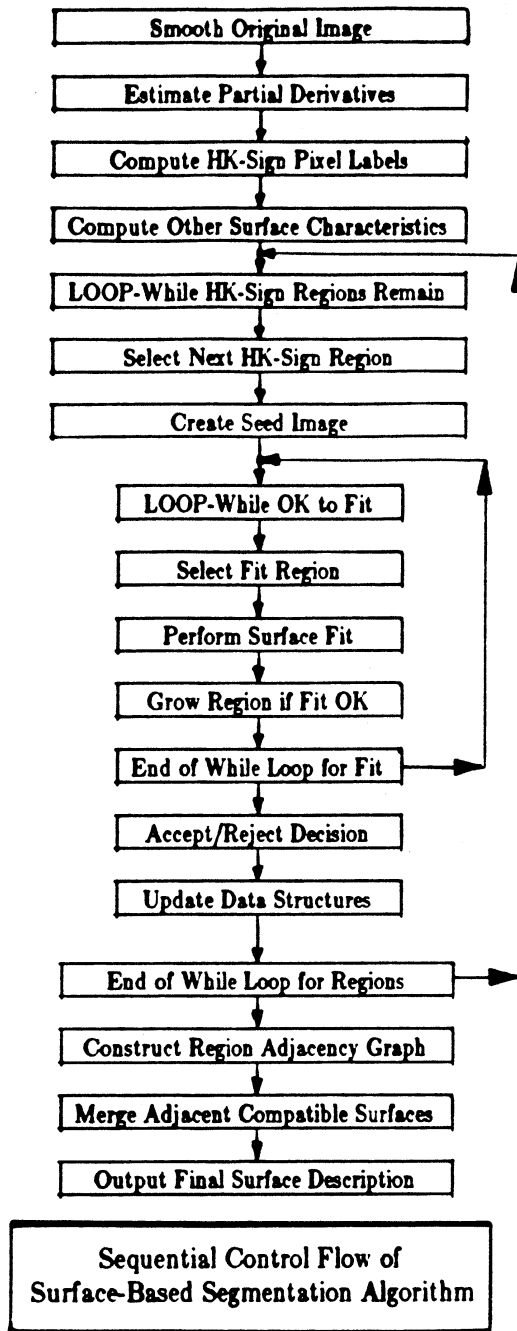
- [1] BEAUDET, P.R. 1978. Rotationally invariant image operators. In *Proceedings of 4th International Conference Pattern Recognition* (Kyoto, Japan, Nov. 7-10). pp. 579-583.
- [2] BESL, P. 1986. *Surfaces in Early Range Image Understanding*, Ph.D. Dissertation, Electr. Eng. Comp. Sci. Dept., University of Michigan, Ann Arbor, MI (Feb.).
- [3] BESL, P., AND JAIN, R. 1985a. Intrinsic and extrinsic surface characteristics. In *Proceedings of Computer Vision and Pattern Recognition Conference* (San Francisco, Calif., June 9-13). IEEE-CS, New York, pp. 226-233.
- [4] BESL, P., AND JAIN, R. 1985b. Three-dimensional object recognition. *Computer Surveys* **17**, 1 (March), 75-145.
- [5] BESL, P., AND JAIN, R. 1986 Invariant Surface Characteristics for 3-D Object Recognition in Depth Maps, *Computer Vision, Graphics, and Image Processing*. in Press.
- [6] BRADY, M., PONCE, J., YUILLE, A., AND ASADA, H. 1985. Describing surfaces. In *Proceedings of 2nd International Symposium on Robotics Research*, Hanafusa H. and Inoue H. (Eds.), MIT Press, Cambridge, Mass.
- [7] BROOKS, R.A. 1983. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-5, 2 (Mar.), 140-149.
- [8] GRIMSON, W.E.L., AND PAVLIDIS, T. 1985. Discontinuity detection for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing* **30**, 316-330.
- [9] HARALICK, R.M., AND SHAPIRO, L.G. 1985. Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing* **29**, 100-132.
- [10] HORAUD, P., AND BOLLES, R.C. 1984. 3DPO's strategy for matching three-dimensional objects in range data. In *Proceedings of the International Conference Robotics*. (Atlanta, Ga., Mar. 13-15). IEEE-CS, New York, pp. 78-85.
- [11] HORN, B.K.P. 1984. Extended Gaussian images. *Proc. IEEE* **72**, 12 (Dec.) 1656-1678.
- [12] MEDIONI, G., AND NEVATIA, R. 1984. Description of 3-D surfaces using curvature properties. In *Proceedings of the Image Understanding Workshop* (New Orleans, La., Oct. 3-4). DARPA, pp. 291-299.

- [13] POTMESIL, M. 1983. Generating models of solid objects by matching 3D surface segments. In *Proceedings of 8th International Joint Conference on Artificial Intelligence* (Karlsruhe, West Germany, Aug. 8-12). pp. 1089-1093.
- [14] ROSENFELD, A., AND KAK, A. 1982. *Digital Picture Processing*, 2nd Ed., Academic Press, New York.
- [15] RISEMAN, E.M., AND ARBIB, M.A. 1977. Computational Techniques in the Visual Segmentation of Static Scenes. *Computer Graphics and Image Processing*, **6**, 221-276.
- [16] TERZOPOLOUS, D. 1985. Computing visible surface representations. AI Memo No. 800, MIT Artif. Intell. Laboratory, Cambridge, Mass. (March)



**Figure 2.** Stimulus-Bound Approach vs. Conventional Approach





**Figure 3 . Sequential Control Flowchart**

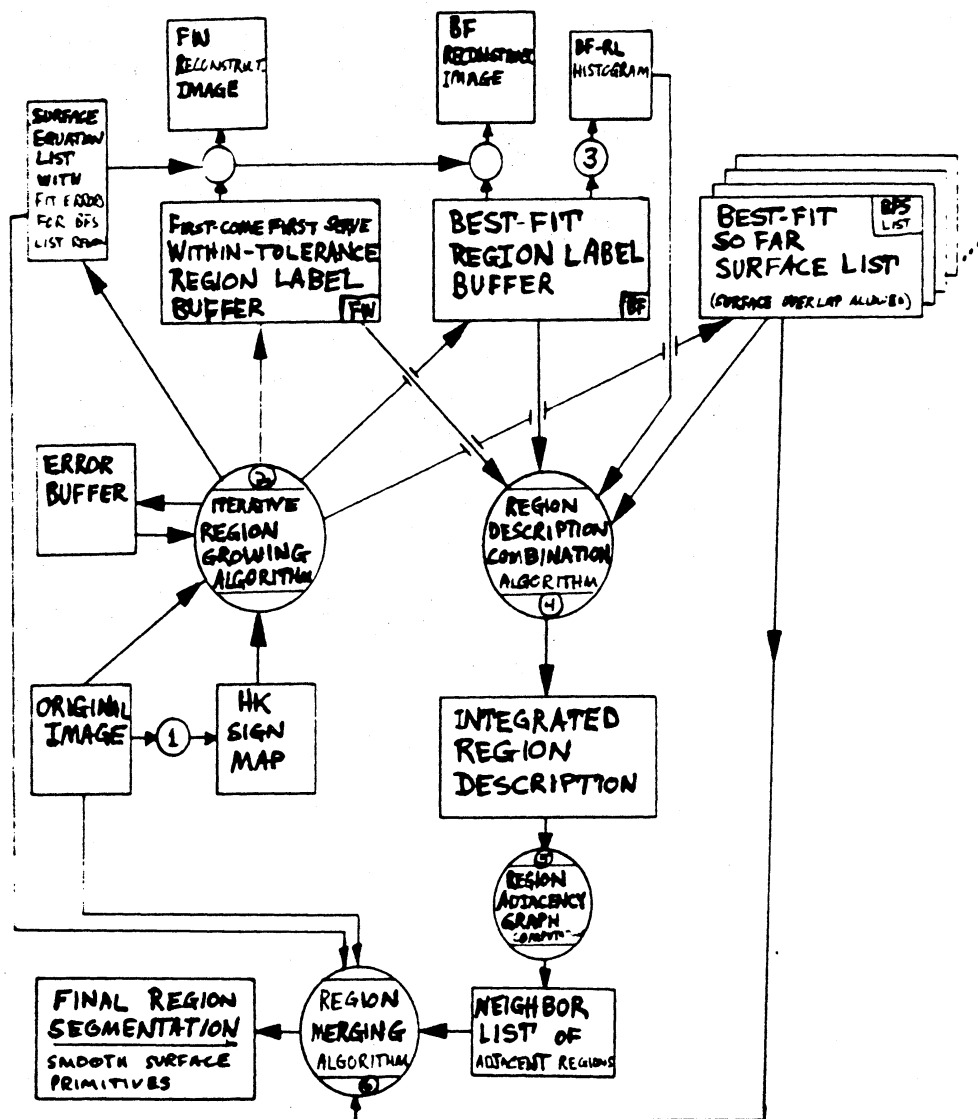
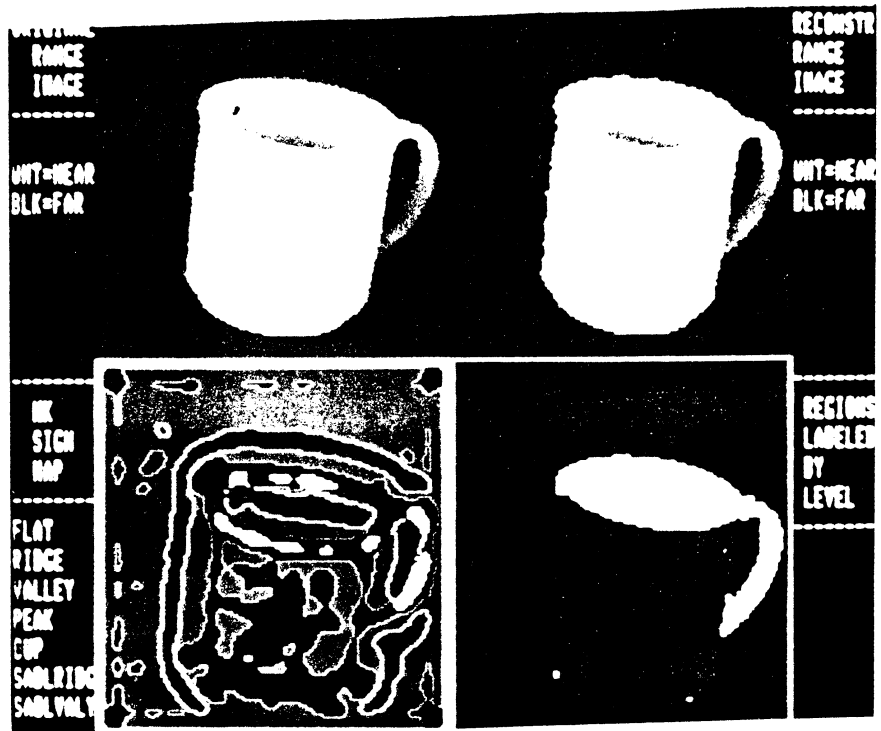


Figure 4. Dataflow Diagram for Segmentation Algorithm



(a) Standard Results for Coffee Cup

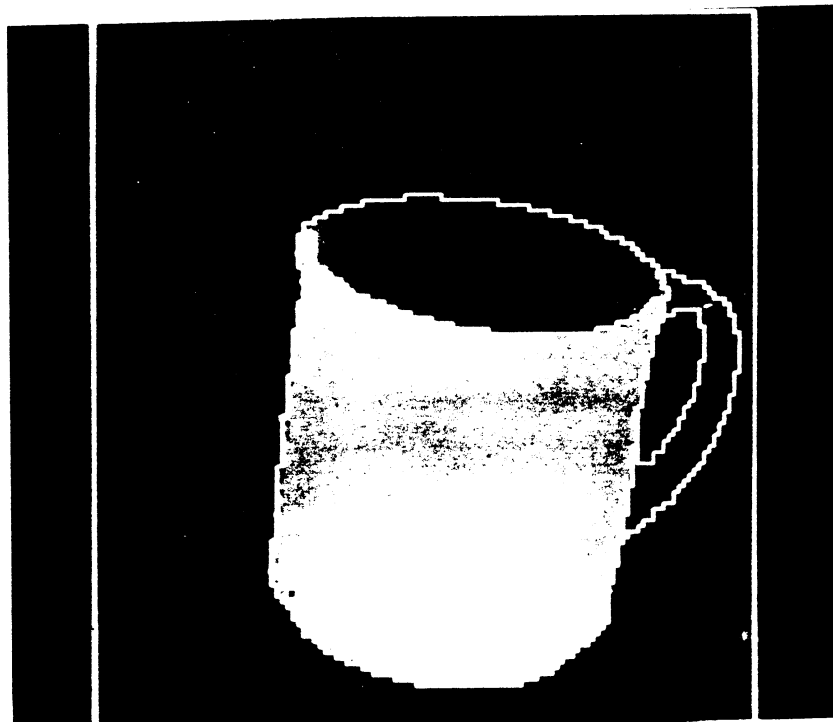
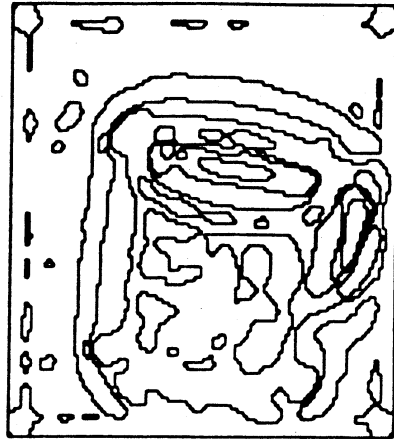
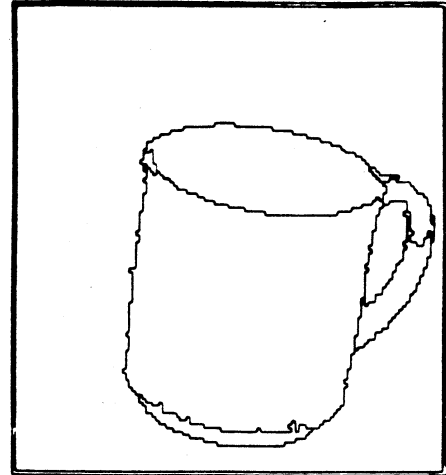


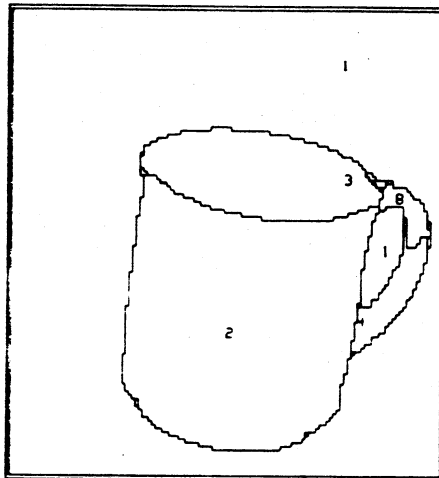
Figure 5. Modified Threshold Results for Coffee Cup



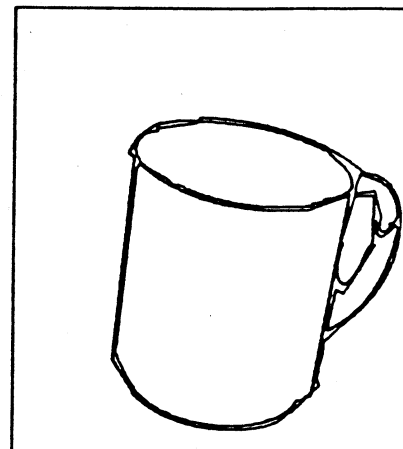
(a) Surface-Curvature-Sign Segmentation



(b) Best-Fit Region Label Buffer Segmentation

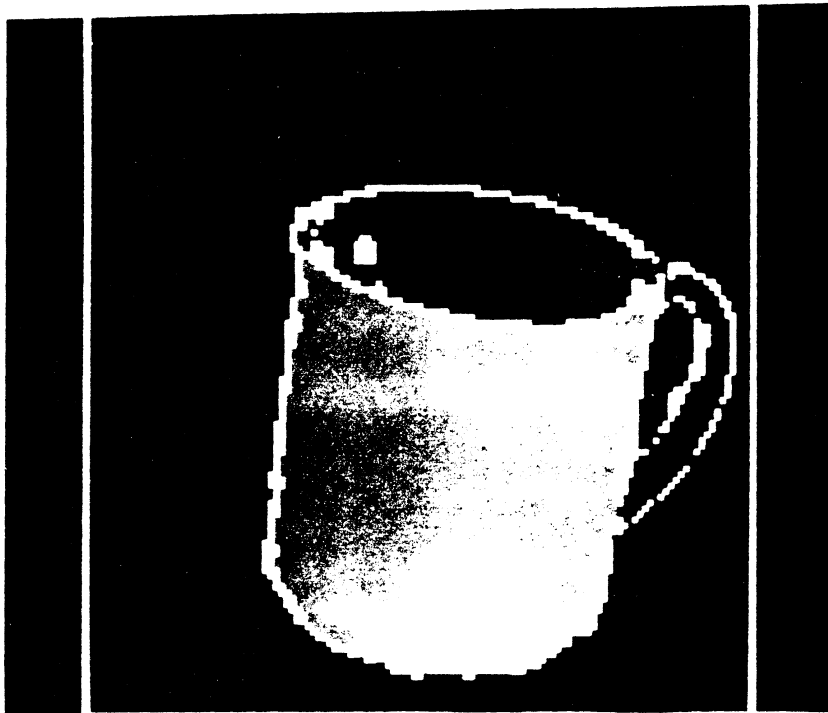


(c) Final Region Segmentation

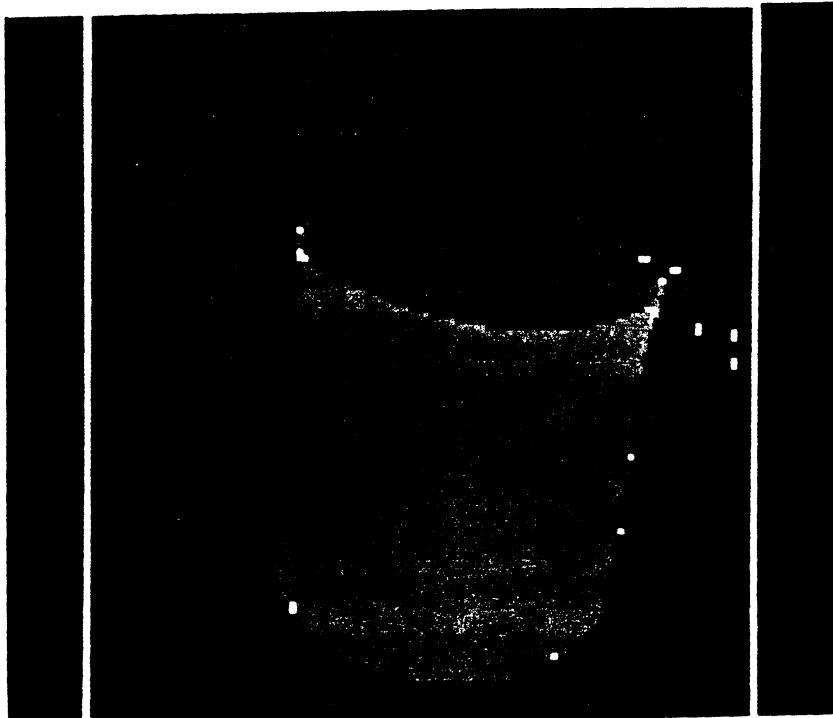


(d) Edge Descriptions of Final Regions

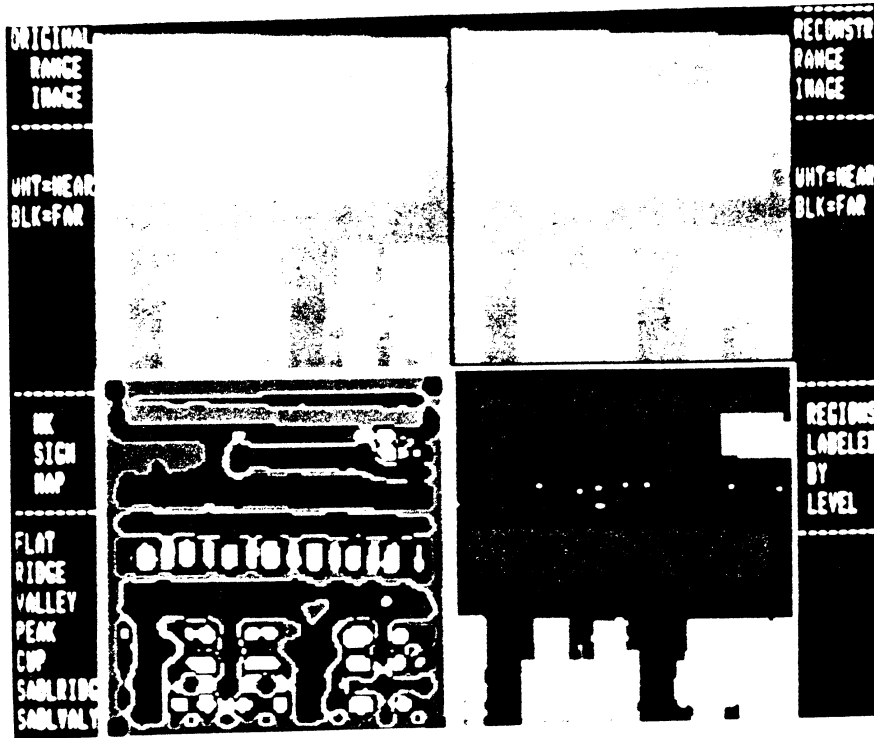
**Figure 6.** Standard Results for Coffee Cup



**Figure 7.** Error Threshold Image Overlaid on Original Image



**Figure 8.** Final Region Label Buffer Reconstruction Image



(a) Standard Results for Keyboard

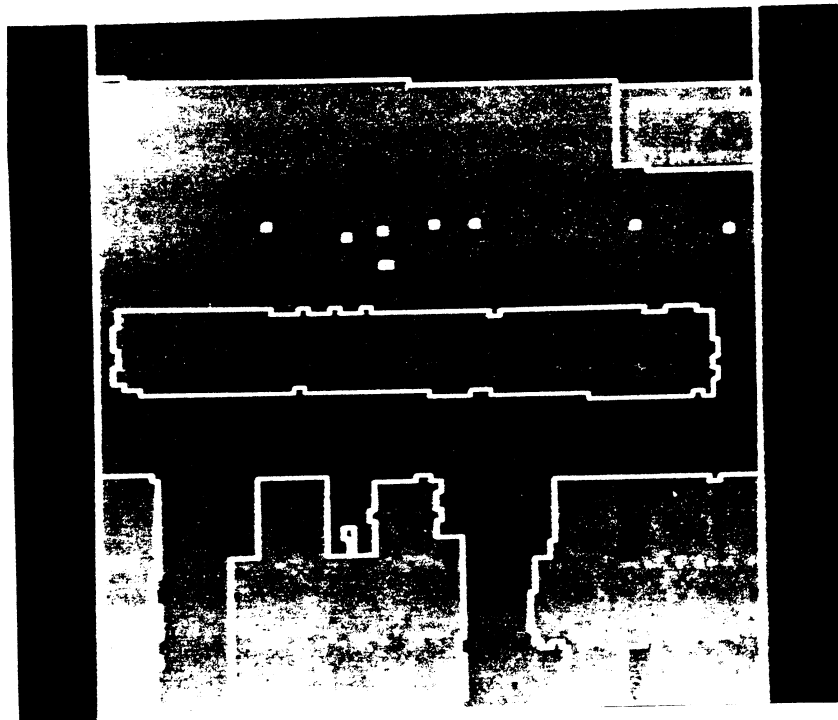
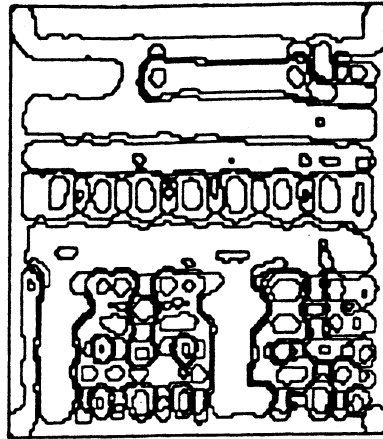
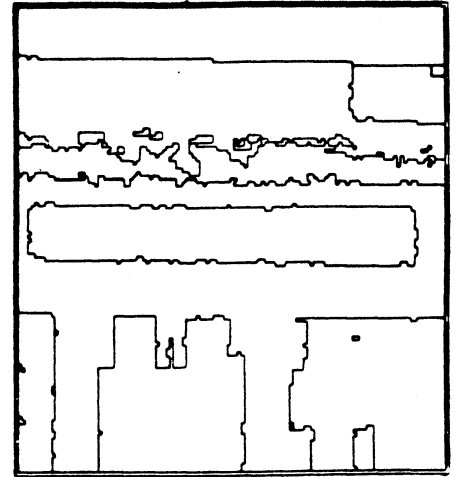


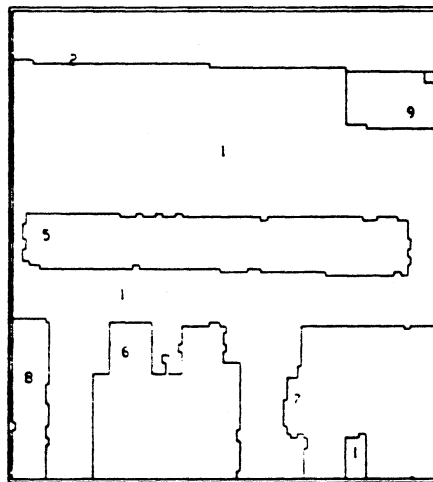
Figure 9. Standard Results for Keyboard



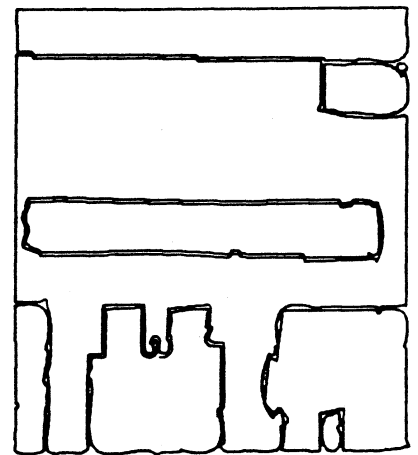
(a) Surface-Curvature-Sign Segmentation



(b) Best-Fit Region Label Buffer Segmentation

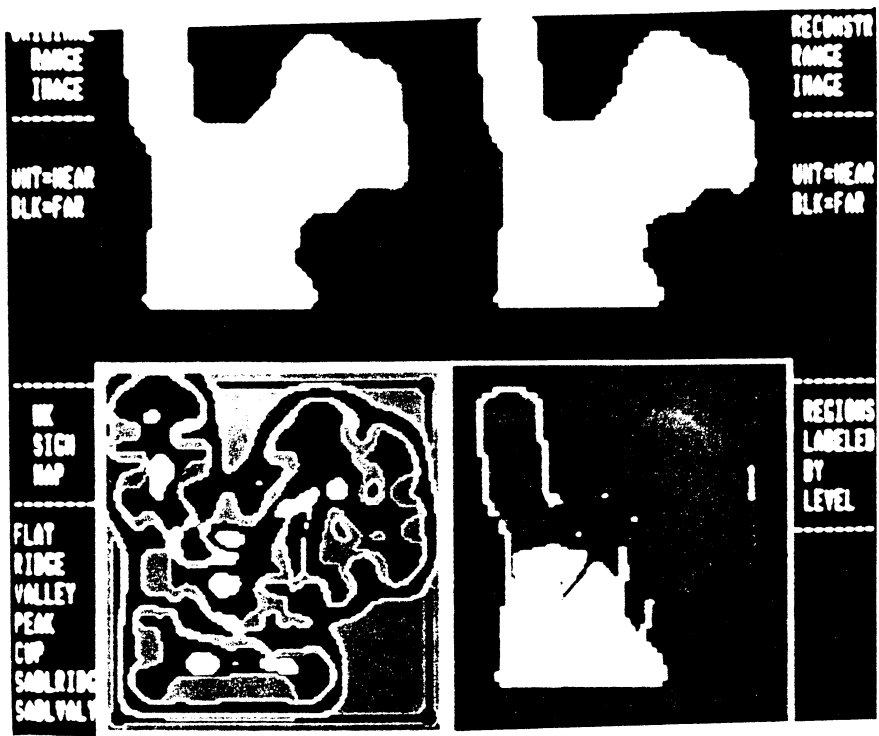


(c) Final Region Segmentation



(d) Edge Descriptions of Final Regions

**Figure 10.** Standard Results for Keyboard



(a) Standard Results for Auto Part

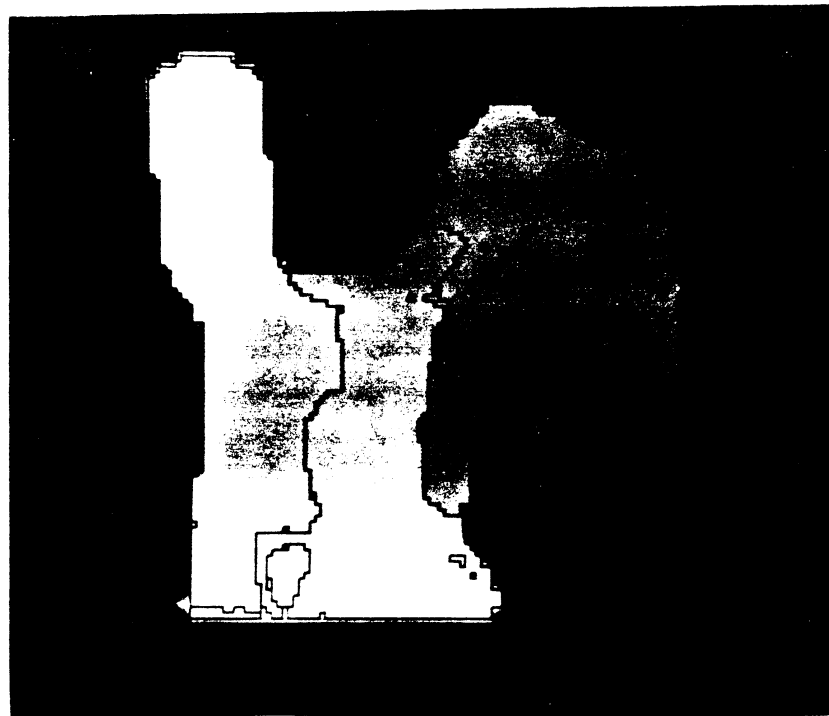
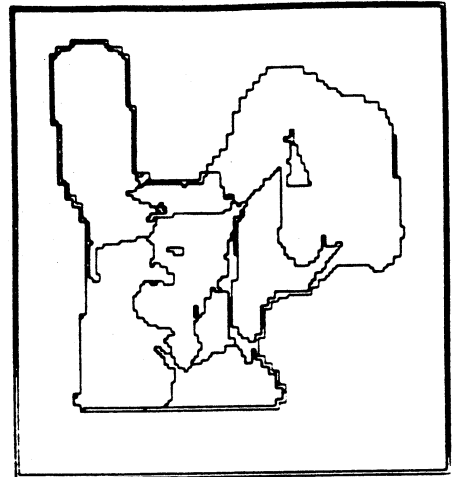


Figure 11. Modified Threshold Results for Auto Part

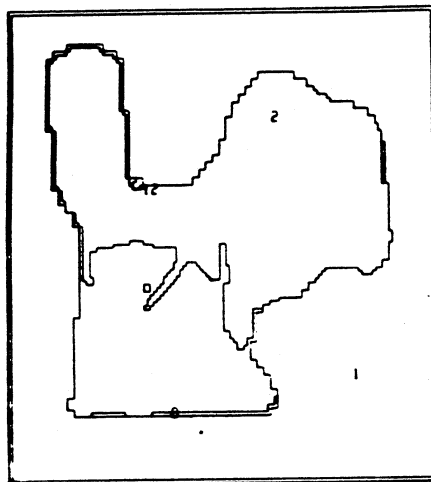




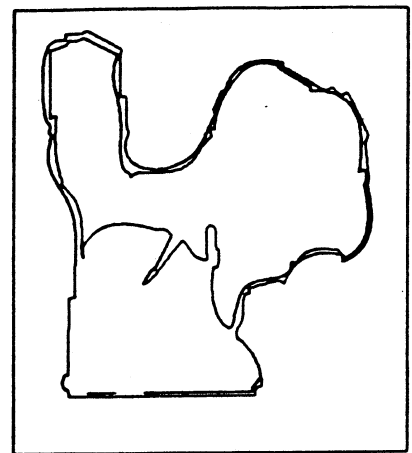
(a) Surface-Curvature-Sign Segmentation



(b) Best-Fit Region Label Buffer Segmentation

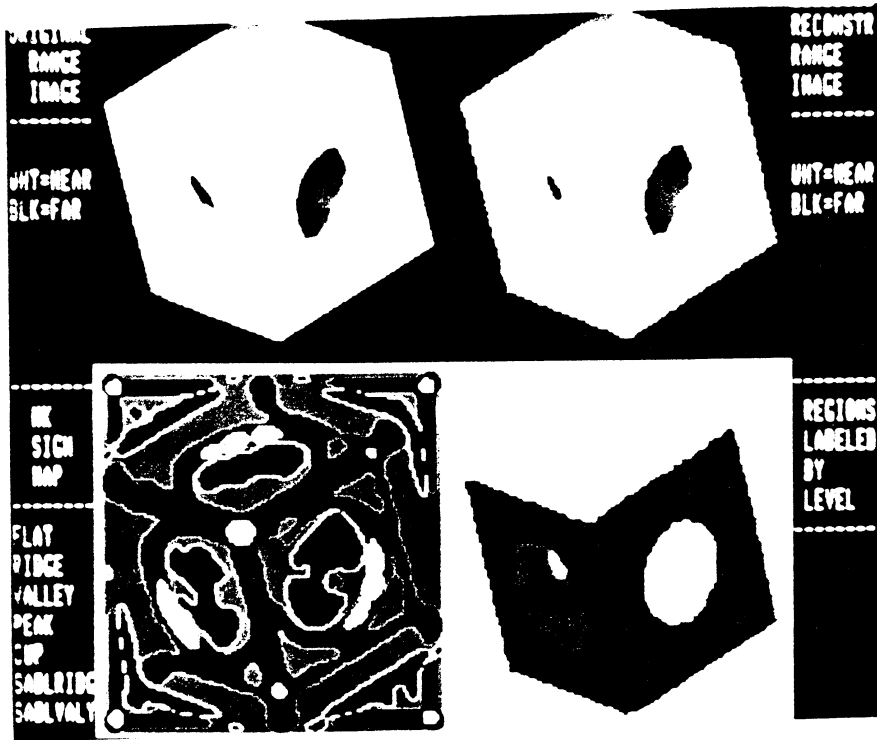


(c) Final Region Segmentation



(d) Edge Descriptions of Final Regions

**Figure 12.** Standard Results for Auto Part



(a) Standard Results for Block with Holes (Low-Noise)

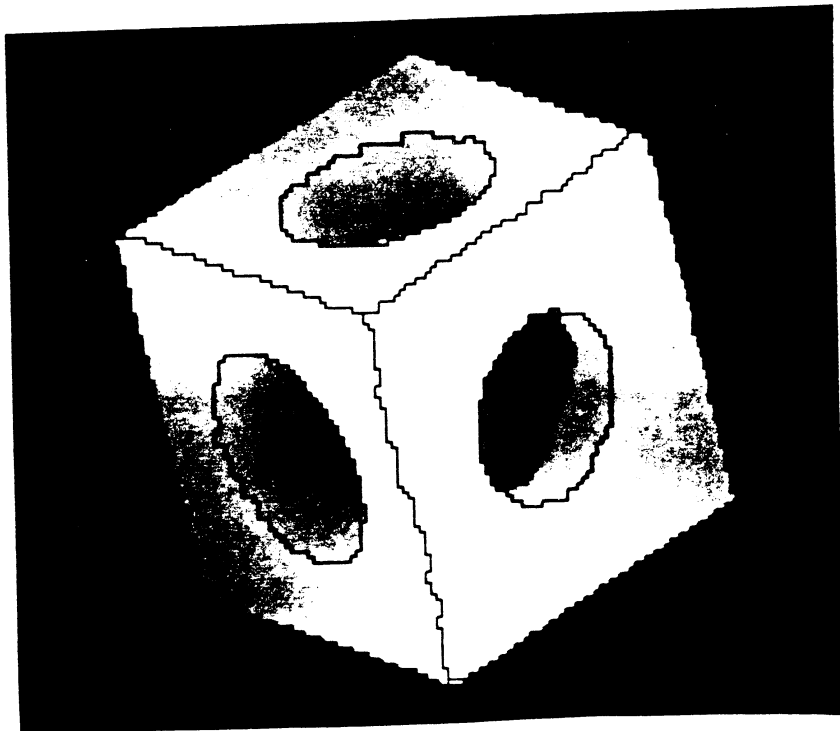
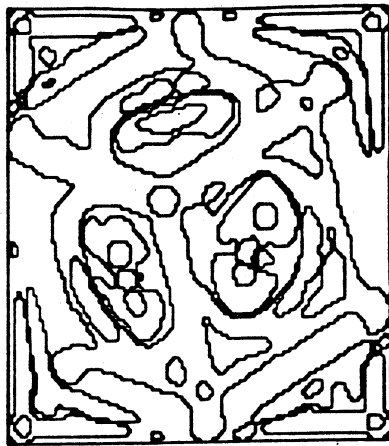
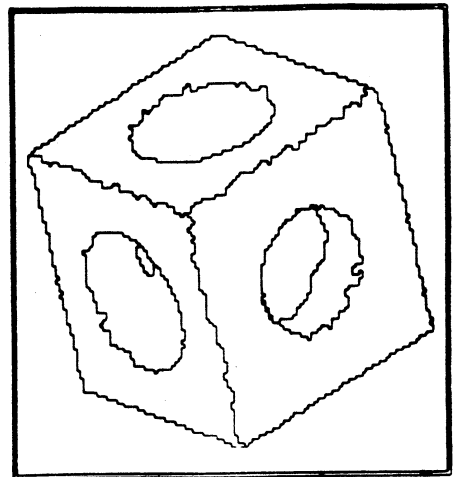


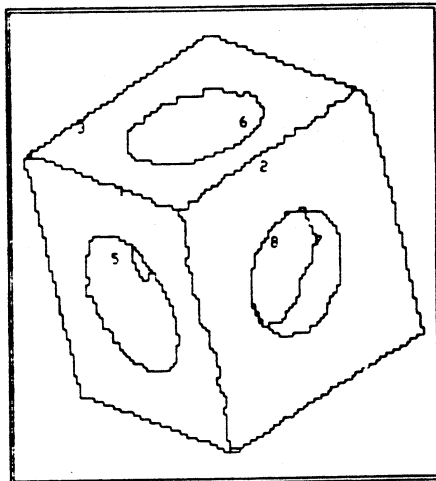
Figure 13. Standard Results for Block (Low-Noise)



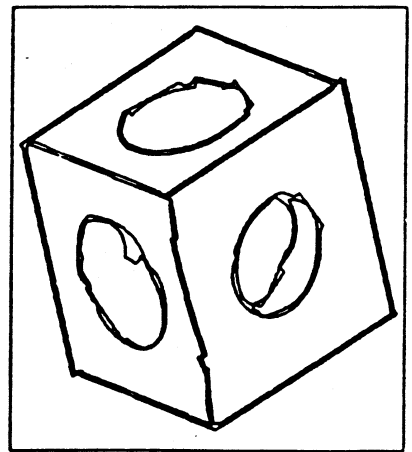
(a) Surface-Curvature-Sign Segmentation



(b) Best-Fit Region Label Buffer Segmentation

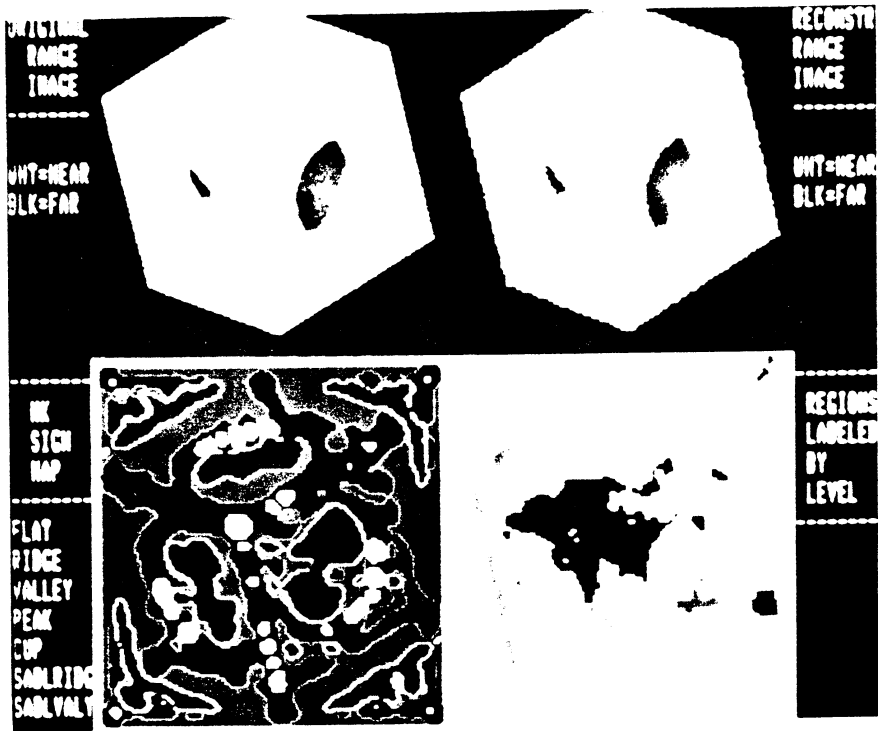


(c) Final Region Segmentation



(d) Edge Descriptions of Final Regions

**Figure 14.** Standard Results for Block (Low-Noise)



(a) Standard Results for Block (High-Noise)

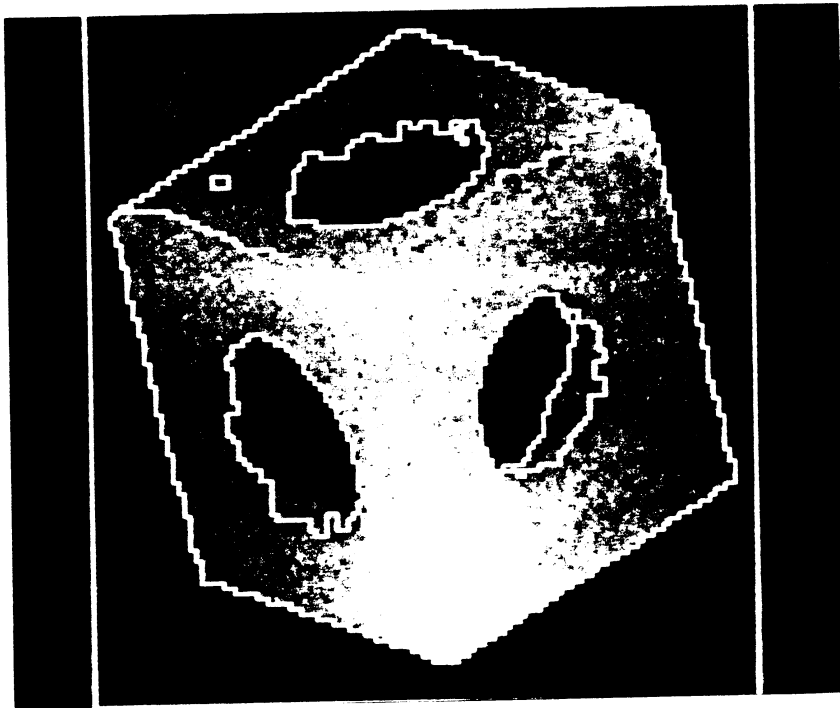
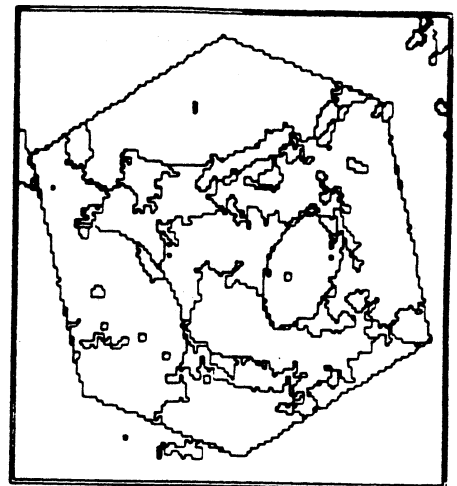


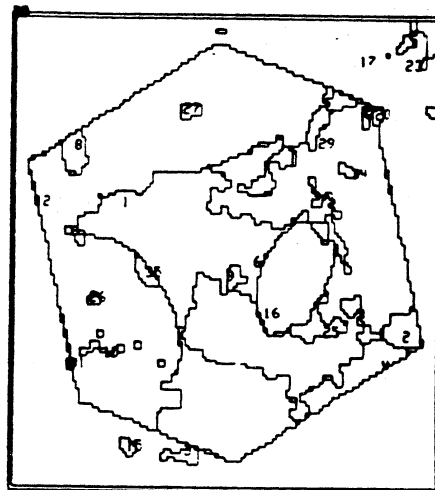
Figure 15. Modified Threshold Results for Block (High-Noise)



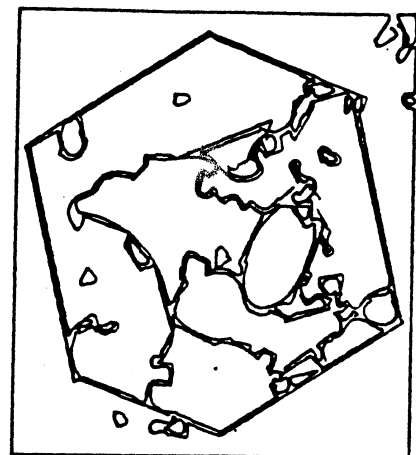
(a) Surface-Curvature-Sign Segmentation



(b) Best-Fit Region Label Buffer Segmentation

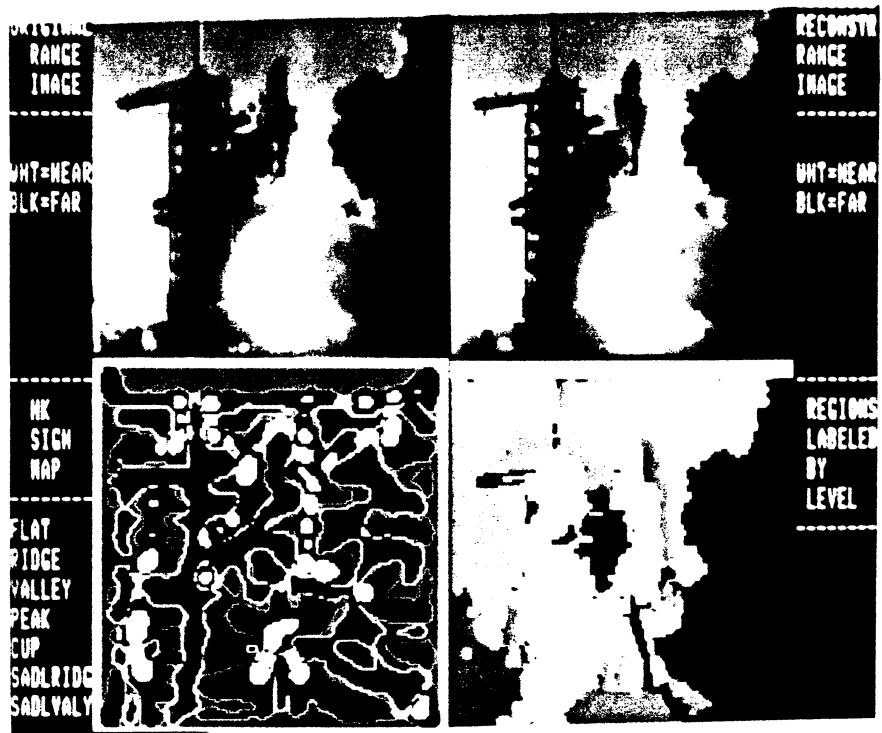


(c) Final Region Segmentation



(d) Edge Descriptions of Final Regions

**Figure 16.** Standard Results for Block (High-Noise)



(a) Standard Results for Space Shuttle

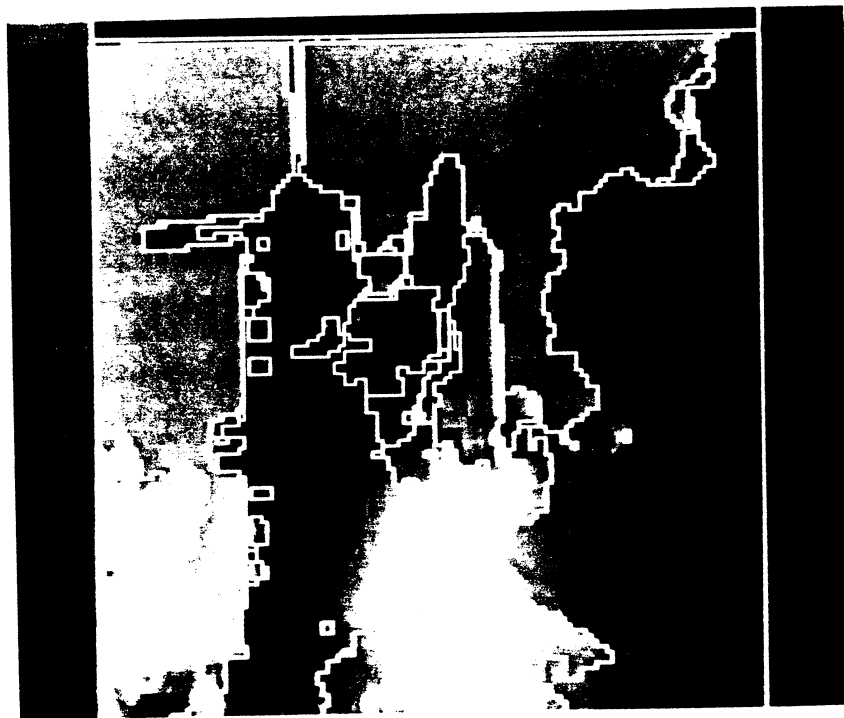
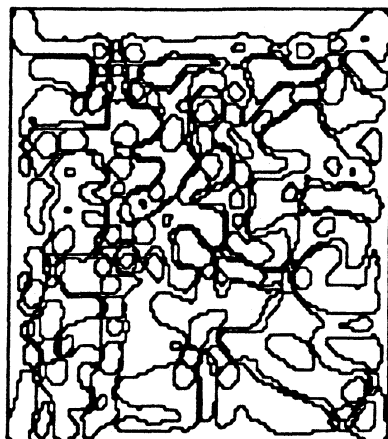
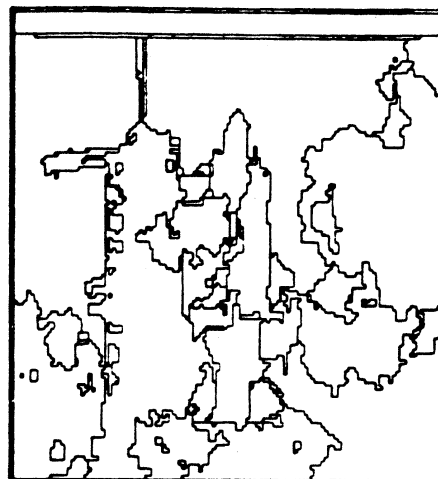


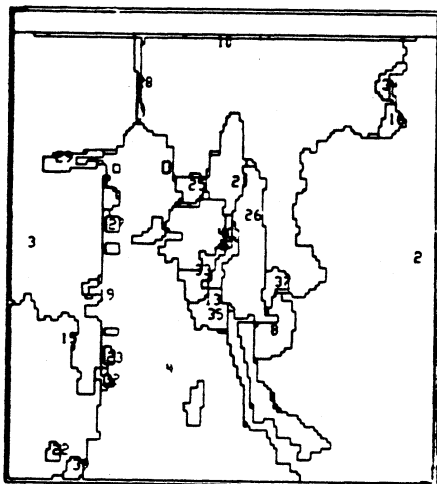
Figure 17. Standard Results for Space Shuttle



(a) Surface-Curvature-Sign Segmentation



(b) Best-Fit Region Label Buffer Segmentation



(c) Final Region Segmentation



(d) Edge Descriptions of Final Regions

**Figure 18.** Standard Results for Space Shuttle