



May 1990

## Segmentation of Range Images as the Search for the Best Description of the Scene in Terms of Geometric Primitives

Aleš Leonardis  
*University of Pennsylvania*

Alok Gupta  
*University of Pennsylvania*

Ruzena Bajcsy  
*University of Pennsylvania*

Follow this and additional works at: [https://repository.upenn.edu/cis\\_reports](https://repository.upenn.edu/cis_reports)

---

### Recommended Citation

Aleš Leonardis, Alok Gupta, and Ruzena Bajcsy, "Segmentation of Range Images as the Search for the Best Description of the Scene in Terms of Geometric Primitives", . May 1990.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-30.

This paper is posted at ScholarlyCommons. [https://repository.upenn.edu/cis\\_reports/547](https://repository.upenn.edu/cis_reports/547)  
For more information, please contact [repository@pobox.upenn.edu](mailto:repository@pobox.upenn.edu).

---

# Segmentation of Range Images as the Search for the Best Description of the Scene in Terms of Geometric Primitives

## Abstract

Segmentation of range images has long been considered in computer vision as an important but extremely difficult problem. In this paper we present a new paradigm for the segmentation of range images into piecewise continuous patches. Data aggregation is performed via model recovery in terms of variable-order bi-variate polynomials using iterative regression. All the recovered models are potential candidates for the final description of the data. Selection of the models is achieved through a maximization of quadratic Boolean problem. The procedure can be adapted to prefer certain kind of descriptions (one which describes more data points, or has smaller error, or has lower order model). We have developed a fast optimization procedure for model selection. The major novelty of the approach is in combining model extraction and model selection in a dynamic way. Partial recovery of the models is followed by the optimization (selection) procedure where only the "best" models are allowed to develop further. The results obtained in this way are comparable with the results obtained when using the selection module only after all the models are fully recovered, while the computational complexity is significantly reduced. We test the procedure on several real range images.

## Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-30.

**Segmentation As The Search For The  
Best Description Of The Scene  
In Terms Of Geometric Primitives**

**MS-CIS-90-30  
GRASP LAB 215**

**Aleš Leonardis  
Alok Gupta  
Ruzena Bajcsy**

**Department of Computer and Information Science  
School of Engineering and Applied Science  
University of Pennsylvania  
Philadelphia, PA 19104-6389**

**May 1990**

**Acknowledgements:**

**The authors wish to thank Luca Bogoni for helpful suggestions regarding the implementation of the procedure, and Helen Anderson, Ulf Cahn von Seelen, and Dr. Max Mintz for comments on an earlier draft of the paper. This research was supported in part by:  
AFOSR Grants 88-0244, AFOSR 88-0296;  
Army/DAAL 03-89-C-0031PRI; NSF Grants  
CISE/CDA 88-22719, IRI 89-06770; ARPA Grant  
N0014-88-K-0630; United States Postal Service and  
Dupont Corporation.**

# Segmentation of Range Images as the Search for the Best Description of the Scene in Terms of Geometric Primitives

**Aleš Leonardis, Alok Gupta and Ruzena Bajcsy**

GRASP Laboratory  
Department of Computer and Information Science,  
University of Pennsylvania, Philadelphia, PA 19104

## **Abstract**

Segmentation of range images has long been considered in computer vision as an important but extremely difficult problem. In this paper we present a new paradigm for the segmentation of range images into piecewise continuous patches. Data aggregation is performed via model recovery in terms of variable-order bi-variate polynomials using iterative regression. All the recovered models are potential candidates for the final description of the data. Selection of the models is achieved through a maximization of quadratic Boolean problem. The procedure can be adapted to prefer certain kind of descriptions (one which describes more data points, or has smaller error, or has lower order model). We have developed a fast optimization procedure for model selection. The major novelty of the approach is in combining model extraction and model selection in a dynamic way. Partial recovery of the models is followed by the optimization (selection) procedure where only the “best” models are allowed to develop further. The results obtained in this way are comparable with the results obtained when using the selection module only after all the models are fully recovered, while the computational complexity is significantly reduced. We test the procedure on several real range images.

## 1 Introduction

Segmentation of images has long been considered in computer vision as an important but extremely difficult problem. One of the reasons being that the definition of segmentation is task oriented and varies for different types of sensors. We view the segmentation process as a data reduction mechanism that requantizes sensory measurements into some predefined primitive elements which encompass the available knowledge. This enables us to infer a symbolic description of the world. Most commonly, segmentation is viewed as a local to global aggregation problem with various similarity criteria [BJ88] employed to achieve a coherent global description. Indeed, this global description is most usefully achieved in terms of global primitives that are easy to extract and are useful for the later processing. This can be accomplished in two ways: one is to actively use the global model as the individual primitives are being developed, in essence recovering the model as aggregation proceeds. The other way is to use a local coherence measure to first classify the data and then use the fitting technique to recover the model. The latter approach, though not limited by the global model at the aggregation stage, essentially isolates the segmentation and the representation stages, with the result that the final description might not correspond to the global model since it played no part in the segmentation process. Besides, the outliers in the data set resulting from misclassification, and the sensitivity of the methods for model estimation may lead to disastrous results [Che89]. A desirable approach is to use both the local coherence measure and the global model to guide the segmentation, corroborating our notion that the problems of segmentation and representation are not separable [BSG90].

In this paper we attempt to provide a definition of segmentation as partitioning the images (range or reflectance) into primitive models by *searching* for the models as they are developed everywhere in the image, such that the description is best in terms of global shape and error. By searching we mean fitting and selecting only those models that best describe the underlying data using the criterion function which takes into account the number of points that are described by a particular model, its goodness-of-fit, and the structural complexity of the model. Our method performs data aggregation via model recovery in terms of variable-order (up to second-order) bivariate patches using iterative regression. Model recovery starts simultaneously and independently at all the regions found to be globally coherent in the initial neighborhood (seed regions). All the recovered models are potential candidates for the final description. To make the method computationally feasible, it is necessary to monitor region growing and discard superfluous regions even before they are fully grown. The major novelty of our approach is in combining model extraction and model selection in a dynamic way, such that only the “best” models are allowed to develop further.

## 1.1 Related work

There are two areas of research that are relevant to our work. One is segmentation and in particular that based on least-squares regression. The other one is optimization of Boolean programming problems used for the selection of optimal description of image in terms of primitives.

The field of range image segmentation has traditionally been explored by researchers by studying invariant differential geometric properties of surfaces, followed by fitting surface or volumetric models to the segmented data, or by using the model to guide the segmentation process. There are numerous methods performing the segmentation by aggregating the local surface models like curvature, surface normals, etc. [BJ85, HJ87], or by detecting the surface discontinuities ( $C_0$  and  $C_1$  discontinuities and smooth boundaries) [Fan88, GL89, SK85].

Approaches based on local differential geometry are the most widely studied techniques for surface segmentation [BPYA85, AB86, PB84, BJ86, SZ88, LT90]. They range from local analysis of the surface to more global interpretation like peaks, pits and passes (saddle points) on a surface [Nac84]. Some of the drawbacks of differential geometric approaches are that they are applicable only in a small neighborhood of the surface and require extensive processing if a global model is to be used later (e.g. in Besl and Jain [BJ88]). Although differential geometric quantities have nice invariant properties, they invariably require smoothing of the data due to the sensor and quantization noise. The undesirable side effect of uniform smoothing is that it alters the underlying surface by smoothing the discontinuities which are vital for surface segmentation. There are several application and sensor dependent range image segmentation techniques that are not of much interest to us since we are interested in a general algorithm. Besl and Jain [BJ85] have summarized the field of 3-D segmentation in their excellent survey.

The methods based on aggregation of local properties cluster data into perceptually or geometrically significant regions with or without considering the final representation in terms of primitives. If a representation in the form of a parametric model is desired, then the model is invoked after the initial clustering. Similarly, purely edge-based methods fit models to closed regions implicitly defined by edges. The fundamental drawback of such approaches is that they isolate the problem of segmentation from the issue of representation. In other words, the model used for representation plays no role in the process of segmentation. To obtain a meaningful segmentation, it is desirable to use the model (representation) to guide the segmentation [BSG90, FH86, BJ88]. One of the motivations for our work is illustrated in figure 1 showing the object (taken from Fan [Fan88]). An edge-based method is unable to segment the two planar surfaces (A and B) joining smoothly by the curved surface C, nor is a model available to describe the union surface S (indicating that segmentation is necessary). Since our approach combines model representation and segmentation, it can successfully segment such an object (figure 13).

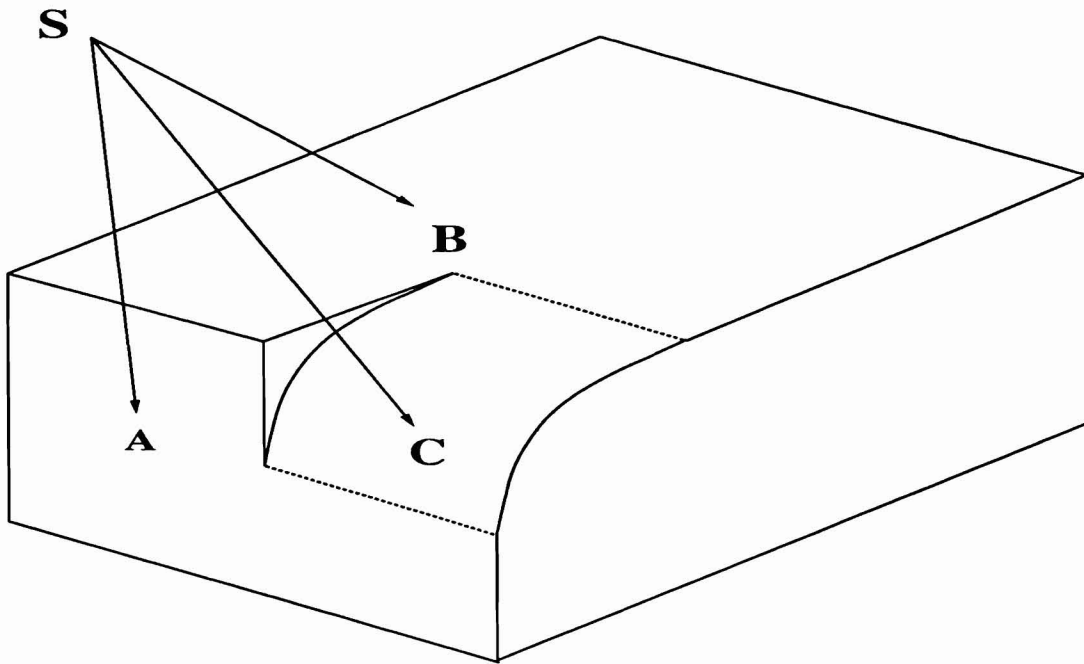


Figure 1: An example of an object that can not be segmented by edge-based approaches.

Much work has been done recently on the problem of reconstructing piecewise-smooth surfaces in one or more dimensions [BZ87, MS85, Ter86, PTK85], which is posed as an optimization problem. In all these approaches the data is weighted uniformly which means that the algorithms do not possess the capabilities to adapt to different conditions in different parts of the image. The global measure provided by the energy function is not able to tell which parts of the image are well described in terms of the underlying models and which are not. Also it is difficult to see how these approaches could be extended to subsequent stages of the vision problem without using models with fewer degrees of freedom. Leclerc [Lec89] developed an interesting concept which can compensate for some of these drawbacks by defining an objective function that is based on the information theoretic notion of minimum length descriptions.

The other area of research that is relevant to our work is optimization of Boolean programming problems [HT85, Pen90]. Perhaps the closest in spirit to our approach to model selection is the one used by Pentland [Pen90]. A detailed overview and comparison of our approach with the existing methods is given in the section on model selection.

The paper is organized as follows. In Section 2 we describe the general framework for segmentation and define our models. Segmentation and model recovery using an iterative refinement approach is explained in Section 3. In Section 4 we explain how the models with the best descriptive power are selected via optimizing the Boolean programming problem. Section 5 describes our

approach in combining model extraction and model selection in a dynamic way, which drastically reduces the computational complexity. In Section 6 we discuss postprocessing steps required to refine the segmentation results. We present some of our experimental results in Section 7. In conclusion we summarize our paradigm and outline the work in progress.

## 2 General Framework

We begin by formally defining the symbols used for the description of our algorithm in this paper. The image space is denoted by  $\mathcal{I}$ . A pixel location  $(x, y) \in \mathcal{I}$  is represented as vector  $\mathbf{x}$ . The input image data at  $\mathbf{x}$  is given by a function in image space  $\mathcal{I}$  as  $g(\mathbf{x})$ . The general segmentation problem is usually stated as follows [HP74, Zuc76]. Given the set of all image pixels and a logical uniformity predicate  $P(\cdot)$ , find a partition  $S$  of the image  $\mathcal{I}$  in terms of set of regions  $R_i$ , where each  $R_i$  is the domain of a model (primitive). Let  $N_R$  be the number of regions in the segmented image, and let  $|R_i|$  be the number of pixels in the region  $R_i$ . The following conditions must hold for the set  $S$ :

$$\bigcup_{i=1}^{N_R} R_i = \mathcal{I} \quad (1)$$

where  $R_i \subseteq \mathcal{I}$  for each  $i$ . Contrary to the conventional definition of segmentation, our primitive based approach permits model domains to overlap partially, therefore :

$$R_i \cap R_j \neq \emptyset \quad (2)$$

for  $i \neq j$  in general.  $R_i$  is a 4-connected set of pixels.

The uniformity predicate

$$P(R_i) = \text{TRUE} \quad (3)$$

for all  $i$ . If  $R_i$  is adjacent to  $R_j$

$$P(R_i \cup R_j) = \text{FALSE} \quad (4)$$

The uniformity predicate  $P(\cdot)$  defines the conformity of all the points in  $R_i$  to the global model (primitive). In the following subsection we discuss the issue of the choice of primitives.

### 2.1 Primitives for Representation

The criteria for selection of primitives have been studied extensively by vision researchers [Bra83, Mar82, Bin82]. The primitives should be invariant to rotation, translation, and scale. *Accessibility*, defined as computability of the primitive is essential, since our goal is to recover the structure from



the input. *Stability* of the primitive with respect to minor changes due to noise or viewpoint, with respect to scale and configuration is important to generate consistent representations. While small changes in scale should not create major changes in the description, a multi-scale representation should be possible. The primitives should have local support, so that occluded parts can still be described.

Besides, primitives should balance the trade-off between data reduction and faithfulness to measured data. They should be generic and data-dependent which is a compromise between the complete knowledge based approach and the one where the primitives possess an enormous number of degrees of freedom in order to model everything. Knowledge about the constituents of the scene can make the segmentation process less dependent on noisy data, thus more robust, but less general. On the other hand, methods that do not constrain their primitives do not achieve any compression or symbolic description. While the number of different scenes is almost non-countable, the number of spatial primitives (planar, convex, concave) is relatively small. This enables us to build models and to find their instances in the scene. It is important for the further processing that they correspond to meaningful segments in terms of physical phenomena or in terms of natural qualitative description (planar, convex, or concave shape, for example). In other words they should possess features which contain perceptually significant information. They represent an intermediate stage in the process of abstraction of information from early levels into successively more complex forms.

However, in all model based approaches we are restricted by the primitives, since they cannot model everything present in the input data. Nevertheless, they can provide approximate descriptions of data even if a model is not present in the vocabulary. For the regions that cannot be accurately represented by the model vocabulary, (for example, if a surface curves faster than the highest order model) it is important that the primitives can be easily combined in a description of patchwise continuous combination of model primitives. In this respect, surface models like bivariate polynomials are better than volumetric models like generalized-cylinders and superquadrics, since they have more local support and higher fidelity to the underlying data.

Spatial primitives like curves and surfaces satisfy the above criteria. Additional criteria for primitive selection, as given by Besl [Bes88] are : 1) Models should approximate well any smooth curve or surface of constant sign-of-curvature over a finite domain. 2) Models should extrapolate accurately to arbitrary points outside the current domain. 3) Models should interpolate between missing points inside the domain. 4) Models should be computed efficiently. 5) Model representation should be compact.

A general parametric model  $f(\mathbf{a}, \mathbf{x})$  can be represented as :

$$\mathbf{f}(\mathbf{a}, \mathbf{x}) = \sum_{l=1}^{p(r)} \mathbf{a}_l f_l(\mathbf{x}) \quad (5)$$

where  $f_l(\mathbf{x})$  are basis functions defined on  $\mathcal{I}$ , and  $\mathbf{a}$  is the parameter vector of the model.

The models we have chosen are the *variable-order bi-variate polynomials* that are linearly parameterizable in the Euclidean space :

$$\hat{\mathbf{f}}(r, \mathbf{a}, \mathbf{x}) = \sum_{i+j \leq r} \mathbf{a}_{ij} x^i y^j \quad (6)$$

where the vector  $\mathbf{a}$  is defined in the parameter space  $\mathcal{A}$ . Dimensions of the parameter space depend on the order of the model  $r$  which is in our case restricted to  $0 \leq r \leq 2$ . Thus our model admits planar and bi-quadratic surfaces. Surfaces of higher order can introduce oscillations, are computationally expensive, and are often unstable during the model-recovery process. If the underlying surface is curving faster than a second-order patch, then it is always possible to break the patch into smaller second-order patches. Our algorithm for model recovery and model selection takes care of this in a unique manner. Even if higher order models (say up to fourth-order as in Besl and Jain, 1988) are considered, there is no guarantee that model will always fit the data, as is typical with any primitive based approach. Instead, bi-quadratic patches have nice properties that they can be used to merge segmented descriptions in order to come up with more global descriptions like concave and convex patches. Second-order patches give descriptions which are perceptually supported. What is more, it is easy to do reasoning with them and to compute discontinuities. The final description in terms of piecewise continuous second-order patches contains global information about the scene that can be described qualitatively (like convex, concave patches) as well as quantitatively (curvature, normals, etc.). The description has local support and can be used to derive quantities and higher level descriptions that are invariant to translation and rotation (Gaussian curvature, critical points, etc.).

### 3 Segmentation and Model recovery

#### Iterative Regression Approach

In this section we describe the process of recovering the primitives from the data. This is performed by following the *iterative regression approach*, used for surface segmentation by [BJ88] and for contour segmentation by [Che89]. Our approach differs from the former's in the selection of initial estimates (seed regions), search for connected compatible points, highest order of the polynomial; and from the latter's in that we do not restrict the connected compatible region to a predefined size and update the model's order during region growing. Our method differs from the Random

Sample Consensus (Ransac) approach of Bolles and Fischler [BF81] in that the seed regions are not selected at random and the model is updated during region-growing depending on the data.

### 3.1 Surface fitting

Let us first introduce the notation of a linearly parameterizable surface patch  $\mathcal{S}(r, \mathbf{a}, \mathbf{x})$  :

$$\mathcal{S}(r, \mathbf{a}, \mathbf{x}) = \{(\mathbf{x}, z) \in \mathcal{I} \times \mathcal{Z} \mid z = \hat{f}(r, \mathbf{a}, \mathbf{x})\} \quad (7)$$

The distance function from a data point  $g(\mathbf{x})$  to the surface  $\mathcal{S}(r, \mathbf{a}, \mathbf{x})$  is given by :

$$d^2(r, \mathbf{a}, \mathbf{x}) = [g(\mathbf{x}) - \hat{f}(r, \mathbf{a}, \mathbf{x})]^2 \quad (8)$$

Let us take a topologically connected set of points  $\mathcal{D}$  which is a subset of  $\mathcal{I}$  and define the total deviation of the points from the surface  $\mathcal{S}(r, \mathbf{a}, \mathbf{x})$  :

$$\chi^2(r, \mathbf{a}, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} d^2(r, \mathbf{a}, \mathbf{x}) \quad (9)$$

Given a set of points  $\mathcal{D}$ , the problem is to find the order  $r$  of the model and the parameters  $\mathbf{a}$  which will minimize the error function  $\chi^2(r, \mathbf{a}, \mathcal{D})$ . Using least-squares regression we get :

$$\chi^2(r, \hat{\mathbf{a}}, \mathcal{D}) = \min_{\mathbf{a} \in \mathcal{A}} \chi^2(r, \mathbf{a}, \mathcal{D}) \quad (10)$$

We use the standard technique for solving the *General Linear Least Squares Problem*.

The solution is given by :

$$\mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^T \mathbf{Y} \quad (11)$$

where  $\mathbf{X}^T$  is the transpose of the matrix  $\mathbf{X}$ . The optimal vector  $\hat{\mathbf{a}}$  is computed as :

$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (12)$$

$$\text{where } \hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_p \end{bmatrix}_{p \times 1}, \quad \mathbf{Y} = \begin{bmatrix} g(\mathbf{x}_1) \\ \vdots \\ g(\mathbf{x}_n) \end{bmatrix}_{n \times 1}, \quad \mathbf{X} = \begin{bmatrix} f_1(\mathbf{x}_1) & \dots & f_p(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ f_1(\mathbf{x}_n) & \dots & f_p(\mathbf{x}_n) \end{bmatrix}_{n \times p} \quad (13)$$

and  $f_l(\mathbf{x})$  are the basis functions as defined in equation 5. The solution depends on the points in  $\mathcal{D}$ . If  $\mathcal{D}$  is determined before the fitting takes place then the schema is called Classify-then-Fit [Che89]. As mentioned earlier, this approach essentially isolates the segmentation and the representation

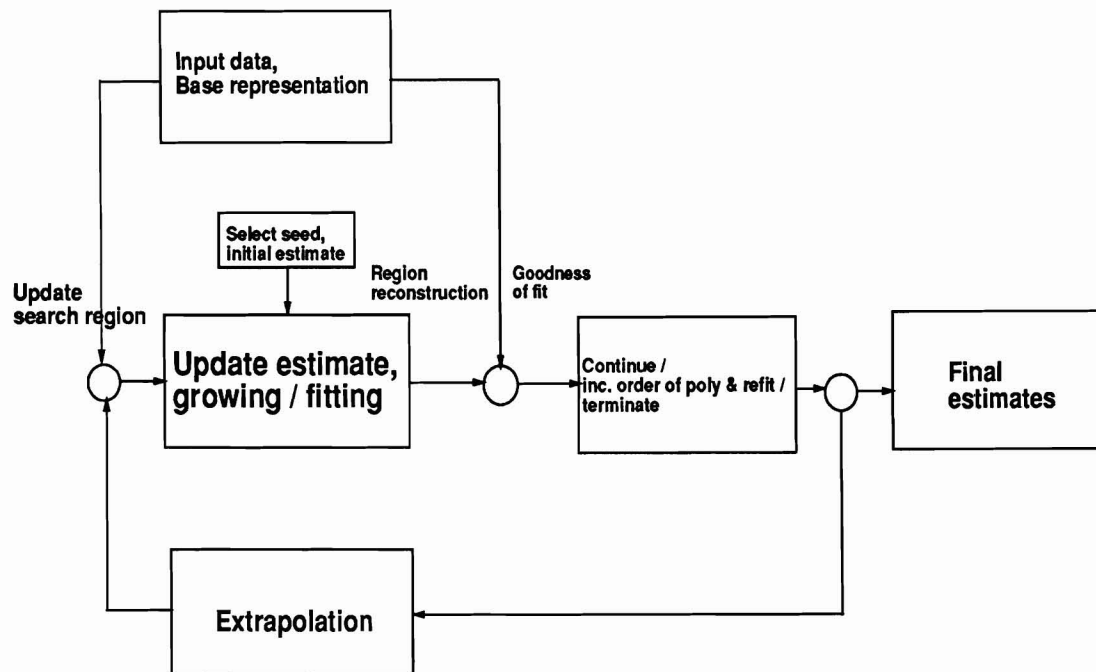


Figure 2: Schema for the iterative regression approach to model-recovery (for one model).

stages, with the result that the final description might not correspond to the global model since it played no role in the segmentation process. Besides, the outliers in the data set resulting from misclassification, and the sensitivity of the methods for model estimation may lead to disastrous results. A desirable approach is to use both the local coherence measure and the global model to guide the segmentation. This is achieved by an iterative procedure combining data classification and model fitting - *the iterative regression method* - with an additional feature that during the process of model recovery even the order of the model can be changed [BJ88]. Our algorithm is outlined below. First we will explain the procedure for one region. The general algorithm which automatically determines the regions (seeds) where the process is initiated is discussed later.

### 3.2 The Segmentation Algorithm (recovery of one model)

The schematic diagram of the algorithm is shown in figure 2. The detailed step-wise description is given below :

1. Initial region (seed) is a small window whose size is determined on the basis of scale considerations and can be adaptively changed depending on the data. It is denoted by  $\mathcal{D}^{(0)}$ .
2. A first-order surface is fitted to the data in  $\mathcal{D}^{(0)}$ . On the basis of results of different statistical techniques (explained in §3.3) we make a decision whether all the data in  $\mathcal{D}^{(0)}$  belong to the

same surface. If they do, the procedure continues with the following steps combining data classification and model fitting. Otherwise the model recovery process is not initiated and the seed-region is rejected.

3. Set the order  $r$  to 0. If the goodness-of-fit is not acceptable then set the order  $r$  to 1. Compute the initial estimates of parameters  $\mathbf{a}^{(0)}$  by fitting the data  $\mathcal{D}^{(0)}$  in  $\mathcal{S}^{(0)}(r, \mathbf{a}^{(0)}, \mathcal{D}^{(0)})$ . The iterative *data classification and model fitting* consist of the following three steps performed in a loop, till the region growing terminates.

- 3.1  $\mathcal{D}^{(s)}$  is updated with all *compatible* points. This is achieved via extrapolation of the current estimate  $\mathcal{S}^{(s)}(r, \mathbf{a}, \mathbf{x})$ . *Compatible points* are defined as :

$$\mathcal{C}^{(s)} = \{\mathbf{x} \mid d^2(r, \mathbf{a}, \mathbf{x}) \leq C \text{ and } \mathbf{x} \in \text{4\_conn\_neighborhood\_of}(\mathcal{D}^{(s)} \cup \mathcal{C}^{(s)})\} \quad (14)$$

where  $C$  is the *compatibility constraint*, used as a local coherence measure. It also acts as a scale parameter by effectively “smoothing” the data. Notice the recursive definition of the compatible points  $\mathcal{C}^{(s)}$  (the set is initially empty) which are connected to the current  $\mathcal{D}^{(s)}$  or to new compatible points  $\mathcal{C}^{(s)}$ . The distance of the connected compatible points from  $\mathcal{D}^{(s)}$  can be controlled by specifying the additional condition in equation 14 that

$$(\mathbf{x} \text{ is no further than } k \text{ pixels from the nearest border point in } \mathcal{D}^{(s)}) \quad (15)$$

- 3.2 Based on  $\mathcal{D}^{(s+1)} = \mathcal{D}^{(s)} \cup \mathcal{C}^{(s)}$ , update the model, and compute the new goodness-of-fit:

$$\mathcal{S}^{(s+1)}(r, \mathbf{a}^{(s+1)}, \mathcal{D}^{(s+1)}) \quad \text{where } \mathbf{a}^{(s+1)} = \mathbf{a}_{\mathcal{D}^{(s+1)}} \quad (16)$$

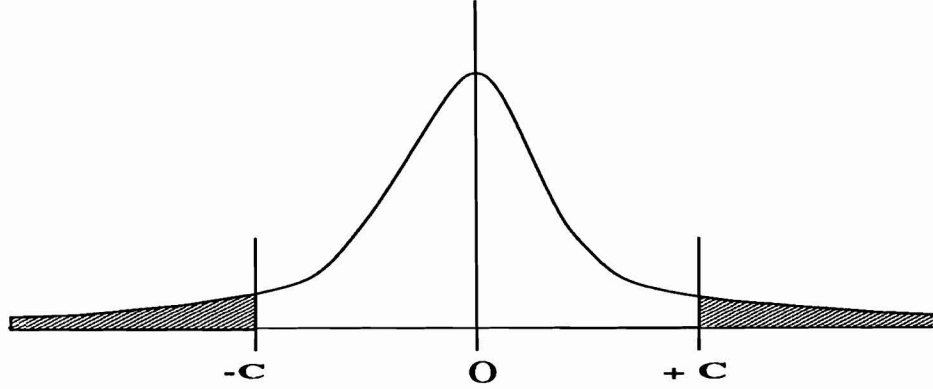
The difference between the old and the new goodness-of-fit is :

$$\rho^{(s+1)} = \chi^2(r, \mathbf{a}^{(s+1)}, \mathcal{D}^{(s+1)}) - \chi^2(r, \mathbf{a}^{(s)}, \mathcal{D}^{(s)}) \quad (17)$$

- 3.3 These two steps are followed by a decision making process :

- i. If  $(\mathcal{C}^{(s)} = \emptyset)$  Goto step 3.3(iii).
- ii. If  $(\rho^{(s+1)} < T_1)$  continue with growing. Goto step 3.1.
- iii. Update the order.  $r = r + 1$ . If  $(r > \text{max\_order})$  Goto step 3.3(v)
- iv. Update model for new  $r$ . If error improves significantly continue with growing. Goto step 3.1.
- v.  $\mathcal{D} = \mathcal{D}^{(s+1)} - \mathcal{C}^{(s)}$ .  $\mathbf{a} = \mathbf{a}_{\mathcal{D}^{(s)}}$  Goto step 4.

4. Done with region growing. Store the model  $\hat{f}(r, \mathbf{a}, \mathbf{x})$  and the region of its extent  $\mathcal{D}$ .

Figure 3: Noise distribution and the role of  $C$ .

---

### 3.2.1 Features of the Algorithm

**Thresholds:** The thresholds for model acceptance and updating the order of the model are determined empirically and kept constant for all the data from the same sensor.

**Termination:** The algorithm always terminates, since the monotonicity requirement for growing regions holds :

$$\mathcal{D}^{(s)} \subseteq \mathcal{D}^{(s+1)} \subseteq \mathcal{I} \quad (18)$$

**Insensitivity to outliers:** The *iterative regression method* is an efficient tool for data-driven extraction of parametric features. Its main advantage is that the performance of the fitting is constantly monitored. The procedure dynamically analyzes data consistency allowing rejection of the outliers. The *compatibility constraint*,  $C$ , which is determined on the basis of the statistical behavior of the sensor prevents the outlying points from being taken into the fitting process (figure 3). This is an important feature since least-squares fitting has undesirable sensitivity to outlying points, and measurement errors are not necessarily normally distributed [BKM86].

**Switching the Order of the Model** A first-order model when grown to its full extent can extend to the neighboring regions along the intersection curve (figures 4 and 8). While it is not a cause for concern in the case shown in figure 8 since the surface is planar, it is necessary to remove the points along the intersection curve for the unbiased growth of the model as a second-order surface in figure 4. We have devised a systematic analytical method for removing these points before the model switches to the second-order. The method, which is general in scope, is explained in section 6.

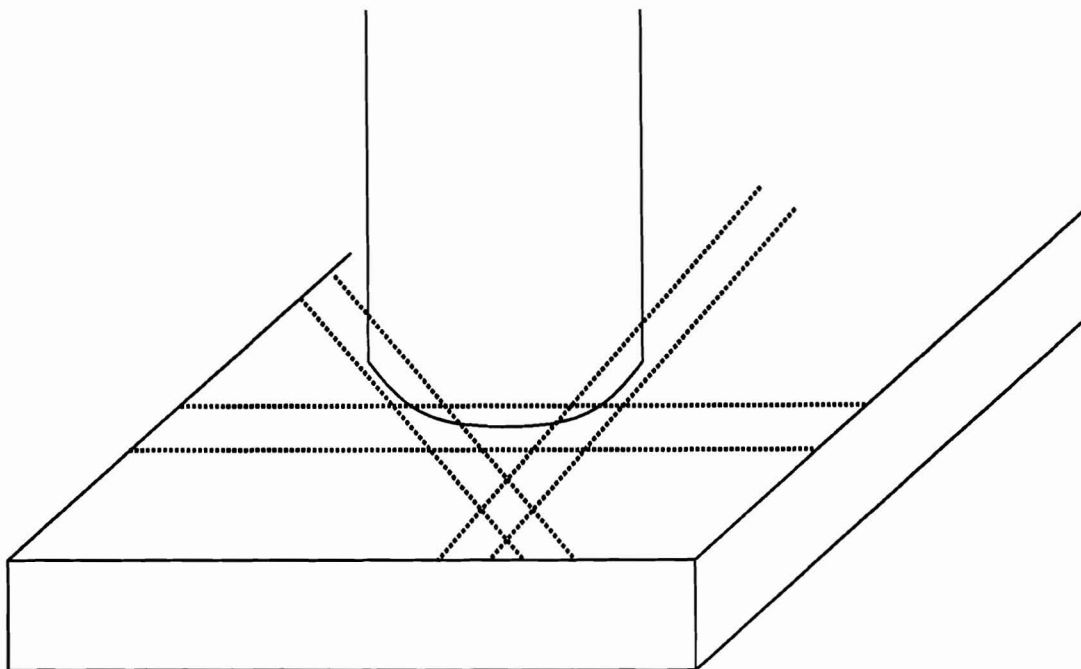


Figure 4: Changing the model - Switching the order.

### 3.2.2 Solving Least Squares Problem

In the step 2 of the segmentation algorithm we accept or reject seed regions based on a global coherence measure, which is the global chi-square error of the first-order least squares fit.

$$\rho^{(0)} = \chi^2(\mathcal{D}^{(0)}) \quad (19)$$

If the seed is accepted we compute the initial estimates of parameters  $\mathbf{a}^{(0)}$  by fitting the data  $\mathcal{D}^{(0)}$  in  $\mathcal{S}^{(0)}(r, \mathbf{a}^{(0)}, \mathcal{D}^{(0)})$ . This is achieved by solving the normal equations:

$$\mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^T \mathbf{Y} \quad (20)$$

Here we describe an efficient way for solving the normal equations [Seb77].

Let us define the augmented matrix  $\mathbf{A}$ .

$$\mathbf{A} = \begin{pmatrix} \mathbf{X}^T \mathbf{X} & \mathbf{X}^T \mathbf{Y} \\ \mathbf{Y}^T \mathbf{X} & \mathbf{Y}^T \mathbf{Y} \end{pmatrix}_{(p+1) \times (p+1)} \quad (21)$$

By simply applying Gaussian elimination to the first  $p$  columns of the augmented matrix  $\mathbf{A}$  we obtain

$$\begin{pmatrix} \mathbf{V} & \mathbf{d} \\ 0 & \rho^{(0)} \end{pmatrix}_{(p+1) \times (p+1)} \quad (22)$$

Since  $\mathbf{V}\mathbf{a} = \mathbf{d}$ , where  $\mathbf{V}$  is an upper triangular matrix, the elements of  $\mathbf{a}$  can be found by back-substitution. The important observation is that the chi-square error can be obtained directly,

$$\rho^{(0)} = \mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (23)$$

without computing the parameters  $\mathbf{a}$ , thus reducing the computational complexity of the algorithm when only the goodness-of-fit is required.

In each iteration we update the model with new compatible points. This is simply accomplished by updating the augmented matrix  $\mathbf{A}$  and performing Gaussian elimination followed by the back-substitution.

### 3.2.3 Computational Complexity:

**Initial estimate:** The computational complexity for computing the first estimate is:

$$\mathcal{O}(np^2) + \mathcal{O}(p^3) \quad (24)$$

where  $n = \|\mathcal{D}\|$  is the number of points and  $p = \|\mathbf{a}\|$  is the number of unknown coefficients. Note that the number of coefficient is limited from above and is in our case less or equal to 6.

Let us explain the equation 24. We need  $np^2$  multiplications to build the matrix  $(\mathbf{X}^T \mathbf{X})$  and  $np$  multiplications to obtain the term  $(\mathbf{X}^T \mathbf{Y})$ . The computational complexity required to obtain the solution vector  $\mathbf{a}$ , and also the covariance matrix if needed, is of the order of  $\mathcal{O}(p^3)$ . In case that  $n \gg p$ , the computational complexity is  $\mathcal{O}(n)$ , which is linearly proportional to the number of points.

**Updating the estimate:** Let  $n'$  denotes the number of points that are added to update the estimate of the vector  $\mathbf{a}$ . The computational complexity for updating the estimate is:

$$\mathcal{O}(n'p^2) + \mathcal{O}(p^3) \quad (25)$$

We add a whole set of new points so the dominant factor in determining the computational complexity becomes the updating of the terms  $(\mathbf{X}^T \mathbf{X})$  and  $(\mathbf{X}^T \mathbf{Y})$ . This can be effectively achieved by storing the  $(\mathbf{X}^T \mathbf{X})$  size  $p \times p$  and vector  $(\mathbf{X}^T \mathbf{Y})$  size  $p \times 1$ . The number of the operations that is needed to update the first term is  $\mathcal{O}(n'p^2)$  and the second  $\mathcal{O}(n'p)$ . If the number of updating points  $n' \gg p$ , then the procedure is asymptotically linear in the number of newly added points.



### 3.3 Complete algorithm for model recovery : Seed selection

The only problem that remains to be explained is where to find the initial regions (seeds) in general. Selection of the seed regions is one of the most important issues in any region growing algorithm since it determines the starting estimates of the parametric models. Placing the seed regions arbitrarily on the image cannot guarantee good description at least for those parts that need to be split to match our primitives. Clearly, this requires the knowledge of those parts *a priori* which is nothing but segmentation. This catch-22 problem can be solved either by intelligently selecting the seeds by computing the primitive-related properties and placing the seeds in the pre-processed image or by placing the seeds everywhere in the image thereby ensuring the best and complete segmented description. Besl and Jain [BJ88] followed the former approach by computing the Gaussian-mean curvature sign maps and selecting only the best patches as seeds. Their procedure involves smoothing the image and computing second-order properties in the local neighborhood of every pixel which is noise sensitive and computationally expensive. Also, uniform smoothing has a major disadvantage of altering the underlying surface at the discontinuities, especially the  $C_1$  (surface normal) type, which are smoothed out to form high curvature continuous patches.

To avoid the problems related to smoothing and local curvature computation, we decided to place fixed size (7x7) seed regions in the image in a grid-like pattern of non-overlapping windows. The seed regions are accepted based on a global coherence measure (step 2 of the segmentation algorithm), which is the global chi-square error of the first-order least squares fit. This should be less than a specified threshold. This constraint ensures that the behavior of the seed is acceptable for the current extent of the region, and that it is not placed on a discontinuity. However, it does not guarantee that a seed accepted for further growth will always grow into an acceptable region, since the global coherence measure can be satisfied on the low strength  $C_0$  and  $C_1$  discontinuities. It is possible to incorporate a planarity check that analyzes the distribution of the residuals to better constrain the seed selection. At present we are not performing such a check since it adds substantial computational burden, and more importantly, because our method is not sensitive to bad starting regions. Such seeds result in regions with high error that are better explained by other well-behaved regions. As explained in a later section, using our model selection procedure it is trivial to discard such regions early in the model recovery process. For now, we limit ourselves to the discussion of model recovery (region growing) as an independent process and outline the complete algorithm :

1. Place 7x7 seeds in a grid-like pattern. If the current attempt of placing a seed region on a window is unsuccessful, then the next attempt is made in a 5x5 overlapping window.
2. For each seed the model-recovery procedure explained in § 3.2 is invoked. The schematic

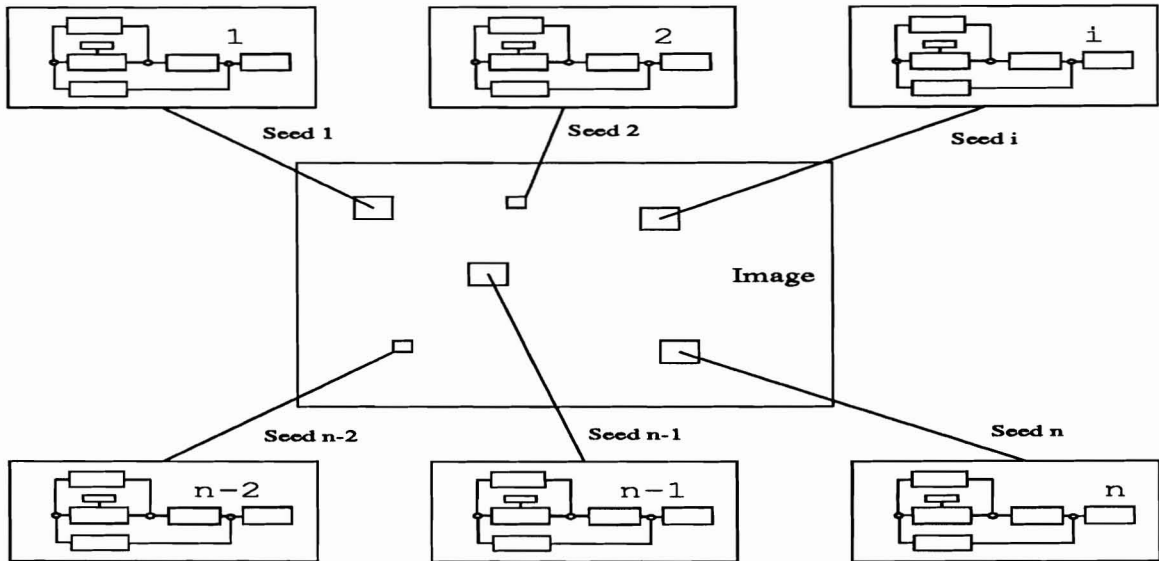


Figure 5: A schematic representation of the region growing process.

diagram is shown in figure 5, where the individual blocks represent the iterative regression schema (figure 2).

Let  $\mathcal{D}_i^{(s)}$  denotes the set of points which are used for the estimation of the parameters  $\mathbf{a}_i^{(s)}$  of the  $i$ -th model at the  $s$ -th iteration. Notice that the regions can overlap partially or completely, so  $(\mathcal{D}_i^{(s)} \cap \mathcal{D}_j^{(s)}) \neq \emptyset$  in general.

The final output of the above algorithm are all the recovered models that are potential candidates for the final description of the data. Selection of the models is achieved through a maximization of quadratic Boolean problem described in the following section.

## 4 Model Selection

After all possible models are recovered we need an efficient procedure for selecting the best description of the image. The growing procedure, as described in the previous section, outputs many different regions of which many are partially or totally overlapped. Intuitively, the method should select the models so that :

- the number of selected models is as small as possible,
- the size of each model (i.e, the cardinality of  $\mathcal{D}$ ) is as large as possible,
- the error measure between the original data and the recovered models is small.

Perhaps the closest in spirit to our approach to model selection is the one used by Pentland [Pen90]. However, there are at least two major differences.

- The objective (saving) function is different since we deal with complex models and not binary silhouettes. This gives us the opportunity to give more preference to a particular description; for example the one which describes more points, or has smaller error, or has lower order model.
- The objective function is not solved by the *continuation method*, where it is not clear how to precisely adjust the steps of the scale parameter. We have developed a faster algorithm whose computational complexity is proportional to the number of selected models and thus drastically speeds up the selection procedure.

#### 4.1 Objective function

Let us first analyze the objective function for one particular model describing the underlying data. The objective function  $F$  is a weighted linear combination of the following terms:

- **Benefit:** Number of points ( $n_i = |D_i|$ ) that are described by the  $i$ -th model.
- **Cost:** Error measure  $\xi_i = \chi_i^2$  of the  $i$ -th model.
- **Cost:** Number of parameters ( $N_i$ ) that are needed to specify the particular model.

$$F(m_i) = K_1 n_i - K_2 \xi_i - K_3 N_i \quad (26)$$

where  $F(m_i)$  is the objective function that we want to maximize and is the function of the model  $m_i$ .  $n_i$  corresponds to the number of points that are explained by the model  $m_i$ .  $\xi_i$  is the error measure between the model  $m_i$  and the data.  $N_i$  denotes the number of parameters for a particular model, which depends on the order of bi-variate polynomial that models the data.  $K_1, K_2, K_3$  are weights which can be adjusted in order to give more preference to a particular description; for example the one which describes more points, or has smaller error, or has a lower order model.

Since many of the models overlap completely or partially, we have to design an objective function that takes into account the interaction between different models. As in Pentland [Pen90], we consider only the pairwise overlaps in the final solution.

The objective function to be maximized for the selection of the “best” description for multiple models has the following form :

$$F(\tilde{\mathbf{m}}) = \begin{bmatrix} \tilde{m}_1 & \dots & \tilde{m}_i & \dots & \tilde{m}_M \end{bmatrix} \begin{bmatrix} c_{11} & \dots & c_{1i} & \dots & c_{1M} \\ \vdots & & \vdots & & \vdots \\ c_{i1} & \dots & c_{ii} & \dots & c_{iM} \\ \vdots & & \vdots & & \vdots \\ c_{M1} & \dots & c_{Mi} & \dots & c_{MM} \end{bmatrix} \begin{bmatrix} \tilde{m}_1 \\ \vdots \\ \tilde{m}_i \\ \vdots \\ \tilde{m}_M \end{bmatrix} \quad (27)$$

$$F(\tilde{\mathbf{m}}) = [\tilde{\mathbf{m}}]^\top [\mathbf{Q}] [\tilde{\mathbf{m}}] \quad (28)$$

where  $\tilde{m}_i$  is a function of the presence of the model  $m_i$ , having unit value for the presence and 0 for the absence of the model in the final description. Diagonal terms express the cost-benefit value for a particular model  $m_i$ :

$$c_{ii} = K_1 n_i - K_2 \xi_i - K_3 N_i \quad (29)$$

Off-diagonal terms handle the interaction between the overlapping models:

$$c_{ij} = (-K_1 \Gamma(m_i, m_j) + K_2 \xi_{i,j})/2 \quad (30)$$

where  $\Gamma(m_i, m_j) = |\mathcal{D}_i \cap \mathcal{D}_j|$  is the number of points that are explained by both models.  $\xi_{i,j}$  corrects the diagonal error terms in case that both models are selected. In the intersection area where both models cover the data the smaller error is taken. The term  $\xi_{i,j}$  is

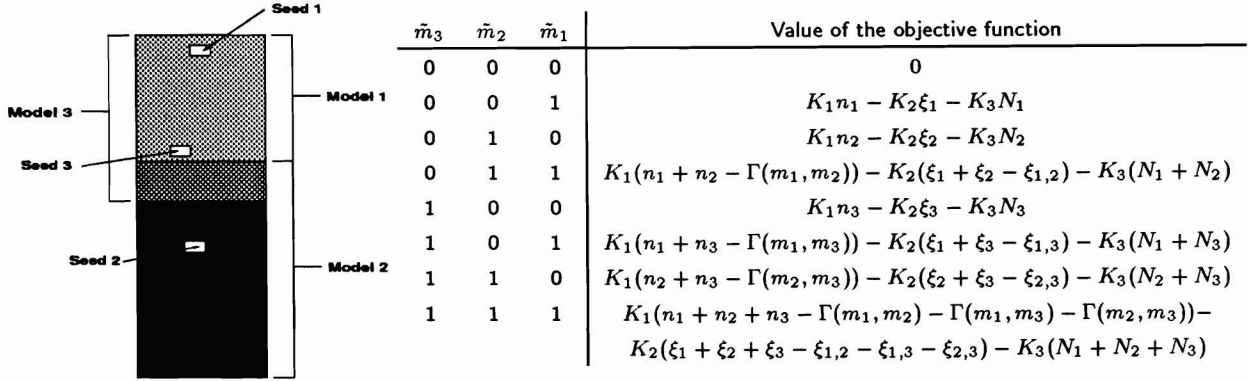
$$\xi_{i,j} = \xi_i + \xi_j - \frac{\sum_{n_i} d_i^2 + \sum_{n_j} d_j^2 - \max(\sum_{\Gamma(m_i, m_j)} d_i^2, \sum_{\Gamma(m_i, m_j)} d_j^2)}{n_i + n_j - \Gamma(m_i, m_j)} \quad (31)$$

$K_1, K_2, K_3$  are weights which can be adjusted in order to take into account the signal-to-noise ratio of the image or to express a preference for a particular type of description.  $K_1$ , which weights the number of points in the model is set to unity and  $K_2, K_3$  are set relative to it.

Notice that the matrix is symmetric. Depending on the overlap of models, the matrix  $\mathbf{Q}$  can be sparse or banded, which can be used to reduce the computations needed to calculate the value of  $F(\tilde{\mathbf{m}})$ .

We would like to emphasize that in contrast with some other approaches, the models that are wholly contained within the bigger models are not a priori discarded but are passed to the selection procedure. The following example dispels the misconception that bigger model always provides a better description.

**An Example:** Table 1 shows three models where model  $m_1$  is wholly contained in the model  $m_3$ , and  $m_2$  and  $m_3$  overlap partially. This situation can occur quite often in real images when



The diagram shows three overlapping regions: Model 1 (top, hatched), Model 2 (middle, dotted), and Model 3 (bottom, solid black). Seeds are placed at the top of each region: Seed 1 in Model 1, Seed 2 in Model 2, and Seed 3 in Model 3. The table below lists the objective function values for all combinations of model presence (indicated by  $\tilde{m}_i$ ).

$\tilde{m}_3$	$\tilde{m}_2$	$\tilde{m}_1$	Value of the objective function
0	0	0	0
0	0	1	$K_1 n_1 - K_2 \xi_1 - K_3 N_1$
0	1	0	$K_1 n_2 - K_2 \xi_2 - K_3 N_2$
0	1	1	$K_1(n_1 + n_2 - \Gamma(m_1, m_2)) - K_2(\xi_1 + \xi_2 - \xi_{1,2}) - K_3(N_1 + N_2)$
1	0	0	$K_1 n_3 - K_2 \xi_3 - K_3 N_3$
1	0	1	$K_1(n_1 + n_3 - \Gamma(m_1, m_3)) - K_2(\xi_1 + \xi_3 - \xi_{1,3}) - K_3(N_1 + N_3)$
1	1	0	$K_1(n_2 + n_3 - \Gamma(m_2, m_3)) - K_2(\xi_2 + \xi_3 - \xi_{2,3}) - K_3(N_2 + N_3)$
1	1	1	$K_1(n_1 + n_2 + n_3 - \Gamma(m_1, m_2) - \Gamma(m_1, m_3) - \Gamma(m_2, m_3)) - K_2(\xi_1 + \xi_2 + \xi_3 - \xi_{1,2} - \xi_{1,3} - \xi_{2,3}) - K_3(N_1 + N_2 + N_3)$

Table 1: An Example: Three overlapped regions and the corresponding objective function table.

the seed of the model (in this case  $m_3$ ) is placed too close to the neighboring region which causes an overflow in the initial stages (initial iterations) of the model recovery process. The values of the objective function for all possible combinations are shown in the table. Since pointwise description is too costly for a significant number of data, at least two models have to be selected to describe the underlying area. So let us concentrate only on two solutions, where  $\tilde{\mathbf{m}} = (1, 1, 0)$  and  $\tilde{\mathbf{m}} = (0, 1, 1)$ . Let us suppose that the models  $m_1$  and  $m_3$  are of the same order. Since  $(n_1 + n_2) = (n_2 + n_3 - \Gamma(m_2, m_3))$  the final decision depends on the error terms. If  $\xi_{1,2} < \xi_{2,3}$  the combination of models  $\tilde{\mathbf{m}} = (1, 1, 0)$  is selected.

## 4.2 Optimizing the objective function - the Algorithm

The variables  $\tilde{m}_i$  are Boolean and denote the presence or the absence of a model in the final description. Since the function  $F(\tilde{\mathbf{m}})$  is quadratic, the problem is known as Boolean quadratic problem. Since the objective function is non-convex, the only way to determine the models which maximize the value of the objective function  $F(\tilde{\mathbf{m}})$  is to calculate the value for all the  $2^M$  possible vectors  $\tilde{\mathbf{m}}$  and choose the one which gives the highest value. This algorithm is exponential in the number of models and thus computationally infeasible. Several different approaches have been proposed to solve the problem faster. Pentland [Pen90] devised a *continuation* method where he weights the negative diagonal terms in the matrix  $\mathbf{Q}$  by a factor  $k_3$  which forces the matrix to become diagonally dominant and thus negative definite. The method is to first solve using a very large value of  $k_3$ , and then, using the previous solution as a starting point, progressively resolve using smaller and smaller values of  $k_3$ , until the solution is reached. He does not mention how the factor  $k_3$ , which can be considered as a scale factor at which parts are recovered, is decreased. It is clear that there is no guarantee that the global maximum will be found, but the reported experimental results show that in most cases the algorithm performs well and gives the expected

results.

We implemented this method using a direct descent algorithm. We observed that if  $k_3$  is decreased in large steps, the solution gets stuck in a local maximum which is significantly lower than the solution reached by decreasing the parameter almost continuously. The computational complexity of the method is proportional to:

$$(\text{no. of steps of } k_3) \times M \times (\text{evaluation of the matrix } \mathbf{Q}) \quad (32)$$

where  $M$  is the number of models involved in a selection process. In evaluating the matrix  $\mathbf{Q}$  we exploited the fact that the matrix can be sparse and banded. Since the computational complexity of the method depends on  $k_3$ -steps, there is an obvious trade-off between the accuracy of the solution and the speed.

While experimenting we made two observations on how the solution develops, which allowed us to design a very efficient algorithm that is based on two assumptions:

- Only one model is chosen at a time,
- Once the model is chosen it cannot be rejected.

### The Algorithm :

Initial values:  $\forall i \tilde{m}_i = 0$ . The initial value of the objective function is 0.

**do**

    old\_value\_of\_objective\_function = new\_value\_of\_objective\_function;

    procedure\_is\_done = **true**;

**for** all the models **do**

        find\_the\_one\_that\_contributes\_the\_most\_to\_the\_value\_of\_objective\_function;

        new\_value\_of\_objective\_function = maximum\_value\_of\_objective\_function;

**end\_for**

**if** (old\_value\_of\_objective\_function < new\_value\_of\_objective\_function)

**Output**( Model that was selected, Value of the objective function);

        procedure\_is\_done = **false**;

**end\_if**

**while** (**not** procedure\_is\_done).

The computational complexity of our method is proportional to:

$$(\text{number of models in the final description}) \times (\text{evaluation of the matrix } Q) \quad (33)$$

The designed algorithm is an excellent compromise between speed and accuracy. Experimental results show that in almost all the cases the algorithm performs well and gives good results both quantitatively and qualitatively. We compared it to the continuation method and except for very small steps of  $k_3$ , where the results were the same, our algorithm selected the models with the higher value of the objective function.

Thus, we now have a complete model recovery procedure that yields all the models in the image, followed by the above model selection procedure which selects the best models according to the global error, order, and the spatial extent of the region. The complete procedure works as shown in Figure 6 (a). As a consequence of the selection process, eventually very few of the regions emerge as acceptable descriptions of the data. However, instead of growing all the regions completely, it is desirable to discard regions as they grow. Also, the computational cost of growing all the regions completely is prohibitive in most cases. These observations suggest incorporating the selection procedure into the recovery procedure to discard redundant and superfluous regions even before they are grown fully. Our final algorithm described in the next section accomplishes this integration.

## 5 Model Recovery and Model Selection

### Combined in a Dynamic Iterative Schema

After describing the two major components of our system, namely, the module for model recovery, and the module for model selection, we now describe how they can be combined in a dynamic way to obtain a fast and efficient method for image segmentation. As explained earlier, to avoid the problems related to smoothing and local curvature computation, seed regions are placed in globally coherent windows everywhere in the image, and models are grown simultaneously and independently for all of them. This way all the regions are grown to their full extent and then selected for the optimal description, as shown in Figure 6a. Since the regions are selected by the optimization procedure after they are fully grown, the resulting segmentation can be claimed to be the best piecewise continuous description of the image. We call this procedure *Recover-then-Select*, for it grows all the regions fully and then prunes them (discards the redundant ones). While the results of this procedure are optimal (we use the word *optimal* to signify the fact that optimization was performed to extract the final regions and that the results are optimal in some sense, and by no means imply that the global maximum was achieved by the procedure), the computational

<i>Feature</i>	<i>Model selection invoked</i>	
	<b>Early</b>	<b>Late</b>
<b>Description</b>	– Less reliable	+ More reliable
<b>Models remaining for further growth</b>	– More	+ Fewer
<b>Processing needed for initial growing</b>	+ Less	– More
<b>Matrix <math>\mathbf{Q}</math></b>	+ Sparse	– Dense

Table 2: Trade-offs in dynamically combining model recovery and model selection procedures.

complexity is prohibitive because all the regions are grown to the maximum before the best among them are chosen. Surely, there must be a way to discard the regions after a few iterations of region growing using the same model selection procedure (with conservative weights, to accept multiple descriptions rather than losing a good one about which we are not confident yet), so that only the active models are grown further. This procedure, performed continuously in a loop (see figure 6b) is called *Recover-and-Select*.

The incorporation of Recover-and-Select paradigm opens up a number of possibilities as to the control of the region-growing procedure. It has the feature of growing only well-behaved regions (in terms of convergence, error, number of compatible points) while at the same time lowering the computational complexity of the procedure. There is a clear trade-off in combining the model recovery module with the module for the selection of the optimal current description. The more the regions are grown, the more reliable is the description they give. But the initial growing is computationally expensive and also results in a less sparse matrix  $\mathbf{Q}$ . However, this reduces further processing since fewer models are selected for further growth. On the other hand, if the growing process is interrupted by the selection of currently optimal models at the early stages, the complexity of the early processing is decreased and the matrix  $\mathbf{Q}$  is sparse due to less overlapping. In this case fewer models are rejected, increasing the complexity of the further processing. These tradeoffs are summarized in Table 2. By properly balancing the two trade-offs a computationally efficient algorithm is obtained.

During the very first iteration, model selection is invoked after all the regions are grown only for a distance of  $k = 20$  pixels (equation 15 in step 3.1 of the segmentation algorithm) from the seed. After that, the restriction on  $k$  is removed, and the model selection procedure is invoked after every iteration of region growing (one iteration of the steps 3.1-3.3). We have found that after the initial selection of regions, depending on the type of surfaces in the image, less than 50% of the regions survive. The model selection weights are kept biased to discard only the completely identical regions. Also, error is weighted more than the order of the regions. Later in the procedure, however,



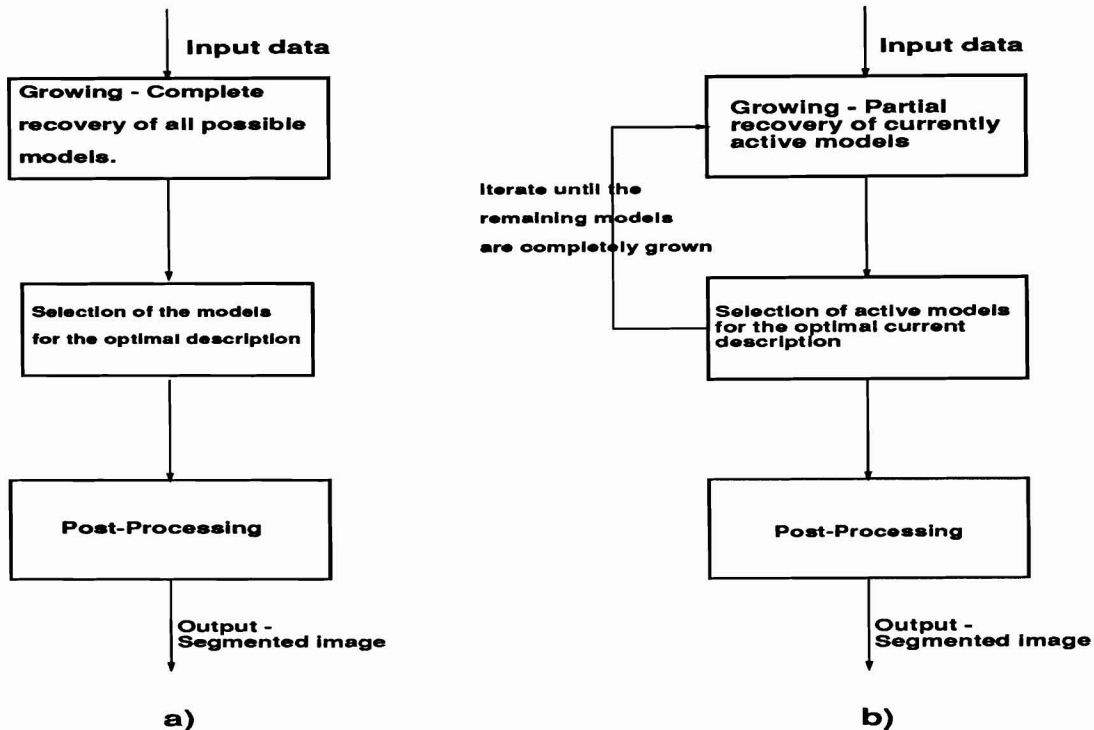


Figure 6: Model recovery and selection: (a) Recover-then-Select (b) Recover-and-Select.

the weights are changed to eliminate duplicate regions describing a patch with considerable overlap and similar global error.

## 6 Post-processing of the Regions

The collection of models obtained by the Recover-and-Select strategy describes the image in terms of primitives with minimum overlap such that all the parts of the image (where seeds were placed) are described by at least one surface. Some of the regions may need refinement of their spatial extent due to the overlaps with neighboring regions. Regions may overlap due to the surface similarities near the border between two patches. It is acceptable in most cases and we make no attempt for further segmentation due to the absence of any additional information.

The second type of overlap is caused due to the compatibility constraint (equation 14). It can result in two cases, shown in figure 8 :

1. Type I: Region A overflows into region B along the intersection curve (region C) due to the lack of topological constraints.
2. Type II: Region A and B overlap near the common boundary.

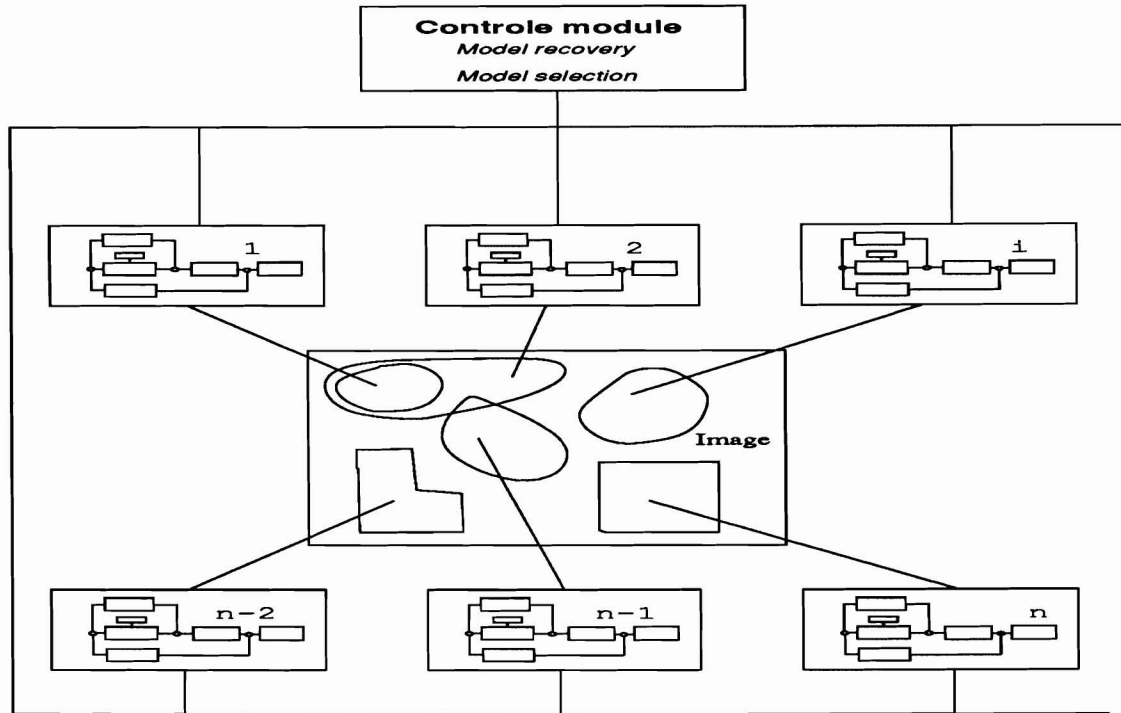


Figure 7: **Recover-and-Select**: A schematic diagram of the closed-loop model recovery and selection process.

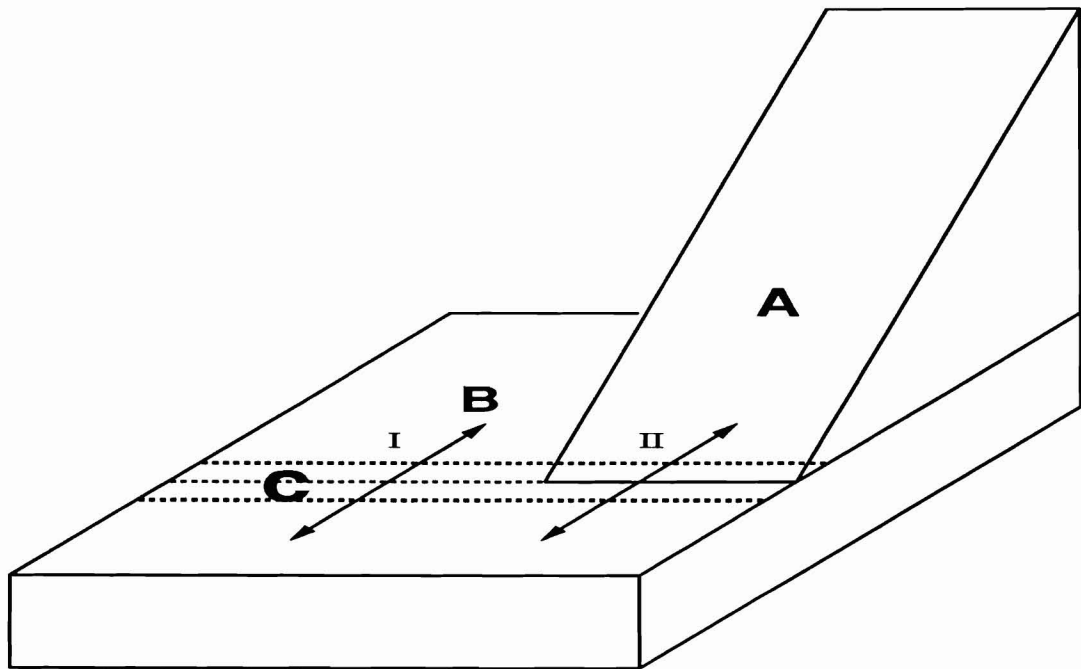


Figure 8: Limitations of the approach due to the lack of topological constraints. They are enforced by the postprocessing stage by following a simple analytical approach.

Both types can be resolved by a systematic procedure. Since the regions are described analytically, it is possible to detect such cases and remove the extraneous points in a clean manner. This is true for all kinds of intersections, including the intersections formed by two second-order regions.

A surface normal check is performed in the overlapped portion of the two regions to determine the type of overlap. If the average angle between surface normals is not significant then there is a smooth overlap between the two regions, otherwise the two patches intersect and need additional processing. We have devised an algorithm to refine the overlapping regions formed at the line of intersection and to assign the points to the right patch. For each pair of overlapping regions the following steps are performed :

1. Determine the pixels  $p_i$  that are on the intersection curve.
2. Compute the normal to the curve at each  $p_i$ .
3. Trace pixels along the normal on both sides of the curve, storing all the traced pixels and looking for the first pixel on both sides that does not belong to the overlapped region.
4. If both end-pixels belong to the same region then assign all the traced pixels along the normal to that region, else if they belong to different regions then distribute the traced pixels between the two regions.

In figure 8, points in C that should only be labeled as belonging to B are assigned to B (case I). In case II, points on the two sides of the intersection curve are distributed between A and B. This procedure has been tested on numerous images with excellent results.

During the post-processing various heuristics can be incorporated to derive the desired high level description. Surfaces curving faster than our second-order model are described as piecewise combination of second-order patches. While the global qualitative description is unique, there is a possibility of a non-unique individual description of the patches that compose the global patch. In other words, more than one kind of decomposition is possible for data that does not exactly match our model, as is the case with any primitive based approach.

## 7 Experimental Results

We tested the Recover-and-Select paradigm on a number of range images of scenes with a combination of different kinds of surfaces. The method is straightforward and is computationally feasible on a sequential machine. All the examples were run on a SUN-4, with the average execution time of less than 2 minutes. The program is twice as fast on the IBM-6000 RISC machine.

Since the pixel values in range images correspond to the real surfaces, it is easy to analyze the results from a geometrical point of view, particularly because our primitives are geometric in nature. Results are shown for seven range images. The images (except the block image) were scanned using a structured lighting laser-scanner with approximately 1mm/pixel spatial resolution and 1.5mm depth resolution. Due to the geometry of the scanner, certain parts of the scene appear as shadow regions (with no data) in the  $2\frac{1}{2}$ D image representation. The compatibility constraint for all the range images was set to 4, which corresponds to a quantization and sensor noise of  $\pm 2$  pixels. The algorithm was run on the raw data without any preprocessing like uniform smoothing. Results are discussed for each image below. All the results coded in gray-levels are grouped such that the top row of the figure (from left to right) shows the original image, its 3-D perspective plot, the reconstructed image from the piecewise continuous segmented patches, and the 3-D plot of the reconstructed image. The images are displayed with the depth value at each pixel from a reference plane appearing larger if the pixel is closer to the camera. The white square in the patch indicates the seed region for that patch. The individual surface patches are displayed in the second row of the figures in the order in which they were selected by the model selection procedure, and are referred to below with their position in the row, counting from left to right.

**Image 1:** This range image consists of a triangular prism and two half-cylinders placed on a box (figure 9). Three planar regions representing the box and the prism and two second-order patches for the cylindrical objects are recovered, as shown in the bottom row of the figure 9. The planar regions on the prism extend along the intersection with the box. Using the procedure described in the previous section, the region of overlap due to intersection of the envelopes was removed and the results are shown in the last three images in the bottom row. The line of intersection between the planes gives the surface-normal ( $C_1$ ) discontinuities.

**The Coffee-mug image:** The convex and concave portions of the body of the cup are recovered as individual second-order patches, as shown in the first two images of the bottom row in figure 10. The handle consists of very curved patches which are modeled piecewise for the given scale (which directly relates to the compatibility constraint). According to the results, the missing parts are better described pixelwise than as parametric patches (due to the scale consideration). It should be noted that the jump ( $C_0$ ) discontinuities are clearly delineated by the neighboring regions.

**Image 2:** The image consists of three cylinders joined to form a globally curved structure. At the regular scale, we obtain three second-order regions (left three images in the bottom row of figure 11, which is the best description of the scene. To demonstrate our claim that the compatibility constraint acts like a scale parameter, we increased its value to 121, (corresponding to noise of  $\pm 11$  pixels), and obtained a single second-order surface describing the “smoothed” data.

**Image 3:** This is a complex image for any surface modeling algorithm, primarily because of

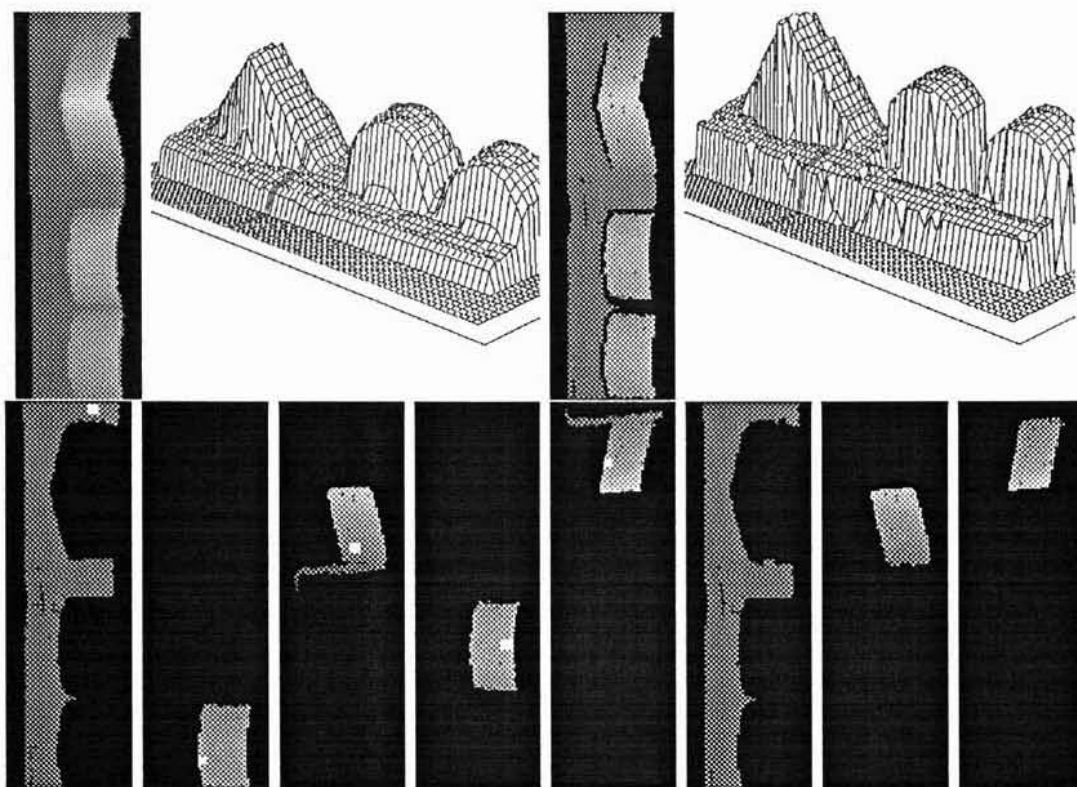


Figure 9: Image 1: The cylindrical surfaces are modeled as bi-quadratic patches.  $C_1$  discontinuities at planar intersections are reliably recovered.

---

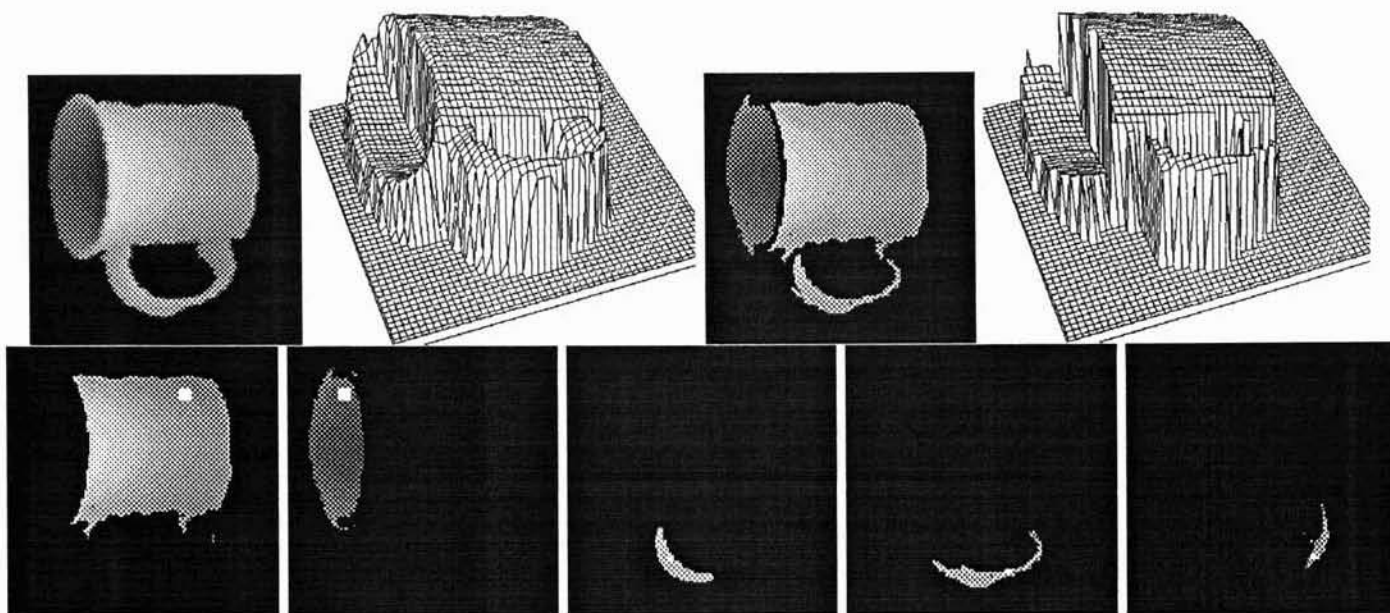


Figure 10: The coffee-mug Image: The highly curved handle is modeled as a combination of the smaller patches.

the smooth boundaries (zero-crossing contours) between the convex and concave patches forming the undulated portion of the scene. The individual patches recovered for the image are shown in the middle row of figure 12, with the first four patches describing the image almost completely. The undulated portion is described as two convex and one concave regions intersecting in the vicinity of the zero-crossing contours. Patch 5 and 7 are the first-order patches flanking the convex second-order patch 3. Adding them to the final description increases the accuracy of description of the convex patch which is curving faster than the bi-quadratic surface. They are selected by putting emphasis on the error term in the model-selection procedure. Patch 6 describes the second-order region that smoothly merges into the planar patch. The merged region is modeled partly by the planar patch and completely by patch 6, which is an acceptable description. One significant result in this example is the approximate detection of the zero-crossing contour by region-growing and not by curvature tracing, which is computationally prohibitive and extremely sensitive to noise. Such a region based description is also useful for qualitative description of the scene in terms of convex, concave and planar patches. The third row in figure 12 shows some of the regions that were rejected during various stages of the Recover-and-Select procedure. In most cases, these regions had bad starting points which passed the seed-selection criterion. It shows that though the seed placement is not perfect, the procedure is robust enough to reject the patches arising out of bad seeds in favor of the patches that are well behaved in terms of the spatial extent, order, and the

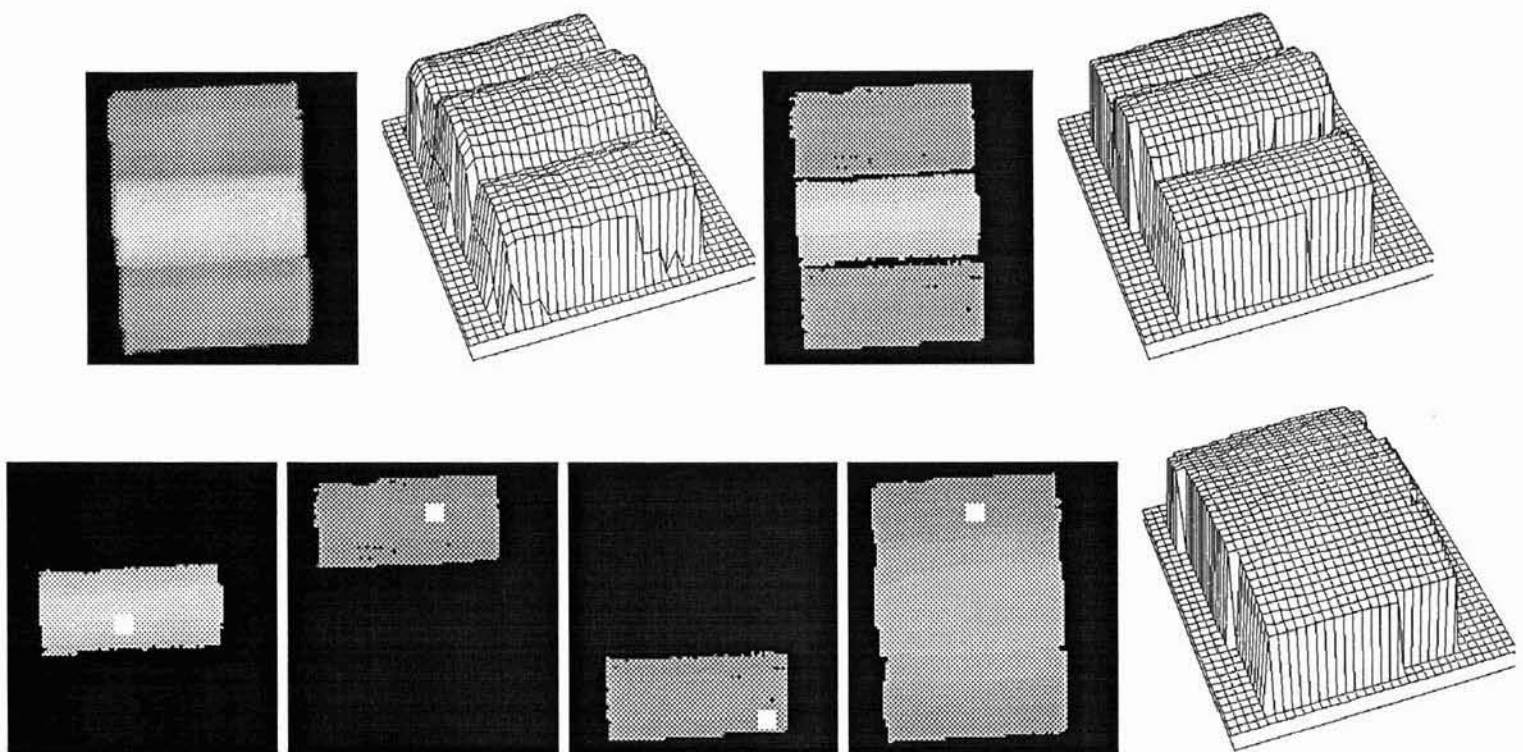


Figure 11: Image 2: Three cylindrical surfaces. Increasing the compatibility-constraint smooths the data to recover one surface.

---

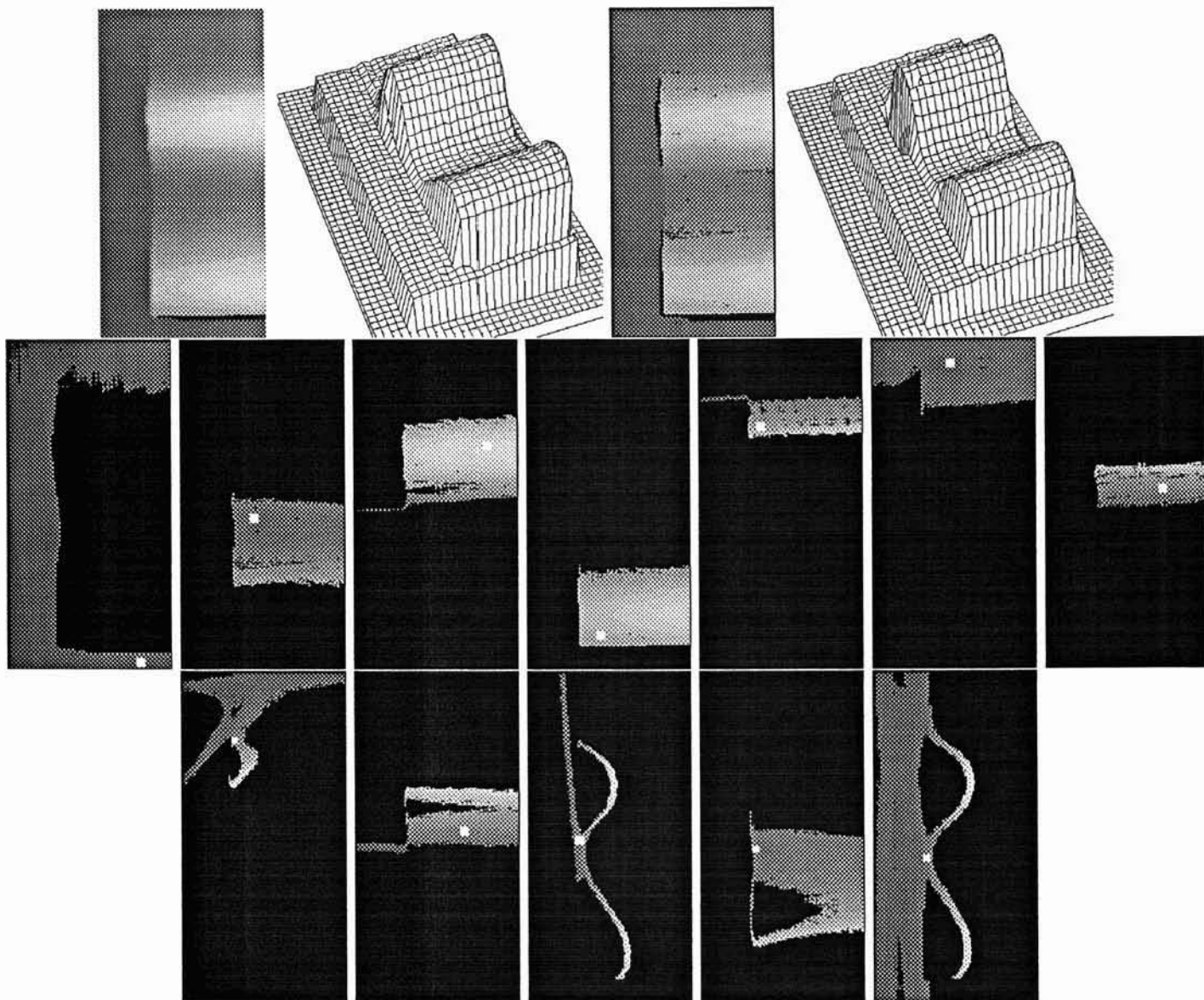


Figure 12: Image 3: Undulated surface smoothly merging into a planar surface. Some of the rejected patches are shown in the bottom row.

---



global error.

**Complex Object:** As mentioned in the introduction, the second-order surface in figure 13 (taken from [Fan88]) is difficult to segment due to the absence of jump or surface normal discontinuities. Our method gives a clean separation of the curved surface from the neighboring planar patches. Such a result is possible only because we search for the best description *everywhere* in the image and allow the models to develop *independently*.

**Block:** This noisy range image of a block (taken from Besl [Bes88]) has both planar and curved patches (figure 14). Due to the noise in the image, higher value of compatibility constraint was used. This has the effect of non-uniform smoothing, where noise points are left out in the final description. The segmentation is as good as the one obtained by Besl using a curvature based approach, but processed at a fraction of the computational cost involved in such elaborate processing.

**NBS object:** This is the range image of a machined object from the National Bureau of Standards. All the surface patches are recovered accurately. This information is useful for initial clustering for automatic or operator-assisted model building from range data.

**Piecewise constant and first-order description of Coffee-mug:** So far we have presented the results of the Recover-and-Select paradigm with models allowed to change up to second-order. Consequently, the final description consists of a combination of piecewise patches up to second-order. Similarly, it is possible to obtain piecewise constant or piecewise first-order descriptions of the scene. Due to the simplicity of our paradigm, it is trivial to restrict the highest allowable order to zero or first-order. It is interesting to note that no other changes need to be made. Starting with the same seed regions, the piecewise constant and piecewise first-order descriptions are obtained for the coffee mug (figure 16). The piecewise constant description is like equidistant contours or planar slices of width determined by the compatibility constraint. The piecewise planar description shows a natural approximation of curved patches by planar patches. The extent of planar patches (along the curvature) is determined by the compatibility constraint.

The above examples show that our method gives acceptable segmentation of objects into patches in most of the cases. The interpretation of these patches is straight forward, with direct applications for surface modeling and detection of both, discontinuities and smooth boundaries. The primitives do not explicitly contain information about the boundaries of the segmented data. This is advantageous, especially in the case of occlusion, but requires a secondary process to determine the actual boundaries.

During the course of experimentation, we observed that the method degrades gracefully if the assumptions which are determined by the choice of primitives are not met. For example, a geometric object like torus is described by numerous bi-quadratic patches which do not result in a simple description, signaling that different kind of primitives should be invoked.

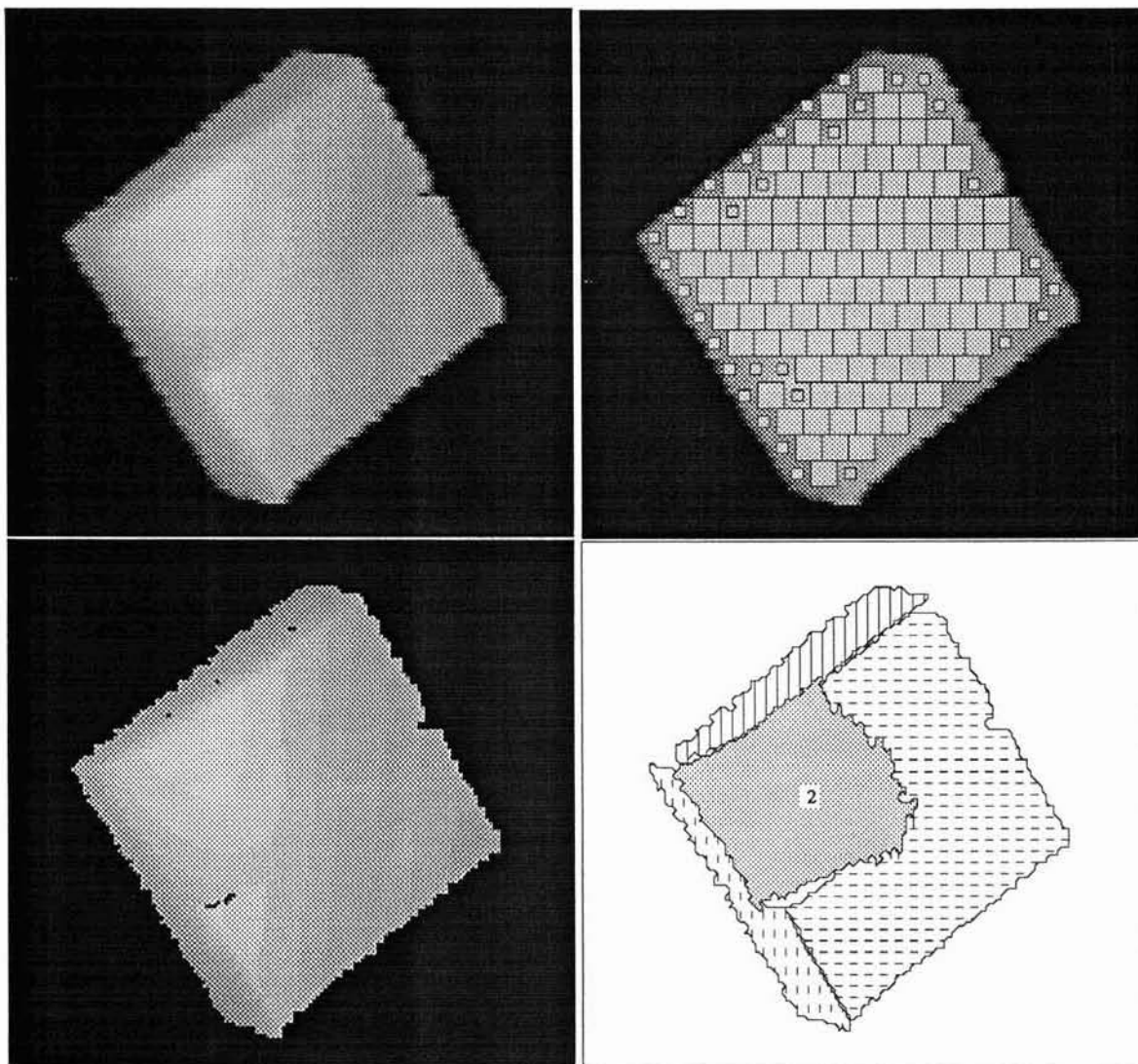


Figure 13: The Complex object: The second-order patch (2) is segmented from the planar patches.

---

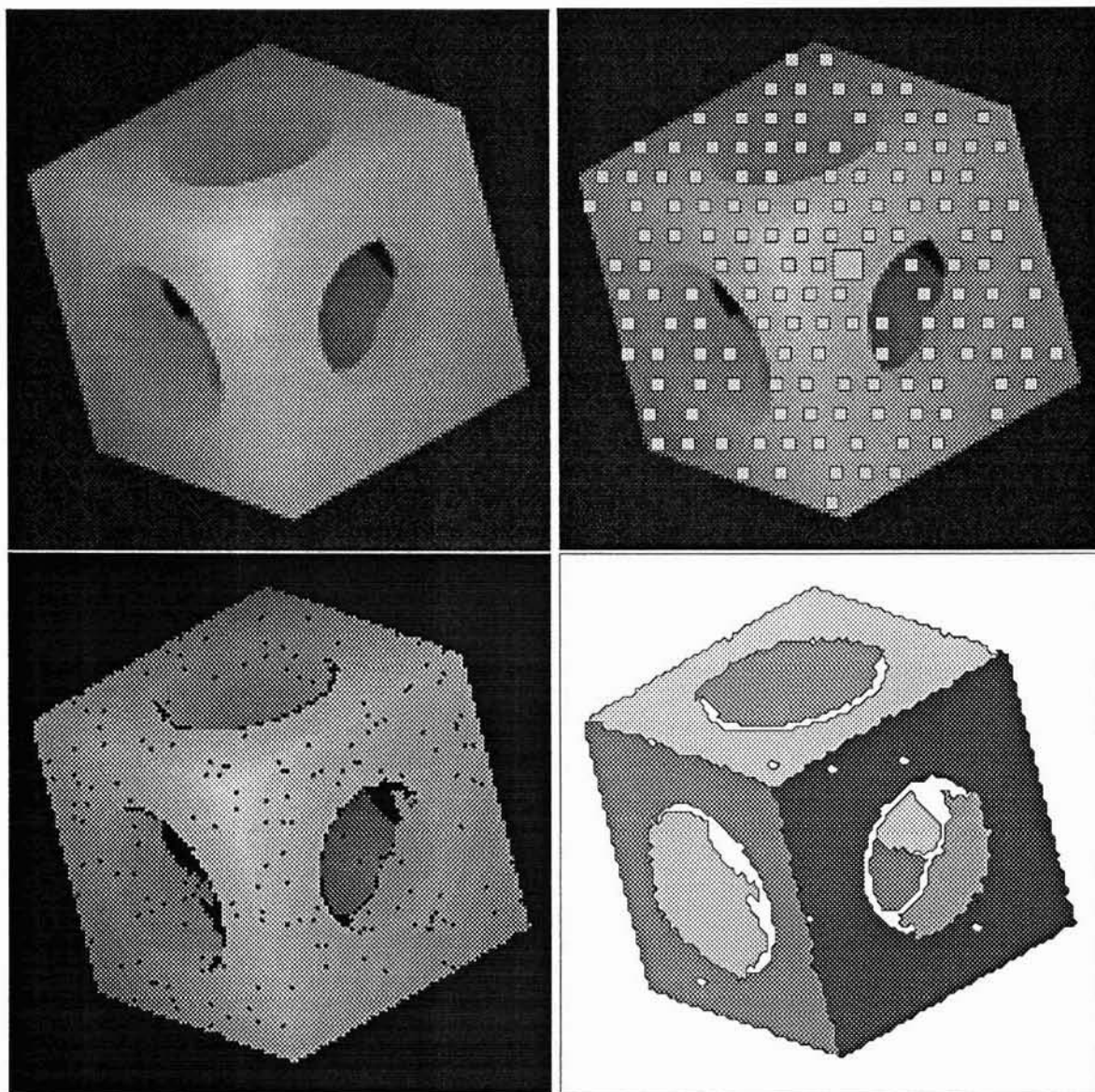


Figure 14: Block: Regions are recovered in the noisy image.

---

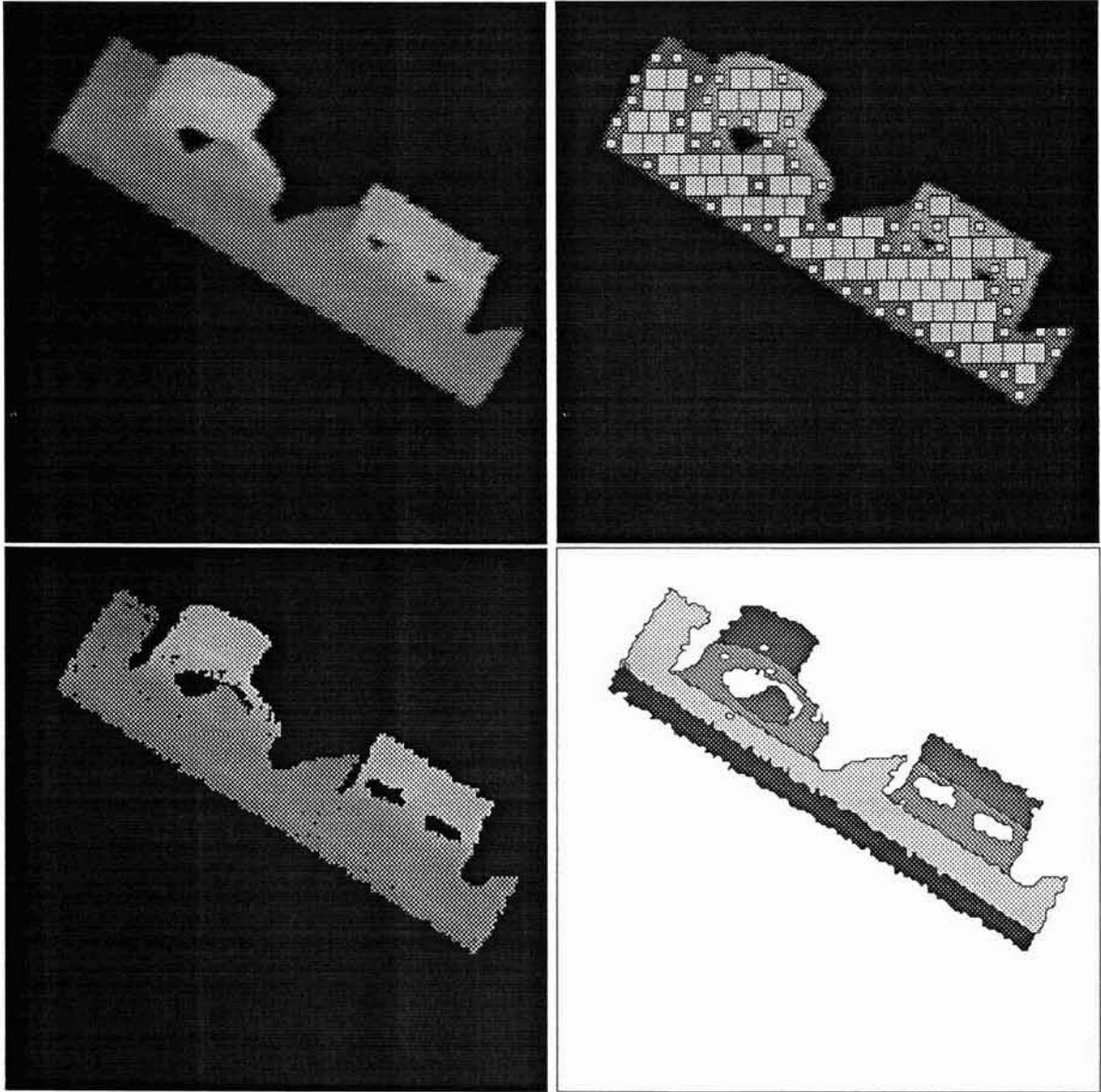


Figure 15: NBS Object: Surface patches are reliably recovered.

---

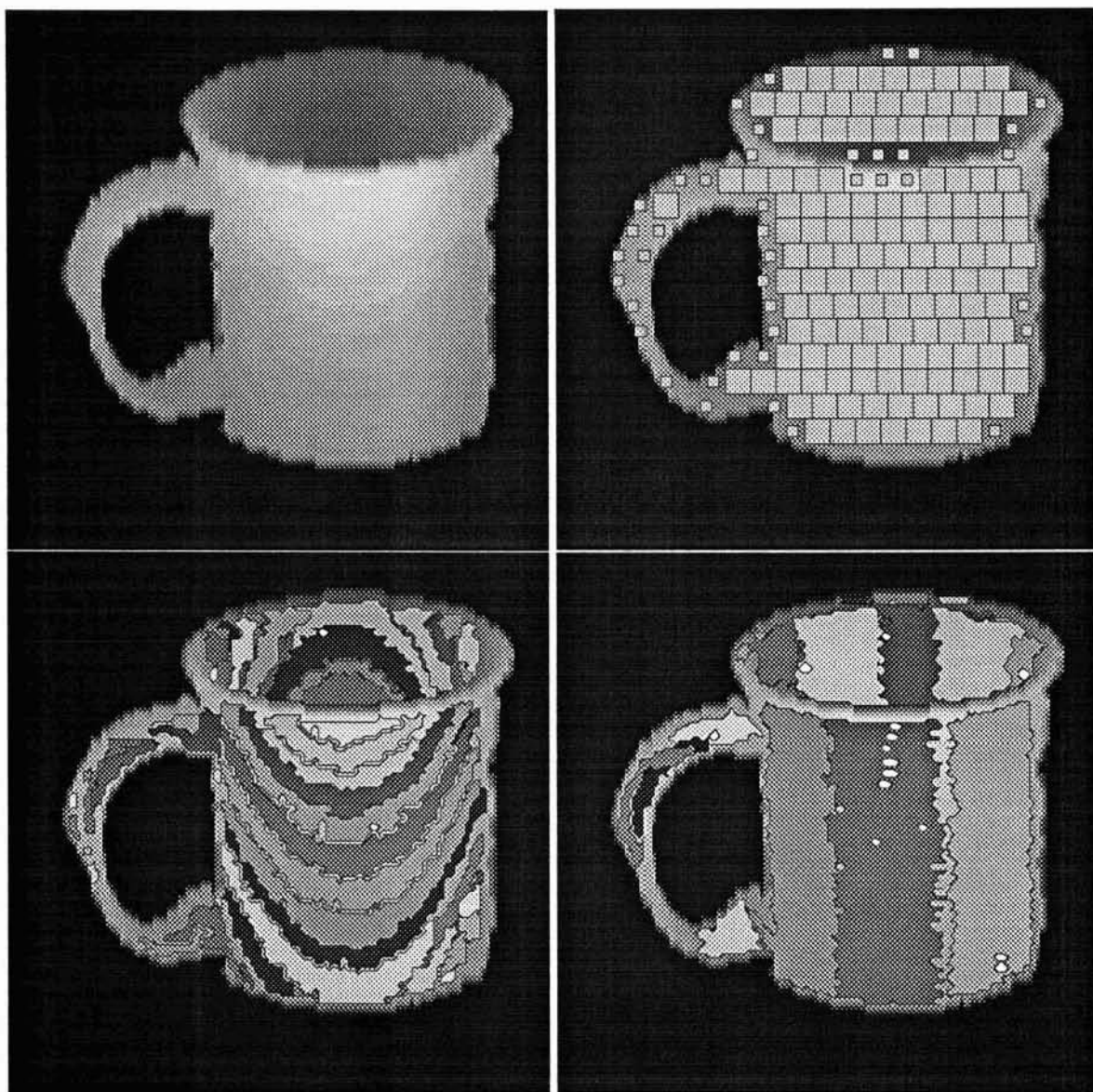


Figure 16: Coffee-Mug: Piecewise constant (left) and planar patches (right).

---

Our future work which is directed toward overcoming some of the problems is discussed in the final section.

## 8 Conclusions and Work in Progress

We believe that our segmentation schema is a tool that will prove useful in many tasks of early vision. We described two procedures where it is clearly shown that the whole is greater than the sum of its parts (synergism). The iterative approach combining data classification and model fitting shows that segmentation and modeling are not two independent procedures but have to be integrated. The procedure which dynamically combines model recovery with model selection proves to be much more efficient than applying the modules one after another.

Another important conclusion that we could draw from our work is that reliable segmentation can only be achieved by considering many competitive solutions and choosing those which reveal some kind of structure in terms of underlying models. We agree with Pentland [Pen90] that fine-tuning of feature detectors does not lead to reliable segmentation, because of the variability of the input data. Initial local estimates, no matter how good they are, do not necessarily lead to a good result, and more global information is needed. Optimization that is performed on the level of primitives rather than on a pixel level not only improves the performance enormously in terms of computational complexity but also gives more reliable results.

An obvious extension is to use the proposed schema to segment intensity images. In intensity images, though smooth boundaries and  $C_1$  discontinuities are not that important, the description can be used to model shading or just to provide a region-based parametric description that effectively achieves data-reduction and generates a middle-level description of image regions for further reasoning in terms of analytical regions instead of coping with unstructured data at pixel level.

The intensity image in figure reffig:ball shows a ball with a circular hole through it. For the compatibility constraint of 49 (noise of  $\pm 7$  pixels), the ball was described as the convex combination of two second-order patches (1 and 2). The reconstructed image is shown in the figure 17, which is a good approximation of the original. By increasing the compatibility constraint by one pixel, almost the entire ball was recovered as one second-order patch (patch 3). Patches 4 and 5 are two of the more than 100 regions rejected during the Recover-and-Select procedure.

An important, implicit feature of our schema is that it automatically selects the *domain of applicability*, which means that it models only the data that can be expressed in terms of predefined models. For textured regions (relative to the scale determined by the size of seed region), the schema that we describe is insufficient. These regions are left out in processing, since no seeds are placed there. They can be further analyzed in the same paradigm, but using different models.

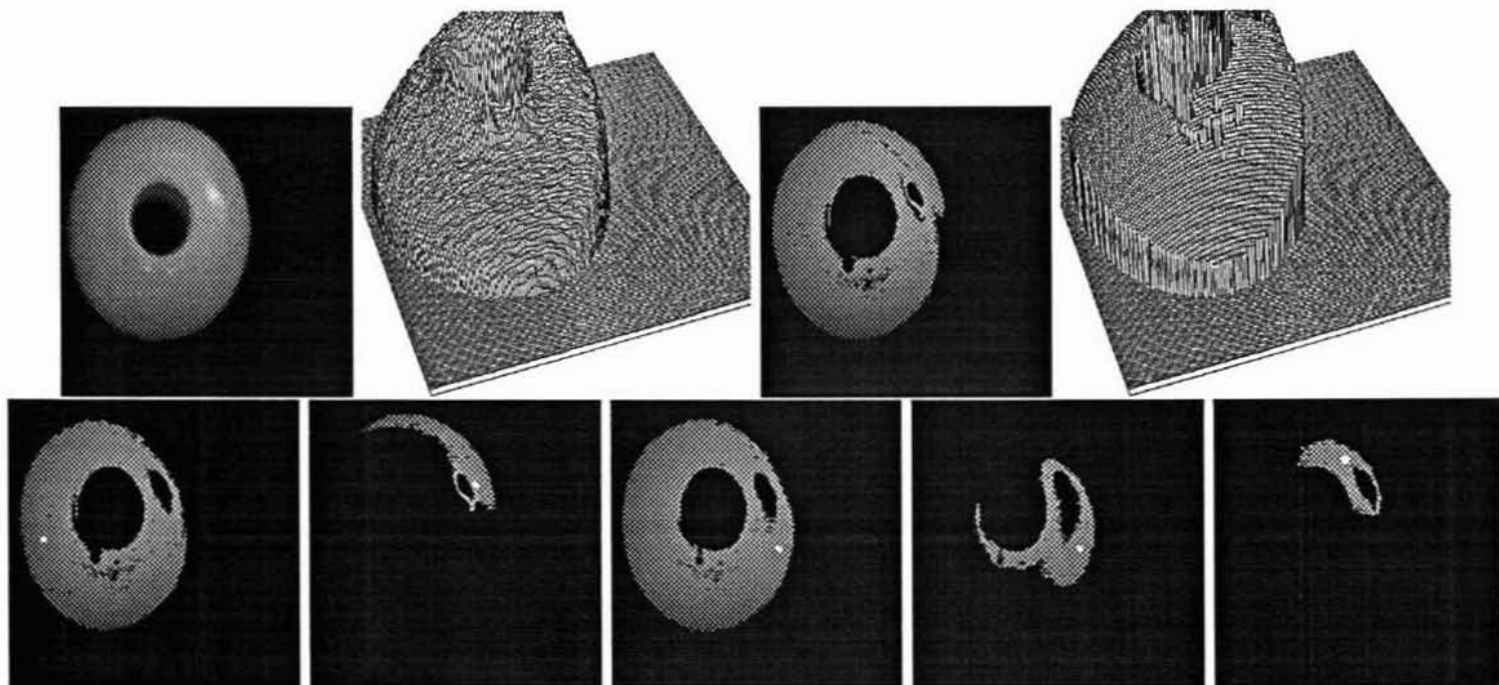


Figure 17: Intensity image: A ball with a hole.

Small regions which have a size that is insufficient to provide enough of information for a reliable estimate can be analyzed on a different scale after new information is acquired.

We plan to apply the same principles to other problems in low-level vision. We have already obtained some preliminary results in detecting curve-structures in the image. The images in figure 18 show the edge elements obtained by multiresolution wavelet transform using different thresholds. The algorithm, similar in spirit, but with some interesting modifications, has detected in all images the same “most” significant structure (the closed outline curve) given by the objective function. This means that the algorithm can perform well in cases of noisy and incomplete information.

The surface segmentation described in this paper will become an integral part of a larger schema for 3-D segmentation of range data using volumetric, surface and curve models [GB90].

Another interesting challenge would be to use the proposed schema as a framework for combining different sources of information. There are two places where we see that the integration of the information could take place. The first one is at the time of decision making in the iterative regression procedure and the second one at the phase of selecting optimal descriptions where different modules could influence one another through the objective function. A module for detecting curve structures in the image would add an important (topological) dimension to the existing schema.

The *active vision* paradigm [Baj88] could use the hierarchy of models that is produced by the selection procedure as a *focus of attention*.

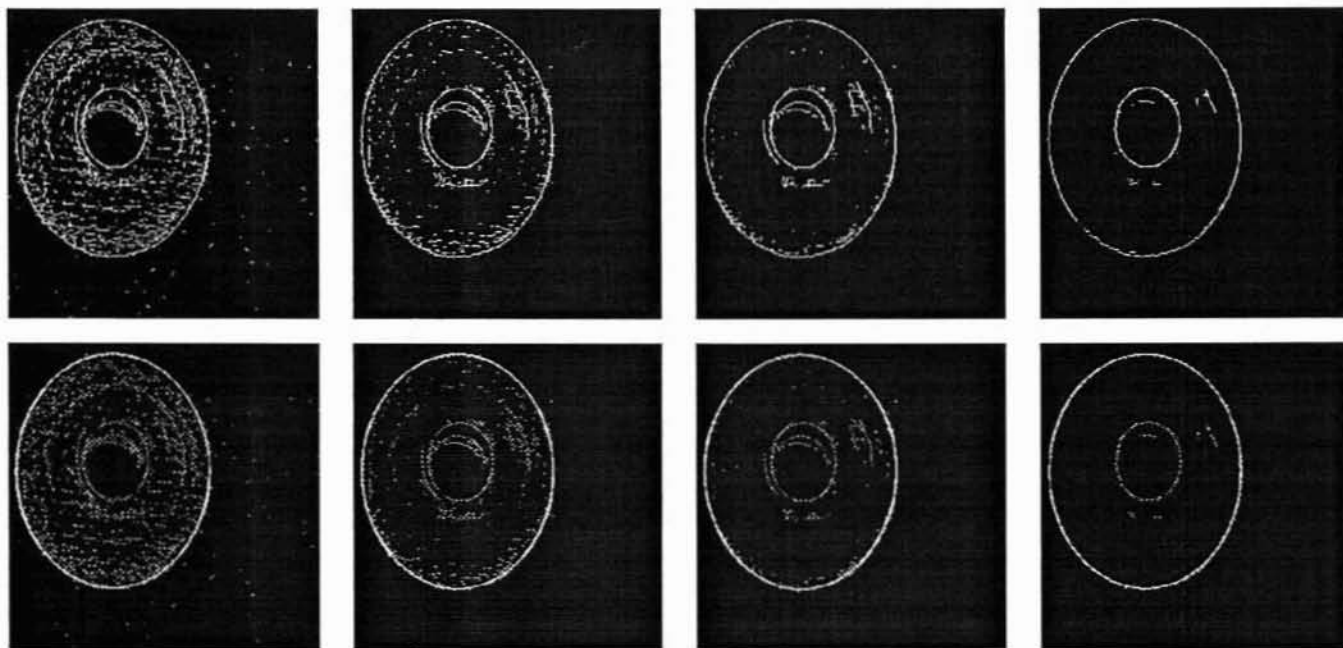


Figure 18: **Structure in edges:** Top row: Edge images using different thresholds. Bottom row: The most significant curve recovered from the corresponding edge image superimposed on the original data.

---

The process of individual region growing as well as the complete model recovery procedure is parallel. Although the procedure is computationally feasible on a sequential machine, like SUN-4, in order to exploit the inherent parallelism, the procedure has been implemented on the Connection Machine.

## Acknowledgements

The authors wish to thank Luca Bogoni for helpful suggestions regarding the implementation of the procedure, and Helen Anderson, Ulf Cahn von Seelen, and Dr. Max Mintz for comments on an earlier draft of the paper. This research was supported in part by : AFOSR Grants 88 0244, AFOSR 88-0296; Army/DAAL 03-89-C-0031PRI; NSF Grants CISE/CDA 88-22719, IRI 89-06770; ARPA Grant N0014-88-K-0630; United States Postal Service and Dupont Corporation.



## References

- [AB86] H. Asada and Michael Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):2–14, January 1986.
- [Baj88] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8), August 1988.
- [Bes88] P. J. Besl. *Surfaces in Range Image Understanding*. Springer-Verlag, 1988.
- [BF81] R. C. Bolles and M. A. Fischler. A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.
- [Bin82] Thomas O. Binford. Survey of model-based image analysis systems. *International Journal of Robotics Research*, 1(1):18–64, Spring 1982.
- [BJ85] Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1), March 1985.
- [BJ86] P. J. Besl and Ramesh C. Jain. Invariant surface characteristics for three dimensional object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33(1):33–88, 1986.
- [BJ88] Paul J. Besl and Ramesh C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), March 1988.
- [BKM86] R. Bajcsy, E. Krotkov, and M. Mintz. *Models of Errors and Mistakes in Machine Perception*. Technical Report MS-CIS-86-26, University of Pennsylvania, 1986.
- [BPYA85] M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing surfaces. *Computer Vision, Graphics, and Image Processing*, 32(1):1–28, 1985.
- [Bra83] M. Brady. *Human and Machine Vision*, chapter Criteria for representation of shape. Academic Press, Orlando,FL, 1983.
- [BSG90] R. Bajcsy, F. Solina, and A. Gupta. *Analysis and Interpretation of Range Images*, chapter Segmentation Versus Object Representation – Are they Separable? Springer-Verlag, New York, 1990.
- [BZ87] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, Massachusetts, 1987.
- [Che89] D. S. Chen. A data-driven intermediate level feature extraction algorithm. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 11(7):749–758, July 1989.

- [Fan88] T. J. Fan. *Describing and Recognizing 3-D Objects Using Surface Properties*. PhD thesis, University of Southern California, August 1988. Technical report IRIS 237.
- [FH86] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-d objects. *The International Journal of Robotics Research*, 5(3):27–52, Fall 1986.
- [GB90] A. Gupta and R. Bajcsy. Part description and segmentation using contour, surface, and volumetric primitives. In *SPIE Conf. on Sensing and Reconstruction of 3D Objects and Scenes*, Santa Clara, CA, February 1990.
- [GL89] G. D. Godin and M. D. Levine. Structured edge map of curved objects in a range image. In *Proceedings of Computer Vision and Pattern Recognition Conference*, 1989.
- [HJ87] R. Hoffman and A. K. Jain. Segmentation and classification of range images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 9, September 1987.
- [HP74] S. L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In *Proc. 2nd International Joint Conference Pattern Recognition*, pages 424–433, 1974.
- [HT85] J. J. Hopfield and D. W. Tank. 'neural' computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.
- [Lec89] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.
- [LT90] P. Liang and J. S. Todhunter. Representation and recognition of surface shapes in range images: a differential geometry approach. *Computer Vision, Graphics, and Image Processing*, 52(1), October 1990.
- [Mar82] D. Marr. *Vision*. Freeman, San Francisco, 1982.
- [MS85] D. Mumford and J. Shah. Boundary detection by minimizing functionals, i. In *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, pages 22–26, San Francisco, CA, 1985.
- [Nac84] Lee R. Nackman. Two-dimensional critical point configuration graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):442–449, July 1984.
- [PB84] J. Ponce and M. Brady. Toward a surface primal sketch. In *Proc. of IEEE Conference on Robotics and Automation*, pages 420–425, March 1984.

- [Pen90] A. P. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4:107–126, 1990.
- [PTK85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.
- [Seb77] G. A. F. Seber. *Linear Regression Analysis*. Wiley, New York, 1977. Chapter 11.
- [SK85] D. R. Smith and Takeo Kanade. Autonomous scene description with range imagery. *Computer Vision, Graphics and Image Processing*, 31(3):322–334, September 1985.
- [SZ88] P. T. Sander and S.W. Zucker. *Inferring Differential Structure from 3-D Images: Smooth Cross Sections of Fibre Bundles*. Technical Report TR-CIM-88-6, McGill University, 1988.
- [Ter86] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4), 1986.
- [Zuc76] S. W. Zucker. Region growing: childhood and adolescence. *Computer Graphics Image Processing*, 382–399, 1976.