



University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

June 1991

Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models

Alok Gupta
University of Pennsylvania

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

Alok Gupta, "Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models", . June 1991.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-45.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/423
For more information, please contact repository@pobox.upenn.edu.

Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models

Abstract

The problem of part definition, description, and decomposition is central to the shape recognition systems. In this dissertation, we develop an integrated framework for segmenting dense range data of complex 3-D scenes into their constituent parts in terms of surface and volumetric primitives. Unlike previous approaches, we use geometric properties derived from surface, as well as volumetric models, to *recover* structured descriptions of complex objects without *a priori* domain knowledge or stored models.

To recover shape descriptions, we use *bi-quadric* models for surface representation and *superquadric* models for object-centered volumetric representation. The surface segmentation uses a novel approach of *searching* for the best piecewise description of the image in terms of bi-quadric ($z = f(x,y)$) models. It is used to generate the region adjacency graphs, to localize surface discontinuities, and to derive global shape properties of the surfaces. A superquadric model is recovered for the entire data set and residuals are computed to evaluate the fit. The goodness-of-fit value based on the inside-outside function, and the mean-squared distance of data from the model provide quantitative evaluation of the model. The qualitative evaluation criteria check the local consistency of the model in the form of residual maps of *overestimated* and *underestimated* data regions.

The control structure invokes the models in a systematic manner, evaluates the intermediate descriptions, and integrates them to achieve final segmentation. Superquadric and bi-quadric models are recovered in parallel to incorporate the best of the coarse-to-fine and fine-to-coarse segmentation strategies. The model evaluation criteria determine the dimensionality of the scene, and decide whether to terminate the procedure, or selectively refine the segmentation by following a global-to-local part segmentation approach. The control module generates hypotheses about superquadric models at clusters of underestimated data and performs controlled extrapolation of the part-model by shrinking the global model. As the global model shrinks and the local models grow, they are evaluated and tested for termination or further segmentation.

We present results on real range images of scenes of varying complexity, including objects with occluding parts, and scenes where surface segmentation is not sufficient to guide the volumetric segmentation. We analyze the issue of segmentation of complex scenes thoroughly by studying the effect of missing data on volumetric model recovery, generating object-centered descriptions, and presenting a complete set of criteria for the evaluation of the superquadric models. We conclude by discussing the applications of our approach in data reduction, 3-D object recognition, geometric modeling, automatic model generation, object manipulation, and active vision.

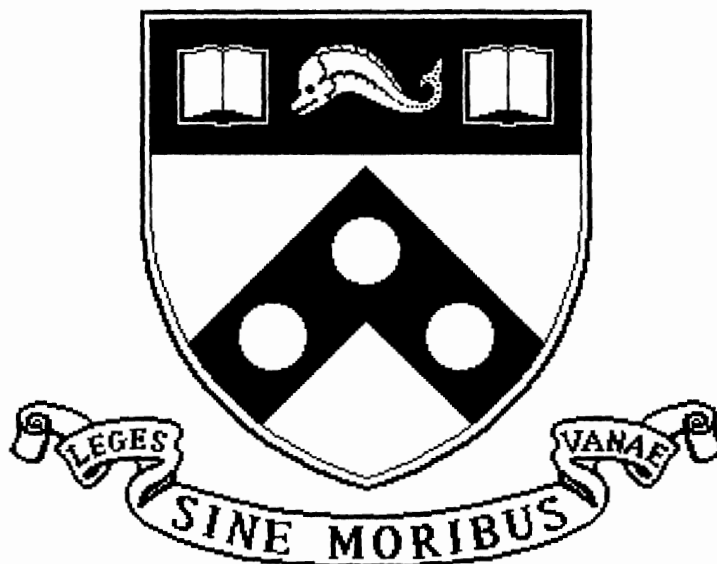
Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-45.

Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models

MS-CIS-91-45

Alok Gupta



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

1991

**Surface and Volumetric Segmentation of
Complex 3-D Objects
Using
Parametric Shape Models**

(Dissertation)

**MS-CIS-91-45
GRASP LAB 267**

Alok Gupta

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

June 1991

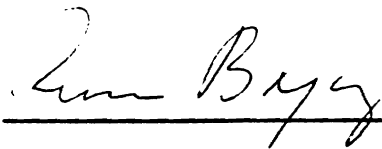
**Surface and Volumetric Segmentation of
Complex 3-D Objects
Using
Parametric Shape Models**

Alok Gupta

A DISSERTATION
IN
COMPUTER AND INFORMATION SCIENCE

Presented to the Faculties of the University of Pennsylvania in
Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy.

1991

A handwritten signature in cursive script, appearing to read "Ruzena Bajcsy", is written over a horizontal line.

Ruzena Bajcsy
Supervisor of Dissertation

A handwritten signature in cursive script, appearing to read "Mitch Marcus", is written over a horizontal line.

Mitch Marcus
Graduate Group Chairperson

To Maharaj Charan Singh Ji, my parents, and Soni.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Professor Ruzena Bajcsy for all the guidance, support and encouragement, and for providing excellent resources in the GRASP lab. Through her tireless efforts the lab has witnessed tremendous growth which afforded me opportunities to exchange ideas with a host of researchers, ranging from students to visiting researchers.

I wish to thank my dissertation committee members, Dr. Max Mintz, Dr. Norman Badler, and Dr. Ramesh Jain for encouraging comments and guidance. Dr. K. Wohn introduced me to the GRASP lab and guided my initial research.

Aleš Leonardis, with whom I had endless brainstorming sessions, collaborated in the work that culminated in an exciting surface segmentation package. Luca Bogoni helped with superquadric evaluation, designing test data, and developing the superquadric library and other support functions. I thank Luca and Aleš for their contributions to this work. I am thankful to Franc Solina for developing a stable superquadric model recovery program that performed as advertised in his thesis. Constantine (Gus) Tsikos' range scanner allowed me the freedom to design my own experiments. Howard Wang helped with superquadric software and analyzed the geometry of the Tsikos' scanner. Dmitry Cherkassky implemented a part of the package on a Connection Machine in an unfriendly C/Paris environment. Gareth Funka-Lea, Pramath Sinha and Celina Imlienska collaborated on the USPS project, which led to the topic of my dissertation.

I am thankful to Raymond McKendall, our resident POSTSCRIPT expert, for providing the valuable (and free) consultation for the succinct display of results in the dissertation. I thank Gaylord Holder, Jennie Kay, Howie Choset, and Jakov Kucan for keeping the PM (picture-manipulation) library functional. I thank John Bradley for providing the X-display and other system support, and Trisha Yannuzzi for taking care of the administrative aspects.

At the risk of missing out a few, I would like to thank the following Graspees for enriching the research environment in the lab: Sanjay Agrawal, Helen Anderson, Mario Campos, George Chou, Peter Corke, Fil Fuma, Janez Funda, Greg Hager, Dave Heeger, Siu-leong Iu, Alberto Izaguirre, Gerda Kamberova, Hajimu Kawakami, Robert King, Sang Wook Lee, Greg Long, Varun Malhotra, Stephane Mallat, Jasna Maver, Eric Paljug, Laurent Peytavin, Marcos Salganicoff, Tarek Sobh, Balaji Srinivasan, Nathan Ulrich, and Yehong Zhang.

The completion of this work owes entirely to the encouragement and support I received from my parents and friends. I cannot express in words my gratitude to my parents, Dr. G. R. Gupta and Mrs. Pushpa Gupta, who always eagerly supported my intellectual pursuits. I also thank my friends who were always there to cheer me up and who helped me maintain my sanity during the final months of the dissertation.

This research was supported in part by: AFOSR Grants 88-0244, 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.

ABSTRACT

SURFACE AND VOLUMETRIC SEGMENTATION OF COMPLEX 3-D OBJECTS USING PARAMETRIC SHAPE MODELS

Alok Gupta

Supervised by Ruzena Bajcsy

The problem of part definition, description, and decomposition is central to the shape recognition systems. In this dissertation, we develop an integrated framework for segmenting dense range data of complex 3-D scenes into their constituent parts in terms of surface and volumetric primitives. Unlike previous approaches, we use geometric properties derived from surface, as well as volumetric models, to *recover* structured descriptions of complex objects without *a priori* domain knowledge or stored models.

To recover shape descriptions, we use *bi-quadric* models for surface representation and *superquadric* models for object-centered volumetric representation. The surface segmentation uses a novel approach of *searching* for the best piecewise description of the image in terms of bi-quadric ($z = f(x, y)$) models. It is used to generate the region adjacency graphs, to localize surface discontinuities, and to derive global shape properties of the surfaces. A superquadric model is recovered for the entire data set and residuals are computed to evaluate the fit. The goodness-of-fit value based on the inside-outside function, and the mean-squared distance of data from the model provide quantitative evaluation of the model. The qualitative evaluation criteria check the local consistency of the model in the form of residual maps of *overestimated* and *underestimated* data regions.

The control structure invokes the models in a systematic manner, evaluates the intermediate descriptions, and integrates them to achieve final segmentation. Superquadric and bi-quadric models are recovered in parallel to incorporate the best of the coarse-to-fine and fine-to-coarse segmentation strategies. The model evaluation criteria determine the dimensionality of the scene, and decide whether to terminate the procedure, or selectively refine the segmentation by following a global-to-local part segmentation approach. The control module generates hypotheses about superquadric models at clusters of underestimated data and performs controlled extrapolation of the part-model by shrinking the global model. As the global model shrinks and the local models grow, they are evaluated and tested for termination or further segmentation.

We present results on real range images of scenes of varying complexity, including objects with occluding parts, and scenes where surface segmentation is not sufficient to guide the volumetric segmentation. We analyze the issue of segmentation of complex scenes thoroughly by studying the effect of missing data on volumetric model recovery, generating object-centered descriptions, and presenting a complete set of criteria for the evaluation of the superquadric models. We conclude by discussing the applications of our approach in data reduction, 3-D object recognition, geometric modeling, automatic model generation, object manipulation, and active vision.

Contents

Acknowledgements	vii
Abstract	ix
1 Introduction	1
1.1 Statement of the Problem	2
1.2 What are Parts?	4
1.3 Shape Primitives	7
1.4 The Segmentation Problem	10
1.4.1 Segmentation in terms of Primitives	11
1.5 The Control Flow for Volumetric Segmentation	13
1.6 Chapter Summary	15
2 Shape Primitives and Segmentation	17
2.1 The Choice of Primitives	17
2.1.1 Criteria for Representation	20
2.2 Our Choice of Primitives	21
2.2.1 The Surface Model: Bi-quadratics	21
2.2.2 Volumetric Model: Superquadric Part-Models	26
2.3 Chapter Summary	31
3 Surface Segmentation: The Search for the Best Description	33
3.1 Segmentation and Model recovery	34
3.1.1 Surface Fitting	34
3.1.2 The Model Recovery Algorithm	35
3.1.3 Features of the Segmentation Algorithm	38
3.2 Model Selection	40
3.2.1 Objective Function	41
3.2.2 Optimizing the Objective Function - the Algorithm	42
3.3 Dynamically Combining Model Recovery and Selection	44
3.3.1 Recover-and-Select	46

3.4	Example Surface Descriptions	47
3.4.1	Scene 1	49
3.4.2	Scene 2	50
3.4.3	The Coffee-mug	52
3.4.4	The Car	54
3.4.5	Object with smoothly merging sides	54
3.5	Chapter Summary	56
4	Deriving Surface Properties from Bi-quadrics	57
4.1	Refining Regions Along Intersection Curves	57
4.1.1	Another Method for Region Refinement	61
4.2	Region Refinement by Relaxing Compatibility Constraint	61
4.3	Bi-quadric Surface Types	62
4.3.1	Transformation to Standard Form	63
4.4	Aligning the Major Axis for Curved Objects	65
4.5	Edge-type Determination	67
4.6	Surface Adjacency Graphs (SAGs)	67
4.7	Chapter Summary	70
5	Superquadrics : Volumetric Part-Models	71
5.1	Introduction	71
5.1.1	Applying Deformations to Superquadrics	73
5.2	Formulation of the Model Recovery Problem	75
5.2.1	Interpretation of the Goodness-of-fit	76
5.3	Euclidean Distance Measure	78
5.4	Apparent Contours of Superquadrics	80
5.5	Superquadric Edges	83
5.5.1	Superquadric Edges from Occluding Contour	83
5.6	Recovery of a Superquadric Model on Range Data	86
5.7	Chapter Summary	89
6	Criteria for Superquadric Model Evaluation: Residual Analysis	91
6.1	Criteria for Model Evaluation	91
6.2	Residual Analysis	99
6.2.1	Residual Analysis by Inside-Outside Function	99
6.2.2	Residual Analysis Along the Viewing (Z) Direction	100
6.3	Residual Clustering for Further Processing	101
6.3.1	Residual Adjacency Graphs (RAGs)	101
6.3.2	Using the Residuals for Superquadric Evaluation	102
6.4	Chapter Summary	104

7	Volumetric Segmentation: The Control Flow	105
7.1	Issues in Volumetric Segmentation	105
7.1.1	What are Superquadrics Modeling?	105
7.1.2	Superquadric Recovery Formulation for Segmentation	106
7.1.3	Coping with the Missing Information	107
7.1.4	Orienting the Initial Superquadric Model	108
7.1.5	Surface Support for the Superquadric Data	109
7.2	The Strategy for Volumetric Segmentation	109
7.2.1	Strategy 1: One Superquadric Model for Every Bi-quadric Surface .	112
7.2.2	Strategy 2: Grouping Convex Surfaces	114
7.2.3	Strategy 3: Global to Local Superquadric Fitting	116
7.3	General Strategy: Integrating the Three Strategies	120
7.3.1	Selection of Part-Models	122
7.3.2	Growing or Extrapolating the Part-Models	123
7.4	Chapter Summary	125
8	Experimental Results	127
8.1	Test Cases: Complexity and Scope of the Paradigm	127
8.2	Examples	129
8.2.1	L shaped Objects	130
8.2.2	Bent Objects	136
8.2.3	The Coffee Mug	142
8.2.4	Scene3: Multiple Clusters	145
8.2.5	The Wrench	148
8.2.6	The NIST Object	152
8.3	Implementation Aspects	155
8.3.1	System Parameters	155
8.4	Discussion	156
9	Summary and Conclusions	159
9.1	Applications and Future Directions	160
9.2	Contributions of the Dissertation	161
A	Linear Least Squares for Surface Fitting	163
A.1	Solving the Least Squares Problem	164
B	Deriving the Standard Forms of Bi-quadrics	167
C	Superquadric Surface Normals	171
	Bibliography	173

List of Tables

3.1	Trade-offs in combining model recovery and model selection procedures. . .	46
4.1	Axis and surface-type determination from the coefficients of the standard form of the bi-quadrics.	64

List of Figures

1.1	3-D parts	4
1.2	Part versus detail	6
1.3	The hierarchy of representation	7
1.4	Edge-Surface-Volume representation of a complex object	9
1.5	Examples of some complex objects	12
1.6	The control flow of the SUPERSEG system	14
2.1	Surface boundaries for part decomposition	22
2.2	Object with smoothly merging sides	26
2.3	Superquadrics: volumetric primitives	27
3.1	Iterative regression approach to model-recovery	36
3.2	Noise distribution and the role of C.	39
3.3	Model recovery and selection	45
3.4	Scene 1	48
3.5	Scene 2	51
3.6	The coffee-mug	53
3.7	The car	54
3.8	Object with smoothly merging sides	55
4.1	Overflow of one region into another along the intersection curve.	59
4.2	Example of intersection cleaning on a real range image	59
4.3	The three bi-quadric surface types.	63
4.4	The Coordinate system for bi-quadric representation	64
4.5	Major axis estimation for curved parts in Scene 1	66
4.6	Edge-type determination for the NIST object	68
4.7	Surface adjacency graph (SAG) for the NIST object	69
5.1	Superquadric shapes	72
5.2	Deformed superquadric shapes	74
5.3	β -expansion and contraction of a superquadric model	77
5.4	Euclidean distance computation	79

5.5	Apparent contours of superquadrics	82
5.6	Superquadric curvature	84
5.7	Superquadric edges	85
5.8	Image, World and Object coordinate systems	87
5.9	Recovery of the superquadric model	87
6.1	Computation of deviation of a point from the superquadric model	93
6.2	The object with missing volume	95
6.3	The composite object	97
6.4	The NIST object	98
6.5	The global model for Scene 3	102
6.6	Residual adjacency graph (RAG) for Scene 3	103
7.1	Orientation of the object coordinate system	108
7.2	The Control Flow of the SUPERSEG system	110
7.3	The composite object	113
7.4	The NIST object: One superquadric model for every surface	115
7.5	The composite object: Grouping convex surfaces together	115
7.6	The composite object: Global to local model growing and shrinking	117
7.7	The composite object: Residuals of strategy 3	118
7.8	The composite object: Volume-GOFs for strategy 3	119
7.9	The hierarchical superquadric representation and recovery structure	123
8.1	The hierarchy of test cases	128
8.2	The L1-shape	131
8.3	The L1-shape: Volumetric segmentation	132
8.4	The L2-shape	133
8.5	The L3-shape	134
8.6	The L4-shape	135
8.7	The Bent1 shape	137
8.8	The Bent1 shape: Volumetric segmentation	138
8.9	The Bent1 shape: Residuals	139
8.10	The Bent2 shape	141
8.11	The Coffee Mug	143
8.12	The coffee mug	144
8.13	Residual adjacency graph (RAG) for Scene 3	145
8.14	Scene 3: Multiple clusters	146
8.15	Scene 3: Volumetric segmentation	147
8.16	The Wrench	149
8.17	The Wrench: Volumetric segmentation	150
8.18	Wrench: Global model	151

8.19 Surface adjacency graph (SAG) for the NIST object	152
8.20 The NIST object	153
8.21 The NIST Object: Volumetric segmentation	154

List of Algorithms

3.1	Recovery of one bi-quadric model.	37
3.2	Bi-quadric model selection.	44
4.1	Region refinement along the intersection curve.	60
4.2	Edge detection and labeling at two intersecting regions.	69
5.1	Recovery of one superquadric model on range data.	88
7.1	Orientation and Z-axis placement of the initial superquadric model.	109
7.2	The control flow for volumetric segmentation.	121
7.3	General Analysis when s_under regions exist.	122

CHAPTER 1

Introduction

For visual discrimination, shape plays a very important role. Human beings exhibit remarkable abilities to simplify the visual input without bringing in domain knowledge or functionality into consideration. A robot using vision for navigation or recognizing objects, has to similarly simplify the visual input to the level that is required for the specific task. To simplify means to partition images into entities that correspond to individual regions, objects and parts in the real world and to describe those entities only in detail sufficient for performing a required task. Usually the first level of simplification entails obtaining part descriptions based on the properties that are independent of the position, orientation, scale and the work domain. Physical shape of an object is an important characteristic that allows us to discriminate between two otherwise identical objects, for example a ball from cube of same color and texture. Shape is the outward appearance or form of an object defined by its boundaries and surfaces. It is therefore possible to model an object's physical shape by geometric primitives in terms of surfaces and volumes. The distinction between surface and volume is only subtle in the sense that surface description captures the abstract notion of a surface as curved or convex or planar, while the volume description gives a higher-level description by combining the surface properties to give the bounding volume as a box or a sphere. So, a piecewise-planar description of a box at surface level can be put together to represent the *global* shape as a box. This indicates that surface models capture *local* shape more accurately than volumetric models, while volume models are better at describing global shapes by ignoring local details.

From the perspective of shape, objects in the real world represent a complex conglomeration of primitive shapes. These primitive shapes can be considered as the building blocks of the larger objects. An object of an arbitrarily complex shape can be decomposed into

numerous pieces based on the primitives shapes. We would like the shapes to be primitive enough to model as large a class of objects as possible. For other objects, only approximate descriptions will suffice. The primary objective of a shape recognition system is to derive a structured description of complex objects in terms of primitive shapes. The resulting decomposition into parts is very useful for the high-level symbolic reasoning object-recognition processes, which can attach domain specific labels to the parts, and reason at a level where the visual input is structured in terms of primitives, rather than cope with the difficulties of low-level vision and huge pile of unstructured data.

Since the shapes have to be *recovered* from raw data, it is not possible to invoke complex models (models with hundreds of degrees of freedom) straight away. It is, however, feasible and perceptually less ambiguous to use simpler but powerful models that can capture the local and global properties of the object shapes, and provide a first approximation to the more complex models. With computability, simplicity, and the utility of the shape representation as our major concerns, we decided to use bi-quadrics and superquadrics as our surface and volumetric models respectively. We develop a control structure to effectively carry out the decomposition of complex objects in range images, and address the numerous issues encountered in a data-driven bottom-up approach.

The dissertation is organized in the following manner. In this chapter, we formally define the shape recognition problem, and give a philosophical overview of the problem. Shape primitives, motivation for segmentation and our approach are discussed in chapter 2. The surface segmentation procedure is described in chapter 3 and the aspects of integration of surface information with superquadric model recovery are presented in chapter 4. Chapter 5 gives a detailed analysis of superquadric models and derives important results useful for volumetric segmentation. The criteria for the evaluation of superquadric models are discussed in chapter 6. The control module, which systematically recovers the superquadric models, is described in chapter 7, and detailed results on real range data are given in chapter 8. Finally, we summarize our approach and discuss the future directions in chapter 9.

1.1 Statement of the Problem

The goal of this research is to *recover structured shape descriptions* of complex three-dimensional objects in range images in terms of significant parts defined by a set of surface and volumetric primitives without *a priori* knowledge about the object or the object domain. By “significant” we mean that the part boundaries are of physical, perceptual or

geometric significance and that part decomposition is *natural*.

In addition to defining the problem as that of part-segmentation, it can also be viewed as that of symbolic representation and data reduction by attaching symbols to data. Other interpretations include geometric modeling and geometric reasoning, where knowledge inherent in the geometric primitives is used to model data.

The phrase “recover structured descriptions” summarizes our approach to the problem of part-segmentation. Instead of matching stored models, the shape vocabulary should include a continuum of shapes that can be recovered from the data. The description should be structured, and obtained in terms of shape primitives without a priori knowledge about the object or the object domain.

This brings in the vital issues of part definition, description and decomposition, each of which addresses the very basis of our research. At the outset, it is important to note that the problem of shape description and decomposition has proved to be extremely difficult mainly because the researchers have either tackled each of the components separately or limited their description to one primitive. We present arguments that the issue of part description and part segmentation¹ are related and *have* to be considered together. This observation leads us to propose surface and volumetric primitives for shape representation, and the control structure to integrate them to obtain the final description.

The complete problem of shape recognition can be posed as a composition of following fundamental subproblems :

1. What are parts and how are they defined?
2. What is the basis of decomposition of shape into parts?
3. How are part definition, description and decomposition related?
4. What types of geometric primitives and how many primitives are enough to generate the desired part description?
5. What is the motivation for selecting a set of primitives and partitioning rules?
6. What are the processes that carry out these decompositions?
7. What is the overall control strategy to arrive at a detailed description of complex objects in terms of chosen primitives?

¹We will use the terms segmentation and decomposition interchangeably.

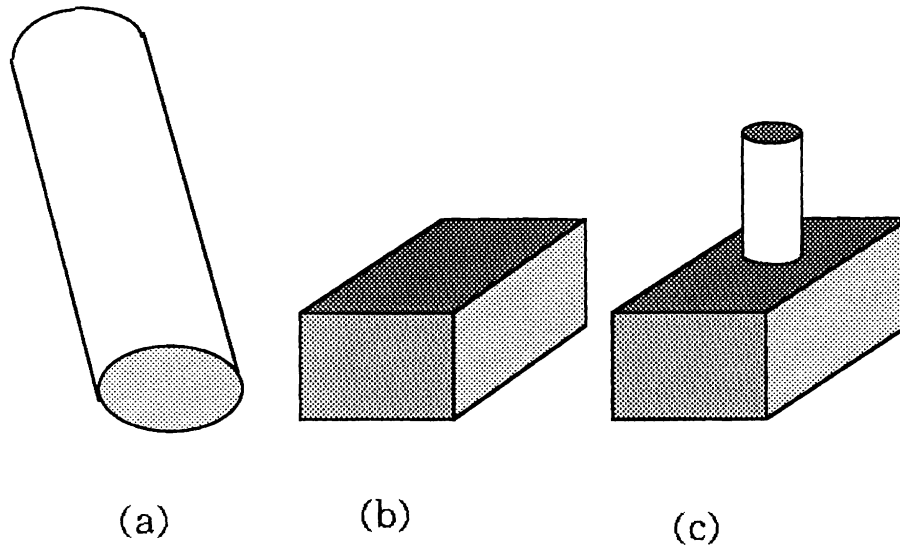


Figure 1.1: **3-D parts:** A cylinder (a) is a single volumetric part consisting of two surface patches. The Box (b) is perceived as a single volumetric part, while three planar patches are seen at surface level. The composite object (c) has two distinct volumetric parts, separated by a concavity at the transversal join.

The first five questions constitute the problem analysis phase, where we attempt to formalize the problem in the most general sense. The last two questions involve important computational and integration issues that will determine the eventual robustness of the system. In this chapter we lay the foundation of our work by giving a general definition of the problem. Other issues are dealt with in detail in the subsequent chapters.

1.2 What are Parts?

Webster's dictionary defines a part as one of the portions into which something is or is regarded as divided and which together constitute the whole. Arnheim [1974] notes that in a quantitative sense, any section of whole can be a part. But this definition does not preserve *structure*. Partitioning by ignoring structure is not of much use in vision [Witkin and Tenenbaum 1983, Hoffman and Richards 1985, Pentland 1987b, Arnheim 1974].

Part definition ultimately depends on the reliability, versatility and computability constraints imposed by the task of shape recognition and may not be unique [Hoffman and Richards 1985]. It is therefore difficult to give a general definition of *part* in the context of

shape recognition. However, a working definition would define a part as an easily *describable* and recognizable portion of a complex shape that is invariant to minor changes in viewpoint (figure 1.1). It brings the notion of description into part definition, emphasizing the fact that two are interrelated. The idea of partitioning a complex object into describable parts is not new in computer vision. It differs in the choice of primitives and the way segmentation is carried out. Traditionally part definition has been either *primitive*-based or *boundary*-based [Bennett and Hoffman 1987, Nevatia and Binford 1977, Hoffman and Richards 1985]. In the literature, primitive-based approaches [Agin and Binford 1973, Nevatia and Binford 1977, Soroka and Bajcsy 1978] have defined objects by cylindrical, polyhedral, conical or spherical shapes. The objective of such systems is to fit parts of complex objects with models in the shape vocabulary. Boundary-based approaches [Hoffman and Richards 1985, Bennett and Hoffman 1987, Koenderink and vanDoorn 1982, Biederman 1985] define parts by outlining the boundaries on surfaces. Beiderman [1985] has emphasized the perceptual basis for part decomposition based on Gestalt principles (nonaccidental properties of 2-D projection of 3-D objects). Others have imposed the requirement of *continuity* [Binford 1982] and *uniformity* [Hoffman and Richards 1985]. In shape decomposition, one tries to follow the principle of orderliness, which means - partitioning things in the simplest possible way. Such partitioning normally reflects the structure of the physical world quite well due to the principle of parsimony [Arnheim 1974].

Segmentation Versus Representation: Decomposition into parts, units or primitives is the basis of scientific methodology. Because of the limits on how much information we can process at a time, we have to simplify and view the world at various levels of abstraction. Our objective is to decompose complex objects into the constituent parts based on the shape. Many reasons have been advanced in favor of such a decomposition. A recognition-by-parts approach is not sensitive to occlusion and is extremely powerful in handling countless configurations of articulated objects. A description in terms of basic shape primitives is more efficient, parsimonious in space consumption, and facilitates structured description of the world. These arguments are supported by the principles of perceptual organization [Biederman 1985].

In computer vision literature the partitioning of images and description of individual parts is called segmentation and shape representation respectively. We have presented arguments in [Bajcsy et al. 1990] that the problem of segmentation and representation are related and *have* to be treated simultaneously. Since the analysis aspect of computer vision

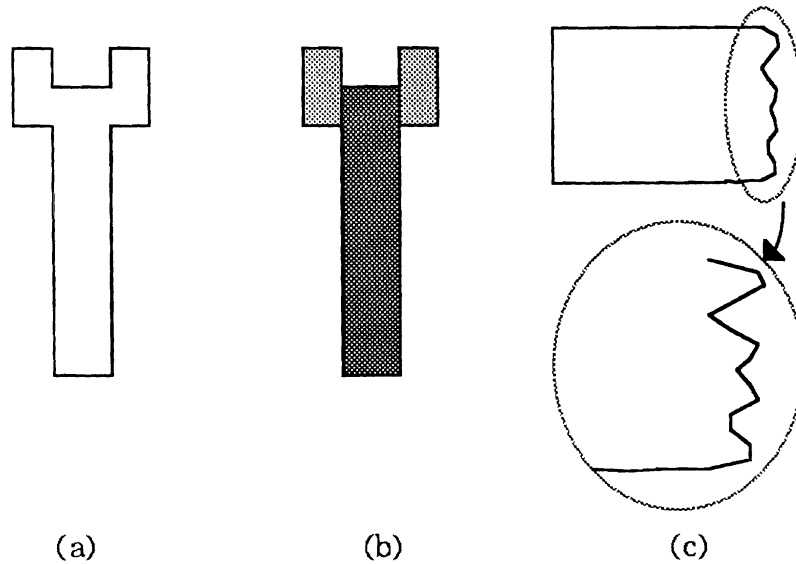


Figure 1.2: **Part versus detail:** Perception of parts depends on scale of the part with respect to the whole. The wrench shape (a) needs decomposition into parts (b). While the jagged boundary on one side of the object (c) can be ignored as a detail. However, at a finer scale, details become parts.

requires symbolic representation of data, for the models to correctly reflect the underlying data it is imperative that the issue of representation be addressed during the process of segmentation. Bennett and Hoffman [1987] have argued that a primitive based part definition confuses the problem of part definition with the separate problem of part description. We consider them to be interdependent; parts are defined the way they are described by shape primitives. Our surface primitive implicitly defines the part boundaries on the smooth surfaces and thus includes the advantages of a boundary-based approach. In this context, it is possible to view the local support enjoyed by the surface descriptions as achieving the goals of a boundary-based approach, while preserving the descriptive power of the primitive-based approaches. However, it might not *always* be possible to obtain complete primitive-based description of arbitrary objects for *all* the parts. Volumetric primitives being influenced by the global shape, *may not* account for all the surface details. Surface primitives ensure that we obtain a part description at a level lower and less global than the volumetric primitive.

An important issue related to the part-whole relationships is the issue of *part* versus *detail*. That a portion of the whole merits an independent description as a part or can be

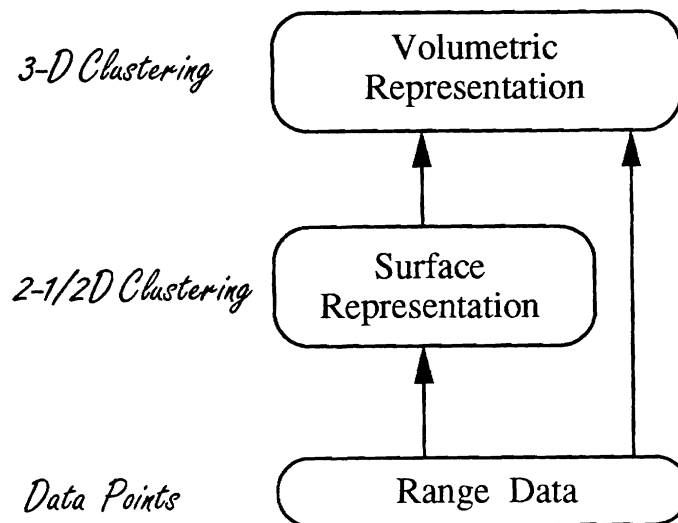


Figure 1.3: **The hierarchy of representation:** The volumetric model is a 3-D representation of the $2\frac{1}{2}$ -D range data.

considered a mere detail is a matter of scale in the bottom-up approach we are adopting. In figure 1.2 the wrench appears to have parts while the wiggles on one side of the other object appear to be details that do not need part level description. However by increasing the scale of the wiggleness with respect to the size of whole we get them as significant parts. Notice that the wrench can be decomposed into four parts instead of three. Ambiguities in final description are common in part-segmentation without a priori knowledge, and the strategy followed by the volumetric segmentation decides which description is preferred.

1.3 Shape Primitives

What are the shape primitives that adequately describe the data? How many primitives are required? Since the objects in the world are of arbitrary complexity, it is not possible to include primitives for all the different shapes as it will never be a complete set. Thus we have to make a judicious choice of primitives that have the capability of describing data at various levels (dimensions), so that description at some level is always possible and computability of primitives is assured. For obtaining a global shape description from single-viewpoint 3-D data requires addressing shape at following levels :

1. **Volumetric level** : Primitives capable of modeling parts in three dimensions are needed to describe global shape of parts.
2. **Surface level** : Surface primitives describe *internal surface boundaries* and *surface patches* which are difficult to model with volumetric primitives, but are vital source of information about recovering part structure.

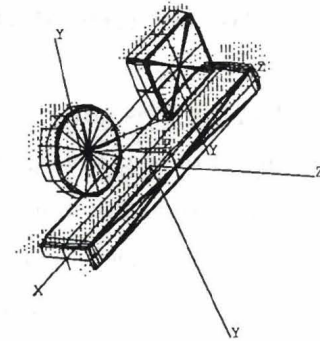
This hierarchy of shape primitives (figure 1.3) allows us to obtain shape descriptions at volumetric and surface levels. Figure 1.4 shows the hierarchy of descriptions for a machined object. It is clear that no one primitive will always capture all the details of shape. For example, if it is not possible to model parts with the selected volumetric primitive, an approximation at volumetric level can be obtained, with more detailed description at surface level. Thus, completeness requirement for a general representation is satisfied by obtaining hierarchical descriptions.

Low-level models like contours and edges have low granularity (edge description in figure 1.4) and are too local to capture or make use of the gross structure of the world. They are sensitive to local changes and difficult to put together in a global context. They are useful when used with surface models. Our hierarchy of representation (figure 1.3) has only implicit information about edges, and no explicit edge models are used.

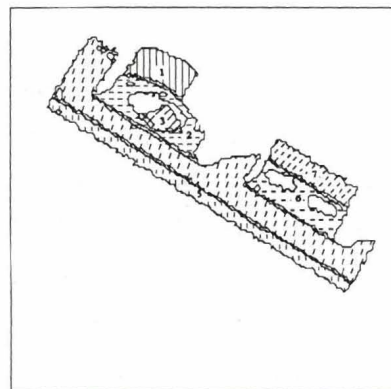
The next level of shape description is achieved by describing local and overall surface characteristics. Surfaces play important role in human perception of shape. A lot of effort in computer vision has been spent on describing complex surfaces as piecewise continuous patches. We have developed a new method for piecewise surface segmentation in terms of variable-order (up to second-order) bi-variate polynomials.

Three dimensional primitives like generalized cylinders and cones, polyhedral models, 3-D Smoothed local symmetries [Brady 1983], and 3-D symmetric axis transform [Nackman and Pizer 1985] have been used by model based vision systems. However, the power of representation varies from model to model. A model allowing deformations is likely to describe objects with fewer primitives than a rigid model which will need more instances to approximate the object. As we shall describe later, volumetric primitives are essential to generate compact object-centered descriptions and to define global part-structure. Superquadric models, our choice of volumetric primitives, generate object centered descriptions by drawing local support from the surface primitives.

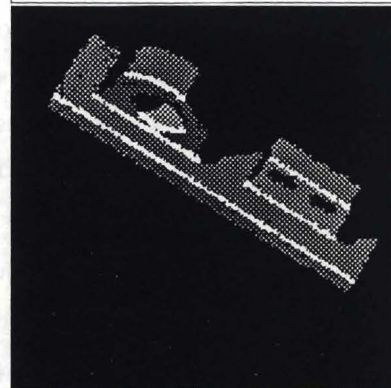
Volumetric description:



Surface description:



Edge description:



Range image:

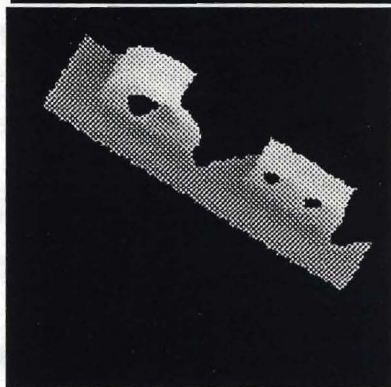


Figure 1.4: **Edge-Surface-Volume representation of a complex object:** NIST Object: The hierarchy of shape primitives highlights different aspects of shape at different levels.

1.4 The Segmentation Problem

The problem then is how to use the primitives to segment the objects into part-structures. In the context of shape recognition, the problem of segmentation can be defined as matching the right kind of shape model with the right parts of data in an image. This brings up the crucial question of facilitating this matching process.

Each of the shape primitive can independently describe the data. The contour-based segmentation is widely studied in pattern recognition and computer vision as 2-D shape recognition problem [Pavlidis 1977, Shapiro 1980, Asada and Brady 1986]. Since we are not explicitly dealing with 2-D segmentation, we are interested in approaches using dense surface information for segmentation. Surface based approaches have been popular with model-based vision systems, as they have local support, and allow 3-D objects to be modeled as collection of surfaces. Volumetric models have proved to be most difficult to recover from image data. Some researchers have used a combination of features to *model* domain specific objects [Kuan and Drazovich 1985, Brooks 1983], exploiting the robustness achieved by combining descriptions at different levels. To facilitate segmentation we believe that a general purpose vision system needs volumetric and surface shape primitives. Difficulty in recovering volumetric models in intensity images is experienced due to the loss of depth information. But the problem has not proved to be any easier even with the availability of depth information [Nevatia and Binford 1977, Kuan and Drazovich 1985, Solina 1987, Boulton and Gross 1987, Rao 1988, Soroka and Bajcsy 1978].

Model based vision systems match the available models in the model database with hypothesized instances of models in the image data. Object models typically used in vision are built as a structured hierarchy of primitive part-models. Since we are addressing the problem at the level of shape-definition only, and not at the object-definition level, we do not have the high-level models that restrict the part-models to a particular configuration. Therefore, the typical model-based vision strategy is too restrictive to be of any use for part segmentation. The essential difference between shape recognition problem and the model-based approach is that we are looking for instances of part-models and not object-models that constrain the part-models to configure in a known order. Consequently, we do not have a notion of “object” as such, but only geometric parts. A cluster of points that cannot be trivially broken into smaller cluster of points is considered an object by us. Thus, neighborhood information provided by the dense $2\frac{1}{2}$ -D image data is vital in defining what we mean by connectivity. 4-connectivity in the neighborhood of a pixel connects the pixel to

the neighboring pixels. This formulation relieves us from generating elaborate neighborhood representations like vornoi diagrams and lets us deal with the more important problem of segmentation of data.

1.4.1 Segmentation in terms of Primitives

There are two basic strategies for segmentation:

1. **Global-to-local:** Proceed from coarse-to-fine discrimination by partitioning larger entities into smaller.
2. **Local-to-global:** Start with local models and aggregate them into larger ones.

Both of these strategies are commonly used in computer vision [Ballard and Brown 1982, Pavlidis 1977]. The advantage of the coarse-to-fine strategy is that one gets a quick estimate about the volume or surface of the object which can be further refined under control of some higher-level process which determines the adequacy of the description. The advantage of this approach is that the amount of detectable detail can be obtained by switching to a different kind of representation or to a finer scale. For example, to describe smaller shape details one might have to go from volumetric to surface representation. At the same time, the criteria for accepting a description can force further splitting of data to match the model. The important idea that these methods convey is that progressive blurring of images clarifies their deep structure. Object-centered descriptions like superquadric models lend themselves especially well to such a strategy, since they describe convex bounding volumes and any deficiency in the description at a desired scale can be modeled by either negative volumes or by decomposing the data into parts. We have developed a control structure that integrates various sources of information to make the coarse-to-fine strategy feasible for superquadrics.

The second strategy, which goes from local to global, starts with local features and incrementally builds larger representations. Models like bi-variate polynomials with good extrapolation capabilities are the right choice for this approach. Although aggregation of low-level models into models of larger granularity is difficult in presence of noise, it is possible to accomplish it iteratively, by incrementally growing the model. We have developed a new approach for the recovery of bi-variate polynomials (up to second-order) using a region growing method based on iterative regression and a model selection method that dynamically selects and allows only the “best” models for further growth.

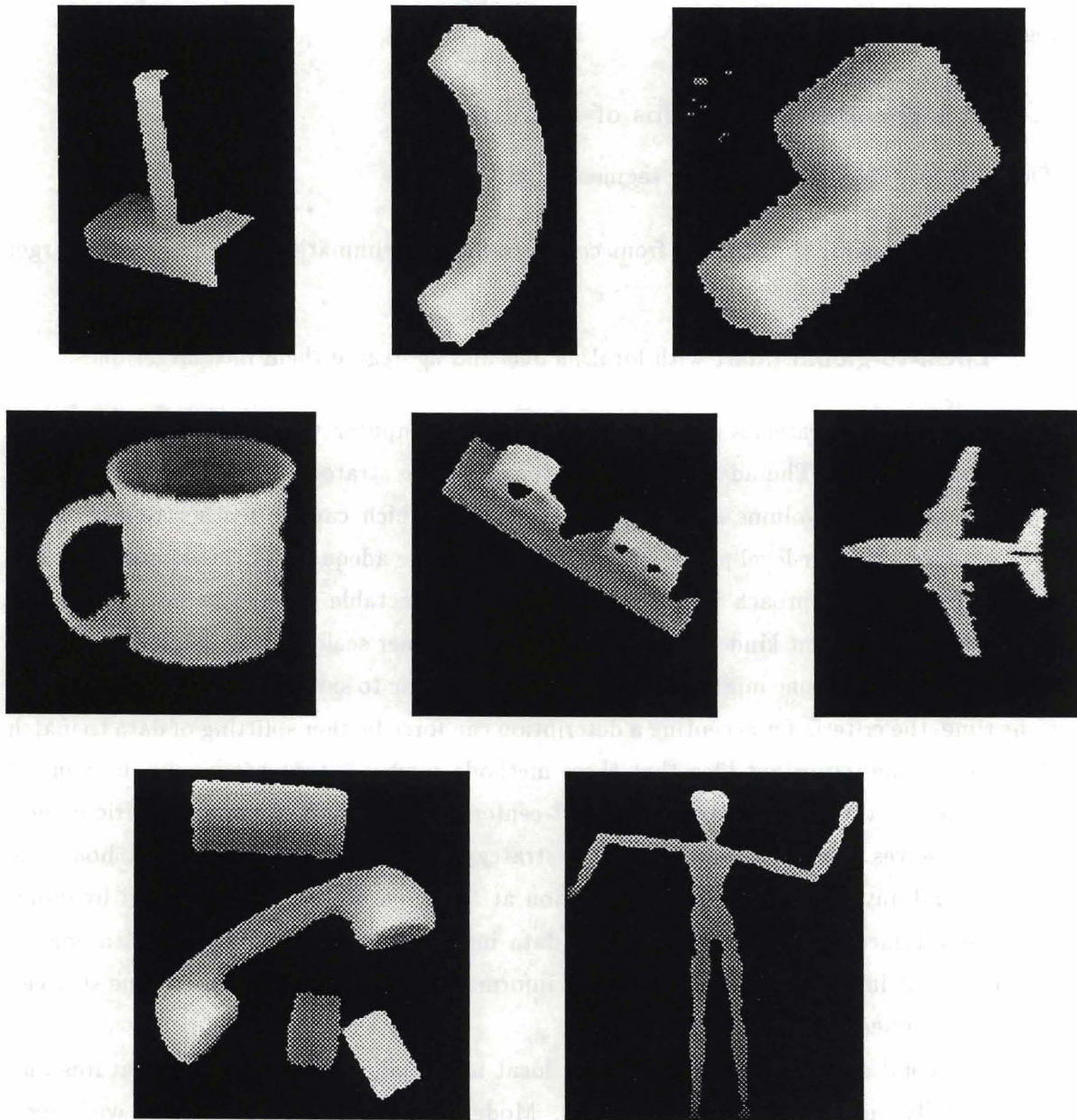


Figure 1.5: **Examples of some complex objects:** Range images of some of the test objects showing the complexity of the segmentation task.

1.5 The Control Flow for Volumetric Segmentation

Given the shape primitives and the modules to recover them, a control strategy is needed to invoke, evaluate and integrate them. The control structure forms the heart of the shape recognition system. The range of input data that we are considering is quite broad, as shown in range images of complex objects in figure 1.5. The input to our system is a dense depth map, scanned by an active range scanner from a single viewpoint. No information about scanner geometry or viewpoint is assumed. Since we are dealing with objects of arbitrary complexity, a general control structure is required.

The control flow of the SUPERSEG (SUPERquadric SEGmentation) system is shown in figure 1.6. The bi-quadric surface segmentation and the recovery of the global superquadric model is done independently and the descriptions are then integrated by the control module. The most important component of the control module is its residual analysis module which matches the current volumetric description with the given data. This matching gives “difference measures” or the residuals that are then used to evaluate the models. While the volumetric model gives a holistic explanation of the whole object it can miss details that are beyond the scope of the model. An overall measure of goodness-of-fit, like the residual from least-squares fit, or the distance measure does not always give an accurate evaluation of the appropriateness of the volumetric model. Although models can have acceptable overall goodness-of-fit, they need not be the acceptable representations of the object. This argues for a measure other than the quantitative measure of goodness-of-fit. The qualitative measures obtained by comparing the input data the recovered volumetric model can point out the limitations of the volumetric model and suggest improvements in segmentation or refinement in shape representation.

We shall describe later that both qualitative (local distribution of residuals) and quantitative measures (normalized deviation of data from the model) are necessary for the complete evaluation of the volumetric models. Based on these measures the control module will either accept the current level of description or generate hypotheses about potential “parts”. The extraction of part-structure proceeds in a systematic global-to-local manner, with global parts discarding the underestimated data and local models growing on the discarded data. This in turn shrinks the global model and makes it converge on a part of the object. Chapter 7 describes the issues involved in designing a control structure to accomplish this.

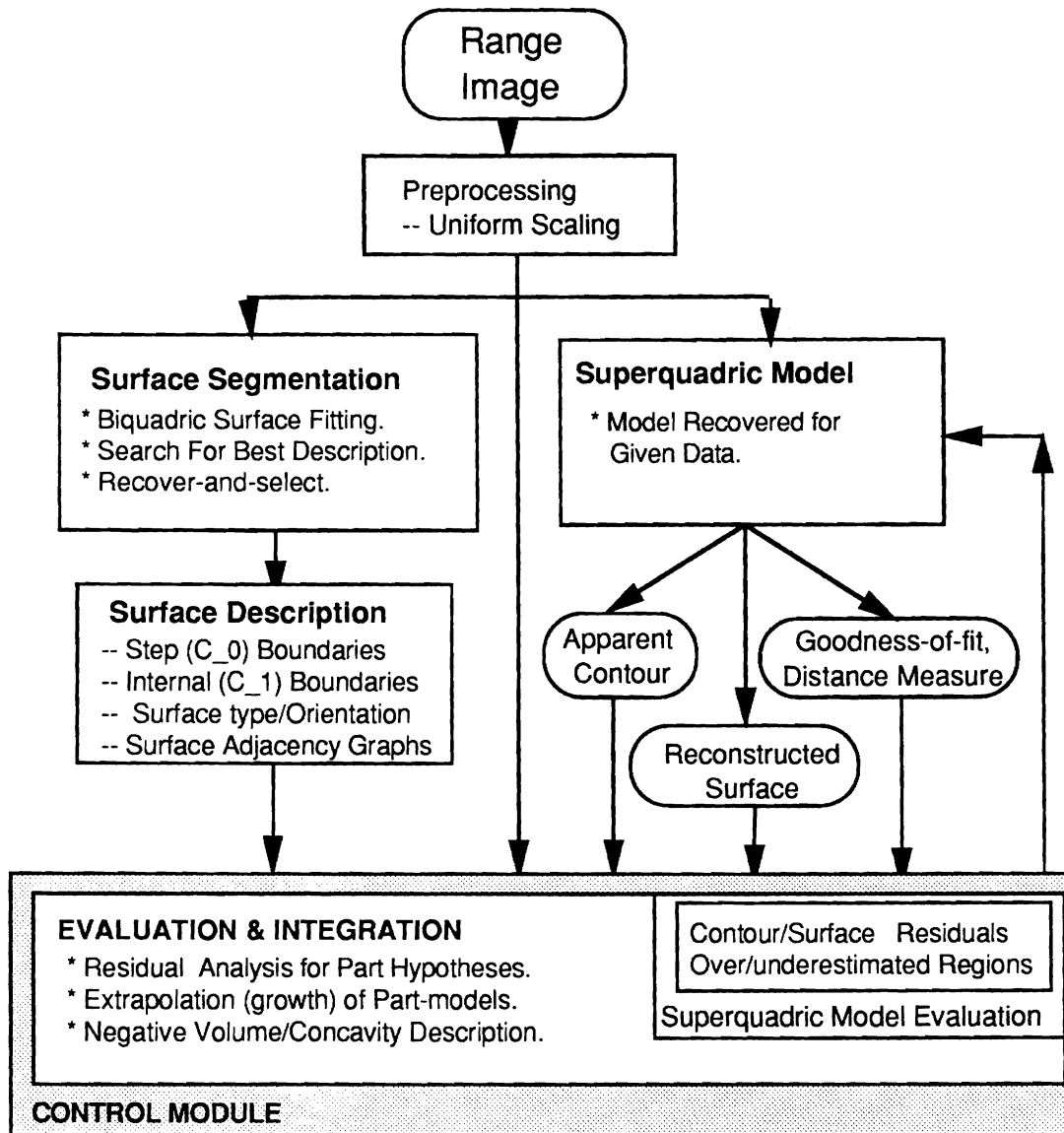


Figure 1.6: **The control flow of the SUPERSEG system:** A schematic diagram of the integrated framework for surface and volumetric segmentation.

1.6 Chapter Summary

We introduced the problem of surface and volumetric segmentation in the context of geometric part-description, and summarized our approach, which is coarse-to-fine at superquadric level and fine-to-coarse at surface level. There are five components of the SUPERSEG system: the bi-quadric surface segmentation module, the module extracting the surface information for volumetric segmentation, the superquadric recovery module, superquadric model evaluation and residual analysis, and the control module to systematically achieve the volumetric segmentation. In the next chapter, we describe the individual shape primitives and the issues involved in decomposing objects using them.

CHAPTER 2

Shape Primitives and Segmentation

In this chapter we analyze the shape primitives with the central idea of using them for part segmentation. We then present the primitives that we have chosen, and address the issue of segmentation using the individual primitives.

2.1 The Choice of Primitives

The choice of primitives can be guided by some general requirements such as a unique decomposition into primitives, that the primitives cannot be further decomposed or that the set of primitives is complete. Some of the shape representation criteria are designed primarily to facilitate object recognition when models recovered from images are matched to a model data base. Unfortunately, all those principles have not been applied to any general shape representation scheme for 3-D objects. A review of computer vision literature which reveals the large variety of geometrical primitives that were investigated for their applicability to shape representation is a testimony to the difficulty of shape description [Besl and Jain 1986]

Another discipline involved in representing shape is computer graphics, but from a synthesis (generating) point of view. Some commonly used 3-D representations in graphics are wire-frame representation, constructive solid geometry representation, spatial-occupancy representation, voxel representation, octree representation, and different surface patch representations. Splines are used for surface boundary representation. But requirements for shape primitives in computer vision are different from the ones for computer graphics. Shape primitives for computer vision must enable the analysis (decomposition) of shape. Common shape primitives for volume representation are polyhedra, spheres, generalized

cylinders, and parametric representations such as superquadrics. Different orders of surface patches (planar, quadratic, cubic) are used for surface representation. For boundary description one can use linear, circular or other second-order models for piecewise approximation, and higher-order spline descriptions. In the rest of this section we will discuss what influences the selection of shape primitives in computer vision.

If only one shape primitive is chosen, the segmentation process is relatively simple. But the resulting segmentation may not be natural! The data can be artificially chopped into pieces to match the primitives. An example of such unnatural decomposition is when a circle is represented piecewise with straight lines or when a straight line is represented with circular segments. If the scene consists of both straight lines and circles, then neither straight lines nor circles alone would enable a natural segmentation. A natural segmentation, on the other hand, would partition an image into entities that correspond to physically distinct parts in the real-world. A solution to such problems is to use more primitives. How many primitives are required for segmentation of more complicated natural scenes is then the crucial question. The larger the number of primitives, the more natural and accurate shape description and segmentation is possible. But the larger the number of primitives, the more complicated the segmentation process becomes. Finding the right primitive to match to the right part of the scene leads potentially to a combinatorial explosion. This argues for *limiting* the number of different shape models.

Another influencing factor on the number of different models is the level or granularity of models. A large number of low-level models is required for scene description because of their small size or granularity. Low-level models can fit to a large variety of data sets but bring little prior information to the problem. Substantial manipulation is required to obtain further interpretation of the data by aggregating low-level models into models of larger granularity which correspond to real world entities. Such aggregation techniques often fail because it is not possible to distinguish data from noise or account for missing data only on the basis of local information. Higher-level models, on the other hand, are prescriptive in the sense that they bring in more constraints and provide more data compression. Higher-level models are not information preserving in the sense that they might miss some important features because they cannot encompass those data variations within their parameterization.

A concise model which adequately describes the data will enable partitioning or segmentation of images into right parts and ignore noise and details. Such a model will have primitive shape models capable of describing shape at both low- and high-levels. In everyday life, people use a default level of representation, called basic categories [Rosch 1978]. Basic

categories seem to follow natural breaks in the structure of the world which is determined by part configuration [Tversky and Hemenway 1984]. Shape representation on the part level is then very suitable for reasoning about the objects and their relations in a scene. For part level description in vision, a vocabulary of a limited number of qualitatively different shape primitives [Biederman 1985] and different parametric shape models have been proposed. Parametric models describe the differences between parts by changing the internal model parameters. In computer vision, the most well known parametric models suitable for representing parts are generalized cylinders but superquadrics with global deformations seem to have some important advantages when it comes to model recovery [Pentland 1986, Bajcsy and Solina 1987]. It is sometimes possible to know a priori that a certain class of geometric models is sufficient to describe observed data. Another possibility is to somehow evaluate the complexity of the scene and the dimensionality of the objects in the scene. Knowing the complexity of the scene can greatly simplify the control structure for segmentation and shape recovery while knowing the dimensionality of objects simplifies the selection of shape models.

The *objective* of a vision system, whether the goal is to avoid obstacles during navigation, to manipulate objects with robotic grippers and hands or to identify objects by matching them to a data base, is another constraint during shape model selection. For object avoidance, only representation of occupied space is necessary, often allowing to largely overestimate the size of obstacles. In addition to location and orientation, grasp planing for robotic hands requires knowing more precisely the size and overall global shape of the object. For object recognition, more specific, identifying features are needed. Different shape primitives are better at representing different aspects of shape and at different scales. Volumetric representation provides information on integral properties, such as overall shape, enabling classification into elongated, flat, round, tapered, bent, and twisted primitives. They can best capture the overall size and volume since they must make an implicit assumption about the shape of the object hidden by self occlusion. Surface representation is better at describing details that pertain to individual surfaces which can be part of larger volumetric primitives. Surface primitives can differentiate planar surfaces versus curved surfaces, concave versus convex, and smooth versus undulated surfaces. On the one hand, occluding boundary representation is a local representation of curvature and surface near the boundaries, on the other hand, by delineating the boundaries of an object from the background, it defines the whole object.

2.1.1 Criteria for Representation

The criteria for selection of primitives have been studied extensively by vision researchers [Brady 1983, Marr 1982, Binford 1982]. 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 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 bi-variate 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 [1988] 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.

2.2 Our Choice of Primitives

Based on the criteria outlined above, we chose bi-quadric shapes as the variable-order surface models and superquadrics as the volumetric part-models. Bi-quadrics achieve $2\frac{1}{2}$ -D clustering, while superquadrics achieve 3-D clustering of the $2\frac{1}{2}$ -D data. Since our task is part-segmentation, which is invariant to scale, size, translation, and orientation, we use superquadrics as a general object-centered model for volumetric segmentation. For curved surfaces of order greater than 2, the piecewise bi-quadric description may not be invariant to orientation due to the fixed Z-axis orientation along the viewing direction. We use the bi-quadric and superquadric models as general-purpose representations to exploit the advantages of surface models and object-centered models. For objects with surface texture, only a coarse segmentation is possible at volumetric level, while detailed segmentation can be achieved at the surface level. In some cases, like the natural scenes, specialized models like fractals need to be employed. Locally deformable models are appropriate for objects with surface details (like human face), but usually require pre-segmented regions where they can be applied. Due to their complexity and representational ambiguities (high degrees of freedom) they are difficult to employ on raw data (if it requires more than one instance of the model for its description) for the purpose of segmentation.

We will now introduce the bi-quadrics and superquadric models, and discuss the segmentation issues as well as the past work in surface and volumetric segmentation using these and other models.

2.2.1 The Surface Model: Bi-quadrics

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

$$f(\mathbf{a}, \mathbf{x}) = \sum_{l=1}^{p(r)} \mathbf{a}_l f_l(\mathbf{x}) \quad (2.1)$$

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

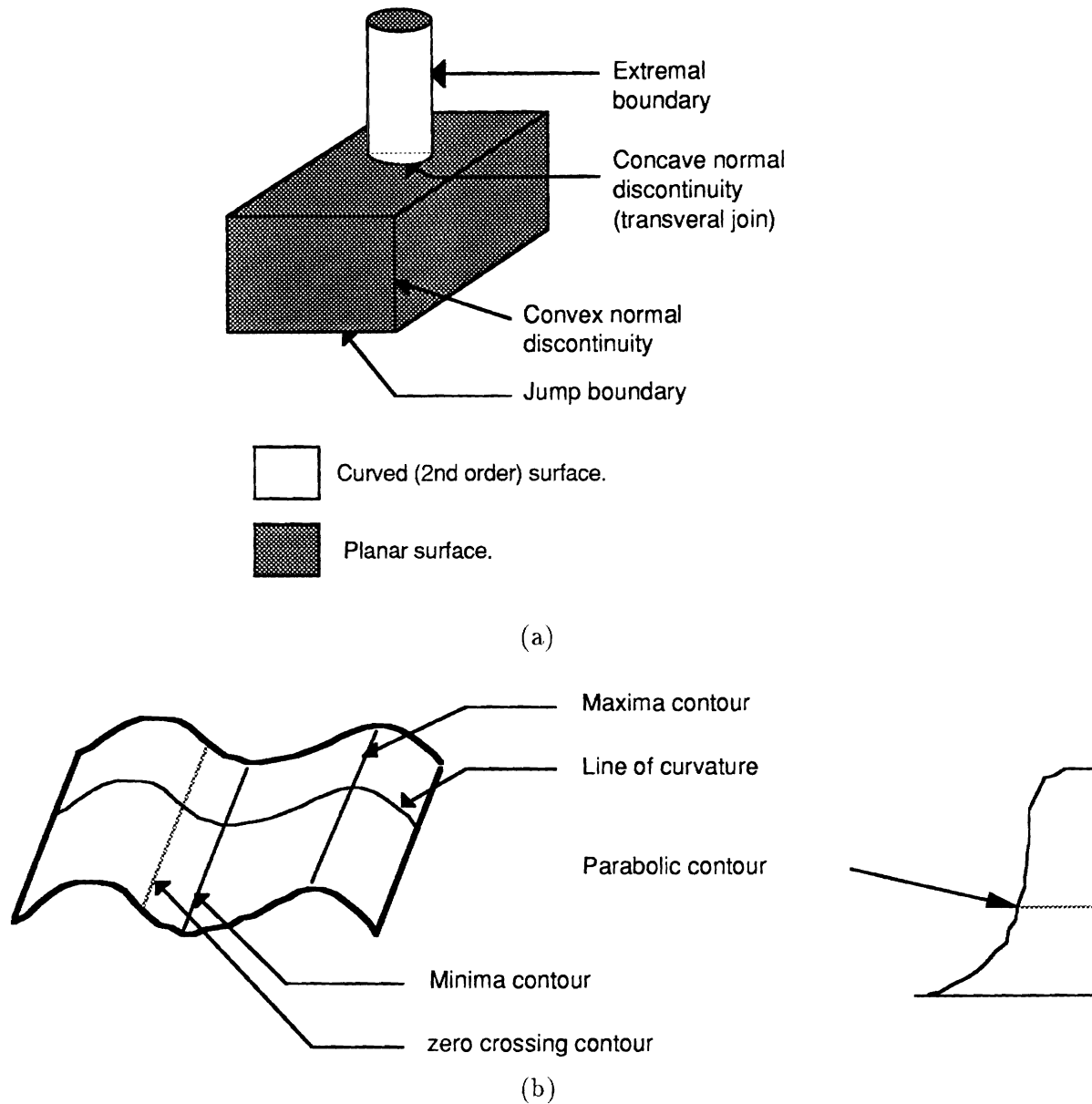


Figure 2.1: **Surface boundaries for part decomposition:** (a) Surface discontinuities (C_0 type) and tangent discontinuities (C_1 type), planar and second-order patches. (b) Smooth boundaries of perceptual significance, like the zero-crossing contours, are also useful as partitioning rules.

the model, and the vector \mathbf{x} denotes a pixel location $(x, y) \in \mathcal{I}$. $p(r)$ is the number of terms in the model of order r . The input image data at \mathbf{x} is given by a function in image space \mathcal{I} as $g(\mathbf{x})$. The models we have chosen are the *variable-order bi-variate polynomials* that are linearly parameterizable in the Euclidean space :

$$\hat{\mathbf{f}}(r, \mathbf{a}, \mathbf{x}) = \sum_{i+j \leq r} \mathbf{a}_{ij} x^i y^j \quad (2.2)$$

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.).

Formulation of the Surface Segmentation Problem

The general segmentation problem is usually stated as follows [Horowitz and Pavlidis 1974, Zucker 1976]. 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 :

$$\bigcup_{i=1}^{N_R} R_i = \mathcal{I} \quad (2.3)$$

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

$$R_i \cap R_j \neq \emptyset, \quad (2.4)$$

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

For all i , the uniformity predicate

$$P(R_i) = \text{TRUE}. \quad (2.5)$$

If R_i is adjacent to R_j ,

$$P(R_i \cup R_j) = \text{FALSE}. \quad (2.6)$$

The uniformity predicate $P(\cdot)$ defines the conformity of all the points in R_i to the global model (primitive). We now examine the various approaches to surface segmentation.

Segmentation by Surface Descriptions

A large portion of computer vision literature is on different methods for surface reconstruction, representation and recognition. The reason for the widespread interest in surface-based object recognition is that this fits well into the prevalent bottom-up approach in vision and that surface is a much more tangible property than volume.

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 a geometric model to guide the segmentation process. There are numerous methods performing the segmentation by aggregating the local surface models like curvature, surface normals, etc. [Besl and Jain 1985, Hoffman and Jain 1987], or by detecting the surface discontinuities (C_0 and C_1 discontinuities and smooth boundaries) [Fan 1988, Godin and Levine 1989, Smith and Kanade 1985].

Approaches based on local differential geometry are the most widely studied techniques for surface segmentation [Brady et al. 1985, Asada and Brady 1986, Ponce and Brady 1984, Besl and Jain 1986, Sander and Zucker 1988, Liang and Todhunter 1990]. They

range from local analysis of the surface to more global interpretation like peaks, pits and passes (saddle points) on a surface [Nackman 1984]. 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, 1988). 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 [1985] 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 [Bajcsy et al. 1990, Faugeras and Hebert 1986, Besl and Jain 1988]. One of the motivations for our approach is illustrated in figure 2.2 showing an object taken from [Fan 1988]. An edge-based method is unable to segment the two planar surfaces (A and B) joined 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 3.8). Thus, edge information is implicit in our primitives. The edges of interest for part segmentation are the C_0 (step edges) and C_1 discontinuities (ramp edges), and the zero-crossing contours on the curved surfaces, as shown in figure 2.1. The zero-crossing contours decompose smooth surfaces into concave and convex parts, and are implicitly detected by bi-quadric primitives.

Much work has been done on the problem of reconstructing piecewise-smooth surfaces in one or more dimensions [Blake and Zisserman 1987, Mumford and Shah 1985, Terzopoulos 1986, Poggio et al. 1985], which is posed as an optimization problem. In all these approaches the data is weighted uniformly which means that the algorithms do not possess

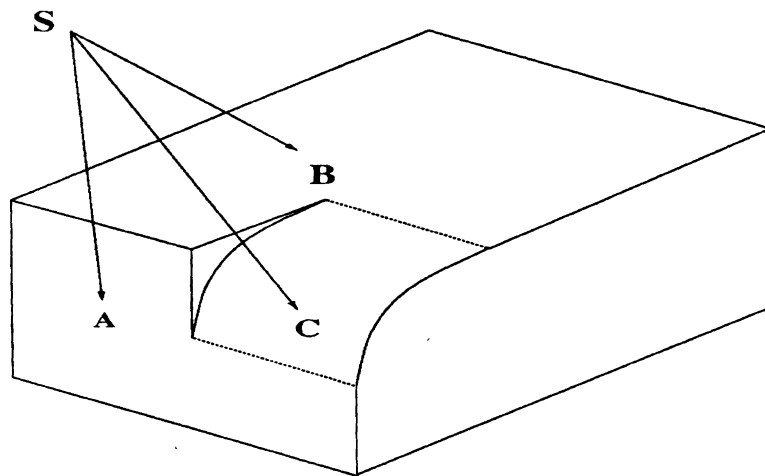


Figure 2.2: **Object with smoothly merging sides:** An example of an object that can not be segmented by edge-based approaches.

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 [1989] 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. Since we want the final segmentation to be geometrically interpretive, such surface reconstruction approaches do not directly relate to our requirements.

2.2.2 Volumetric Model: Superquadric Part-Models

Parametric models like generalized cylinders and their derivatives have been used as volumetric primitives by vision researchers because they give compact overconstrained estimate of overall shape. This overconstraint comes from using models defined by a few parameters to describe a large set of 3-D points. The Volumetric primitives we are using are the *superquadric part-models*. Superquadrics (figure 2.3) have been used in vision [Pentland 1986, Pentland 1987b, Solina 1987, Boulton and Gross 1987] to represent natural part-structure. Pentland [1987b] argues that superquadric part-models possess *descriptive adequacy* though they do not account for every detail of the image data. Also, they are *stable*

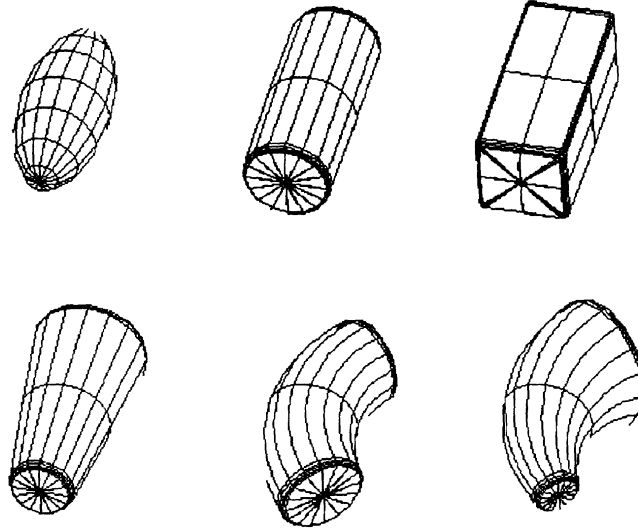


Figure 2.3: **Superquadrics: volumetric primitives:** Clockwise from top : ellipsoid, cylinder, box, tapered and bent model, bent model, tapered model

with respect to scale, noise, and configuration. Solina [Solina 1987, Solina and Bajcsy 1990] has developed a model recovery procedure to fit tapered and bent models to given data. Our SUPERSEG system uses his formulation for the recovery of a single superquadric model for a given set of 3-D points.

Generalized cylinders [Klingenberg 1978] proposed for use in vision by Binford [1971] have been used as volumetric primitives for their rich vocabulary of shapes. However, this vocabulary of shapes is very difficult to recover from vision data, thereby limiting the actual vocabulary to simple linear-straight-homogeneous-cylinders. Deformable models based on generalized cylinders [Terzopoulos et al. 1988, Koenderink and vanDoorn 1979] or superquadrics [Terzopoulos and Metexas 1990] have the disadvantage that they too complex and have so far shown to work only on pre-segmented data. The descriptions generated by our method can be used as starting approximations for the complex deformable models. Superquadric primitives can model only a subset of generalized cylinders shapes, a good compromise for the representation and computational effectiveness. They are capable of modeling global tapering and bending deformations, and are recovered effectively by a stable numerical procedure.

Superquadrics are a family of parametric shapes with a rich vocabulary of part-models

that encompass shapes ranging from cylinders and parallelepipeds to spheres. The representational power is further increased by introducing deformations like bending and tapering along the major axis.

Definition : A superquadric surface is defined by a vector \mathbf{x} sweeping a closed surface in space by varying angles η and ω in the given intervals :

$$\mathbf{x}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\varepsilon_1}(\eta) \cos^{\varepsilon_2}(\omega) \\ a_2 \cos^{\varepsilon_1}(\eta) \sin^{\varepsilon_2}(\omega) \\ a_3 \sin^{\varepsilon_1}(\eta) \end{bmatrix} \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{array}$$

Parameters a_1 , a_2 , and a_3 define the superquadric size in x,y and z direction (in object centered coordinate system) respectively. ε_1 is the squareness parameter in the latitude plane and ε_2 is the squareness parameter in the longitude plane. Based on these parameter values superquadrics can model a large set of standard building blocks, like spheres, cylinders, parallelepipeds and shapes in between. If both ε_1 and ε_2 are 1, the surface defines an ellipsoid. Cylindrical shapes are obtained for $\varepsilon_1 < 1$ and $\varepsilon_2 = 1$. Parallelepipeds are obtained for both ε_1 and ε_2 are < 1 . We have restricted the model recovery procedure to fit the models with $0 \leq \varepsilon_1, \varepsilon_2 \leq 1$.

Segmentation using Superquadrics

Many different methods for partitioning into volumetric primitives have been proposed in computer vision. The common problem with all the volumetric primitives is that, though they are quite rich representations, they are extremely difficult to recover from the real image data. Superquadrics being convex models (except for the bent models), derive a piecewise convex description of the global volume. Therefore, it is natural to consider them as bounding volumes that combine information along convex discontinuities on the surface. One of the drawbacks of the previous methods using superquadric models is that they are unable to combine information along the convex discontinuities (and hence their examples have smooth convex blobs as parts). This is due to the separation of the representation stage from the segmentation process. We now present a review of some of the previous approaches using superquadric models.

Solina [1987] has described a global-to-local method of segmentation using superquadric recovery procedure. His goal was to decompose objects or scenes into parts which can be represented with a single superquadric model enhanced with global deformations such as tapering and bending. When several parts or objects made up of multiple parts were

present, a suitable distance measure was used to decide which 3-D points should be included in a particular volumetric model and which points should be excluded. The method works on some examples, but not on an arbitrary complex object, since it is difficult to constrain the minimization procedure to take part-structure into account.

Pentland [1988] has described a two-part procedure to recover segmented descriptions of complex objects. His approach is first to recover part-structure by matched filtering and maximum likelihood estimation, and then to describe parts by superquadrics using a least squares procedure. Only occluding boundary data is used, though he noted that surface information will be useful in extracting complete part-structure. The procedure is extremely slow on sequential machines. Another method by Pentland [1987a] uses range data to recover part-structure. The method works by skeletonizing the parts, recovering the models along the skeleton (for efficiency's sake), and then performing refinement of the initial models by a gradient descent procedure. The final description is obtained by selecting the "best" models among the global models. The method can account for occlusion of a model due to other parts, and is inherently parallel, but it requires skeletonization (which was done by hand) of the parts to be computationally feasible. This means that the method assumes that coarse segmentation of the part-structure is available, which is a difficult problem in case of range images (where parts may not appear in the silhouette).

In a system developed for the purpose of automating the sorting of mail pieces for the United States Postal Service, superquadrics were used only for modeling and classification, while the segmentation of the postal scene was performed using edge-based methods [Gupta et al. 1989b, Gupta et al. To appear, Bajcsy et al. 1990b]. The procedure segmented objects at jump boundaries, and recovered superquadrics for individual objects after reasoning about the depth of the object from the available information. Ferrie et al [1989] also use superquadrics only as a final modeling primitive, and not as a means to arrive at a segmentation. The surface segmentation is performed by following a differential-geometric approach developed by [Sander and Zucker 1988], and edges are detected at surface intersections to form closed convex patches. The convex patches are then modeled by superquadrics recovered using Solina's formulation. Due to the isolation of the modeling primitive from the segmentation, the models may not correspond to the segmented data. The problems inherent in such approaches is that a one-to-one correspondence between superquadric models and the surface segmentation is assumed, which is not true in case of objects with planar faces. This problem is also evident in the superquadric descriptions achieved by the procedure developed by Darrel et al [1990]. Our paradigm solves this problem by following

a systematic global-to-local volumetric segmentation *using* the superquadrics to drive the segmentation, and not just as final modeling primitives. Biederman [1987], in his theory of Recognition-By-Components has suggested an edge and volumetric primitive (generalized cylinders) based approach for describing complex objects in intensity images. He however, does not describe any procedure to recover such complex part-structure.

The process of obtaining superquadric models uses least squares minimization for recovery of model parameters. An important advantage for ease of model recovery is that the superquadric surface is defined by an analytic function, differentiable everywhere. Superquadric shapes form a subclass of shapes describable by generalized cylinders. Shape deformations like bending and tapering can be defined with global parametric deformations. Superquadrics with parametric deformations encompass a large variety of natural shapes yet are simple enough to be solved for their parameters. Due to their built-in symmetry, superquadric models predict the shape of occluded parts conforming with the principle of parsimony - among several hypotheses select the simplest [Gombrich 1972]. Except for bending, the shape vocabulary consists of convex objects.

An issue to be resolved by the control module is, how to deal with *concavities*, *cavities*, and *holes*? Cavities form when a significant chunk of volume is taken away from the object leaving a dent enclosed by the remaining object (bowl or cup). Solina [Solina 1987, Solina and Bajcsy 1990] developed a recovery procedure to identify the presence of cavities in segmented objects and model them as superquadrics. Concavities (a circular cut-out of a box) form by a similar process but they are not enclosed completely by the object, so they are visible in the 2-D projection of the object. If a model exists for a concavity or hole (like for objects with cylindrical hole), it can be modeled as a negative volume. For example, the circular cut-out can be modeled as a boolean subtraction of a box and an elliptical cylinder, such that the points on the box that belong to the cylinder are not considered as part of the model. The superquadric inside-outside function presents a convenient formulation of negative volume. Thus, the descriptions can be combined in the constructive solid geometry (CSG) sense, where superquadrics represent the primitive models and the regularized intersection, union and subtraction are formulated in terms of the inside-outside function. Our hierarchical representation of the superquadric part-structure enables us to directly perceive the description as composed of CSG operations [Requicha 1980, Woodwark 1989, Kapur and Mundy 1989].

2.3 Chapter Summary

The two shape primitives that we are using for part-segmentation are bi-quadrics and superquadrics. In the next two chapters we describe a novel approach to surface segmentation and techniques for using the surface information for volumetric segmentation. The recovery procedure for one superquadric model for the given data is described in chapter 5. The complete problem of performing volumetric part-segmentation is described in chapter 6 and 7.

CHAPTER 3

Surface Segmentation: The Search for the Best Description

In this chapter, we present a novel method for surface segmentation in range images¹. Following our segmentation paradigm, we view surface segmentation as a local to global aggregation process, needing various similarity criteria 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 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 may lead to disastrous results [Chen 1989]. 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 [Bajcsy et al. 1990].

The uniqueness of our approach lies in defining surface segmentation as partitioning the range data 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

¹The work described in this chapter was done jointly with Aleš Leonardis.

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. The method performs data aggregation via model recovery in terms of variable-order (up to second-order) bi-variate 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 this approach is the development of an extremely simple and robust control structure that combines model extraction and model selection in a dynamic way, allowing only the “best” models to develop further.

The procedure has three major components, viz. model recovery, model selection, and the control structure that dynamically combines model recovery and selection. After discussing these components, we present results on real range images.

3.1 Segmentation and Model recovery

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 [Besl and Jain 1988] and for contour segmentation by [Chen 1989]. Our approach differs from [Besl and Jain 1988] in the selection of initial estimates (seed regions), search for connected compatible points, highest order of the polynomial; and from [Chen 1989] 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 [1981] in that the seed regions are not selected at random and the model is updated during region growing depending on the data.

3.1.1 Surface Fitting

Using the notation introduced in chapter 2, we now formulate the surface recovery problem. The *variable-order bi-variate polynomials*, linearly parameterizable in the Euclidean space are :

$$\hat{f}(r, \mathbf{a}, \mathbf{x}) = \sum_{0 \leq i+j \leq r} \mathbf{a}_{ij} x^i y^j \quad (3.1)$$

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-quadric surfaces.

A linearly parameterizable surface patch $\mathcal{S}(r, \mathbf{a}, \mathbf{x})$ can be written as :

$$\mathcal{S}(r, \mathbf{a}, \mathbf{x}) = \{(\mathbf{x}, \mathbf{z}) \in \mathcal{I} \times \mathcal{Z} \mid \mathbf{z} = \hat{\mathbf{f}}(r, \mathbf{a}, \mathbf{x})\} \quad (3.2)$$

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

$$d^2(r, \mathbf{a}, \mathbf{x}) = [g(\mathbf{x}) - \hat{\mathbf{f}}(r, \mathbf{a}, \mathbf{x})]^2 \quad (3.3)$$

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

$$\chi^2(r, \mathbf{a}, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} d^2(r, \mathbf{a}, \mathbf{x}) \quad (3.4)$$

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 SSE function $\chi^2(r, \mathbf{a}, \mathcal{D})$. Using least-squares regression we get :

$$\chi^2(r, \hat{\mathbf{a}}, \mathcal{D}) = \min_{\mathbf{a} \in \mathcal{A}} \chi^2(r, \mathbf{a}, \mathcal{D}) \quad (3.5)$$

We use the standard technique for solving the *General Linear Least Squares Problem*. The solution is given in detail in Appendix A. 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 [Chen 1989]. As mentioned earlier, this approach 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 role in the segmentation process. Our 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 [Besl and Jain 1988].

3.1.2 The Model Recovery Algorithm

Surface segmentation starts at a seed region by iteratively growing it as outlined in algorithm 3.1. The schematic diagram of the algorithm is shown in figure 3.1. An important

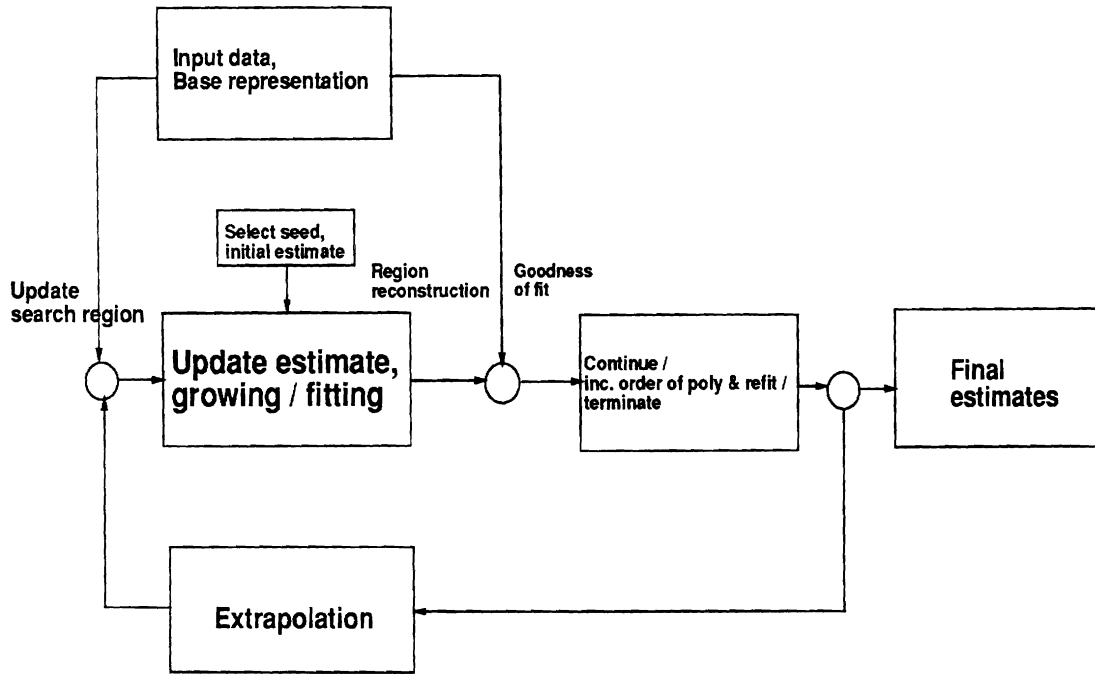


Figure 3.1: **Iterative regression approach to model-recovery:** For one model.

problem in region growing method is how to select the seed regions to start the local to global aggregation. It is even more critical in an iterative approach since the starting region determines the initial estimates of the parametric models, which determine the entire course of its growth. Placing the seed regions arbitrarily on the image cannot guarantee a complete description of the image, since some of the regions can be left undescribed. Placing seed regions intelligently, such that all the regions are covered requires a priori knowledge of those parts, which is nothing but segmentation. This catch-22 problem can be solved either by smart selection of the seeds by computing the primitive-related properties and placing the seeds in the pre-processed image or by taking the brute-force approach of placing the seeds everywhere in the image thereby ensuring that all the possibilities are considered. Besl and Jain [1988] 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.

1. Initial region $\mathcal{D}^{(0)}$ (seed) is a small window whose size is determined on the basis of scale considerations and can be adaptively changed depending on the data.
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 :

$$\mathcal{C}^{(s)} = \{\mathbf{x} \mid d^2(r, \mathbf{a}, \mathbf{x}) \leq C \text{ and } \mathbf{x} \in \text{4_conn_neighborhood_of}(\mathcal{D}^{(s)} \cup \mathcal{C}^{(s)})\} \quad (3.6)$$

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 3.6 that

$$(\mathbf{x} \text{ is no further than } k \text{ pixels from the nearest border point in } \mathcal{D}^{(s)}) \quad (3.7)$$

- 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:

$$\mathcal{S}^{(s+1)}(r, \mathbf{a}^{(s+1)}, \mathcal{D}^{(s+1)}) \quad \text{where} \quad \mathbf{a}^{(s+1)} = \mathbf{a}_{\mathcal{D}^{(s+1)}} \quad (3.8)$$

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

$$\rho^{(s+1)} = \chi^2(r, \mathbf{a}^{(s+1)}, \mathcal{D}^{(s+1)}) - \chi^2(r, \mathbf{a}^{(s)}, \mathcal{D}^{(s)}) \quad (3.9)$$

- 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} .

To avoid the problems related to smoothing and local curvature computation, we place fixed size (7x7 or 5x5) 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. Such a check is, however, not required 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, and are discarded by the model selection procedure. Thus, the complete model recovery procedure consists of the following two steps :

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 (algorithm 3.1) is invoked.

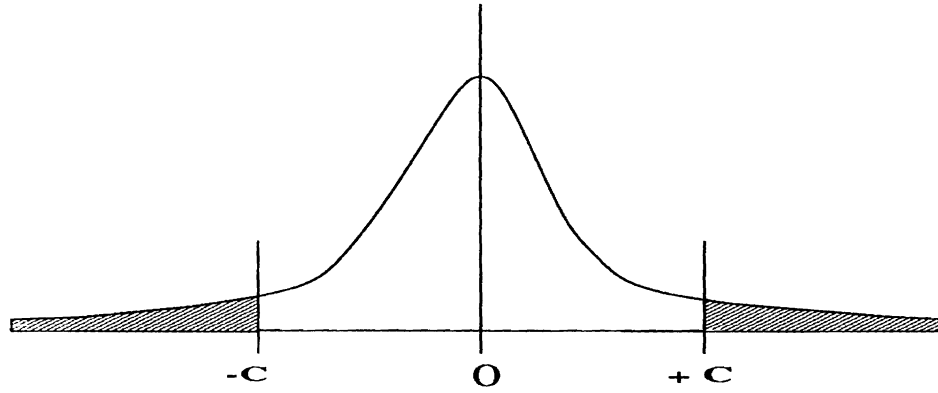
Let $\mathcal{D}_i^{(s)}$ denotes the set of points which are used for the estimation of the parameters $\mathbf{a}_i^{(s)}$ of the i -th model at the s -th iteration. Notice that the regions can overlap partially or completely, so $(\mathcal{D}_i^{(s)} \cap \mathcal{D}_j^{(s)}) \neq \emptyset$ in general.

3.1.3 Features of the Segmentation 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 :

$$\mathcal{D}^{(s)} \subseteq \mathcal{D}^{(s+1)} \subseteq \mathcal{I} \quad (3.10)$$

Figure 3.2: Noise distribution and the role of C .

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 3.2). This is an important feature since least-squares fitting has undesirable sensitivity to outlying points, and measurement errors are not necessarily normally distributed [Bajcsy et al. 1986].

Computational Complexity:

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

$$\mathcal{O}(np^2) + \mathcal{O}(p^3) \quad (3.11)$$

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 3.11. 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:

$$\mathcal{O}(n'p^2) + \mathcal{O}(p^3) \quad (3.12)$$

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.

The final output of the segmentation algorithm consists of all the recovered models that are potential candidates for the final description of the data. Selection of the models is achieved by maximizing a quadratic Boolean function described in the following section.

3.2 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 [1990] . 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 a 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.

3.2.1 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 ($n_i = |\mathcal{D}|$) 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.

$$F(m_i) = K_1 n_i - K_2 \xi_i - K_3 N_i \quad (3.13)$$

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 . ξ_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 [1990], 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 :

$$F(\tilde{\mathbf{m}}) = \begin{bmatrix} \tilde{m}_1 & \dots & \tilde{m}_M \end{bmatrix} \begin{bmatrix} c_{11} & \dots & c_{1M} \\ \vdots & & \vdots \\ c_{M1} & \dots & c_{MM} \end{bmatrix} \begin{bmatrix} \tilde{m}_1 \\ \vdots \\ \tilde{m}_M \end{bmatrix} \quad (3.14)$$

$$F(\tilde{\mathbf{m}}) = [\tilde{\mathbf{m}}]^T [\mathbf{Q}] [\tilde{\mathbf{m}}] \quad (3.15)$$

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 :

$$c_{ii} = K_1 n_i - K_2 \xi_i - K_3 N_i \quad (3.16)$$

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

$$c_{ij} = (-K_1 \Gamma(m_i, m_j) + K_2 \xi_{i,j})/2 \quad (3.17)$$

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

$$\xi_{i,j} = \xi_i + \xi_j - \frac{\sum_{n_i} d_i^2 + \sum_{n_j} d_j^2 - \max(\sum_{\Gamma(m_i, m_j)} d_i^2, \sum_{\Gamma(m_i, m_j)} d_j^2)}{n_i + n_j - \Gamma(m_i, m_j)} \quad (3.18)$$

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.

3.2.2 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 [1990] 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.

The optimization is performed by a direct descent algorithm. We observed that if k_3 is decreased in large steps, the solution gets stuck at 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:

$$(\text{no. of steps of } k_3) \times M \times (\text{evaluation of the matrix } \mathbf{Q}) \quad (3.19)$$

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 3.2 that is based on two assumptions:

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

Algorithm 3.2 is computationally inexpensive. The computational complexity of our method is proportional to:

$$(\text{number of models in the final description}) \times (\text{evaluation of the matrix } \mathbf{Q}) \quad (3.20)$$

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 3.3 (a). As a consequence of the selection process, eventually very few of the regions emerge as acceptable descriptions of the data. However,

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

```

old_value_of_objective_function = new_value_of_objective_function;
procedure_is_done = true;
for all the models do
    find_model_contributing_most_to_the_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( Selected model, Value of the objective function);
    procedure_is_done = false;
end_if
while (not procedure_is_done).
```

Algorithm 3.2: Bi-quadric model selection.

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.

3.3 Dynamically Combining Model Recovery and Selection

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 3.3a. 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

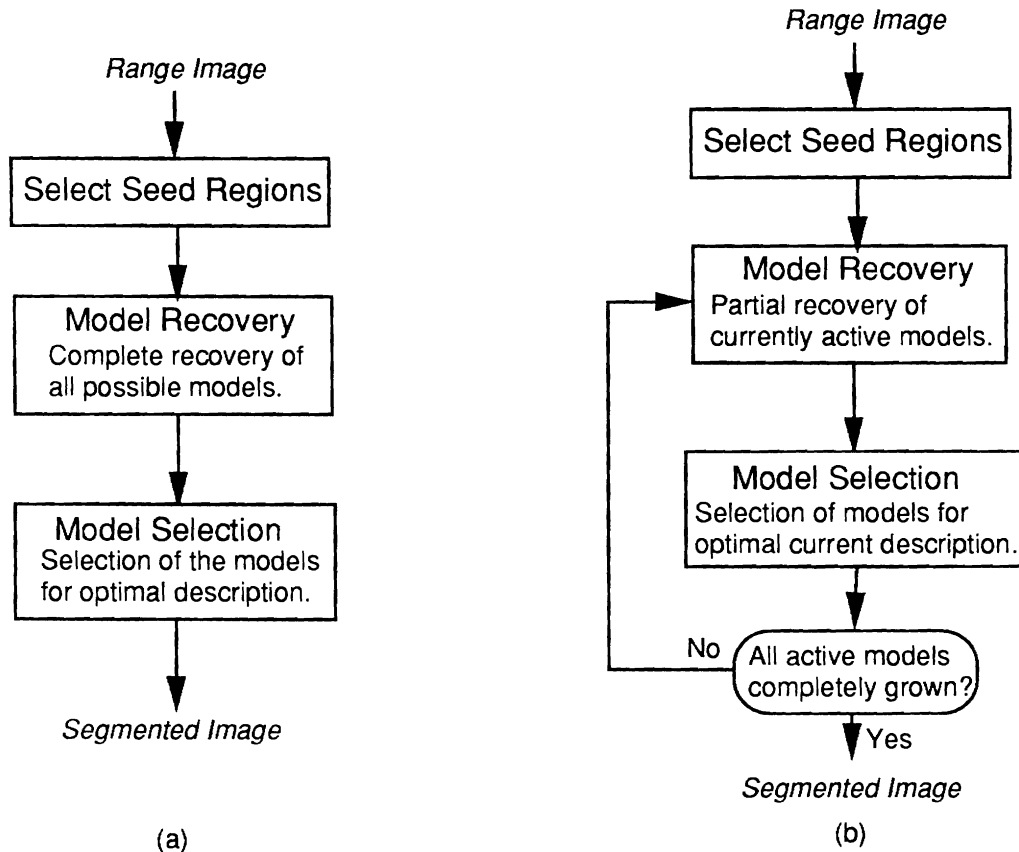


Figure 3.3: **Model recovery and selection:** (a): The Recover-then-Select paradigm. All the models are fully recovered before model selection is invoked. (b): The Recover-and-Select paradigm. Models are selected before they are fully grown, thereby reducing the number of active models.

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 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 3.3b) is called *Recover-and-Select*.

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

Table 3.1: Trade-offs in combining model recovery and model selection procedures.

3.3.1 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 **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 **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 3.1. 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 3.7 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, the weights are changed to eliminate duplicate regions describing a patch with considerable overlap and similar global error.

3.4 Example Surface Descriptions

The Recover-and-Select paradigm was tested 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 an IBM-6000 RISC machine or a Sparcstation-2.

In this section we present a few examples that illustrate the most important aspects of the surface segmentation algorithm. Results on other objects are displayed in chapter 8. The images 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 ± 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.

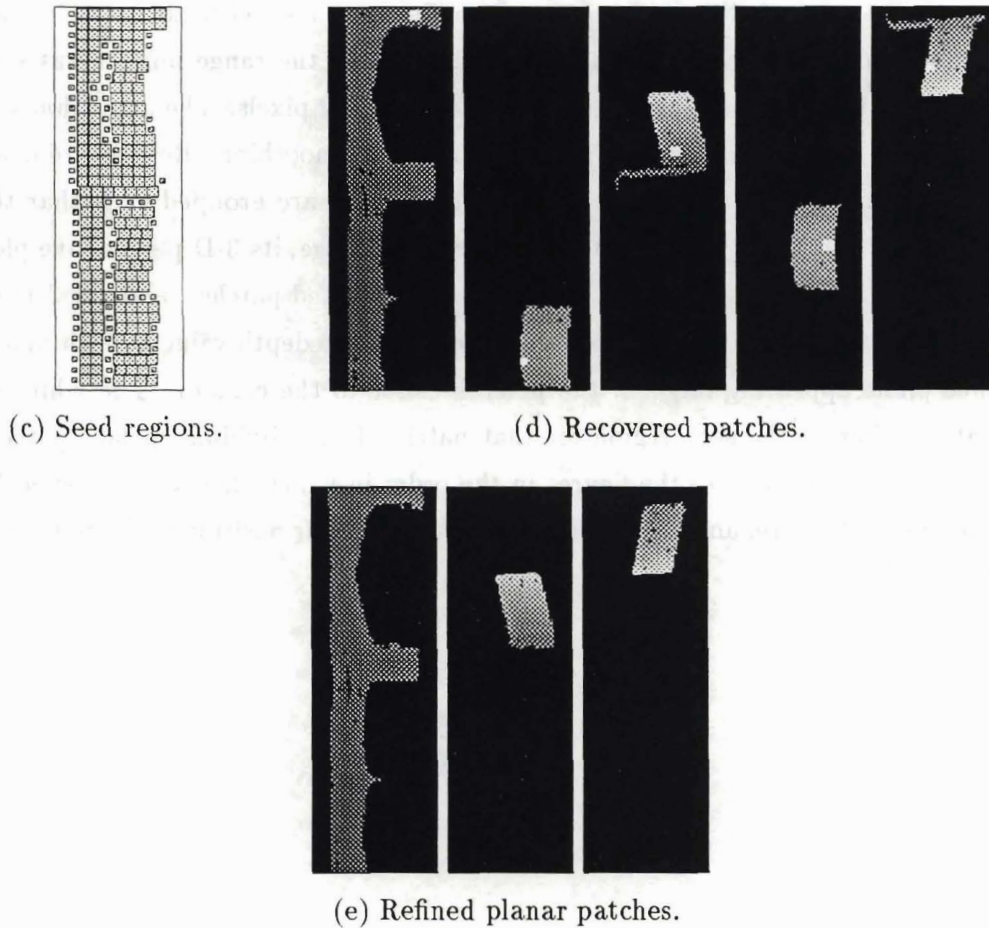
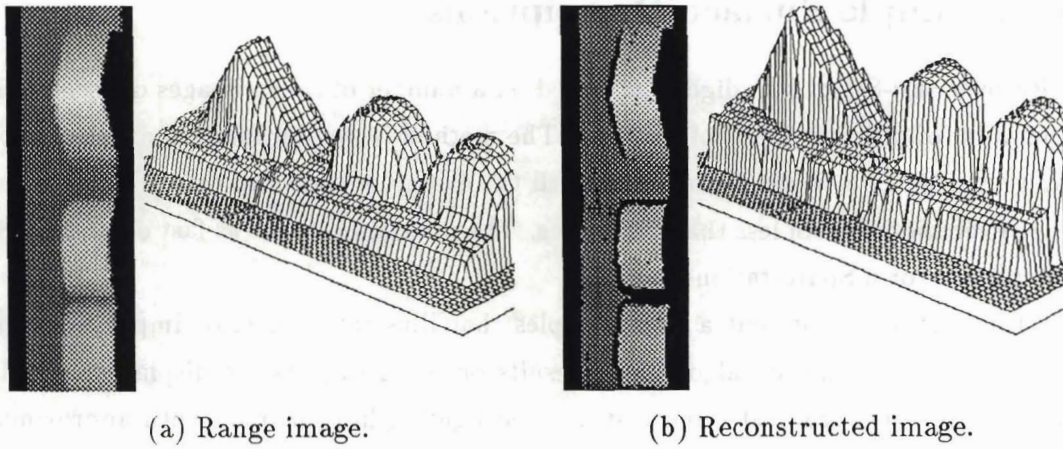


Figure 3.4: **Scene 1:** The cylindrical surfaces are modeled as bi-quadric patches. C_1 discontinuities at planar intersections are reliably recovered. Refinement of the patches is done by using a technique described in the next chapter.

3.4.1 Scene 1

This range image consists of a triangular prism and two half-cylinders placed on a box (figure 3.4). 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 3.4. The planar regions on the prism extend along the intersection with the box. Using the procedure for region refinement described in chapter 4, the overlap due to the intersection of the regions 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.

3.4.2 Scene 2

This is a complex image for any surface modeling algorithm, primarily because of the smooth boundaries (zero-crossing contours of the lines of curvature) 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 3.5, 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 is a first-order patch flanking the convex second-order patch 3. Adding it to the final description increases the accuracy of description of the convex patch which is curving faster than the bi-quadric surface. It is 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 3.5 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 global error.

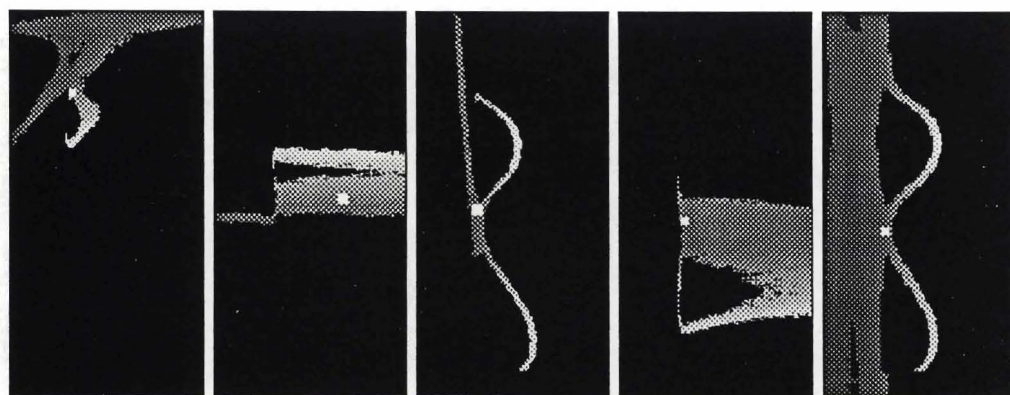
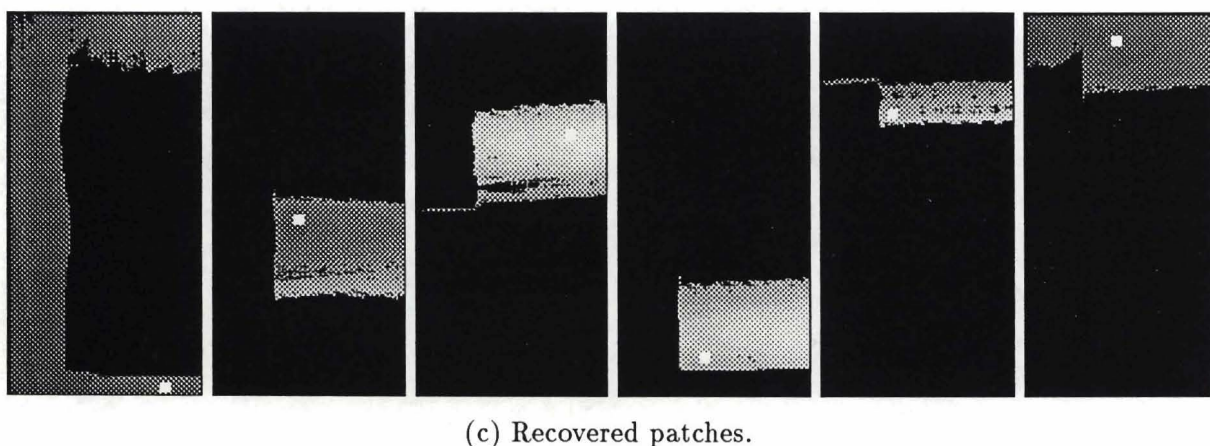
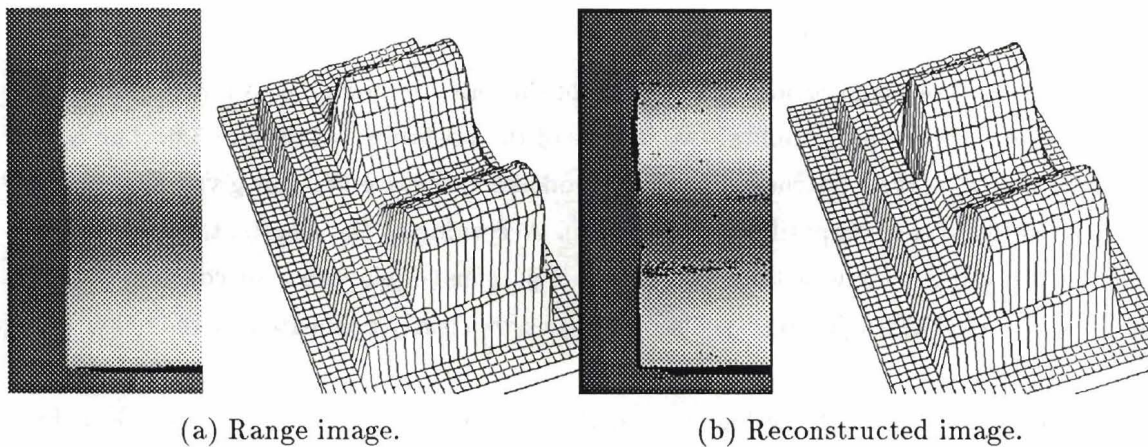


Figure 3.5: **Scene 2:** Undulated surface smoothly merging into a planar surface: Segmentation is achieved in terms of convex and concave parts of the surface. Some of the rejected patches are shown in the bottom row.

3.4.3 The Coffee-mug

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 3.6. 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.

It is possible to restrict the highest order during the model recovery to zeroth or first, resulting in piecewise-constant and piecewise-planar descriptions respectively. This restriction can be trivially imposed in the recover-and-select paradigm by changing the *max_order* value to 0 or 1 in step 3.3(iii) of algorithm 3.1. Starting with the same seed regions, the piecewise constant and piecewise first-order descriptions are obtained for the coffee mug (figure 3.6). 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.

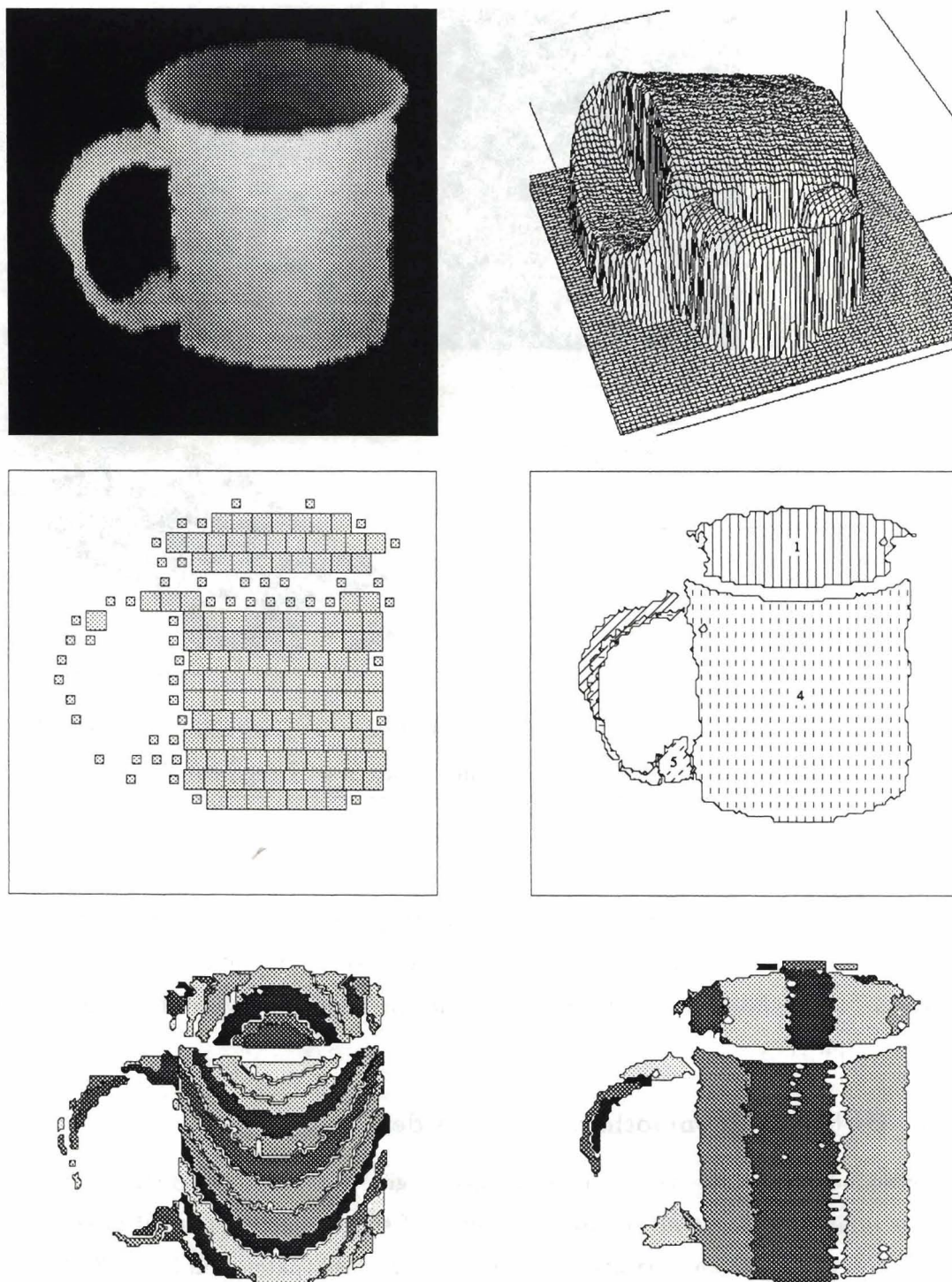


Figure 3.6: **The coffee-mug:** Top: Range image and its 3-D plot. Center: Seed regions and the bi-quadric segmentation. The highly curved handle is modeled as a combination of the smaller patches. Bottom: The piecewise-constant (left) and piecewise-planar (right) approximation of the curved surfaces.

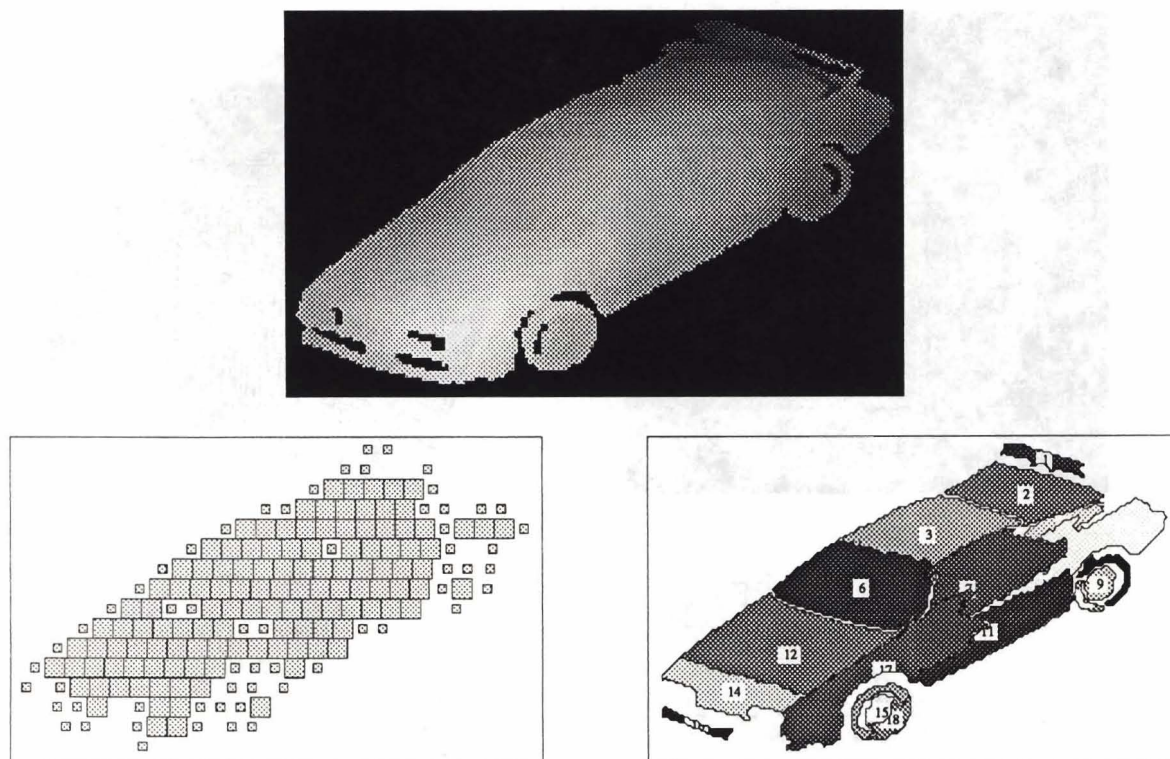


Figure 3.7: **The car:** Top: Range image. Bottom: Seed regions (left) and the final segmentation (right). Surfaces constituting the car are neatly segmented.

3.4.4 The Car

The surfaces constituting the car in figure 3.7 (range image provided by USC) are segmented neatly, even though the boundaries separating them are not sharp (as evident from the placement of the seeds). The surface details on the side of the car are described by overlapping patches.

3.4.5 Object with smoothly merging sides

As mentioned earlier, the second-order surface (region C) in figure 3.8 (taken from [Fan 1988]) is difficult to segment due to the absence of step or surface normal discontinuities. Our method gives a clean separation of the curved surface (patch 2 in figure 3.8) 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*.

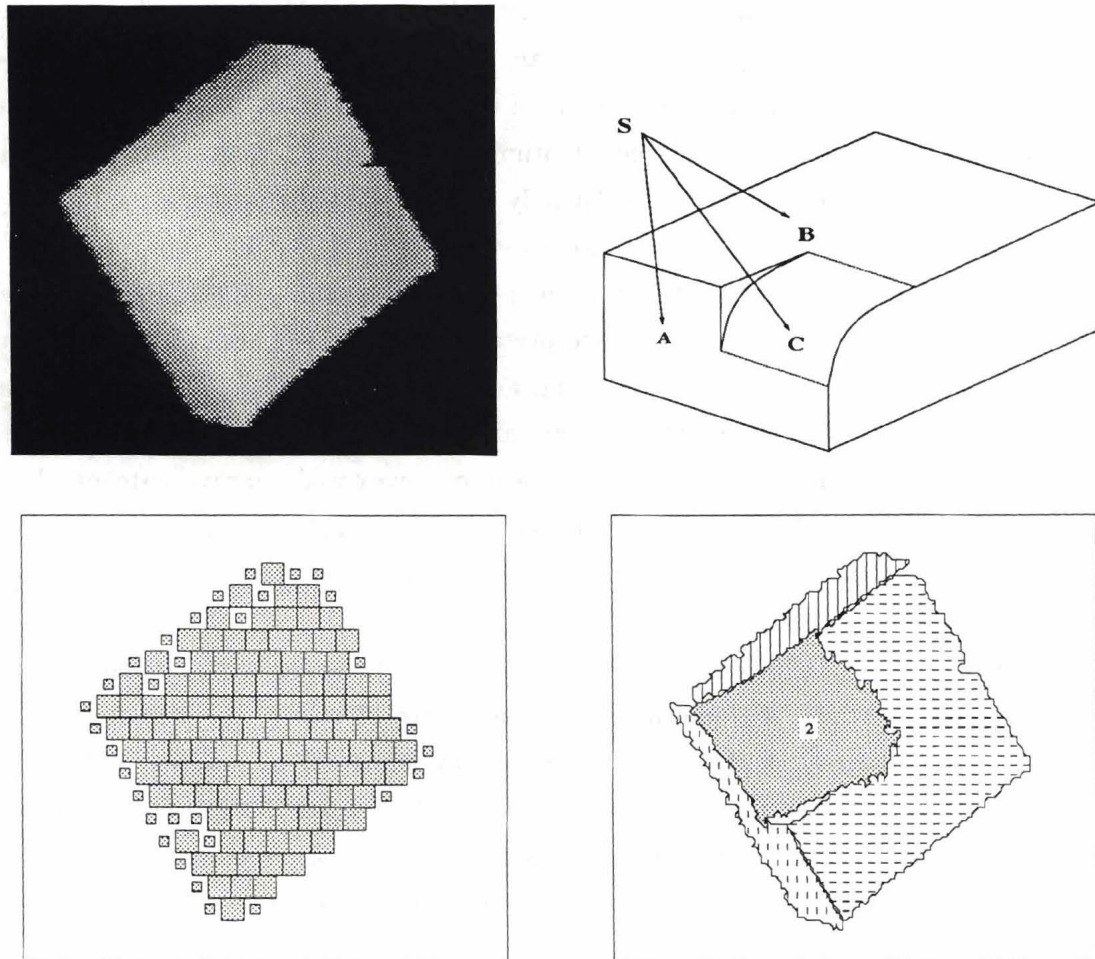


Figure 3.8: **Object with smoothly merging sides:** Top: Range image and a sketch of the object. Bottom: The seed regions and the final segmentation. The second-order patch is segmented from the planar patches.

3.5 Chapter Summary

We presented a novel approach for local to global data aggregation in terms of bi-quadric patches. The iterative approach combining data classification and model fitting shows that segmentation and modeling are not two independent procedures but have to be integrated. 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. 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.

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 segmentation occurs at the surface normal discontinuities or at the zero-crossing contour to divide the surface into convex and concave patches. In the next chapter, we will analyze these descriptions to label surface discontinuities as convex or concave, and also present methods to interpret the bi-quadric patches for further use in volumetric segmentation.

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-quadric patches which do not result in a simple description, signaling that different kind of primitives should be invoked. 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.

CHAPTER 4

Deriving Surface Properties from Bi-quadrics

So far we have a piecewise, and possibly overlapping segmentation of the range data into bi-quadric patches. In order to use this description for the recovery of superquadric models, we have to refine the segmentation and derive as much information about the surface attributes as possible. The only relevant refinement that needs to be done concerns the overlapping of the bi-quadric patches along their intersection curve. The important surface attributes are region-adjacency information, edge localization, and determining if it is convex or concave, major axis determination, and determining if the surface is convex or concave. In this chapter we will derive analytical expressions to compute these important surface attributes and discuss their relevance in superquadric fitting. We start with presenting our general algorithm for the refinement of surface patches which is also used for determining edge-type at the intersection of two patches.

4.1 Refining Regions Along Intersection Curves

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. There are two kinds of overlaps that need to be treated differently.

The first kind occurs when two regions have significant amount of overlapping domain due to the smoothness of the underlying data with no clear segmentation possible (as in figure 3.5). It is not possible to resolve the overlapping portion without considering

additional constraints or domain knowledge. In this case, the overlapping points are decided by the volumetric segmentation described in chapter 7.

In this section, we are concerned with the overlapping points occurring due to the compatibility criteria employed during the region-growing process. The compatibility constraint (C in equation 3.6) only accounts for conformity of the point to the model and not the shape of the domain. Thus, the region A's geometrical domain consists of all the points that satisfy the compatibility constraint, including the points lying along the intersection curves of all the regions that intersect with A. This can further result in two cases as illustrated in figure 4.1. Region A and B are both planar patches, with area C consisting of the common points between A and B. The intersection curves for the image of scene 1 are shown in figure 4.2(a). Planar patches 1 and 2 intersect with planar patch 3, as also the second-order patch 5. The two cases are:

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

Although the type I overlap is geometrically correct, it is perceptually unacceptable, and must be removed before the region A can be useful in any way. Type II overlap, on the other hand, is mostly harmless, since it extends only a few points along the region boundary and does not significantly alter the interpretation of the segmented regions. In fact, it provides useful information about the connectivity of the regions and implicitly detects ramp edges (which exist at the intersection) by distinguishing them from the step edges along which two regions would never overlap.

Both types of overlaps 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. The analytical computation of the intersection curve of two quadric surfaces is quite involved and is well understood in CAD literature [Luh and Krolak 1965, Comba 1968, Mahl 1972, Levin 1976, Levin 1979]. Fortunately, the problem of two intersecting bi-quadric surfaces is simpler, since the intersection is at most a second-order curve lying in XY plane.

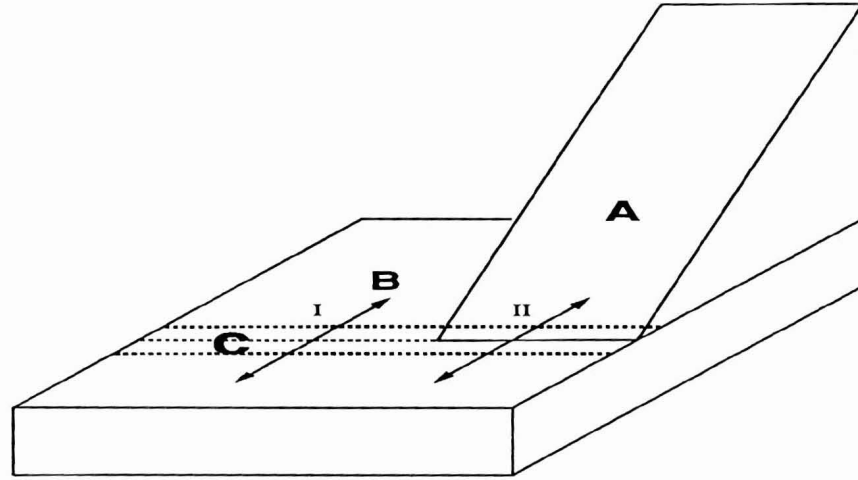


Figure 4.1: Overflow of one region into another along the intersection curve.

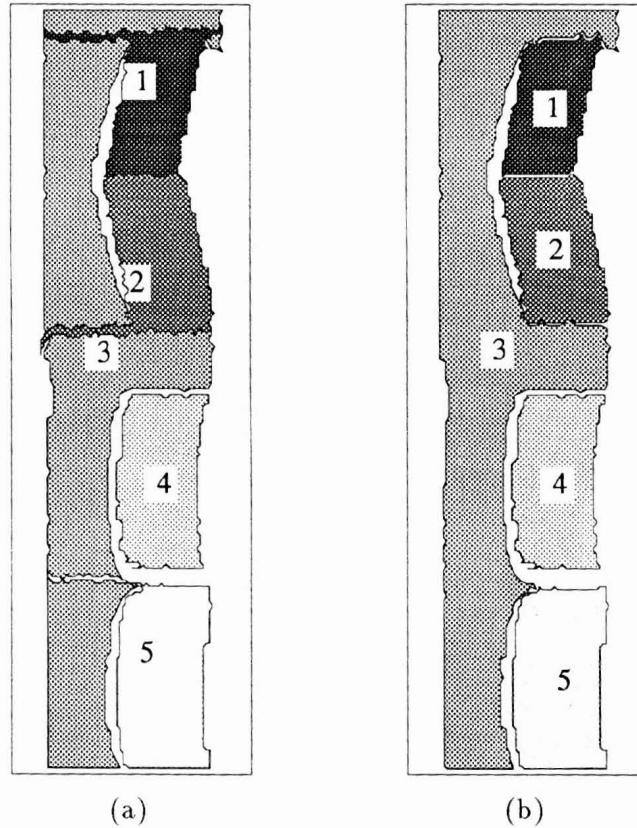


Figure 4.2: **Example of intersection cleaning on a real range image:** (a): Output of the Recover-and-select procedure for scene1. (b): Output after Refining patches 1, 2, 3 and 5 along the intersection curves.

1. Determine the pixels p_i that are on the intersection curve.

2. **foreach**(p_i) **do**

Compute the normal to the curve.

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.

If (both end-pixels belong to the same region)

then /* Type I intersection */

Assign all the traced pixels to that region

else /* Type II intersection */

Distribute the traced pixels between the two regions.

3. Intersection cleaning completed.

Algorithm 4.1: Region refinement along the intersection curve.

For two surfaces $S(r, \mathbf{a}, \mathbf{x})$ and $S(r, \mathbf{b}, \mathbf{x})$ of up to second-order, the intersection curve is given by :

$$C(\mathbf{a}, \mathbf{b}, \mathbf{x}) = \sum_{1 \leq i+j \leq r} \mathbf{a}_{ij} x^i y^j - \sum_{1 \leq i+j \leq r} \mathbf{b}_{ij} x^i y^j = 0 \quad (4.1)$$

Thus, given a point (x, y) , it is possible to check if it lies on the intersection curve of the two surfaces. The curve in equation 4.1 is planar and at most second-order. The analytical form of the curve generator also gives the closed form solution for the normal to the curve, which is used for refining the surface labeling.

The equation of the normal at (x, y) is given by :

$$N(X, Y) = a'X + b'Y + c' = (X - x) \frac{\partial C}{\partial y} - (Y - y) \frac{\partial C}{\partial x} = 0 \quad (4.2)$$

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 refinement along their intersection curve. 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 steps in algorithm 4.1 are invoked.

In figure 4.1, points in C that should only be labeled as belonging to B are assigned to B (case I). For case II, points on the two sides of the intersection curve are distributed between A and B by the algorithm. In figure 4.2, the intersection curves of regions 1, 2 and 5 with region 3 are cleaned to give the refined regions. This procedure has been tested on numerous images with excellent results. The method of intersection tracing is also used later in determining the type of edge at region intersections.

4.1.1 Another Method for Region Refinement

The post-processing described in this section is required because of the absence of the constraints to limit the domain of the geometric patch. By definition, the domain extends to all the pixels satisfying the geometric model. Another way to deal with this problem is to prevent it from occurring in the first place. From the observation that intersection regions are invariably narrow in width (determined by the value of the compatibility constraint), we can prevent the geometric patches to grow into regions that are not sufficiently wide. This has the effect of inhibiting the region growth along narrow appendages. It is simple to incorporate such a check in the region growing process. We have implemented it by simply requiring that every new pixel incorporated in the patch should have at least p pixels in its 5×5 neighborhood. The only drawback of this constraint is that it puts a non-zero lower bound on the width of the acceptable regions.

4.2 Region Refinement by Relaxing Compatibility Constraint

Due to the iterative regression approach, a new point is included in the domain of the region only if it satisfies the equation 3.6. Most of the data points that do not satisfy this strict compatibility criterion can be included in the region by relaxing the compatibility constraint C after the termination of the recover-and-select procedure. However, the surface parameters are not reestimated to take those points into account. This has the effect of including the missing points that narrowly miss the compatibility test, and giving clean segmentation results. Another way could be to interpolate or extrapolate the surface patch to include such points.

4.3 Bi-quadric Surface Types

The bi-quadric surface segmentation consists of a piecewise description in terms of following two types of patches expressed in their general form as:

1. *Planar patches*: Zeroth or first-order, represented as,

$$z = ax + by + c$$

2. *Curved patches*: Second-order, represented in general form as,

$$z = ax^2 + by^2 + cxy + dx + ey + f \quad (4.3)$$

There are three types of curved bi-quadric patches, viz. elliptic paraboloid, hyperbolic paraboloid, and the elliptic cylinder, as shown in figure 4.3. The surface type can be determined from the sign of the quantities that are invariant with respect to the translation and rotation transformations. For bi-quadrics, the sign of the following invariant quantity determines the surface type :

$$J = \begin{vmatrix} a & c/2 \\ c/2 & b \end{vmatrix} = ab - \frac{c^2}{4}. \quad (4.4)$$

A second-order surface can therefore be classified as :

$$J > 0 \quad : \text{Elliptic Paraboloid}$$

$$J = 0 \quad : \text{Parabolic Cylinder}$$

$$J < 0 \quad : \text{Hyperbolic Paraboloid} \quad (4.5)$$

Because there is no second-order term involving z in a bi-quadric surface, they belong to the class of non-central quadrics. The general form of the bi-quadric surface has xy term signifying the rotation of the X and Y axes in the world coordinate system (see figure 4.4). In addition, the linear terms constitute the translation component. Since every bi-quadric has a standard form without the cross-terms, it is possible to transform the general form into the standard form and obtain the orientation information as a result.

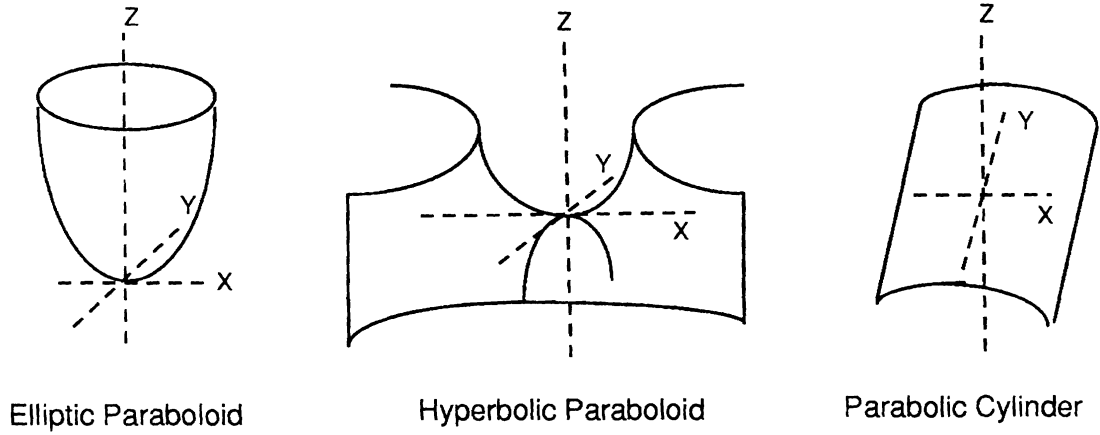


Figure 4.3: The three bi-quadric surface types.

4.3.1 Transformation to Standard Form

The transformation is accomplished by translating and orienting the patch to remove the linear and cross-multiple terms in the equation. Appendix B outlines the procedure to reduce the bi-quadrics to their standard forms.

The standard forms of the three basic bi-quadric surfaces are given by:

$$\text{Elliptic Paraboloid: } z = \frac{x^2}{a'^2} + \frac{y^2}{b'^2}$$

$$\text{Hyperbolic Paraboloid: } z = \frac{x^2}{a'^2} - \frac{y^2}{b'^2}$$

$$\text{Parabolic Cylinder: } z = \frac{x^2}{a'^2} \tag{4.6}$$

Based on the second-order coefficients, the surface types can be classified, as shown in table 4.1. We are now ready to use the standard form of the three types of bi-quadrics to derive the orientation and the type of surface embedding, to facilitate volumetric segmentation.

a'	b'	$ a' > b' $		$ b' > a' $		bi-quadric type
		Type	Axis	Type	Axis	
+	+	concave	Y	concave	X	elliptic paraboloid
-	-	convex	Y	convex	X	
+	-	concave	Y	convex	X	hyperbolic paraboloid
-	+	convex	Y	concave	X	
+	0	concave	Y	-	-	parabolic cylinder
-	0	convex	Y	-	-	
0	+	-	-	concave	X	
0	+	-	-	convex	X	

Table 4.1: Axis and surface-type determination from the coefficients of the standard form of the bi-quadrics.

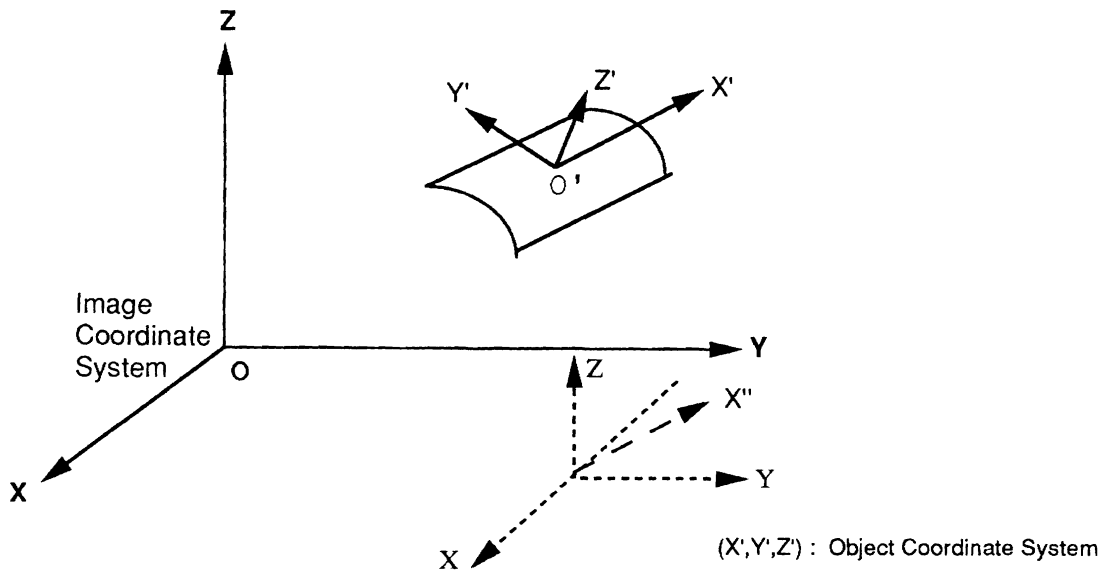


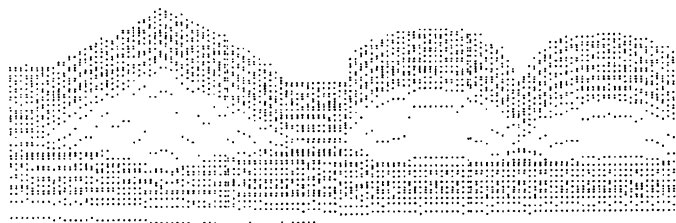
Figure 4.4: **The Coordinate system for bi-quadric representation:** The axis determination algorithm chooses X' as the correct orientation for the surface since its projection X'' makes least angle with X .

4.4 Aligning the Major Axis for Curved Objects

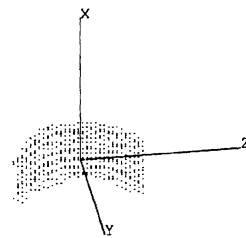
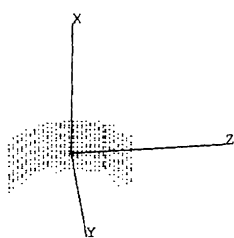
Curved objects in 3-D space are modeled by bi-quadrics (in $2\frac{1}{2}$ -D space) as one of the three basic surface types. The restriction of a fixing Z-axis along the viewing direction has a disadvantage that the curved surfaces like cylinders have no corresponding bi-quadric model when their axis is not orthogonal to the viewing direction. The model then obtained for the cylinders is either elliptic paraboloid or hyperbolic paraboloid, which is only an approximation of the underlying surface. In addition to providing surface support to the curved surfaces, these models have the information about the curvature of the surface. This information can be used to derive the axis in 3-D space that corresponds to the bi-quadric axis along which the surface curves the least or the most. This information is helpful in orienting the major axis of the initial approximation for the superquadric model recovery. For example, for the cylinders with greater diameter than height, this is the only way to get the correct the major axis (shown for the two cylindrical parts in scene 1 in figure 4.5). The superquadric axis placement follows the rule of thumb that the axis of least inertia (corresponding to the largest eigenvalue of the moment matrix) is the major axis (figure 4.5(b)). This heuristic is not true for cylinders with larger diameter than height, and may result in a box-like model instead of a cylindrical model. Aligning the Z-axis using the bi-quadric coefficients (figure 4.5(c)), as explained below, results in the initial model model to converge quickly to a cylindrical shape.

Figure 4.4 depicts a scenario where a cylindrical object is modeled by a elliptic paraboloid. The axis for which the coefficient in the standard form is larger, is the one along which the surface has higher curvature, so the axis orthogonal to it is the right axis for the superquadric major axis (Z direction). Referring to table 4.1, we note that if $|a'| > |b'|$, then the chosen bi-quadric axis is Y, and whichever axis in the object centered system makes least angle with Y axis will be considered as the Z-axis for superquadric alignment. Thus, for cylindrical shapes, we get the correct orientation by enforcing the axis direction to comply with the bi-quadric curvature. This is extremely useful for the superquadric model recovery, because the optimization procedure is unable to change the orientation of the model drastically if the initial estimates of the orientation are not correct.

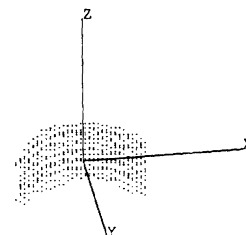
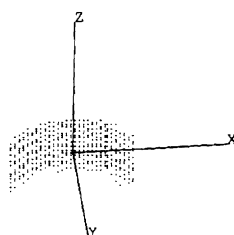
The surface-type information is also available from the standard form, as shown in table 4.1. However, for hyperbolic paraboloids, additional processing is required if a concave surface is modeled by them. The superquadric surface normal test or the Z-residuals derived later can be used to determine if a patch is concave, and the model can be correctly oriented.



(a) All range points.



(b) Z-axis aligned along the eigenvector with largest eigenvalue.



(c) Z-axis determined by bi-quadric orientation.

Figure 4.5: **Major axis estimation for curved parts in Scene 1:** The object centered system is oriented along the eigenvectors. (a) Aligning the Z-axis along the eigenvector with largest eigenvalue results in incorrect initial model which will never converge to a cylinder. (b) Using the bi-quadric information to align the Z-axis gives the correct initial model which converges quickly to a cylindrical model.

4.5 Edge-type Determination

By edges we mean the embedding of the intersection curve on the two surfaces in 3-space. An edge exist where the domains of two analytical surfaces intersect. Whether they meet to form a convex edge or concave edge (in a global sense), is what interests us here. This information is of great importance in later processing when regions connecting along concave edges will discard any hypothesis predicting their union in a volumetric sense.

The property of convexity and concavity is a function of surface embedding and hence an extrinsic second-order property. We have adapted the surface refinement method for intersection cleaning to detect and label edges reliably. The procedure involves intersection traversal like before and does ramp detection in a one-dimensional signal. A one-dimensional mask, $[1 -2 1]$, is applied in the neighborhood of every overlapping point to compute second-order differences. A negative value indicates presence of a convex edge, while a positive value indicates a concave edge. The procedure is given in algorithm 4.2. The method is demonstrated on the NIST object in figure 4.6. The edge labeling is shown in the adjacency graph in figure 4.7. The edge information is used to label the edges of the surface adjacency graphs.

4.6 Surface Adjacency Graphs (SAGs)

The region adjacency graph is a *simple graph* (No self-loops or parallel edges), with nodes representing each region and edges representing the edge-types between pairs of intersecting regions. If two regions overlap (but do not intersect, as in two smoothly merging surfaces) then they are marked as non-intersecting, and allowed the possibility of combination by the globally convex volumetric model. If step edges exist between two regions, then there is no edge between the nodes representing the surfaces. Thus, if we remove all the concave edges from a SAG, we are left with graph (possibly disconnected, if parts connected only along a concave edge) that only convex edges. This graph encodes information about the consistency of surface level combinations, although this information may inhibit some of the possible combinations at global level. We shall explain in chapter 7, how the information about edges is used by the coarse to fine strategy of the control structure to allow for the possibility of global combination of surfaces that form concave edges at their intersections.

The surface graph for the NIST object is shown in figure 4.7. An interesting aspect of the SAG is that by removing concave edges, it divides the data set into convex con-

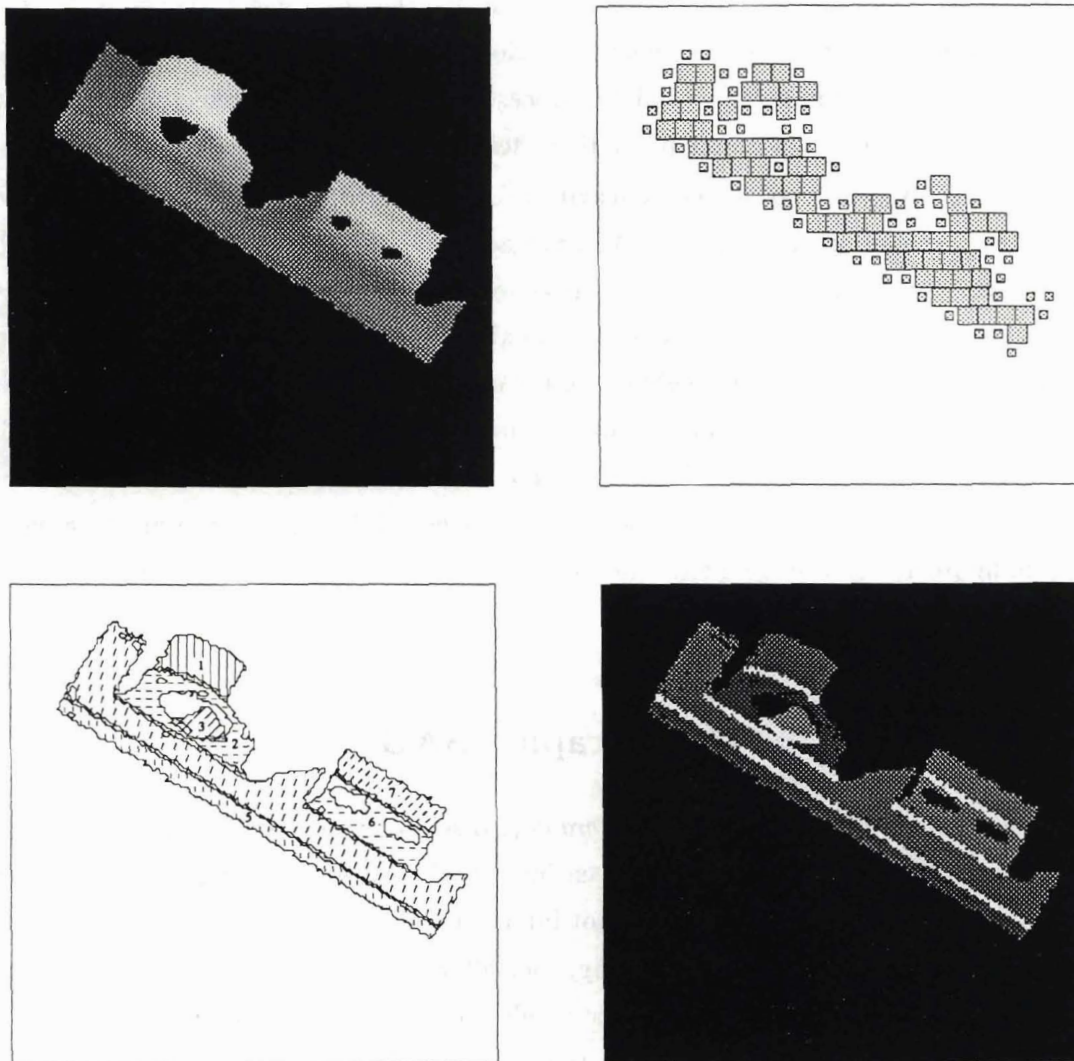


Figure 4.6: **Edge-type determination for the NIST object:** Top: The range image, and its seed regions. Bottom: Surface segmentation and the C_1 (surface normal) edges marked at the overlapping parts of the surfaces. Following a procedure similar to the intersection cleaning, the edges are marked as convex or concave.

1. Determine the pixels p_i that are on the intersection curve.
2. **foreach**(p_i) **do**
 - (a) convex_pixels = concave_pixels = 0.
 - (b) Compute the normal to the curve.
 - (c) 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.
 - (d) Apply $[1 \ -2 \ 1]$ mask in the k neighborhood of p_i .
 - (e) **If** (negative) **then**
 Embedding = CONVEX
 convex_pixels = convex_pixels + 1
else
 Embedding = CONCAVE
 concave_pixels = concave_pixels + 1
3. **If**(concave_pixels > convex_pixels) **then** Edge_embedding = CONVEX
else Edge_embedding = CONCAVE

Algorithm 4.2: Edge detection and labeling at two intersecting regions.

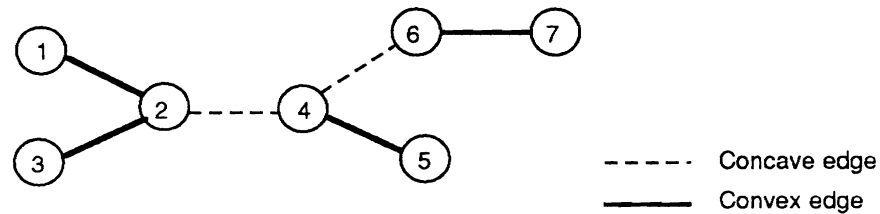


Figure 4.7: **Surface adjacency graph (SAG) for the NIST object:** The removal of concave edges splits the graph into three connected components, corresponding to three parts in the object.

nected components, which can be viewed as independent data sets. Thus a SAG is useful for connected component analysis as well as for checking the combination hypotheses for consistency.

4.7 Chapter Summary

We described techniques for the refinement of surface patches, so that the surface clustering can be used by the volumetric segmentation as the first approximation of the data. The information contained in bi-quadric surface segmentation can be of immense use for volumetric segmentation. In addition to providing the surface support, bi-quadrics contain information about axis orientation for curved surfaces, convex component analysis for segmenting parts along concave discontinuities, and the embedding of the surface in 3-D space. We derived the analytical expressions to compute these important surface attributes and discuss their relevance in superquadric fitting. The analysis presented in this chapter is of vital importance in designing the control strategy for volumetric segmentation described in chapter 7.

CHAPTER 5

Superquadrics : Volumetric Part-Models

We now have a piecewise description of the range data in terms of bi-quadric patches. However, our goal is to recover volumetric descriptions of data by clustering them into piecewise-convex or combination of positive and negative convex parts. Superquadric models give volumetric object-centered descriptions of the object parts. In this chapter we will describe the superquadric model, formulate the model-recovery problem, and derive some results that are useful for obtaining the volumetric segmentation. We will first give the definition of deformable superquadrics as given by Solina [Solina 1987, Bajcsy and Solina 1987, Solina and Bajcsy 1990], and then develop some useful results about the interpretation of the inside-outside function, computation of the true Euclidean distance, derivation of the occluding contour generator on superquadrics, computing edges on superquadrics, and the formulation of the superquadric recovery problem to provide volume, surface and occluding-contour constraints.

5.1 Introduction

Superquadrics are a family of parametric shapes that have been used as primitives for shape representation in computer vision [Pentland 1986, Solina 1987, Boulton and Gross 1987] and computer graphics [Barr 1981, Barr 1984]. Superquadrics are like lumps of clay that can be deformed and glued together into realistic looking models.

Definition : A superquadric surface is defined by a vector \mathbf{x} sweeping a closed surface in space by varying angles η and ω in the given intervals :

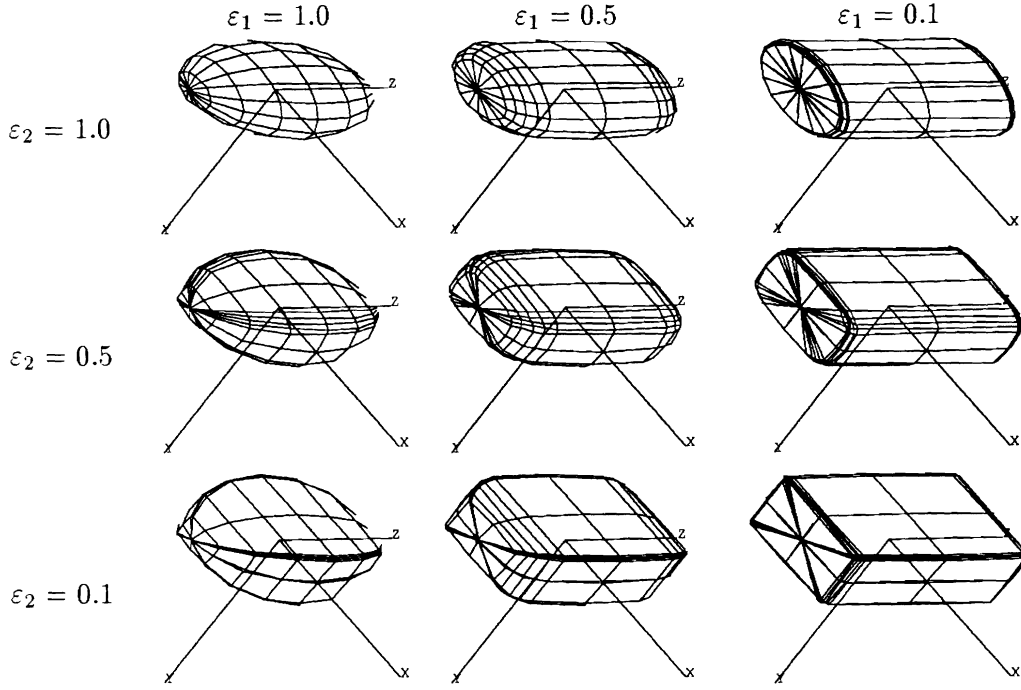


Figure 5.1: **Superquadric shapes:** Superquadric model vocabulary between $0.1 \leq \varepsilon_1 \leq 1.0$ and $0.1 \leq \varepsilon_2 \leq 1.0$.

$$\mathbf{x}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\varepsilon_1}(\eta) \cos^{\varepsilon_2}(\omega) \\ a_2 \cos^{\varepsilon_1}(\eta) \sin^{\varepsilon_2}(\omega) \\ a_3 \sin^{\varepsilon_1}(\eta) \end{bmatrix} \quad \begin{array}{l} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{array} \quad (5.1)$$

Superquadric implicit equation can be derived from the above equation by eliminating η and ω :

$$\left(\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} = 1. \quad (5.2)$$

Parameters a_1 , a_2 , and a_3 define the superquadric size in x, y and z direction (in object centered coordinate system) respectively. ε_1 is the squareness parameter in the latitude plane and ε_2 is the squareness parameter in the longitude plane. Based on these parameter values superquadrics can model a large set of standard building blocks, like spheres, cylinders, parallelopipeds and shapes in between (figure 5.1).

If both ε_1 and ε_2 are 1, the surface defines an ellipsoid. Cylindrical shapes are obtained for $\varepsilon_1 < 1$ and $\varepsilon_2 = 1$. Parallelopipeds are obtained for both ε_1 and ε_2 are < 1 . We have

restricted the model recovery procedure to fit the models with $0.1 \leq \varepsilon_1, \varepsilon_2 \leq 1.0$.

5.1.1 Applying Deformations to Superquadrics

The representational power of superquadrics increase further by applying various global *deformations* on the basic model (incorporated by Solina [Solina 1987]). Deformations that we have included in our vocabulary are *tapering* and *bending*. The tapering and bending transformations applied to the vector $\mathbf{x}(\eta, \omega)$ in the forward direction are given below:

Tapering : Linear tapering along z axis transforms the superquadric (x, y, z) to (X, Y, Z) by following transformation :

$$\begin{aligned} X &= f_x(z) x \quad \text{where} \quad f_x(z) = \frac{K_x}{a_3} z + 1 \\ Y &= f_y(z) y \quad \text{where} \quad f_y(z) = \frac{K_y}{a_3} z + 1 \\ Z &= z \end{aligned} \tag{5.3}$$

where $-1 \leq K_x, K_y \leq 1$.

Bending : Bending deformation transforms the superquadric surface vector by following transformation :

$$X = x + \cos_\alpha(R - r), \quad Y = y + \sin_\alpha(R - r), \quad Z = \sin_\gamma\left(\frac{1}{k} - r\right).$$

Where r is the projection of x and y components onto the bending plane $z - r$:

$$r = \cos(\alpha - \tan^{-1}(\frac{y}{x}))\sqrt{(x^2 + y^2)}$$

Bending transforms r into

$$R = k^{-1} - \cos(\gamma)(k^{-1} - r),$$

Where γ is the bending angle

$$\gamma = zk^{-1}. \tag{5.4}$$

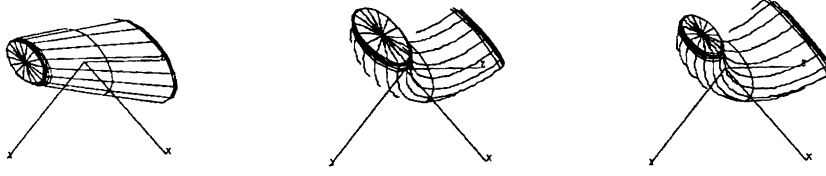


Figure 5.2: **Deformed superquadric shapes:** Tapered, bent, and tapered and bent shapes for a cylindrical model ($\varepsilon_1 = 0.1$ and $\varepsilon_2 = 1.0$).

Combination of Tapering and Bending: The two independent deformations are applied by computing the corresponding homogeneous transformation matrices. It is possible to apply both the transformations to a superquadric model one by one. since matrix multiplication is not commutative, the order in which deformations are applied is important. The model recovery procedure has adopted the following structure to transform an object centered superquadric model to a deformed superquadric in general position and orientation :

$$\mathbf{X} = \text{Translation}(\text{Rotation}(\text{Bending}(\text{Tapering}(\mathbf{x})))) \quad (5.5)$$

Thus bending and tapering introduce two parameters each in the final superquadric equation, bringing total parameter count to 15. The tapered and bent shapes for a cylindrical model are shown in figure 5.2.

Computing the deformations in the reverse direction transforms a point (X, Y, Z) on a deformed model into (x, y, z) on the basic superquadric model by satisfying the following implicit equation of the deformed model :

$$\left(\left(\frac{X - \cos(\alpha)(R' - r')}{\left(\frac{K_x\gamma}{ka_3} + 1\right)a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{Y - \sin(\alpha)(R' - r')}{\left(\frac{K_y\gamma}{ka_3} + 1\right)a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{\gamma}{ka_3} \right)^{\frac{2}{\varepsilon_1}} = 1. \quad (5.6)$$

where:

$$\begin{aligned} \gamma &= \arctan\left(\frac{Z}{(k^{-1} - R')}\right) \\ R' &= \cos\left(\alpha - \arctan\left(\frac{Y}{X}\right)\right)\sqrt{X^2 + Y^2} \\ r' &= k^{-1} - \sqrt{Z^2 + (k^{-1} - R')^2} \end{aligned} \quad (5.7)$$

5.2 Formulation of the Model Recovery Problem

The Modified Inside-outside Function: The inside-outside function for the superquadric model can be modified by adding the exponent ε_1 to the equation 5.2, to cancel out the effect of low values of ε_1 during the model recovery process [Solina 1987] :

$$F(x, y, z) = \left[\left[\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left[\frac{z}{a_3} \right]^{\frac{2}{\varepsilon_1}} \right]^{\varepsilon_1} \quad (5.8)$$

This modification does not alter the shape of the superquadric model, but significantly improves the recovery of cylindrical objects, The inside-outside function (IO function for short) determines where a point lies relative to the superquadric surface. If $F(x, y, z) = 1$, point (x, y, z) lies on the surface of the superquadric. If $F(x, y, z) < 1$, the point lies inside and if $F(x, y, z) > 1$, the point lies outside the superquadric.

Solina [Solina 1987] has formulated the superquadric model recovery problem in general position and orientation by using Euler angles ϕ, θ, ψ to define the orientation and p_x, p_y, p_z to define position of the superquadric in a world coordinate system. The optimization procedure minimizes the inside-outside function of deformed superquadrics in general position given by :

$$GOF = \sqrt{a_1 a_2 a_3}(R), \quad (5.9)$$

where

$$R = \frac{1}{N} \sum_{i=1}^N (F(x, y, z; a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \phi, \theta, \psi, p_x, p_y, p_z, K_x, K_y, k, \alpha) - 1) \quad (5.10)$$

Thus, the formulation imposes two constraints on the recovering model:

1. **Volume Constraint:** The $\sqrt{a_1 a_2 a_3}$ factor provides for the smallest volume satisfying the surface constraint.
2. **Surface Constraint:** The condition that a point should satisfy the inside-outside functions provides the constraint for a point to lie on the superquadric model.

When an arbitrary collection of points (non-convex) is presented to the above formulation, and there is no model that will satisfy the surface constraint, the model averages out

the inside-outside function value to leave certain points outside the model ($F > 1$, underestimated) and some inside the model ($F < 1$), overestimated. If the concavities (or convex deficiencies) are significant, then cluster of points have values away from the ideal value of 1. The Goodness-of-fit is simply the normalized sum of the inside-outside function values at all the points. To use this normalized value of F for model evaluation, we have to assign a meaning to it. In other words, what does it mean for a point to have a goodness-of-fit value? It is certainly not related to the Euclidean distance in the sense that two equidistant points from the superquadric model can have different inside-outside function value. We now describe the significance of the goodness-of-fit measure based on the IO function.

5.2.1 Interpretation of the Goodness-of-fit

The outermost exponent ε_1 in the inside-outside function F was added by Solina [Solina 1987] to cancel out the effect of ε_1 in the equation. This modification resulted in better recovery of cylindrical objects. Solina noted only the qualitative effect of the modification, and no mathematical justification was given for it. We provide an explanation which gives an intuitive meaning to the values of inside-outside function, and makes it possible to use this measure for model evaluation.

Consider a superquadric $S_1 = (X_1, Y_1, Z_1)$ defined by explicit superquadric equations. Take an arbitrary point $P(x, y, z)$ in space, and scale the three axes of S_1 by a factor β such that the point P lies on the scaled superquadric $S_2 = (X_2, Y_2, Z_2)$:

$$S_2(\eta, \omega) = \begin{bmatrix} \beta a_1 \cos^{\varepsilon_1}(\eta) \cos^{\varepsilon_2}(\omega) \\ \beta a_2 \cos^{\varepsilon_1}(\eta) \sin^{\varepsilon_2}(\omega) \\ \beta a_3 \sin^{\varepsilon_1}(\eta) \end{bmatrix} \quad \begin{matrix} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{matrix} \quad (5.11)$$

We will prove that F and β are related. The implicit form of $S_2(\eta, \omega)$ can be written as :

$$\left[\left(\frac{x}{\beta a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{\beta a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left[\frac{z}{\beta a_3} \right]^{\frac{2}{\varepsilon_1}} = 1. \quad (5.12)$$

Solving for β yields :

$$\beta = \left[\left[\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left[\frac{z}{a_3} \right]^{\frac{2}{\varepsilon_1}} \right]^{\frac{\varepsilon_1}{2}} \quad (5.13)$$

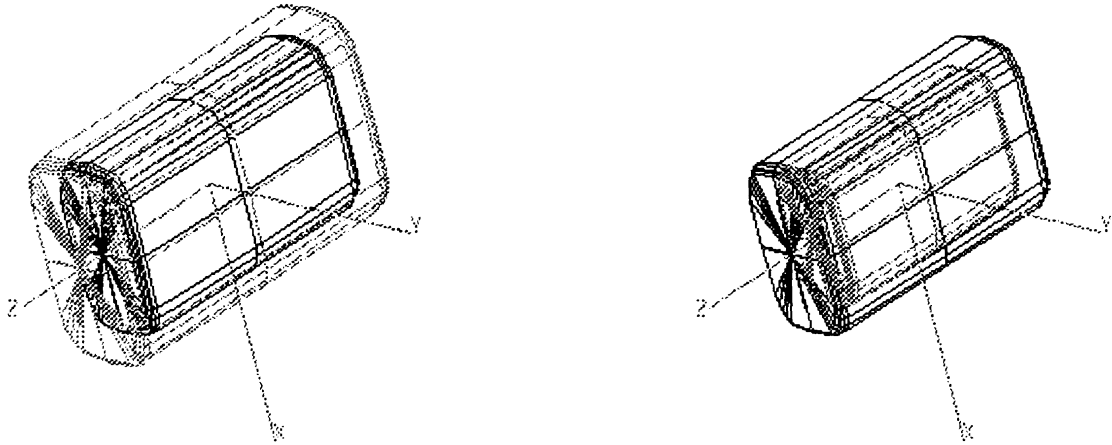


Figure 5.3: β -expansion and contraction of a superquadric model: left: expansion for $\beta = 1.2$; right: contraction for $\beta = 0.8$.

It follows from the definition of F that :

$$F = \beta^2. \quad (5.14)$$

This result shows that the value of inside-outside function F for a point (x, y, z) is nothing but square of the factor by which the axes of superquadric S_1 have to be scaled to make it pass through (x, y, z) . This factor can be seen as the amount a superquadric has to be *expanded* or *contracted* (figure 5.3) to make it pass through an arbitrary point in 3 space. This result provides an intuitive explanation for the values of F , with values > 1 indicating expansion and < 1 indicating dilation of the superquadric.

The obvious question to ask is if this explanation can be extended to the tapered or bent models? Since tapering is defined in terms of a_3 (the dimension along the major axis), it is not possible to obtain a closed form solution for β . So the above interpretation is only approximately true for tapered models. For the models with bending deformation, however, the interpretation is valid. Since the minimization problem is formulated in terms of inside-outside function, its values are available with the model parameters, and does not require explicit computation.

5.3 Euclidean Distance Measure

The formulation of the superquadric recovery procedure in terms of minimization of inside-outside function is *not* the same as the minimization of the distance function :

$$d = \sqrt{(x - x_1)^2 + (y - y_1^2) + (z - z_1^2)} \quad (5.15)$$

Where d is the distance of a point (x, y, z) from the superquadric. So the Euclidean distance is not computed at any stage of model recovery. It is important to note that the inside-outside function and the distance measure are not related in the sense that two points at the same distance from the superquadric surface do *not* have the same value of F in general.

The distance of an arbitrary point in 3-D space from a given superquadric model is difficult to compute because of multiple solutions of the analytical formulation of the problem as the non-linear root finding problem. Further, it is not possible to obtain a closed form solution for the problem. We have posed it as a minimization problem, that iteratively minimizes d for a given point and a given deformed superquadric. In any minimization problem it is imperative to have a close initial approximation. Superquadric surfaces are parameterized by η and ω , and are convex for the points outside the model. Thus the problem is formulated as :

Problem definition : Given (x_1, y_1, z_1) , minimize the following function of two variables :

$$d(\eta, \omega) = \sqrt{(x(\eta, \omega) - x_1)^2 + (y(\eta, \omega) - y_1)^2 + (z(\eta, \omega) - z_1)^2} \quad (5.16)$$

Where $x(\eta, \omega), y(\eta, \omega), z(\eta, \omega)$ are the position vectors of the deformed superquadric

To ensure convergence to the right solution, a close initial approximation is obtained by extending the expansion/contraction approach introduced in the previous section (figure 5.3). Corresponding to the point $P(x_1, y_1, z_1)$ in 3-D space, there is a point $G(x_2, y_2, z_2)$ on the original superquadric S_1 :

$$\begin{aligned} x_2 &= x_1/\beta, \\ y_2 &= y_1/\beta, \\ z_2 &= z_1/\beta. \end{aligned} \quad (5.17)$$

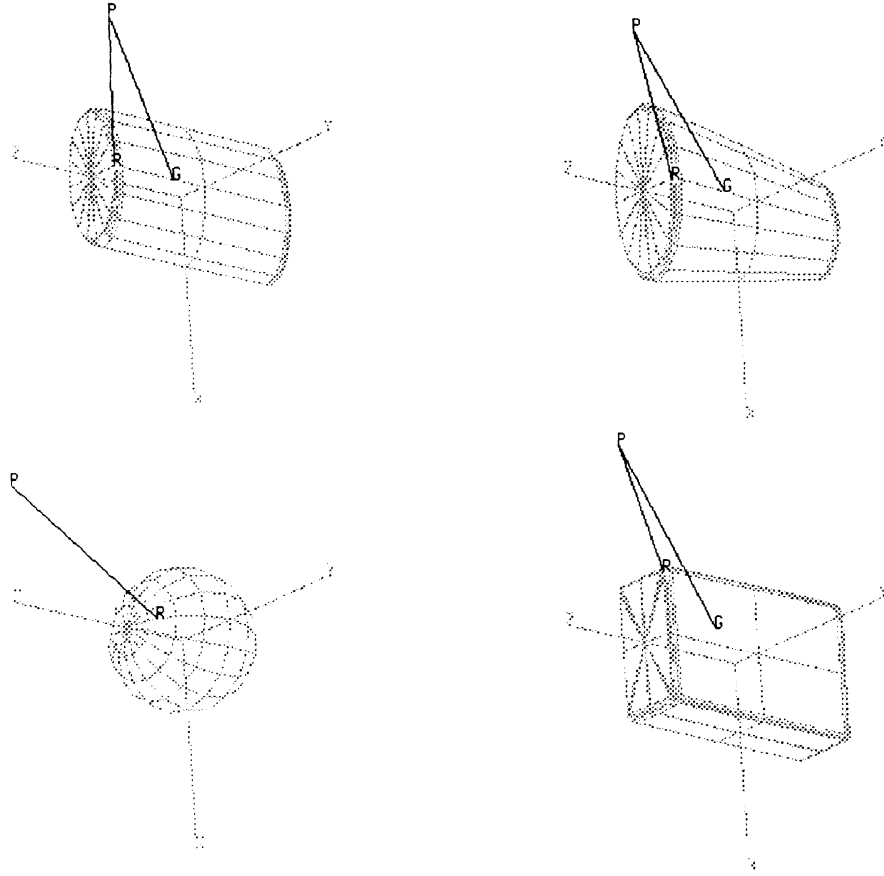


Figure 5.4: **Euclidean distance computation:** The distance (PR) of a point P from the model is determined using an initial guess (PG) based on β -expansion/contraction. The point G lies on the straight line connecting the point P and the origin of the model. The superquadric recovery formulation corresponds to the distance PG and not to PR . Excepting the spherical model (bottom left), where points G and R coincide, PG is an overestimation of the real distance PR .

The point G in Cartesian coordinate system can be written as $G(\eta, \omega)$ in the parametrized form. Thus, initial approximation of η and ω is easily obtained. If the superquadric in consideration is deformed then deformations are ignored since we are interested in only an initial approximation. This method essentially traces the locus of η and ω on superquadrics by varying β but keeping other parameters constant. Thus the points P and G correspond to the same η and ω values, and G is likely to be very close to the point $R(\eta', \omega')$ such that R is the point closest to P . Figure 5.4 gives examples of initial guess to find the minimum distance on different superquadric shapes.

The objective is to find R . The function d of two variables is minimized given the initial approximation η and ω , using a gradient-descent method. The method requires only function values, a finite-difference method is used to estimate the gradient internally. Though d is differentiable at all points (even with deformations), we have found that supplying external gradient values does not speed up the iterative process in general. The method was found to be accurate up to sixth decimal place for experimental data. We can settle for lower accuracy for faster convergence. The method has been successfully tested on deformed superquadrics.

The optimization function represented in equation 5.16 is convex for the points lying outside the superquadric model. For the points inside the model, distances are also minimized in two orthogonal directions past the edge of the model, since the presence of a strong edge is the reason for non-convexity of the distance function. Two methods for the computation of superquadric edges are described later in the chapter.

5.4 Apparent Contours of Superquadrics

Definition: The **Contour-generator** (or occluding contour) defined as the locus of the points (a closed curve) on the superquadric surface where the surface normal vector is perpendicular to the viewpoint vector.

Let $\mathbf{V} = (V_x, V_y, V_z)$ be the viewpoint vector, and $\mathbf{N} = (n_x, n_y, n_z)$ be any surface normal vector (see appendix B for surface normal computation). The occluding contour is then given by :

$$\mathbf{V} \cdot \mathbf{N} = 0 \quad (5.18)$$

We now derive a closed form solution for the contour generator on a non-deformed superquadric surface :

$$V_x n_x + V_y n_y + V_z n_z = 0 \quad (5.19)$$

Substituting for \mathbf{N} gives :

$$\frac{V_x}{a_1} \cos^{2-\varepsilon_1}(\eta) \cos^{2-\varepsilon_2}(\omega) + \frac{V_y}{a_2} \cos^{2-\varepsilon_1}(\eta) \sin^{2-\varepsilon_2}(\omega) + \frac{V_z}{a_3} \sin^{2-\varepsilon_1}(\eta) = 0. \quad (5.20)$$

Solving for η gives the closed form solution for generating the apparent contour :

$$\eta = \tan^{-1} \left(\left(-\frac{a_3}{V_z} \left(\frac{V_x}{a_1} \cos^{2-\varepsilon_2}(\omega) + \frac{V_y}{a_2} \sin^{2-\varepsilon_2}(\omega) \right) \right)^{\frac{1}{2-\varepsilon_1}} \right). \quad (5.21)$$

When $V_z = 0$, the contour generator becomes:

$$\omega = \tan^{-1} \left(\left(-\frac{a_2 V_x}{a_1 V_y} \right)^{\frac{1}{2-\varepsilon_2}} \right) \quad \text{and} \quad -\pi/2 \leq \eta \leq \pi/2. \quad (5.22)$$

For the degenerate case ($V_x = V_y = 0$) the contour is given by $\eta = 0$; $-\pi \leq \omega < \pi$.

Figure 5.5 shows the apparent contours of non-deformed box and cylindrical superquadric models generated by the above equation. Unfortunately, there is no closed form solution for a general deformed superquadric, as the surface normal vector N has to undergo deformation by the following rule (derived by [Barr 1984]) :

$$N' = \det \mathbf{J} \mathbf{J}^{-1T} \mathbf{N} \quad (5.23)$$

where \mathbf{J} is the Jacobian of the deformed superquadric. To trace the apparent contour of a deformed superquadric, we have to vary the angles η and ω systematically, and accumulate points on the contour in such a way that a closed contour is formed (shown for a tapered box in figure 5.5). This contour can be orthographically projected on the image coordinate system to make comparisons with the image contour.

Due to the closed form of the 5.21, it can be used to derive an objective function to provide the occluding contour constraint during the model recovery. The constraint would force the occluding contour points (if known in advance) to lie along the occluding contour of the model. We tested such a formulation by adding it to the optimization function in equation 5.9, but did not observe any significant improvement in convergence. The reason being that the equation 5.21 is valid *only* when a point lies *on* the superquadric model. For the points away from the surface of the model, there is no closed form solution for the surface normal and hence a general formulation is not possible.

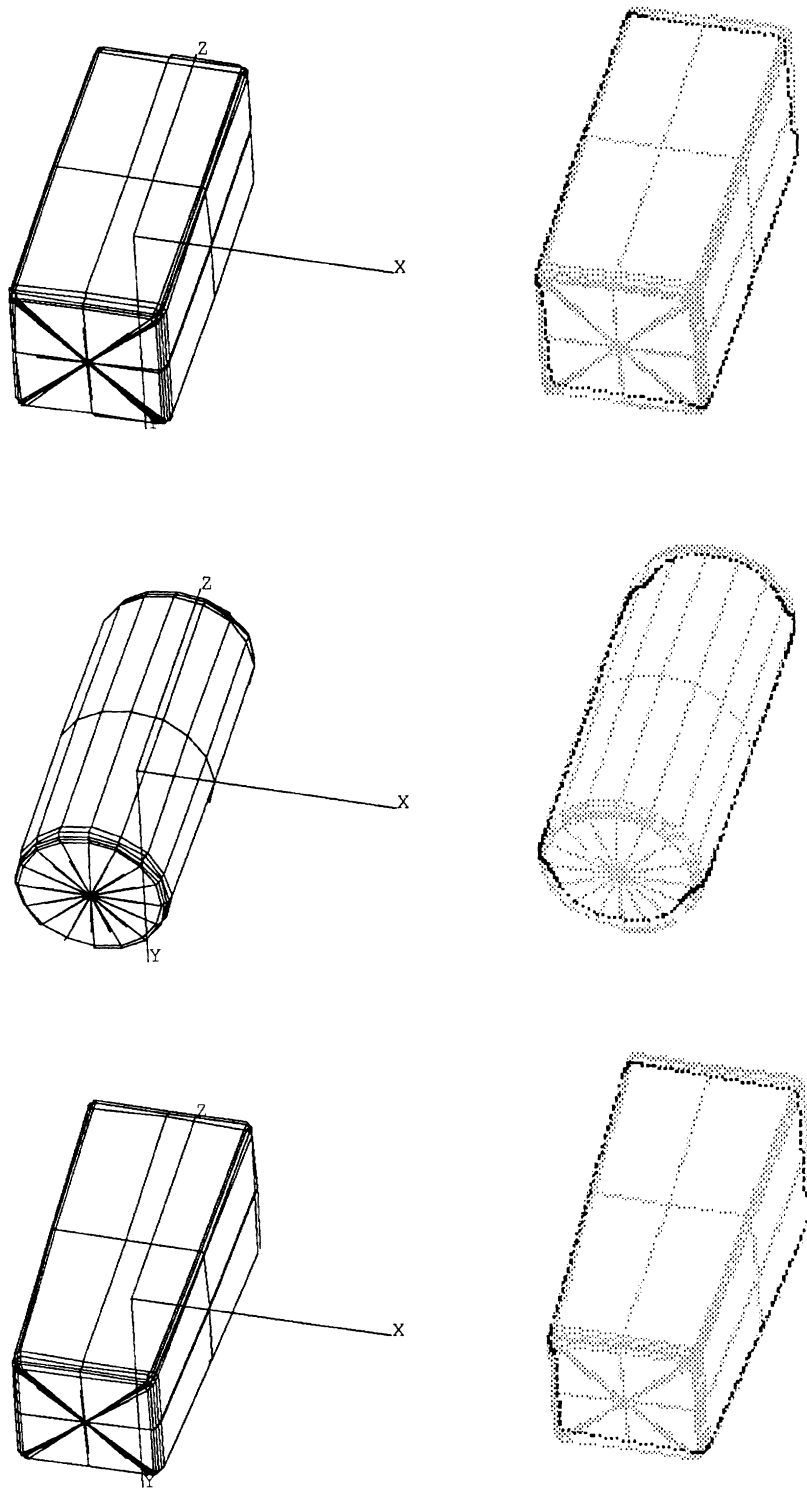


Figure 5.5: **Apparent contours of superquadrics:** Shown for a box and a cylinder, and a tapered box. Left: The superquadric model; right: The occluding contour superimposed on the model.

5.5 Superquadric Edges

By limiting the shape parameters between 0.1 and 1.0, the analyticity of the superquadric surface is maintained. Thus, there are no C_1 (surface normal) discontinuities, but the points of high curvature represented by the shape factor can be considered as “analytical edges.” We now present an algorithm for edge extraction on deformable superquadrics. On a non-deformed superquadric, the two 2-D cross-section contours (corresponding to $\eta = 0$ and $\omega = 0$), contain complete information about edges. Of course, existence of an edge depends on what we consider an edge. Typically, a shape parameter value of more than 0.5 indicates smooth surface, otherwise an edge exists. Interestingly, the edges *do not* change in terms of angles (although their location changes), when tapering and bending deformations are applied to the model. This means that we need only consider the non-deformed case to extract angles corresponding to the edge location. An edge exists where the curvature is maximum or equivalently the rate of change of curvature is zero. Since the superquadric equation is differentiable everywhere on the surface, the first, second and third derivatives are computed and plugged into the 2-D curvature and rate of curvature equations :

$$\kappa = \frac{x'y'' - y''x'}{(x'^2 + y'^2)^{3/2}}$$

$$\kappa' = \frac{(x'^2 + y'^2)^{3/2}(x'y''' - y'x''') - \frac{3}{2}(x'y'' - y'x'')\sqrt{(x'^2 + y'^2)}(2x'x'' + 2y'y'')}{(x'^2 + y'^2)^3} \quad (5.24)$$

Curvature and the rate of change of curvature values along the $\eta = 0$ contour for $0 \leq \omega \leq \pi/2$, are shown for $\varepsilon_1 = 0.1$ & ε_2 varying between 0.1 and 1.0 in figure 5.6. Six angles (2 for η and 4 for ω) are required to completely describe edges for a general superquadric model. Edges for a box, its deformations, and a cylinder are shown in figure 5.7.

5.5.1 Superquadric Edges from Occluding Contour

The superquadric model and its edges in figure 5.7(a) lead to an interesting observation. Notice that the edges exist at the occluding contour of the model. The question then arises, can we use the closed-form formulation of the occluding contour in equation 5.21 to directly derive the η and ω instead of taking the curvature-based approach? The answer, fortunately, is yes, thereby providing us with an elegant method to determine superquadric edges. Also, since the edge locations do not change when deformations are applied, the equation 5.7 is adequate to compute edges in general.

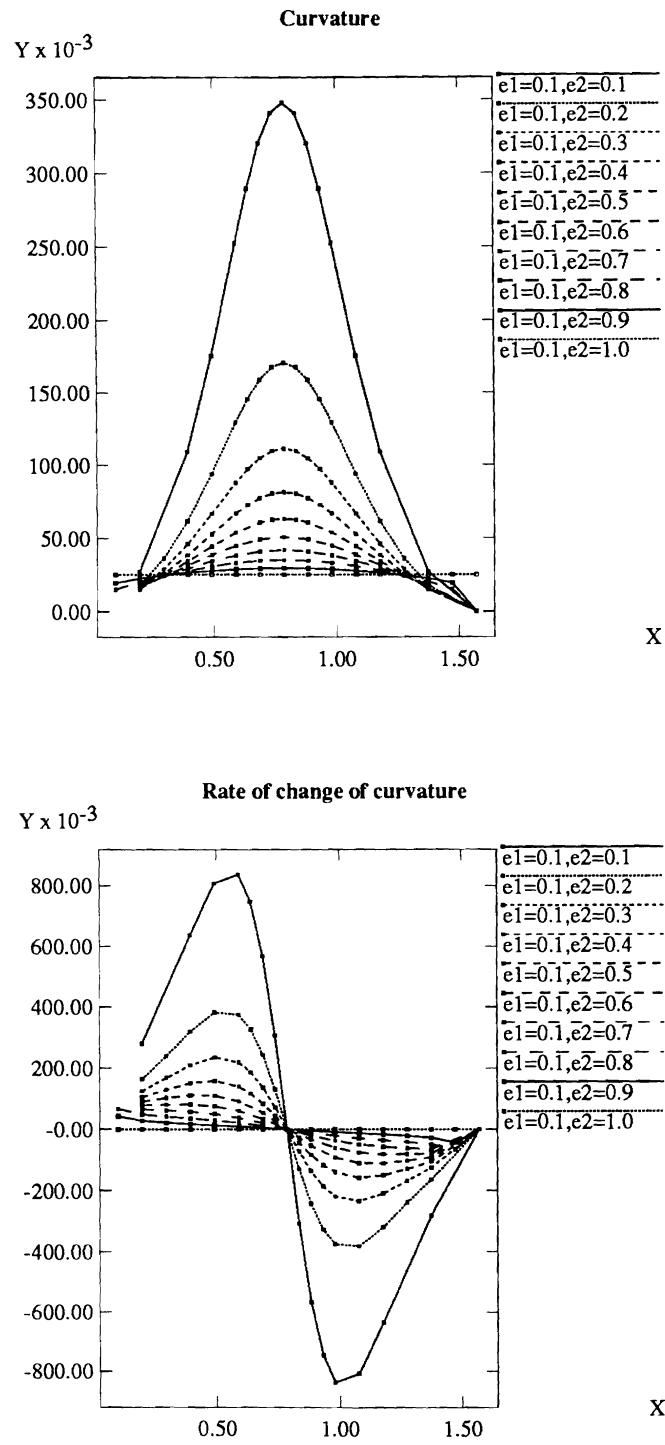


Figure 5.6: **Superquadric curvature:** Curvature (top) and the rate of change of curvature (bottom) along the $\eta = 0$ contour for $0 \leq \omega \leq \pi/2$, for $\varepsilon_1 = 0.1$ & $0.1 \leq \varepsilon_2 \leq 1.0$.

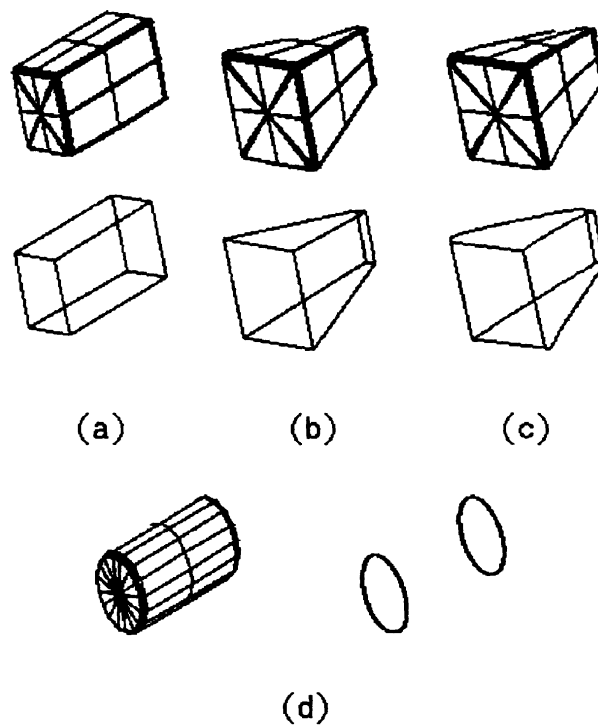


Figure 5.7: **Superquadric edges:** Edges of (a) a non-deformed box, (b) its tapered model, (c) its tapered and bent model, and (d) a cylinder. Same edge angles were used to compute edges on the deformed models.

The edge angle η_e can be obtained from the equation 5.20 by making $\omega = 0$:

$$\begin{aligned} \frac{V_x}{a_1} \cos^{2-\varepsilon_1}(\eta) + \frac{V_z}{a_3} \sin^{2-\varepsilon_1}(\eta) &= 0. \\ \eta_e &= \tan^{-1} \left(\left(-\frac{a_3 V_x}{a_1 V_z} \right)^{\frac{1}{2-\varepsilon_1}} \right), \quad \omega = 0. \end{aligned} \quad (5.25)$$

The other edge angle ω_e , is obtained by making $\eta = 0$ in the equation 5.20:

$$\begin{aligned} \frac{V_x}{a_1} \cos^{2-\varepsilon_2}(\omega) + \frac{V_y}{a_2} \sin^{2-\varepsilon_2}(\omega) &= 0. \\ \omega_e &= \tan^{-1} \left(\left(-\frac{a_2 V_x}{a_1 V_y} \right)^{\frac{1}{2-\varepsilon_2}} \right), \quad \eta = 0. \end{aligned} \quad (5.26)$$

A corner exists at (η_e, ω_e) . Equations 5.25 and 5.26 give the analytical solutions for computation of edges (or high curvature contours) and corners for a superquadric model.

5.6 Recovery of a Superquadric Model on Range Data

The model recovery algorithm as formulated by equation 5.9 starts with fitting an ellipsoidal shape on 3-D points and converges on a shape that minimizes the least-squares error. A stepwise description of the procedure is given in algorithm 5.1. The coordinate systems are shown in figure 5.8. Recovery occurs in world coordinate system, whose origin is at the centroid of the data points and oriented the same way as the image coordinate system. This has the effect of starting iterations with (0,0,0) position vector for the object centered system, which was empirically found to converge faster. The initial ellipsoidal model for the NIST object, and the model after 15 iterations are shown in figure 5.9. Clearly, the global model is unacceptable as a volumetric description of the NIST object, and reflects the need for further segmentation.

Solina [Solina 1987] showed that the solution space is convex near the optimal solution, and the model generally converges to perceptually acceptable shapes. We have found the procedure to be stable numerically, but having difficulties in recovering cylindrical shapes when the Z-axis (axis along a_3 dimension) is not aligned along the axis of the cylinder. The method also converges faster if the initial orientation is close to the final one, specially when the data is not complete or symmetric due to occlusion. We will address these problems later in the context of volumetric segmentation and provide efficient solutions to the problem of model orientation and Z-axis determination.

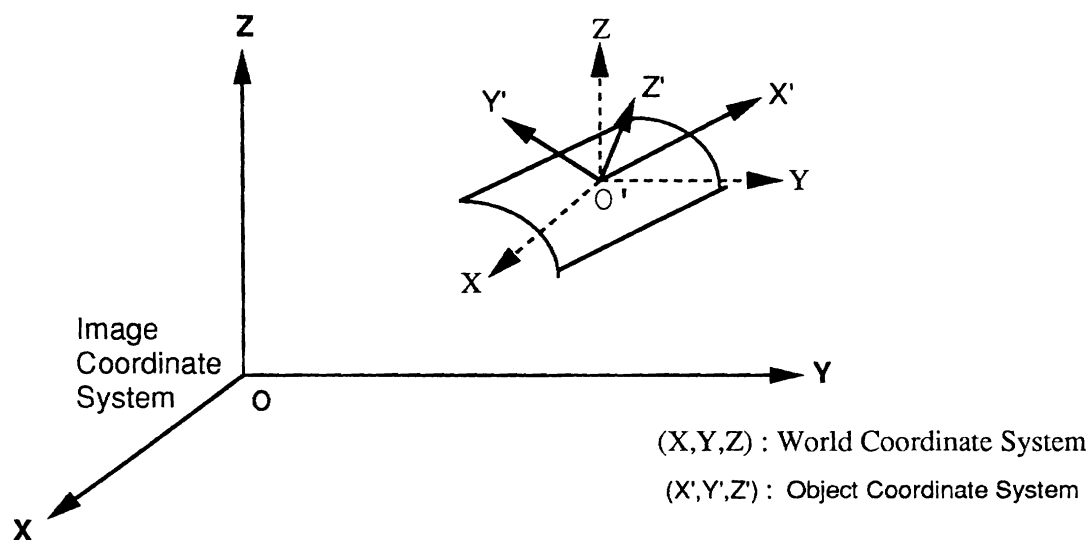


Figure 5.8: **Image, World and Object coordinate systems:** The representation and recovery space for representation and segmentation.

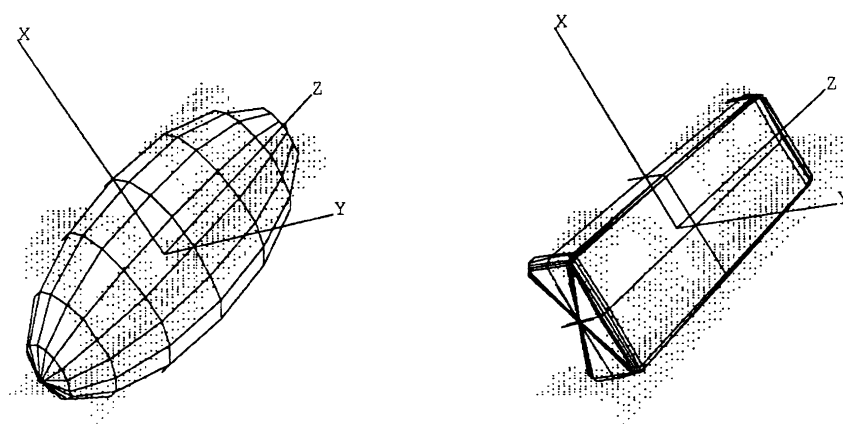


Figure 5.9: **Recovery of the superquadric model:** NIST object. Left: The initial ellipsoidal model is oriented along the eigenvectors of the moment matrix. Right: The model after 15 iterations of the Levenberg-Marquardt method.

1. Convert range image points from Image coordinate system (ICS) $z = f(x, y)$ to World coordinate system $f(x, y, z) = 0$, centered at the object centroid and oriented as image coordinate system (figure 5.8).
2. Compute eigenvectors and eigenvalues of the moment matrix of the 3-D points. Orient the Z-axis of the superquadric along the eigenvector with least moment of inertia.
3. Object centroid gives position (p_x, p_y, p_z) , and the eigenvectors give orientation (Euler) angles (ϕ, ω, ψ) placing the object coordinate system (OCS) with respect to the world coordinate system (WCS).
4. Compute extremities of 3-D points in OCS to estimate the size parameters (a_1, a_2, a_3) of the initial estimate of the superquadric S_0 .
5. Set $\varepsilon_1 = \varepsilon_2 = 0$ (ellipsoid), $K_x = K_y = 0$ (no initial tapering), and $k = 0.0001, \alpha = 0$ (no initial bending).
6. Enable desired deformations. (Tapering in our case).
7. Perform iterative non-linear minimization (Levenberg-Marquardt method, [Press et al. 1988]). Termination is decided by m (maximum number of iterations), and the least-squares error:
 - (a) The first or second time error decreases marginally:
 $(\text{Error}(S_{i-1}) - \text{Error}(S_i)) < 0.1$,
 the procedure terminates even if $i < m$.
 - (b) Procedure terminates if,
 for some $n > m$, $\text{Error}(S_n) \leq E_{\min}$,
 where $E_{\min} = \text{Min}(\text{Error}(S_0) \dots \text{Error}(S_m))$.
8. Model recovery done.

Algorithm 5.1: Recovery of one superquadric model on range data.

5.7 Chapter Summary

After describing the deformable superquadrics as defined by Solina [Solina 1987, Bajcsy and Solina 1987], we developed some useful results about interpretation of the inside-outside function, computation of the true Euclidean distance, derivation of the occluding contour generator on superquadrics, computing edges on superquadrics, and formulated the superquadric recovery problem to provide volume and surface constraints. Using this formulation, a model for the given set of points can be obtained. In our global to local approach for volumetric segmentation this model gives the first volumetric estimate of the data-set. In the next chapter we analyze the global model for its adequacy in describing the data, and develop an exhaustive set of criteria to completely evaluate the model.

CHAPTER 6

Criteria for Superquadric Model Evaluation: Residual Analysis

We now have a fine-to-coarse surface segmentation procedure and a procedure to recover the global superquadric model for the given data. Consequently, for the given data set, we have the piecewise bi-quadric description and a global superquadric model. The superquadric model recovery formulation lacks the segmentation capability. All our efforts from this chapter on are directed towards developing a control structure that will segment the given data set by constant evaluation of the intermediate superquadric approximations of the data and by using the information from the biquadric segmentation and other geometric constraints.

In this chapter we begin the design of the control flow of the volumetric segmentation, so that the procedure can recognize the correct strategy for approaching the segmentation problem starting with the global superquadric model. We first present a set of criteria for the complete evaluation of a superquadric model and then demonstrate how they can be generated, evaluated and used by the volumetric segmentation module.

6.1 Criteria for Model Evaluation

A superquadric model obtained by least-square fitting the inside-outside function is an over-constrained estimation of data, with more constraints than parameters. Like any parametric approach the goal is to describe a large chunk of data by a few parameters. The recovery procedure assigns equal weight to each point, no matter where the point lies in 3-D space, with the central goal of satisfying the volume and surface constraints. As discussed in the

previous chapter, the superquadric recovery procedure is formulated to provide :

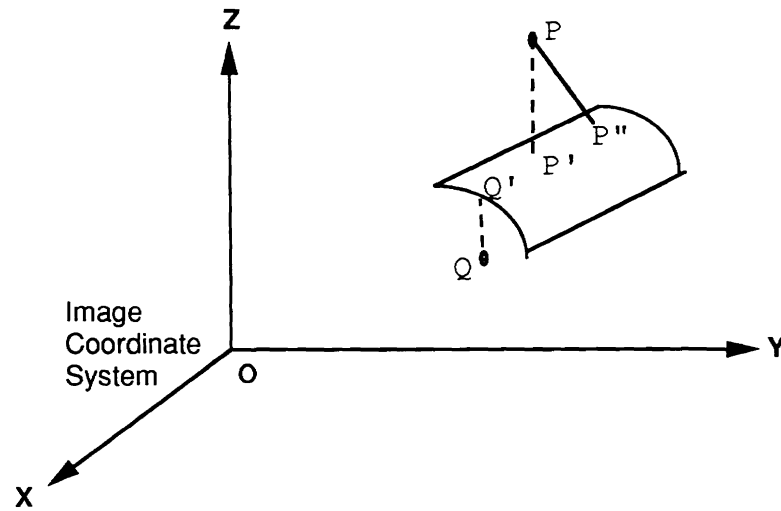
1. **Volume Constraint:** The $\sqrt{a_1 a_2 a_3}$ factor provides for the smallest volume satisfying the surface constraint.
2. **Surface Constraint:** The condition that a point should satisfy the inside-outside functions provides the constraint for a point to lie on the superquadric model.

Our definition of volumetric segmentation imposes acceptance criteria for the recovered models, which must be satisfied before a model is considered to be adequate for the given data. These acceptance criteria reflect the scale considerations and specify how much of error can be tolerated in the final description. The acceptance criteria dictate that all the data points must correspond to the model within the given error tolerance. When an arbitrary collection of points (non-convex in general) is presented to a single-model formulation, and there is no model that will satisfy the surface constraint, the model averages out the inside-outside function value to leave certain points outside the model ($F > 1$, underestimated) and some inside the model ($F < 1$, overestimated). If the concavities (or convex deficiencies) are significant, then clusters of points have values significantly different from the ideal value of 1. In such cases, the recovered model is not a satisfactory description of the underlying data, and the presence of such clusters signals the need for decomposition of data into smaller pieces to satisfy the modeling constraints.

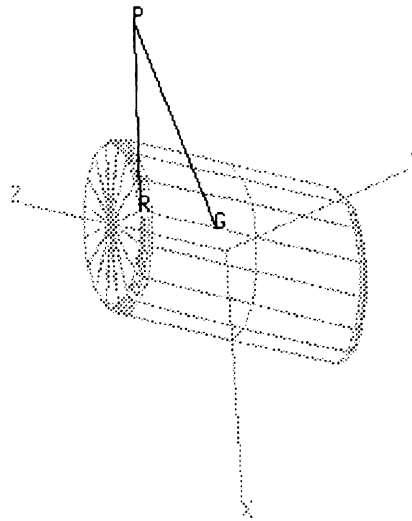
Thus, it is imperative that the model be fully analyzed, both qualitatively (using local distribution of residuals) and quantitatively (using global error measures), to aid in further course of action. As will be discussed later, the existence of residuals and their distribution is key to developing a systematic segmentation procedure. We have identified the following measures for model evaluation in the context of the shape recognition problem :

Quantitative Measures: Deviation of data points from the model surface can be measured by the following two methods (illustrated in figure 6.1):

1. Goodness-of-fit (\mathcal{G}) measure based on the inside-outside (IO) function (without the volume factor). In figure 6.1(b), the IO function value corresponds to PG and not to the minimum distance PR. A 0.1 value of F indicates 10% expansion/contraction of the model and is generally a good cut-off for evaluating the quality of a fit.
2. Average deviation along Z direction: In figure 6.1(a), the distance of point P from the model is PP' along Z while the minimum distance is PP'' . PP' is usually an



(a)



(b)

Figure 6.1: **Computation of deviation of a point from the superquadric model:**
 (a) The Z-residual is computed along the viewing direction in the image coordinate system. PP' is the distance along Z, while PP'' is the minimum distance. (b) The IO residual is based on the inside-outside function, measuring the distance corresponding to PG and not PR.

overestimation of PP'' . A value of 2 to 3 pixels is a good threshold for considering a fit acceptable at individual points.

These thresholds were determined from the empirical observations of the fits, and taking into account the quantization and sensor noise. However, relying on these thresholds for the evaluation of a recovered model can be misleading. Quantitative measures are normalized, global, and least-squares numerical values of the measuring quantity. Thus, a high value of the average deviation indicates a bad fit, although a good value may not always be due to a good fit. An acceptable global error can result from models with local details that are averaged out in the global consideration. Sometimes these details may need negative volume descriptions or further segmentation. This necessitates analysis of the type and the distribution of residuals. An example of such a case is shown in figure 6.2, where the data points in region 5 are overestimated by the global model having an acceptable global error-of-fit. Region 36 shows the points that are estimated within the error tolerance, and region 27 shows the overestimation of the boundary of the object in Z direction. The residual of region 5 can trigger further segmentation or provide for the negative volume fitting. Thus, local residuals and their distribution contains useful information about the quality of segmentation. Therefore, in order to evaluate individual residual regions, we present the following qualitative measures.

Qualitative Measures: The deviation of individual points from the model can be used to generate maps of the residuals to form clusters of points with identical description by the superquadric model:

1. IO residual-map: Classifies clusters of points that are outside, inside or on the model in terms of the inside-outside function.
2. Z residual-map: Classifies clusters of points that are outside, inside or on the model when analyzed along the viewing (or scanning) direction.
3. ED residual-map: Classifies clusters of points along the direction of the true Euclidean distance.

We discuss each one of these methods separately in next section. First, we need to describe all the different types of clusters (regions) that can be generated in a residual map. Residuals are computed by projecting the model in the image coordinate system, and

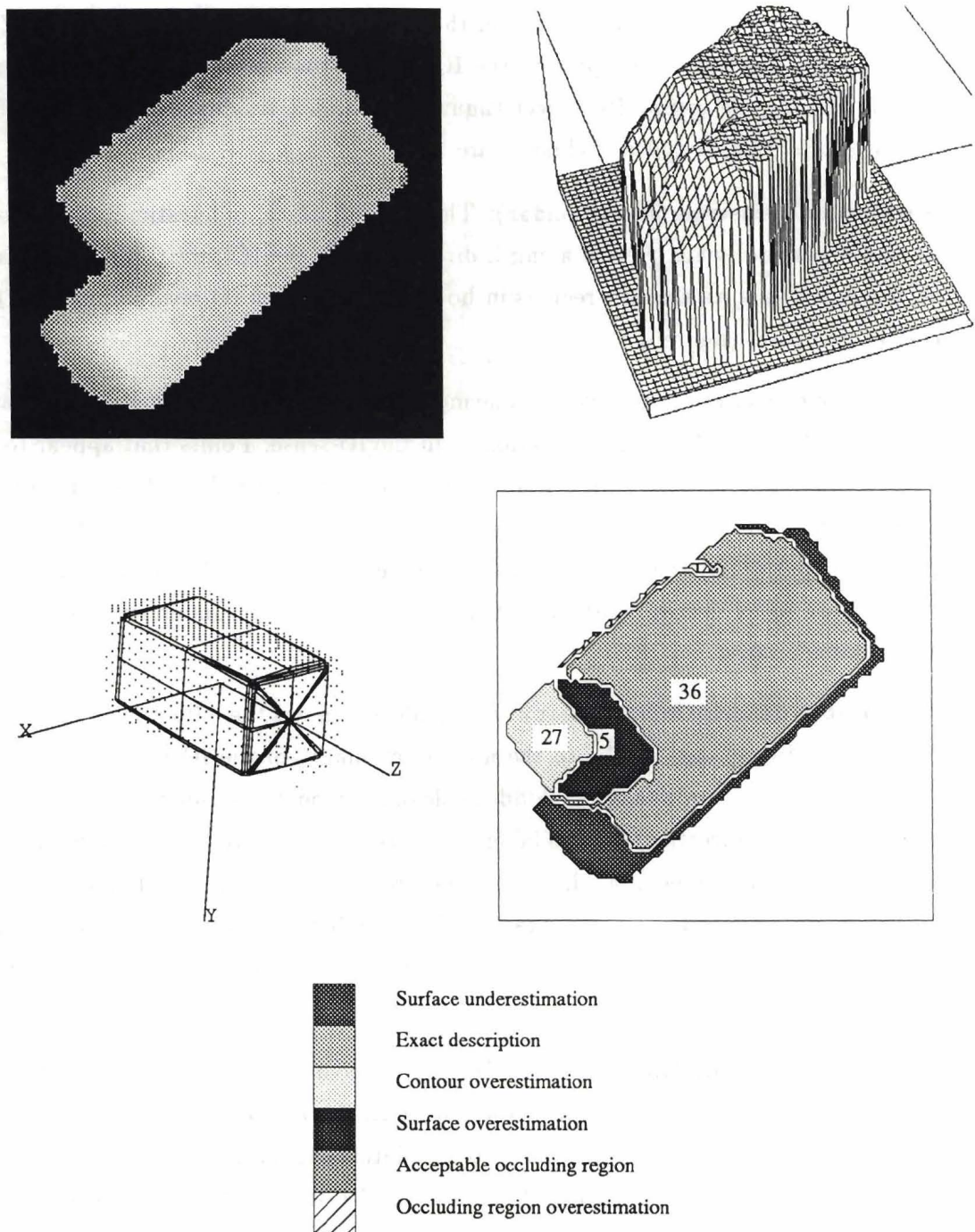


Figure 6.2: **The object with missing volume:** Top: Range image and its 3-D plot. Center: The global model and its Z-residual. The missing volume results in local residuals but the global model has acceptable error-of-fit. Bottom: Legend for the interpretation of the residuals, used throughout the dissertation.

making comparisons between data points and the model surface using one of the IO, Z or ED measures. For illustration, we present the IO residual-map and the Z residual-map of the global model for the composite object (figure 6.3), and Z residual-map for the NIST object (figure 6.4). The six types of clusters are :

1. **Surface underestimation (s_under):** The model surface underestimates the point (leaves it outside) when viewed along Z direction or in the IO sense. Regions 41 and 48 represent exist as **s_under** regions in both Z-residual and IO-residual maps of the composite object (figure 6.3).
2. **Surface overestimation (s_over):** The model surface overestimates the point (leaves it inside) when viewed along Z direction or in the IO sense. Points that appear to be underestimated in IO sense can actually appear overestimated along Z, because of the directionality constraint of the Z residuals. For example, the underestimated region for the composite object in figure 6.3 has a number of points hidden behind the model that appear to be overestimated along Z (region 14) but are underestimated in the inside-outside sense.
3. **Acceptable description (s_exact):** The model estimates the data points within the specified tolerance. Again, due to the non-directionality of the IO function, some of the points that are modeled by the hidden side of the model (the side that is not visible from the viewing direction) will be labeled as acceptable, whereas the Z-residual map will show them as overestimated. Due to the presence of parts, the NIST object and the composite object have small **s_exact** regions, while the global model for the object in figure 6.2 is a good approximation for the majority of the surface points (region 36).
4. **Contour overestimation (c_over):** Due to the symmetry and shape constraints of the rigid model, the projection of the model on the image coordinate system can result in overestimation of the silhouette of the data. These regions predict extra data which does not exist in the image. Region 46 in Z-residuals of the global model for the NIST object is such a region. Note that **c_over** is similar in both Z-residuals and IO-residuals (regions 6 and 9, and 7 and 11 respectively).
5. **Acceptable Occluding regions (occ_ok):** When data is decomposed to arrive at a piecewise description, it is desirable to allow the volumes of the models to occlude each other such that the occluded model *underestimates* or *exactly describes* the data

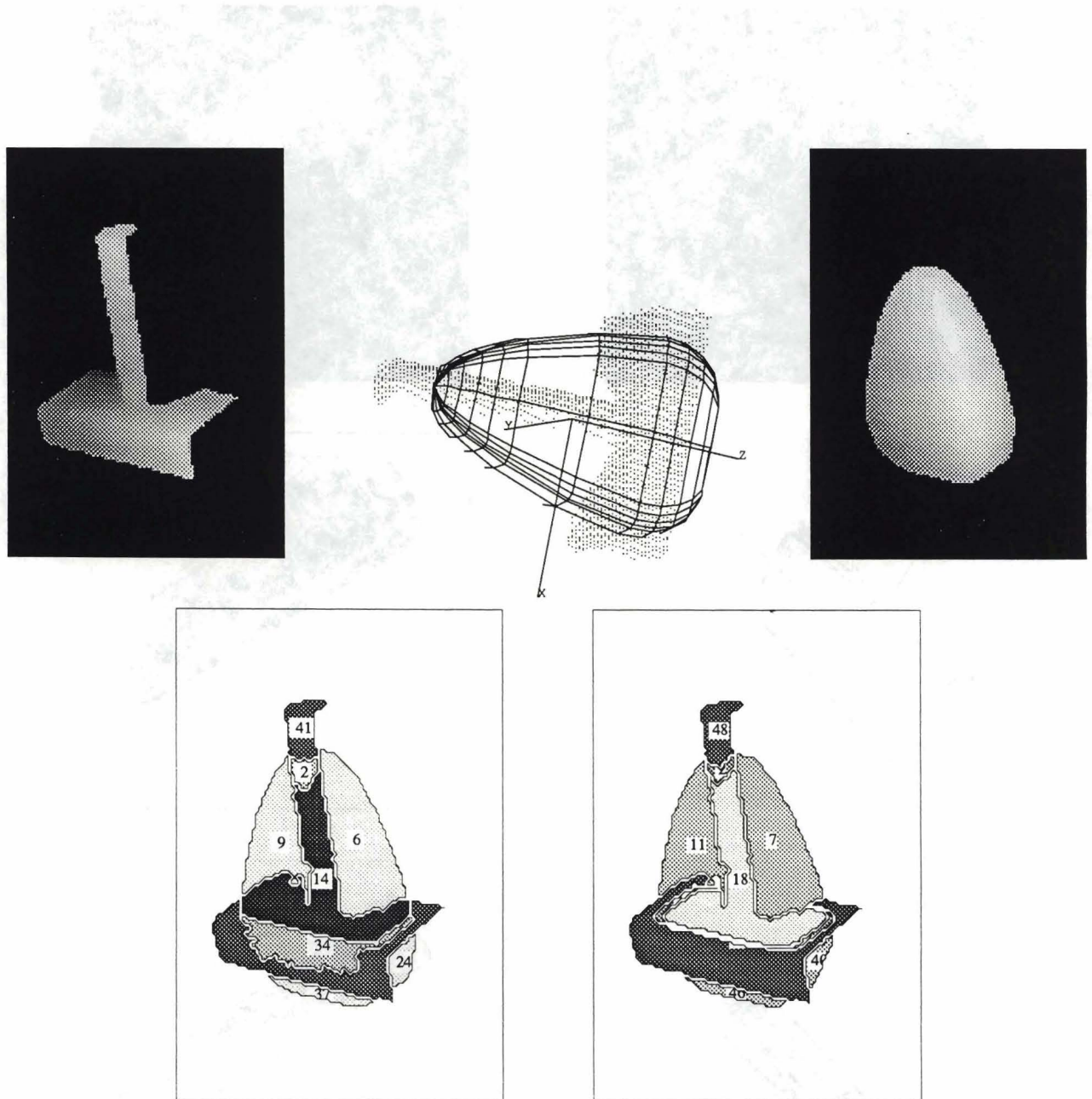


Figure 6.3: **The composite object:** Top: The range image, the global model, and its projection. Bottom: The Z residual-map and the IO residual map.

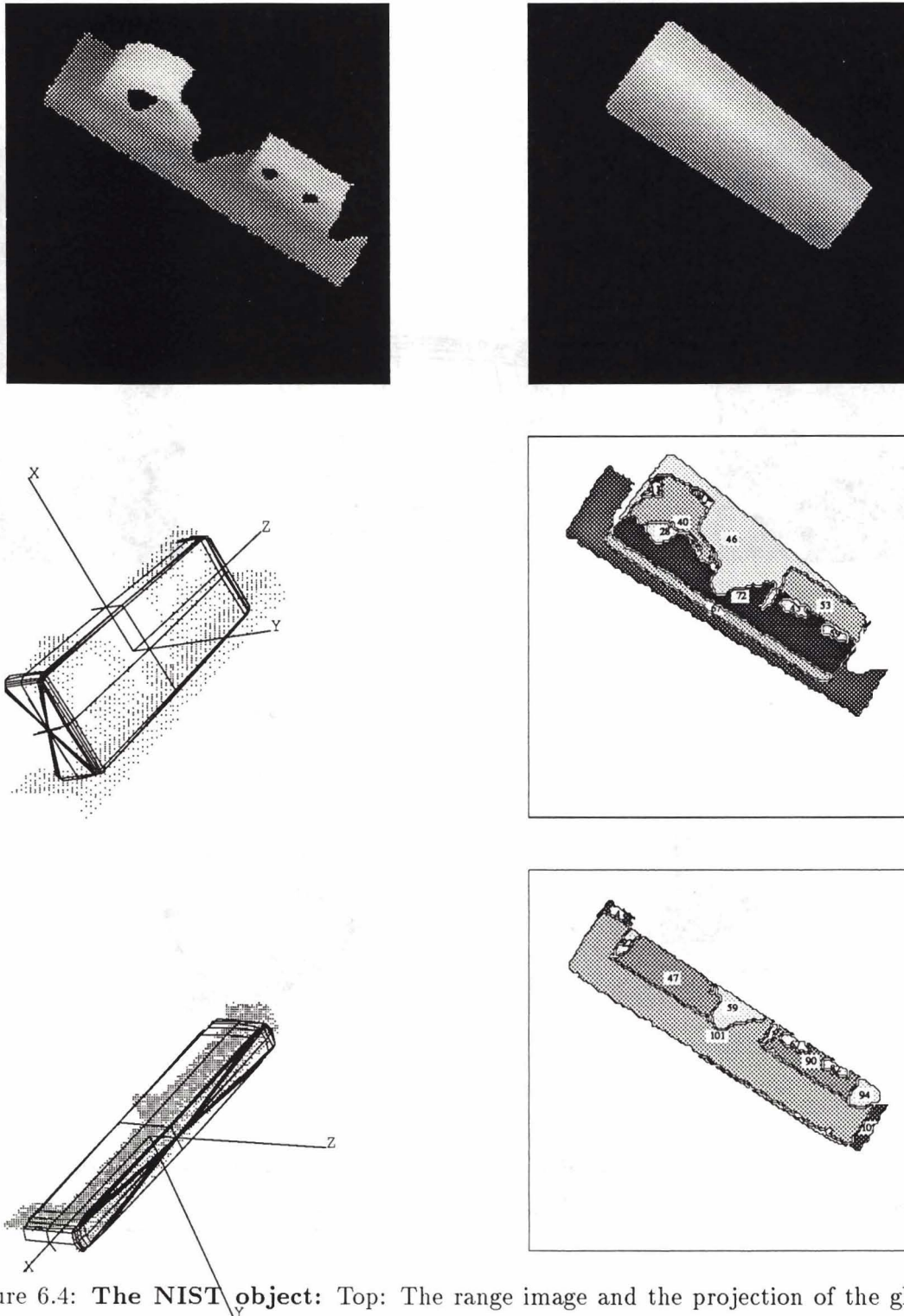


Figure 6.4: **The NIST object:** Top: The range image and the projection of the global model; Middle: The global model and the Z residual map for the global model; Bottom: The model for the base of the object and its Z residual map showing occluding regions

points that do not belong to the model. For example, the base in the NIST object shown in the bottom row of figure 6.4 underestimates the points belonging to the other parts. The `occ_ok` regions (47 and 90) are labeled to show that the occlusion is acceptable and that those residuals should not adversely affect the model for the base.

6. **Occluding region overestimated (`occ_over`):** Regions where the model overestimates the data points not belonging to it. It is clear that this is undesirable and therefore counts against the model during evaluation.

6.2 Residual Analysis

The residuals generated by comparing the recovered model on the given data form the basis of our coarse-to-fine volumetric segmentation approach. The residuals of the superquadrics can be referred to as the deficiency in describing the convexity of the object. We now describe methods of generating the residuals based on the inside-outside function (IO-residual maps) and those based on the point-to-point correspondence along the viewing direction (Z-residual maps). Euclidean distance can be used in IO-residual maps instead of the inside-outside function value.

6.2.1 Residual Analysis by Inside-Outside Function

The inside-outside function value corresponds to the expansion/contraction factor β for the given point, and therefore is not described along a fixed direction. As described earlier, its direction is along the line connecting the point with the center of the model. An important consequence of this is that the point can associate itself with the surface not *visible* from the viewing direction. Although it is a strong indication of the presence of concavities, it can be misleading in the case of thin objects. Together with the analysis along the viewing (Z) direction, it provides a check for the existence of concavities or combination of concave surfaces (in global sense). Clusters of points that are correctly modeled according the IO-residual map but are overestimated according to the Z-residual map, belong to the hidden the side of the model, and therefore need to be discarded if a convex combination of surfaces is desired. This check is similar to the surface normal check for the detection of concavities, but does not require explicit computation of additional quantities like surface normals. The quantity of deviation at individual points can be the real Euclidean distance instead of the

inside-outside value, although the latter is available as part of the recovery procedure while the former requires explicit computation.

Generating the IO Residual-map: The analytical formulation of the inside-outside function allows for an inexpensive computation of this map at the data points. This gives `io_s_over`, `io_s_exact`, and `io_s_under` regions. For the `io_c_over` regions, where superquadric predicts data (along the viewing direction), we need to compute the projection of the superquadric occluding contour in the image coordinate system following the procedure described in chapter 5.

6.2.2 Residual Analysis Along the Viewing (Z) Direction

Since we are aiming to describe the single viewpoint data, it is clear that we want to minimize the modeling error along the Z direction. This fact, however does not contradict the formulation of the superquadric recovery in terms of the IO function, since any formulation assumes that the data is inherently describable by the superquadrics without requiring segmentation. Hence, even with the Z-distance formulation of the superquadric recovery, segmentation will still be required. Thus we isolate the recovery procedure and the residual analysis formulation to achieve best results.

Given that we want the Z-distance residual, let us define *underestimation* of surface as the points that are outside of the visible superquadric surface, and *overestimation* of the surface as the points that are inside the visible surface (beyond a certain acceptable `z_tolerance` value). Additionally, the non-existent points that are described by the superquadric, are due to the symmetry constraint (if a superquadric model does not exist) or the presence of concavities.

Generating the Z residual-map: For the purpose of comparing the superquadric model with given surface points to generate a difference map, we have to compute the distance of every given point from the superquadric surface *along* the given direction. There are two ways to accomplish this:

1. Compute the distance analytically, if possible, else numerically.
2. Reconstruct the superquadric surface in the scanner coordinate system and then perform point by point comparison in z direction to compute the difference map.

Due to the absence of a closed-form analytical solution to compute the distance of a point from a superquadric surface (even when the vector along which the distance is to be computed is given), we backproject the model into the scanner coordinate system and make point by point comparison to generate the residual map. This approach has many advantages over a numerical method that only computes the distance of a point from the superquadric model. A complete backprojection along the viewing direction gives us an immediate assessment of the extrapolation of the model into non-existent data.

6.3 Residual Clustering for Further Processing

Following the above analysis, each pixel can be marked to be of the basic six types. To use these individual points as clustered units, it is important to label the connected pixels as a single cluster. A 4-connected neighborhood is used to enforce pixel connectivity in a cluster. The resulting labeling produces clusters of each type, so that they can all be referenced as units. Once we have the clustered residuals they can be treated as a graph structure and their connectedness at cluster level can be determined.

6.3.1 Residual Adjacency Graphs (RAGs)

A region-adjacency graph is constructed with each node representing a region and edges labeled according to the following relationships between a pair of regions:

1. Connection between two valid non-`c_over` regions.
2. Connection between two valid regions (one is `c_over`).
3. Connection with an invalid region (region too small).
4. Connection with a background/hole region.
5. Connection with an occluding region.

A region is valid if it has more than a minimum number of points (usually 1 or 2). Figure 6.6 shows it for the global model of scene3 (figure 6.5) containing 4 clusters of data. The RAG encodes connectivity information of the residual regions, and therefore can be used to break isolate data clusters that are not connected in the image. This can be easily accomplished by removing the edges corresponding to cases 2 (one of the region is `c_over`) and 4, and analyzing the graph for connected components. Although similar

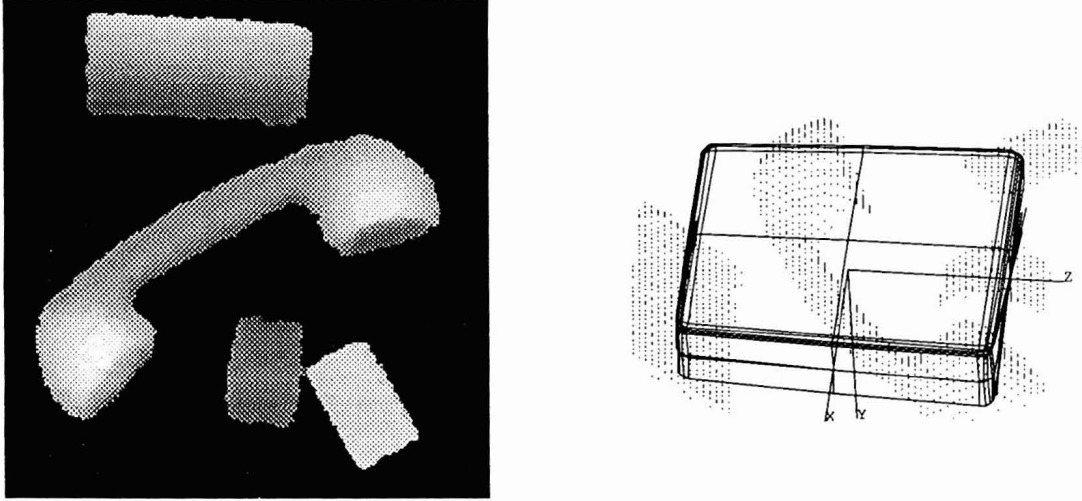


Figure 6.5: **The global model for Scene 3:** The range image and its global model.

in concept to the surface adjacency graph (SAG) described earlier, the RAG is generated for every new data-set. Thus RAG dynamically changes as the models evolve and data clusters become disconnected during the segmentation process, but are actually connected in the SAG sense. Therefore, we need the RAG description even though surface graphs can do the same initial decomposition. The RAG for scene3 breaks the scene into four independent parts and initiates independent models on each one of them. Notice that during the connected-component analysis, the edges connecting with occluding regions are not removed. It has the advantage of providing continuity of data along occluded regions, as also isolating the regions that are solely formed of occluded points. We will later show that during the second iteration for scene3, such a case occurs with the global model describing the handset of the phone.

6.3.2 Using the Residuals for Superquadric Evaluation

Having described the 6 types of residuals and methods of generating them, we now discuss the issue of using them to evaluate a model. For a data set of cardinality n , we know the number of points that are exactly described, underestimated, and overestimated. The absolute numbers are not of much use in evaluation, since they are size dependent. To enforce scaleability and size invariance, we use the relative measures. This information is stored as percentage of data that is exactly described, overestimated or underestimated. Various thresholds can be put on these fractions to define the acceptance criteria. Similarly,

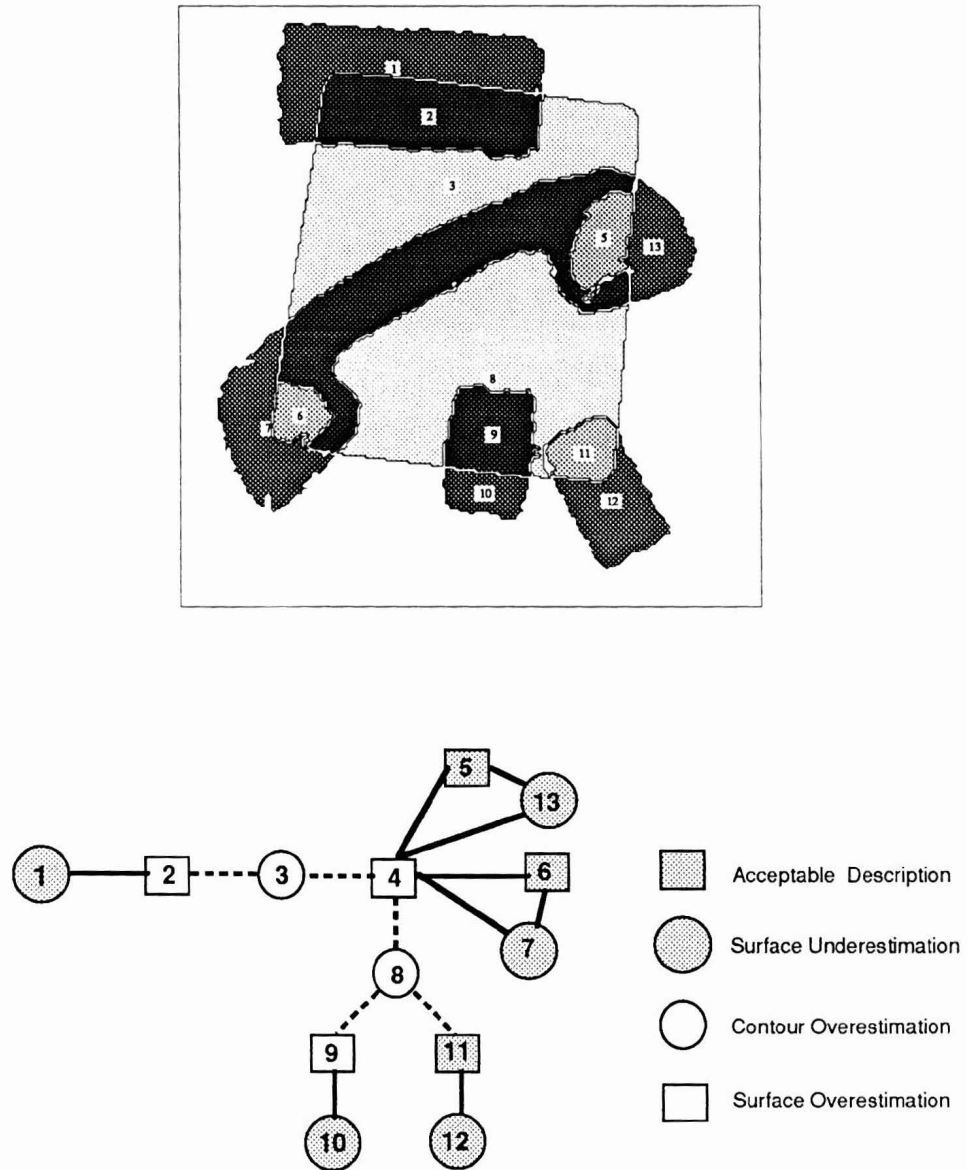


Figure 6.6: **Residual adjacency graph (RAG) for Scene 3:** Top: Z-residual map for the global model of Scene3, and its RAG (bottom). Connected component analysis gives four independent clusters for further processing.

the `c_over` can be studied relative to the original data. Usually there is a 10 to 20% `c_over` regions due to noisy data near the edges and due to the fact that superquadric only approximately follow the boundaries of real data. A 20% value means that in order to describe 100 points in data, an overestimation of 20 points occurred in volumetric sense. Similarly, an acceptance condition for `occ_over` regions can be enforced. A combination of these conditions gives the acceptance criteria to the control module, and defines the termination conditions for the model recovery.

In addition to the residuals for the entire data, residuals for individual surfaces can also be obtained to further refine the acceptance criteria. The residuals are computed only for the domain of the biquadric surface included in the superquadric model. Thus, an acceptable model will describe all its constituent surfaces with high confidence. It is also possible to selectively enforce the thresholds, for example, some of the surfaces (eg. smaller than a fixed size) can be ignored during the evaluation, while others can be given more weight. Together with the globally-relative acceptance criteria outlined above, the surface-relative criteria form a comprehensive criteria for superquadric model evaluation.

6.4 Chapter Summary

Given a recovered superquadric model, we developed a set of criteria for complete superquadric evaluation. Both quantitative, as well as the qualitative measures are required to exhaustively evaluate a superquadric model. Residual analysis forms the basis of the global to local volumetric segmentation, and in the next chapter we present the issues involved in accomplishing it.

CHAPTER 7

Volumetric Segmentation: The Control Flow

We continue with the design of the volumetric segmentation module in this chapter. The description available at this stage is in the form of the piecewise bi-quadric patches along with the information about the surface-type, orientation and edges. Also available is the global superquadric model and a set of criteria to exhaustively evaluate it given the original dataset. The main task of the control structure for volumetric segmentation can be defined as systematically integrating surface descriptions with the global-to-local superquadric recovery approach, evaluating the intermediate descriptions, and deciding on the strategies for segmentation.

We begin this chapter by addressing the important issues in superquadric-based volumetric segmentation of real range data obtained from a structured lighting range scanner. We will then discuss the control structure for integrated segmentation procedure in detail.

7.1 Issues in Volumetric Segmentation

7.1.1 What are Superquadrics Modeling?

Surfaces have more local support and hence provide reliable intermediate-level clustering in terms of bi-quadrics. Unfortunately, the analytical correspondence between biquadrics and superquadrics is minimal since biquadrics belong to the class of non-central quadrics while superquadrics are more appropriately non-linear deformations of ellipsoids which belong to the class of central quadrics¹. In chapter 4, we established some correspondence between

¹The appropriate term for superquadrics is superellipsoids since they do not cover general quadrics.

the two for the purpose of axis alignment for curved objects. Essentially, we look at a superquadric model as not only a surface description, but also as an object-centered volumetric description describing a convex cluster of points as close as possible to its surface. For example, a superquadric with $\varepsilon_1 = \varepsilon_2 = 0.1$ gives a convex combination of planar patches that actually meet at C_1 (surface normal) discontinuities. But from the superquadric point of view, the discontinuity is smoothed out, since the model is differentiable everywhere. An important ramification of this unique formulation is that our approach to segmentation departs from the standard feature-based techniques and generalized cylinder-based formulations.

7.1.2 Superquadric Recovery Formulation for Segmentation

An important difference between model-based approaches and our formulation lies in how models are matched or recovered. Since our models represent a continuum of shapes, the techniques relating to stored model matching are not useful. Instead of *matching* a stored model, we allow the model to *recover* all its parameters starting from a basic ellipsoidal shape. The formulation is based on surface and volumetric constraints and not on any features like edges. The reason being that the local continuity analysis is of little use in invoking a globally differentiable superquadric model. As mentioned earlier, the edges of a superquadric model have to be viewed as high curvature contours or as occluding contours. Thus, an edge-based approach has to be formulated in terms of occluding contours, providing the surface constraint and the occluding contour constraint. As noted before, the occluding contour formulation does not improve the model recovery capability. In any case, no matter how the model recovery is formulated, the most difficult aspect of using the recovery-based formulation is that the domain of the model (data points for which the model has to be recovered) has to be defined before the model is recovered. This rules out any possibility of segmentation during the model recovery phase.

Therefore segmentation has to be performed by a process that is separately formulated. Skeletonization is a popular approach [Nevatia and Binford 1977, Pentland 1987b, Terzopoulos et al. 1988, Rao 1988], but is sensitive to occlusion and requires the knowledge of internal boundaries for complex objects. The difficulties involved in a reliable detection of internal boundaries in range images renders them impractical for our use. In the absence of any domain knowledge we want to provide for the possibility of occlusion due to parts, and handle it in a perceptually significant manner. Besides, skeletonization works best for curved surfaces and in general it is ambiguous and distracting when the volume consists of

planar surfaces.

7.1.3 Coping with the Missing Information

Data can be absent due to occlusion with other parts, and shadows cast due to the scanner geometry. Shadows due to scanner geometry are considered as missing data and the model is allowed to predict data on them. In the context of residual analysis, it was mentioned that occlusion due to other parts in the scene can be handled easily by considering the residuals for those points to be acceptable. However, occlusion can prevent a part from constraining the model to get its right size and position, although shape and orientation are less sensitive to occlusion.

An additional source of missing data in single view data is self-occlusion. An object in general position and orientation gives an idea of its volume or global shape. If the self-occlusion is such that it hides the volume of the object, then the view is degenerate and additional data or reasoning is required to get volume estimates. Given a non-degenerate view, superquadrics fill in the missing data by imposing symmetry constraints. There are no other options available because superquadrics are symmetric models, and available data constrains the model and predicts a symmetrical hidden shape. This is not a problem if the object is to model the available data and the object in consideration is indeed symmetric in superquadric sense. In fact, this only predicts the hidden side of the object and is the best guess given the single view. Additionally, it provides pointers for where to look for additional information by predicting the existence of information. A secondary procedure can verify the prediction by manipulation or imaging the hidden side if possible, depending on the application. Generally speaking, self-occlusion is not a problem if the viewpoint is non-degenerate for the data, but in a complex scene objects can be in their general or degenerate viewing position (along a given viewing direction) and hence it is not possible to make the general viewpoint assumption for all the parts in general. This means that the possibility of existence of degenerate views of parts has to be taken into account when analyzing the scene. The degeneracy of the view affects different kinds of objects differently. A box appears as just a plane, resulting in a flat “volumetric” model. A cylinder, on the other hand, poses such problem only if viewed head-on, in which case only the cross-section information is available. If the curved surface is visible, then the cross-section information of a cylindrical object can be extracted from the surface curvature by the superquadrics.

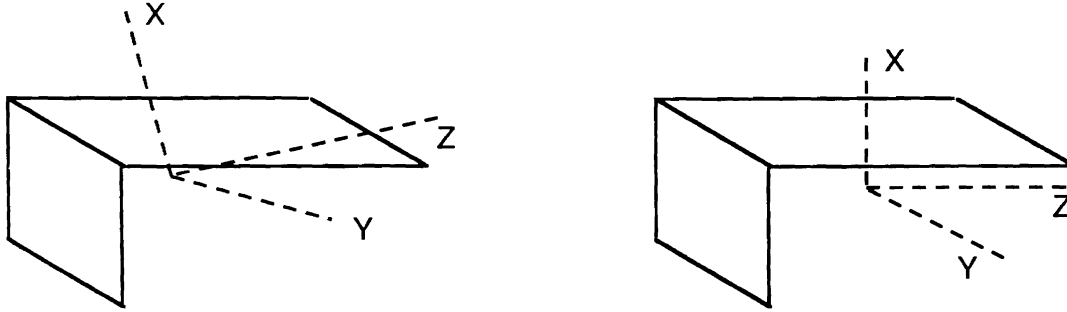


Figure 7.1: **Orientation of the object coordinate system:** The eigenvectors give a moment based estimation of the orientation (left). Orienting it along one of the surfaces is a better estimate for the least-squares based optimization procedure (right).

7.1.4 Orienting the Initial Superquadric Model

As a rule of thumb, the initial model (the ellipsoid) is oriented along the eigenvectors of the moment matrix, and the Z axis is aligned along the eigenvector with maximum eigenvalue (corresponding to least axis of inertia). The model is sensitive to the selection of the Z axis (axis along a_3 dimension) in the object-centered coordinate system for cylindrical models, requiring that cylinders curve only along the Z axis. In addition, it is helpful to orient the model as close to its final orientation as possible. Due to the least-squares-based recovery process, and the initial axis estimation using eigenvectors of the moment matrix (which is biased due to the self-occlusion in single view data), the optimization procedure can get stuck at the local minima. Instead, if the model is oriented such that it corresponds to one of the constituent surfaces (with the assumption that the orientation of any one of the surfaces is also the most likely final orientation), the recovery procedure exhibits quick convergence to the correct model² (figure 7.1). We have empirically tested this heuristic and found it to give consistently better solution than the uncorrected one. Since the initial guess is close to the final orientation, the model convergence is also improved. The problem of selecting the Z-axis in case of curved surfaces is resolved by biquadric surfaces. Algorithm 7.1 gives the general approach for the orientation of the object coordinate system with respect to a known world coordinate system.

²We use the words “correct” and “acceptable” based on the study of residuals, and not on our subjective evaluation.

1. Compute eigenvectors and eigenvalues of the moment matrix for the 3-D points in world coordinate system. Orient the object centered system along the eigenvectors.
2. **If** (single curved patch)
 - then**
 - Align Z-axis along the axis with least coefficient in the standard form of the biquadric.
 - else if** (single planar patch)
 - then** Align Z-axis along the eigenvector with maximum eigenvalue.
 - else** /* cluster of patches */
 - Orient the object coordinate system in the same sense as the coordinate system of the largest patch in the cluster. If curved surfaces present then prefer that orientation (to correctly align the Z axis).

Algorithm 7.1: Orientation and Z axis placement of the initial superquadric model.

7.1.5 Surface Support for the Superquadric Data

The volumetric segmentation procedure considers only those data points that have biquadric surface support. The support is exhibited by the inclusion of a point in one or more surface patches. The logic behind this requirement is that if a data point cannot gather surface support then it can be excluded from the volumetric consideration as well. It also has the desirable effect of leaving out the outliers (filtered by the iterative regression approach of biquadric recovery) that can be distracting for the least-squares procedure for the superquadric recovery.

7.2 The Strategy for Volumetric Segmentation

A schematic diagram of our approach for volumetric segmentation is shown in figure 7.2. The surface and superquadric recovery modules are applied independently to the range data. The surface segmentation is refined to obtain surface patches that can be used by the volumetric segmentation, and surface adjacency, edge-type, and surface orientation information are extracted from the standard form of the biquadrics. Surface segmentation is considered final in the sense of $2\frac{1}{2}$ -D description. Residuals defined in the previous chapter are generated for the global superquadric model.

The objective of the control module is to evaluate the global superquadric model and

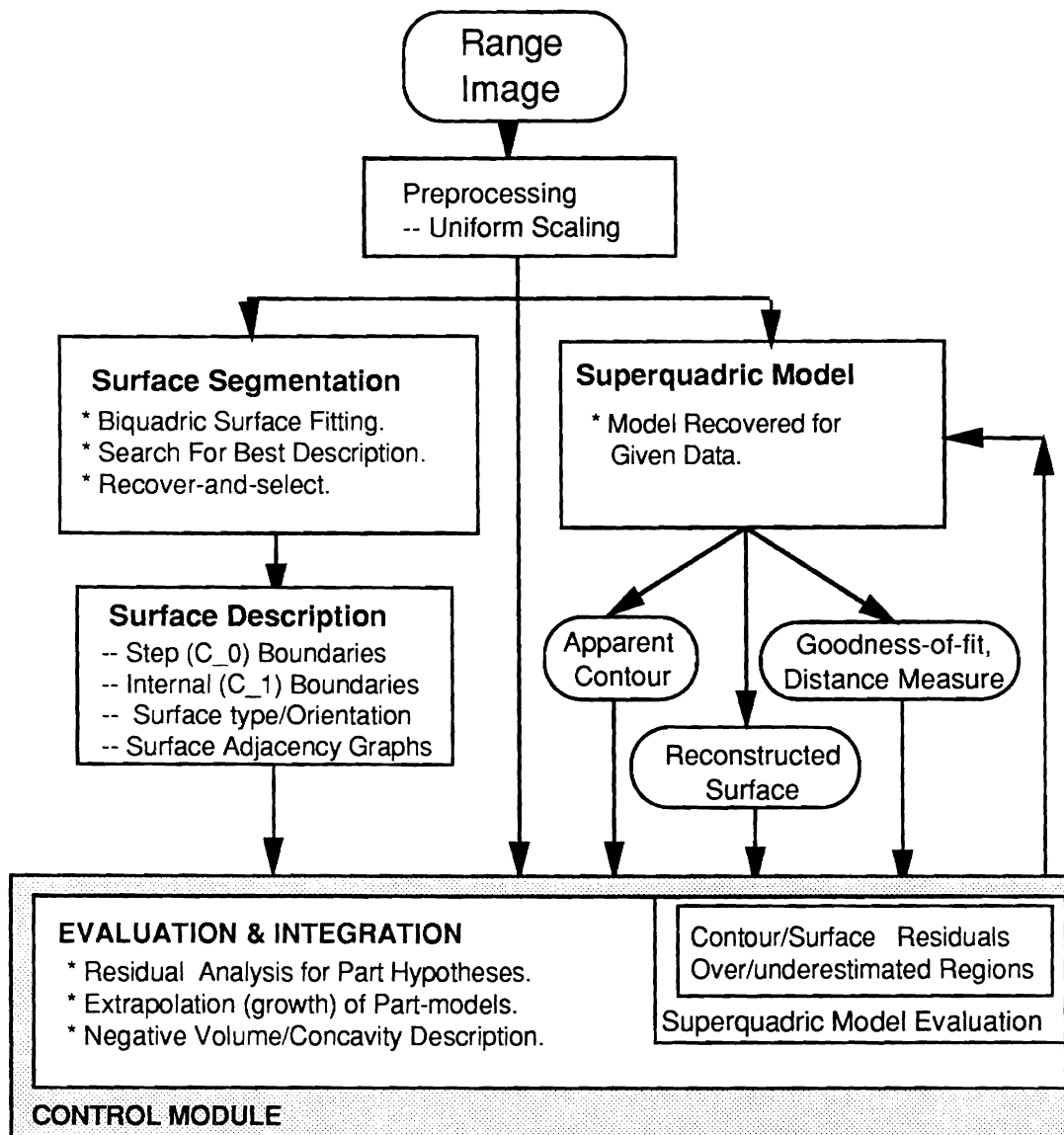


Figure 7.2: The Control Flow of the SUPERSEG system: An integrated approach for surface and volumetric segmentation.

devise the appropriate strategy to either segment the data by hypothesizing part-models as indicated by the residuals, or terminate the procedure, or generate negative volume descriptions. The segmentation at surface level can be used to guide the volumetric segmentation, but when to rely on surface information, and what surface information to use is not clear at a first glance. Depending on whether or not to invoke surface information, there are three basic strategies:

1. For every surface model one superquadric model is recovered. This strategy has been most popular with researchers interested only in using superquadrics for final modeling of a segmented description arrived at by using techniques not involving the superquadric model. Thus the superquadric model may or may not be the right model for the segmented data. This is the weakest strategy to follow, but we will describe the advantages of integrating it as part of a more complex strategy.
2. Segmenting the object at concave discontinuities found during the segmentation, and recovering a superquadric model for each convex component. This is not a general strategy, and will not always work since it assumes that surface segmentation is final in volumetric sense as well, and groups of surfaces can be combined together to form a volume. However, we want the control structure to identify the situations in which it will work, so that it can make use of the surface information to derive the correct segmentation.
3. By following a global to local approach, driven solely by residual analysis, and without the help of surface patches, it is possible to generate part hypotheses at the `s_under` regions, place local superquadrics (part models) there, and let them grow (extrapolate) as the global model shrinks by discarding the points that were underestimated by the earlier fit.

The third strategy is the most general one. However, it is too slow and tends to generate more false hypotheses than if surface information was also taken into account. Crucial information that surfaces provide is the existence of step edges and the concave surface normal discontinuities, as also the orientation of individual surface patches, which is of crucial importance in orienting the part (seed) superquadric models placed on `s_under` regions. We will first explain the three strategies as independent methods and then describe the integration of their best features to obtain a general control structure. We will illustrate the results of following these strategies for the composite object in figure 7.3.

7.2.1 Strategy 1: One Superquadric Model for Every Bi-quadric Surface

If the biquadric surface segmentation is complete at the superquadric level in the sense of 1-to-1 correspondence between the biquadric and superquadric surfaces, then there is no more segmentation necessary at the superquadric level. A superquadric model can then be recovered for each bi-quadric surface patch. Cases where this will work include: Curved surface patches where each patch provides enough information to constrain the superquadric recovery. This is not a general strategy, as it does not allow convex combination across convex discontinuities, and hence admits a small class of objects. The other researchers using the superquadric modeling methods [Ferrie et al. 1989, Pentland 1987b, Darrell et al. 1990, Terzopoulos and Metexas 1990] have made the assumption that a 1-to-1 correspondence exists between the surface descriptions and superquadric models and thus are not able to handle volumetric parts (like boxes) with convex discontinuities.

The result of fitting one model for every surface to the composite object gives an unrealistic looking description shown in figure 7.3 (bottom left). The description for the NIST object is shown in figure 7.4. However, this strategy can be used by the general segmentation procedure for the following purposes:

1. Some of the planar patches are described by the biquadric recovery procedure as second-order polynomials. This can be due to the distortion of data or noise in a tilted planar surface. Planar surfaces require different consideration than curved surfaces, and hence it is necessary to be confident about the fact that a surface is second-order or planar. The strategy-1 provides an unambiguous *planarity check* by computing the dimensions of the data in the surface domain by following the superquadric fitting procedure. This approach is similar to the approach based on computing eigenvectors of the moment matrix for planarity check [Hoffman and Jain 1987], but performs better because it also gives an estimate of the size and shape of the surface to make a decision about the planarity of the patch. For this test, we keep the Z-axis along the shortest axis (along the axis of maximum inertia). Thus, if the a_3 dimension of the object is small then it can be concluded that the patch is globally planar. The surfaces labeled 1,3, and 4 in the composite object (figure 7.3) result in flat superquadric models, while the curved surface gives a box-like model.
2. By fitting a superquadric to every surface, we get an estimation of the global orientation of the patch in 3-D. This information is not present in the biquadric parameters because the Z-axis is fixed for biquadric patches. The surface orientation information

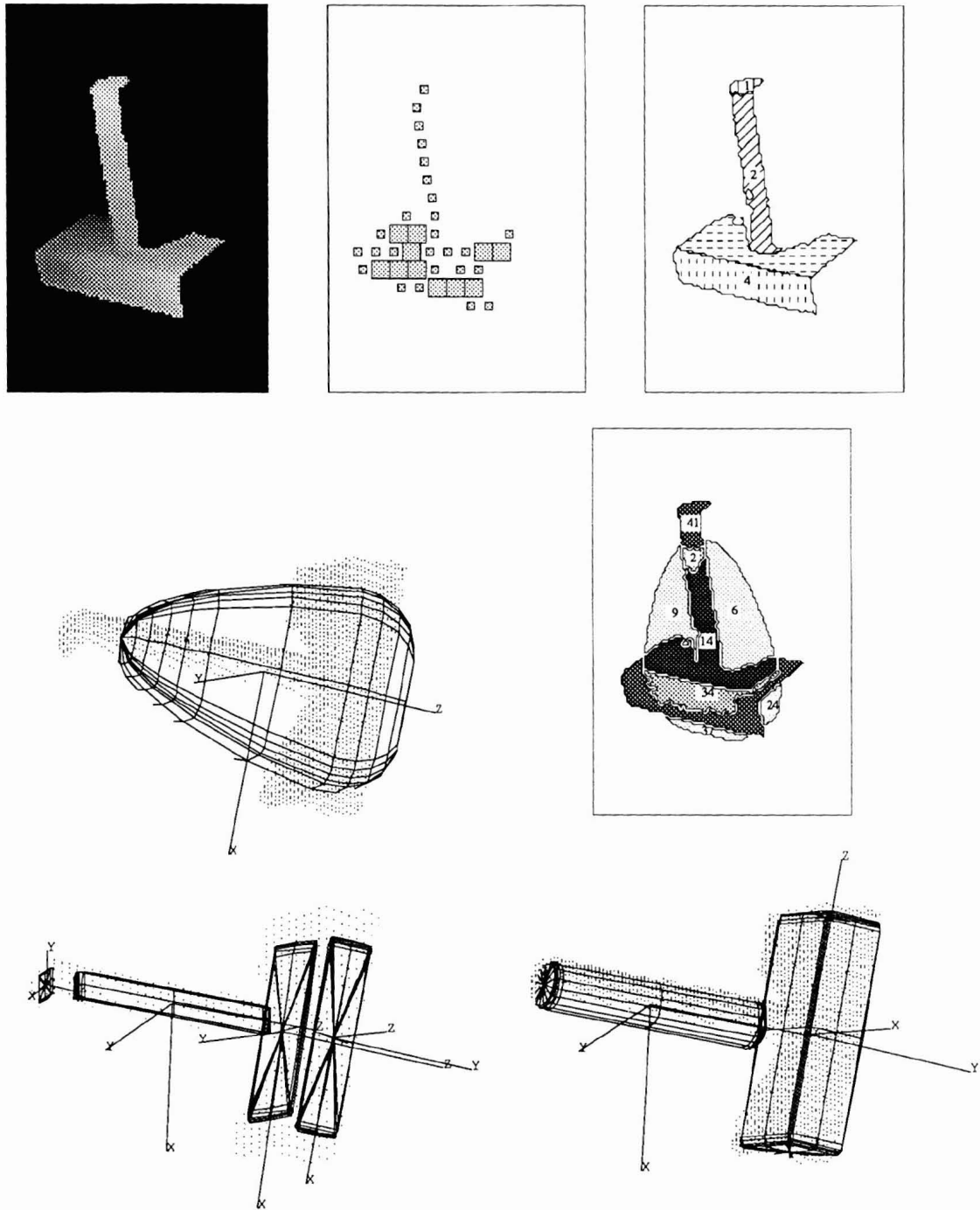


Figure 7.3: **The composite object:** Top: Range image, bi-quadric seed regions, bi-quadric segmentation. Center: The global model and its Z-residual map. Bottom: Left: Result of strategy 1: One superquadric model for every surface. Right: Result of strategy 3: Shrinking the global model.

can be used in orienting the initial estimates of superquadrics hypothesized by the control module if the surface is included in part or completely by the hypothesized model, as explained earlier and shown in figure 7.1.

3. Surfaces patches can also provide starting data regions for growing or extrapolating the superquadric models. Although, extending the surface patches in this manner will not work in general because the patch may not correspond to any superquadric crosssection.
4. Recovering a superquadric model for every surface gives an intermediate description, which is final in surface sense but not in the sense of the optimal volumetric description.

The primary reason that we do not consider this description as the final description, even for the patches that have superquadric models at surface level (for example the composite object), is that it does not maximize the positive volume of the data. The box in the composite object is better represented as one volume rather than two “flat” volumes. Thus, the control structure enforces the minimum volume constraint for a given set of points (in the model recovery formulation), but it aims to maximize the positive volume by extending the model as much as possible without creating negative volumes.

7.2.2 Strategy 2: Grouping Convex Surfaces

A more sophisticated strategy will allow the models to combine surfaces along convex discontinuities, thereby allowing a scenario where individual surfaces need to be grouped to get a volumetric fitting (as in the example of the composite object). If the grouping of surfaces is sufficient to form the volumetric model, then segmentation of surfaces is not necessary. But this is not true in general, because the surface patches may need further segmentation for a volume description, for example, an L-shaped object needs the surface with the L-shape to be broken into at least two parts. Therefore, a more elaborate strategy is needed that will recognize the possibility of further surface segmentation, and at the same time use the convexity information if possible. The composite object can be completely segmented by grouping the convex connected components together, as shown in figure 7.5.

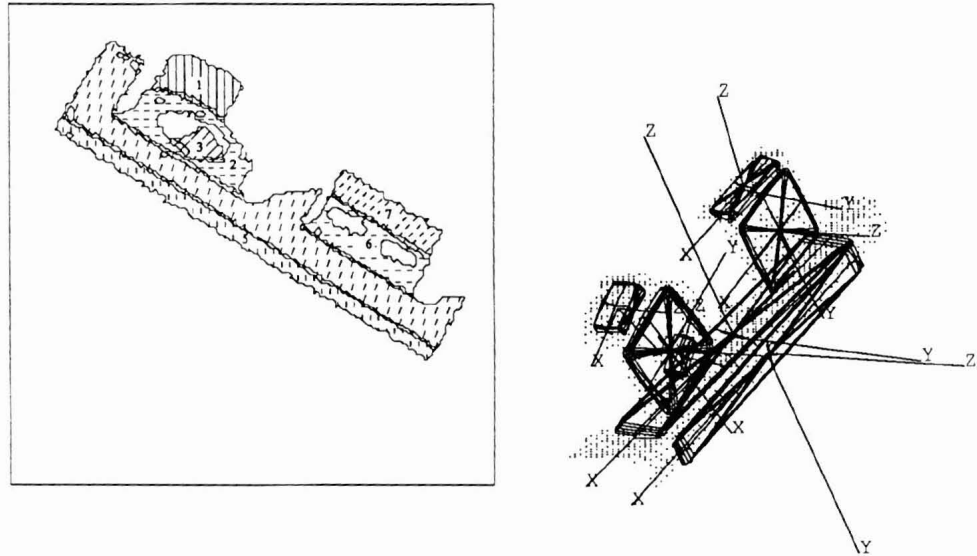


Figure 7.4: **The NIST object: One superquadric model for every surface:** Left: The bi-quadratic surface segmentation. Right: The corresponding superquadric models for individual surfaces.

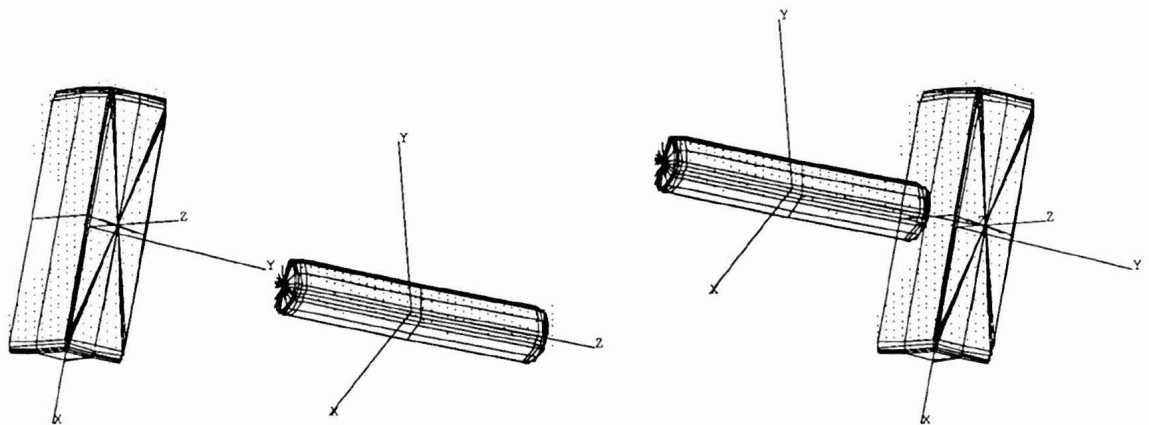


Figure 7.5: **The composite object: Grouping convex surfaces together:** The concave edge at the traversal join decomposes the object into a box and a cylinder.

7.2.3 Strategy 3: Global to Local Superquadric Fitting

By following a global to local approach, driven solely by residual analysis, it is possible to generate part hypotheses at the `s_under` regions, place local superquadrics there, and let them grow (extrapolate) as the global model shrinks by discarding the points that were underestimated by the earlier fit. No surface information is assumed to be available.

The strategy is illustrated for the composite object in figure 7.6 where the global model (untapered), begins to discard the underestimated points on the cylinder (corresponding to OU-region 41). We will later describe a procedure for selecting the starting OU-regions. The global model shrinks iteratively as the local model on the cylinder accepts the points rejected by the global model. When the global model converges to the box shape, it has no `s_under` residuals to discard in the direction of the local model, so it stops shrinking and the local model stops growing. Note that residuals change drastically as the model begins to approach the cluster of points that it can model without significant `s_over` or `s_under` residuals. The final models look exactly the same as those obtained using the convex-combination of the surface patches. The goodness-of-fit values and the qualitative residuals of the global and part models are shown in figure 7.7. Notice that the qualitative residuals (measured as percentage of total data points) of the global model improve as the model converges to the final acceptable description of a box. The residuals of the part-model (corresponding to the cylinder) remain acceptable throughout the iterative process. However, it is important to note that the goodness-of-fit based on the volume considerations (which is the actual optimization function) cannot be used to evaluate models recovered on different data sets. For example, the volume-GOF actually increases in case of the part-model (figure 7.8) because of the increase in the *volume* and not due to increase in the value of the inside-outside function (the surface constraint). Thus, our quantitative evaluation criteria considers the global goodness-of-fit without the volume factor (equation 5.10 instead of equation 5.9).

This strategy is general in application and fits in our global to local approach to the volumetric segmentation. In addition, it can use the surface information effectively by rejecting false hypotheses early in the segmentation process. We are now ready to present our general control structure for volumetric segmentation. We will now present our integrated approach, combining the elements of strategies 1, 2 and 3 to give an efficient control structure that systematically generates volumetric descriptions.

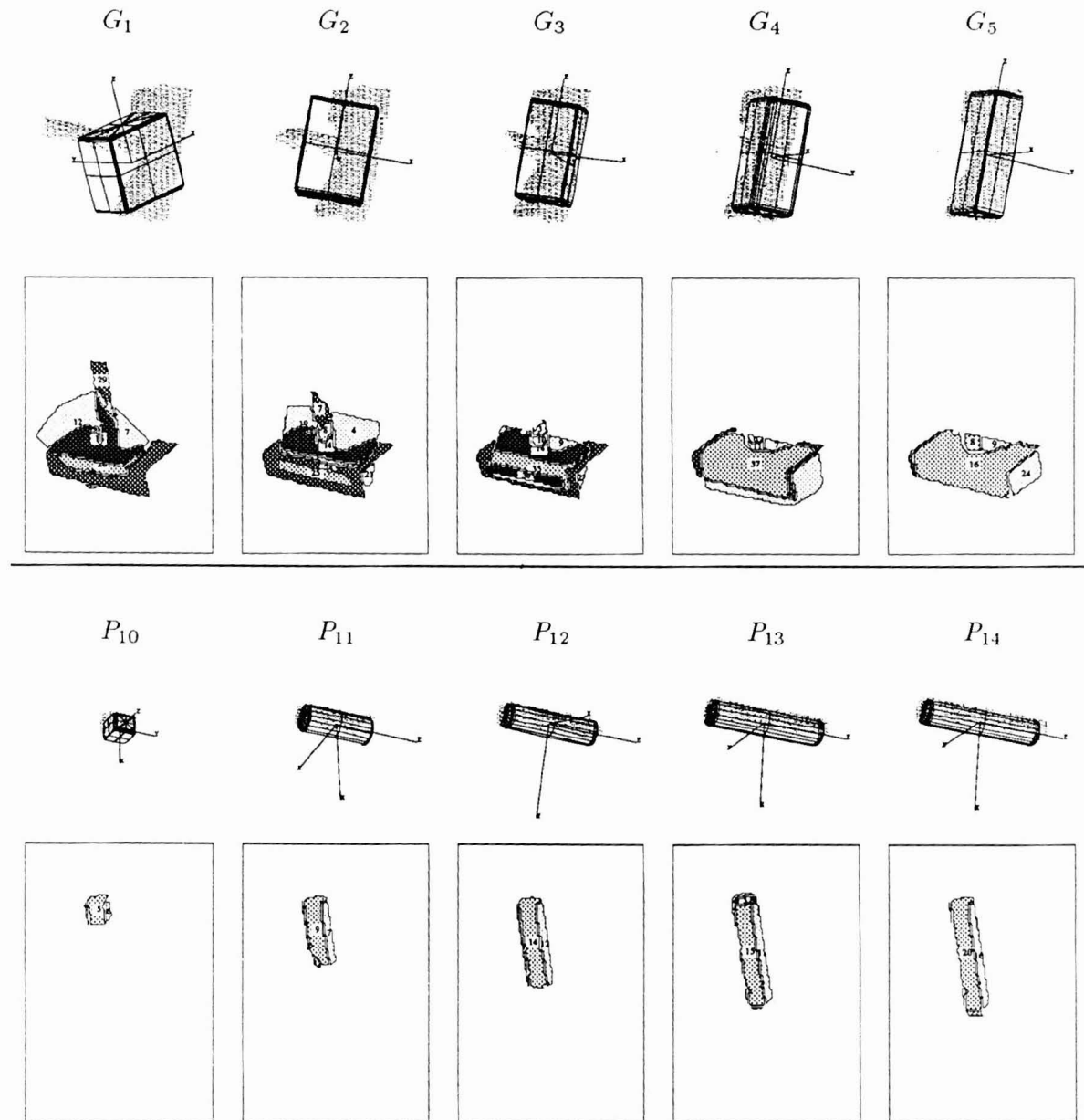


Figure 7.6: **The composite object: Global to local model growing and shrinking:** Top: Shrinking the global model and the corresponding residuals; Bottom: Growing the local model and the corresponding residuals.

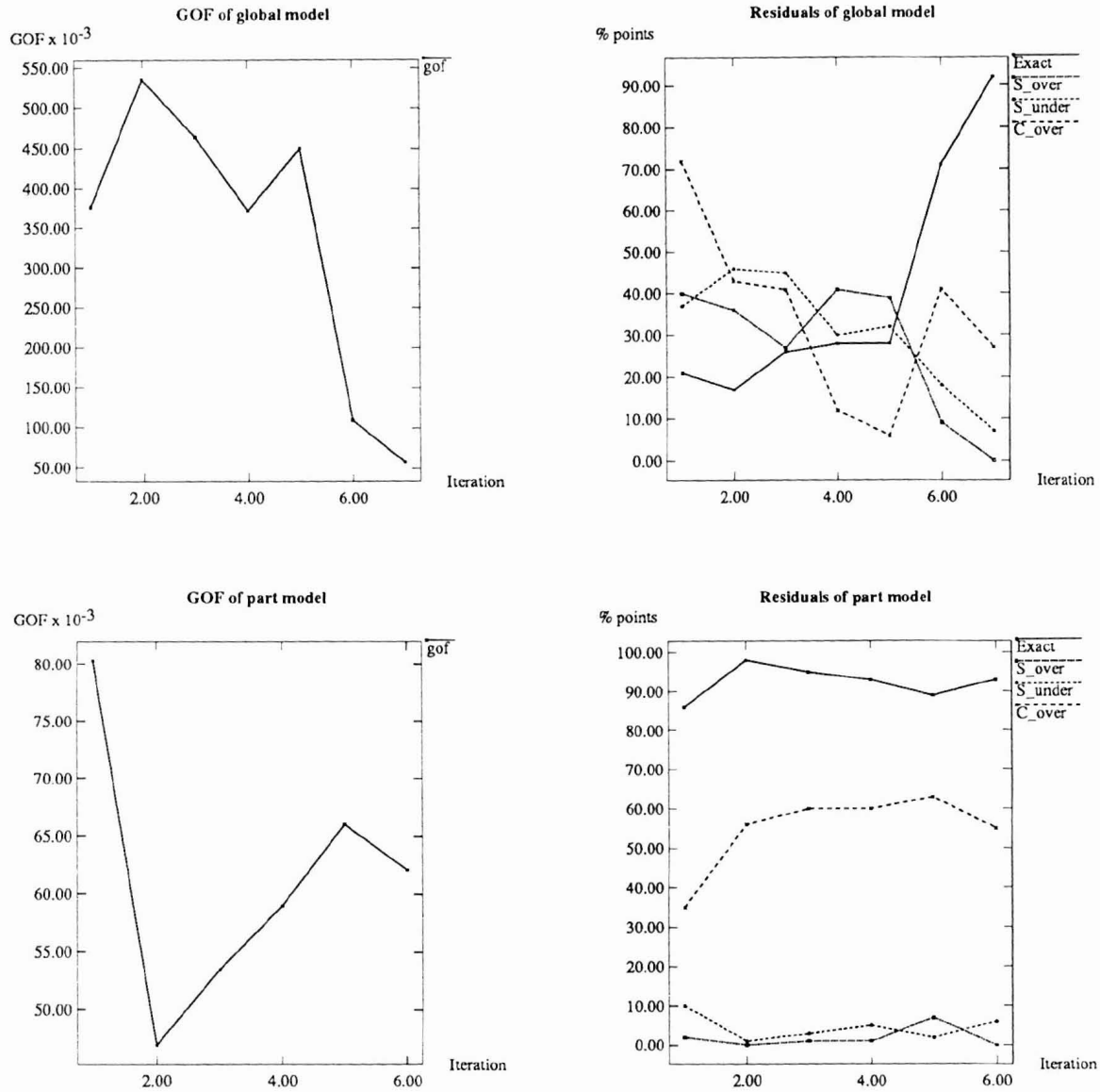


Figure 7.7: **The composite object: Residuals of strategy 3:** Top: For the 7 iterations of the global model. The goodness-of-fit values (left), and the residuals of the global fit as percentage of data points. Bottom: For the 6 iterations of the part-model corresponding to the cylinder. The goodness-of-fit values (left), and the residuals of the model.

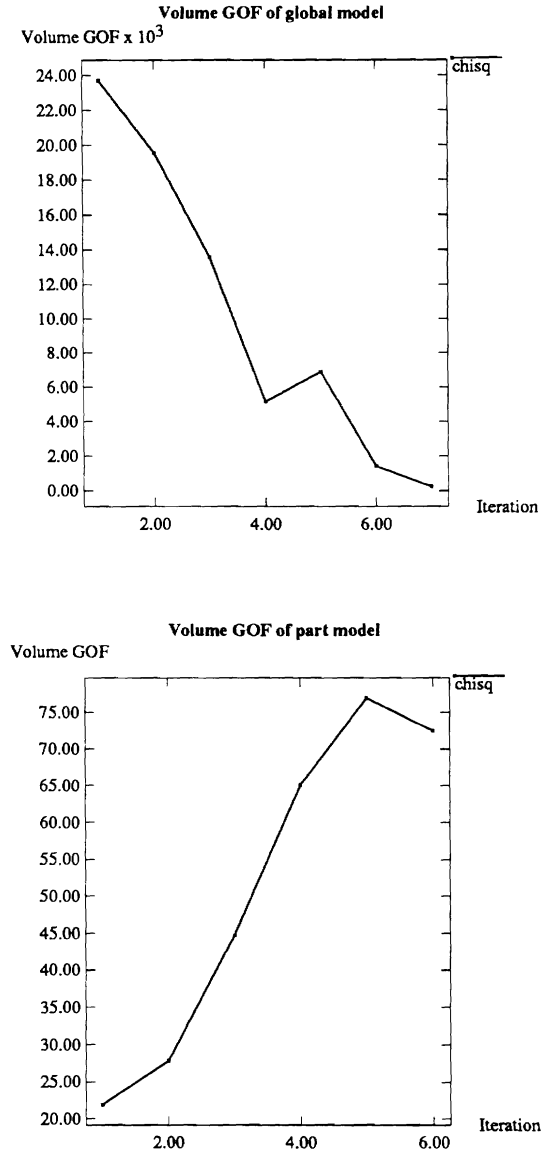


Figure 7.8: **The composite object: Volume-GOFs for strategy 3:** Top: 7 iterations of the global model. Bottom: 6 iterations of the part model. The Volume GOF of the part model increases as the model grows in size, while the non-volume GOF (figure 7.7 (left)) effectively decreases. For this reason, the volume factor is ignored during evaluation of the models and only the non-volume GOF is used.

7.3 General Strategy: Integrating the Three Strategies

As noted earlier, some of the surfaces require further segmentation to conform to any superquadric model in a combination of surface patches. This is the most difficult scenario and the one we focus most of our attention on. But at the same time, we don't want to ignore the possibility of cases 1 and 2, which simplify the volumetric description for simpler objects. Therefore, our strategy is to control the flow of the segmentation in such a way that the easier strategies can be identified early on, but not become a significant overhead if the object turns out to be more complex.

The global strategy beginning with the biquadric segmentation is outlined in the algorithm 7.2. Before describing our approach let us introduce the hierarchical structure of the superquadric model representation in the control module. In Figure 7.9, the global model for the complete data set is G_0 , and $[G_1...G_n]$ represent the evolution of the global model as it is refined by the global-to-local process. The first refinement gives G_1 with l part-models, $P_{10}..P_{l0}$, describing clusters of data taken away from G_0 . Thus, P_{i0} models are stored as children of G_1 and are inherited by all the subsequent refinements of G_1 . The global model either terminates as one model, or can break into more global models $[G_{n1}...G_{np}]$ as dictated by the dynamic connected-component analysis. This is the reason, Residual adjacency graphs are needed although Surface graphs also represent data connectivity. Each of the global models then behaves just like G_0 , but inherits all the children of its previous iterations. The children are likewise represented as a chain of evolution from P_{10} to P_{1m} . A child model can also become a global model if further segmentation becomes necessary for the data in its domain. Thus, the representation is recursive by definition, reflecting the control flow of the procedure which is recursive in its global to local approach.

The control structure branches out to the appropriate strategy depending on the relative amount of each type of residual. Algorithm 7.2 shows all the cases that can exist after the residual analysis. Each of these cases indicate the type of data being modeled, and the appropriate strategies can be invoked. For example, if only surface overestimations (**s_over**) are the significant residuals in the Z-residual map, then the underlying data is concave and requires negative volume descriptions. Flat objects result in significant surface underestimations. However, if a flat object has to be broken into parts, it will also have contour overestimations along the Z direction.

The most general case is when surface underestimations (**s_under**) exist along with other residuals. The algorithm for the general case is outlined in algorithm 7.3. The two most

1. Recover bi-quadric models for the range data using the recover-and-select paradigm (Algorithm 3.1).
2. Recover global superquadric model (S_G) for the data (Algorithm 5.1).
3. Recover superquadric models for each bi-quadric surface.
4. Analyze (S_G) for the Z-residuals and perform connected component analysis of the OU-regions.
5. **if**(multiple clusters)
then foreach (cluster i)
 Determine orientation. (Algorithm 7.1).
 Recover global superquadric S_G^i
 goto step 4.
else /* single cluster */
 If (Fit == OKAY)
 output current S_G
 else if (fit == SU) /* **s_under** regions exist */
 Provide for flat object during the general analysis.
 else if (fit == CO) /* **c_over** regions exist */
 Invoke contour constraint.
 else if (fit == SO) /* **s_over** regions exist */
 Analyze for negative volume.
 else if (fit == CO_SO) /* **c_over** and **s_over** regions exist */
 Invisible side of superquadric modeling data.
 else if (fit == SO_SU) /* **s_over** and **s_under** regions exist */
 Object with surface details. Do general analysis.
 else if (fit == SU_CO) /* **s_under** and **c_over** regions exist */
 Do general analysis of underestimated regions.
 else if (fit == SU_SO_CO) /* **s_under**, **s_over**, and **c_over** regions exist */
 Do general analysis of underestimated regions.
6. Done with volumetric segmentation.

Algorithm 7.2: The control flow for volumetric segmentation.

1. Global model is S_G and its children are $[S_{P1} \dots S_{Pn}]$.
2. Grow the already existing active children into **s_under** regions of S_G . If a child is done, mark it as inactive.
3. Find new children of S_G , by taking away points from S_G , in the **s_under** regions in the same coordinate direction as the already existing children.
4. Refit S_G on the new data set.
5. Go to step 4 of the control flow algorithm.

Algorithm 7.3: General Analysis when **s_under** regions exist.

important aspects of the control procedure in algorithm 7.3 are selection of data clusters to start part-models, and growing the models placed in those clusters in a controlled manner. We now discuss these issues in detail.

7.3.1 Selection of Part-Models

s_under regions corresponding to the underestimate points are the indication of concavities or part-structure in the global cluster of data. They *protrude* from the global model, suggesting the existence of a separate part, part of which is underestimated due to global averaging by the minimization procedure. Unfortunately, since the global model tends to average out residual errors, these regions can be odd-shaped and can be elongated in one direction, or surround the object completely (as in flat objects). We want to start the part-models at regions that constrain the model at least in two dimensions and allow a good approximation of the orientation of the model by extracting it from the orientation of the constituent surfaces.

For this purpose, the **s_under** regions are positioned in the object-coordinate system of the parent model in such a way that the region extremities are known in terms of the parent model. Thus, if a region extends beyond an axis completely, like the region 41 for the composite object in figure 7.3, and its size is large enough to place a superquadric model, then it is selected as a part-model. Additionally, the direction (which is negative Z-axis for the composite object) in which the residual lies is also stored, such that all the residuals in that direction are considered as part-models. This has the desirable effect of removing data from the parent model only in one direction, thereby letting the model shrink in a stable

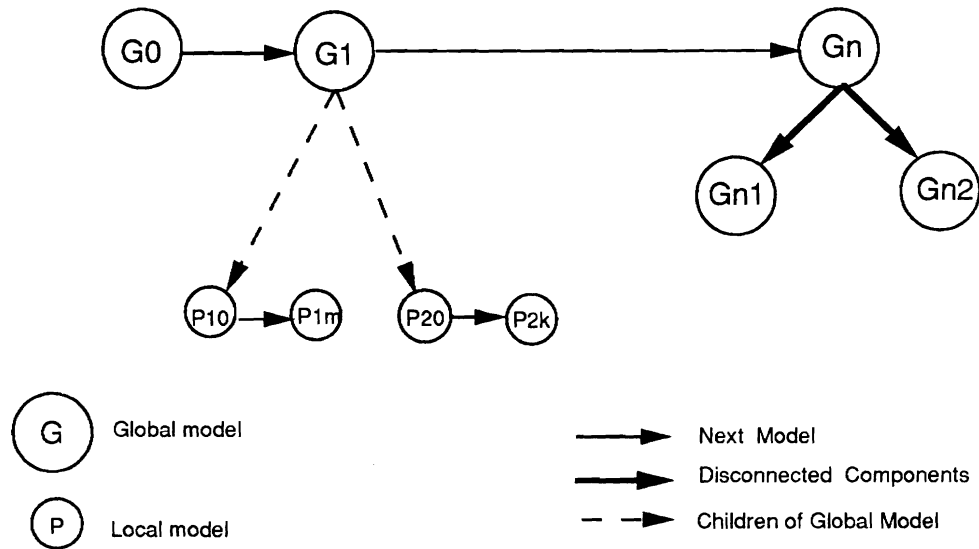


Figure 7.9: The hierarchical superquadric representation and recovery structure

manner. Later, when the residuals in a particular direction do not exist any more (due to the current global shape), other direction of shrinking is chosen following the same method.

This strategy can be easily refined to include surface information in deciding which **s_under** regions are best starting points. For the composite object, the surface 1 is completely underestimated by the global model and the curved surface is partly included, thereby preferring the portions of the **s_under** region due to the cylinder to the region formed due to the box. Other considerations include size and location of the regions, and the types of surfaces they underestimate.

7.3.2 Growing or Extrapolating the Part-Models

The part-models placed at the **s_under** regions are constrained only along the direction they have data. The directions (with respect to the part-model's coordinate system) along which the model is not constrained are the prime candidates for extrapolation of the model. Thus a model can grow only if it is not constrained in the direction of the available data, which is the case with the global-local segmentation approach. The extrapolation has the effect of changing the cross-section of the model (the a_1 and a_2 dimensions) and the length (a_3)

of the model. Either of the shape parameter for a box shaped object will not change, while both or one of them will change for a curved object. But if a curved surface is present, then the surface segmentation will result in a curved patch and the local model can be placed on the entire curved patch rather than a part of it. Thus, the real issue is in growing of part-models having planar patches. The question is, how to extrapolate a model given an initial set of data, initial orientation and the neighboring unclaimed data?

Superquadric Cross-section

Before describing the possible extrapolation strategies, it is instructive to study the XY cross-section of superquadrics. Given a superquadric model described by equation 5.2, the cross-section at $z = d$ is given by :

$$\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}} = \left(1 - \left(\frac{d}{a_3}\right)^{\frac{2}{\epsilon_1}}\right)^{\frac{\epsilon_1}{\epsilon_2}} \quad (7.1)$$

The cross-section is a super-ellipse (left side of the equation), starting at $z = 0$, and disappearing at $z = a_3$. ϵ_2 determines the shape of the super-ellipse. Note that the cross-section changes non-linearly along the Z-axis, and it always terminates. Hence, the typical generalized-cylinder based methods assuming linear non-terminating cross-section are not useful here. For $\epsilon_1 = 1$, the variation in the area of the cross-section is maximum as z progresses from 0 to a_3 . At the other extreme, for $\epsilon_1 = 0.1$, the cross-section remains constant (~ 1) for almost the entire length of the model, dropping to zero for values very close to $z = a_3$. Thus, the objects like cylinders and boxes, for which the cross-section does not vanish at any point, are modeled with $\epsilon_1 = 0.1$; while the curved objects have values of ϵ_1 close to 1.

In the context of model extrapolation, the growth in the length of the model (along Z-axis), is a function of ϵ_1 and a_3 , while changing the cross-section at given value of Z depends on ϵ_2, a_1 , and a_2 .

Growing the part-models

One approach is to check the neighborhood connectivity, and slowly increment the model domain by constantly evaluating the refined model. In general, all the parameters (including translation and orientation) are allowed to change, though at times it may be necessary to fix some of the parameters. If changes result in an increased error or in a model that is not acceptable, then the previous model is considered final and a new model is started at the

underestimated points. However, this approach has a problem that it extends the model equally in all directions, which may not work in general.

A more controlled method would extrapolate the model only in one dimension and ignore data altogether in other directions. By direction, we mean the six axes directions of the object centered coordinate system. We can divide the problem into extrapolating along the length of the model (Z-axis) or the cross-section of the model. For box shaped objects ($\varepsilon_2 < 0.5$), the growth along a particular direction is possible by simply observing the coordinates of the potential data point and comparing it with the dimensions of the part-model. This has the effect of extending the model along a desired dimension only and ignoring the rest. The model is thus grown along only one dimension at a time and constantly evaluated. Notice that the model grows only if there are unclaimed data points along that dimension *and* including those points does not conflict with the points already accepted by model. We will show results of this approach in the next chapter. The method can handle tapering deformation but not global bending, due to the ambiguities inherent in bent models.

Following the above considerations, the control structure recursively shrinks and grows global and part models respectively, and terminates individual global models when their fit is acceptable. The global models shrink till they converge on a part or disappear after the data is completely accounted for. Part-models grow for as long as they satisfy the acceptance criteria. In the next chapter we present detailed results of applying the general control structure on complex objects, and discuss the salient aspects of our segmentation schema.

7.4 Chapter Summary

We now have a general control structure that identifies various possibilities for segmentation and guides the segmentation procedure accordingly. We discussed the issues relating to volumetric segmentation of $2\frac{1}{2}$ -D data, devised efficient solutions for overcoming the problem of model extrapolation, which is difficult to perform for an object-centered model. The role of surface segmentation was discussed, and the information derived from biquadrics was fully incorporated in the control structure.

CHAPTER 8

Experimental Results

We now have a systematic procedure for the surface and volumetric segmentation of dense range data. To test the SUPERSEG system thoroughly and to highlight its strengths and weaknesses, we designed a set of experiments comprising of objects of varying complexities. We will begin with discussing the scope of the object classes that can be handled by the system and then present the results for objects representing each one of the categories.

8.1 Test Cases: Complexity and Scope of the Paradigm

Objects in the real world are of varying complexities and present most demanding requirements on a general segmentation procedure which has no knowledge of the object domain or domain properties. The complete scenario can be divided into different classes as shown in figure 8.1. Notice that the division reflects the control flow of the volumetric segmentation system. Thus, the control flow maps favorably to the complexity of the input and is able to systematically analyze it.

The simplest case is that of isolated objects on the given background depth, with each object corresponding to a superquadric model. In general, the objects can be of varying complexity from single part to multiple part objects. Single part objects can again either be superquadric-modelable or not. Same is true at part-level for the complex objects. The part arrangement and levels of parts add to the complexity of the objects. Our objective in this chapter is to test the robustness of the paradigm by running an implementation of the SUPERSEG system on range images scanned with the GRASP lab structured lighting range scanner (spatial resolution = 1mm and depth resolution = 1.5mm).

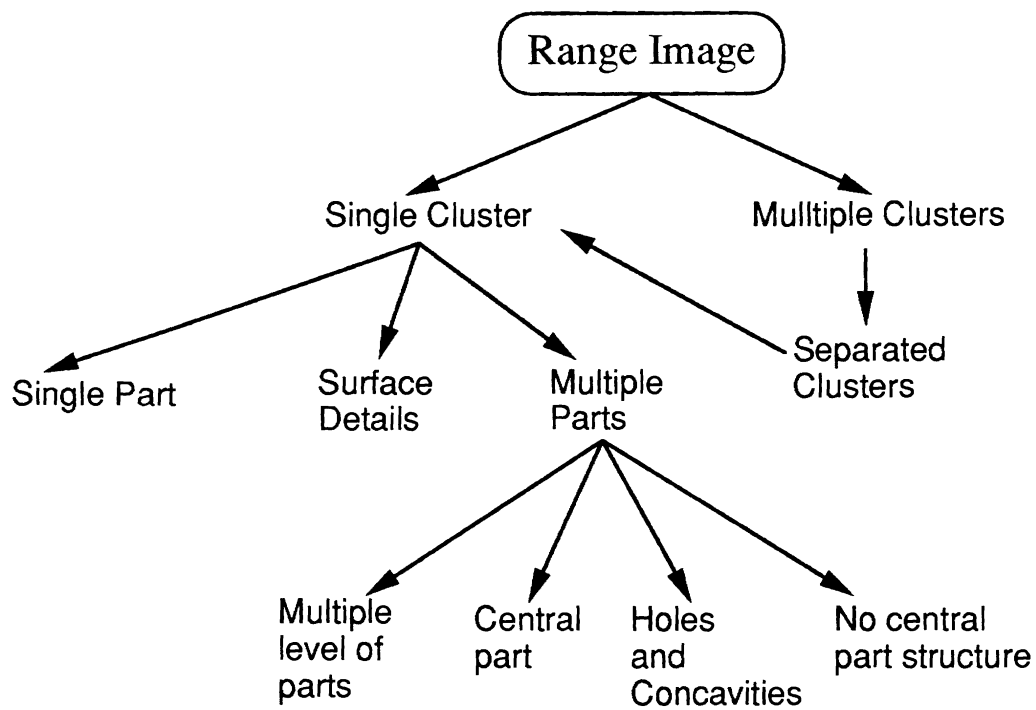


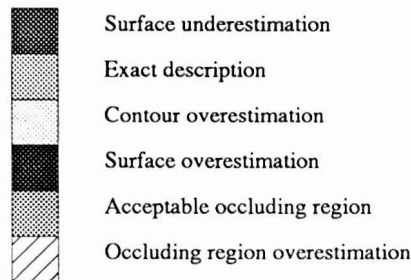
Figure 8.1: **The hierarchy of test cases:** The complexity and scope of the paradigm are representative of the control flow of the system.

8.2 Examples

We first present the case of L-shaped objects with varying relative sizes of parts to demonstrate the robustness of the system. The two examples of bent objects demonstrate that the part-description capability of the system is tied to the acceptance criteria which directly evaluates all the recovered models. The coffee-mug is an object with a hole and a cavity, along with concave and convex surfaces. A negative volume description is generated for the body of the mug. The NIST object is a machined object with holes, and planar and curved surfaces. The wrench is an example of an object with multiple parts, and some of its parts are in a degenerate viewing position. The scene3 has multiple objects (clusters) in the scene and shows the segmentation using independent global models.

Representation of Results: The input data is shown as a gray-level image with gray values corresponding to the range values along the viewing (Z) direction. Thus, the points closer to the camera appear brighter. Since the objects are scanned along Z , this representation shows all the data that is available to the segmentation modules. In addition, a 3-dimensional perspective plot is also generated for every object. The OU regions (the residual maps) are shown for the intermediate and final descriptions, where appropriate. The superquadric models are displayed in a coordinate system different from the image coordinate system to get a better viewing angle. So their orientation does not directly correspond with the orientation of the objects in the image coordinate system. The models are shown on the entire data set to highlight the segmentation.

The POSTSCRIPT display of surface segmentation and the residuals show a pixel-wide gap between adjacent non-overlapping regions. The gap merely illustrates that the two regions are separate (no overlapping domain) and that they actually extend equally to cover the gap completely. However, no gap is shown between the adjacent seed regions. The following legend is followed throughout this chapter for the display of the residuals:



8.2.1 L shaped Objects

We show the results for a continuum of L-shaped objects of varying sizes and scale. The surface segmentation into planar patches is not final for these examples, thereby requiring the control module to reject the convex-combination hypothesis, and perform controlled growing of the part models to achieve the final segmentation.

L1-shape: Although the surfaces form convex edges, the surface segmentation into three parts cannot be combined in convex sense due to the shape of surface 3 (figure 8.2), which requires further segmentation. The residuals of the global model contain significant regions of `c_over`, `exact`, `s_under`, and `s_over` regions. The segmentation starts with the biggest `s_under` region lying completely outside the model along the Z-axis.

The two iterations leading to an acceptable segmentation are shown in figure 8.3. The `s_under` region (# 4 in figure 8.2) of the residual of the global model is selected for placing the part-model (also called the local model), as shown in figure 8.3(c). The global model's second iteration is shown in (a), which allows the local model to grow into the points in the new `s_under` (# 3 in (a)) region. The grown model is shown in (d). The residuals indicate that the local model describes data very well and has contour overestimation due to the missing data and an acceptable occluding region due to the viewing direction. The global models terminates at third iteration to give the model shown in (b). This has the effect of terminating the local model as well.

L2-shape: Similar processing for the L2-shape gives two volumetric parts (figure 8.4), with the global model converging to the bigger part. Notice that the surface segmentation breaks one of the planar patches into smaller regions, which could be distracting to a control structure completely dependent on the surface information.

L3-shape: The part-model starts growing at the smaller part, but due to the noise on the face consisting of surfaces 4 and 5, the part-model stops growing after first iteration. A second part-model is initiated to model the `s_under` region by the recursive structure of the control module.

L4-shape: Even with an oversized main part of the L-shape, a part-model is recovered for the smaller part, while the global model converges to the bigger part.

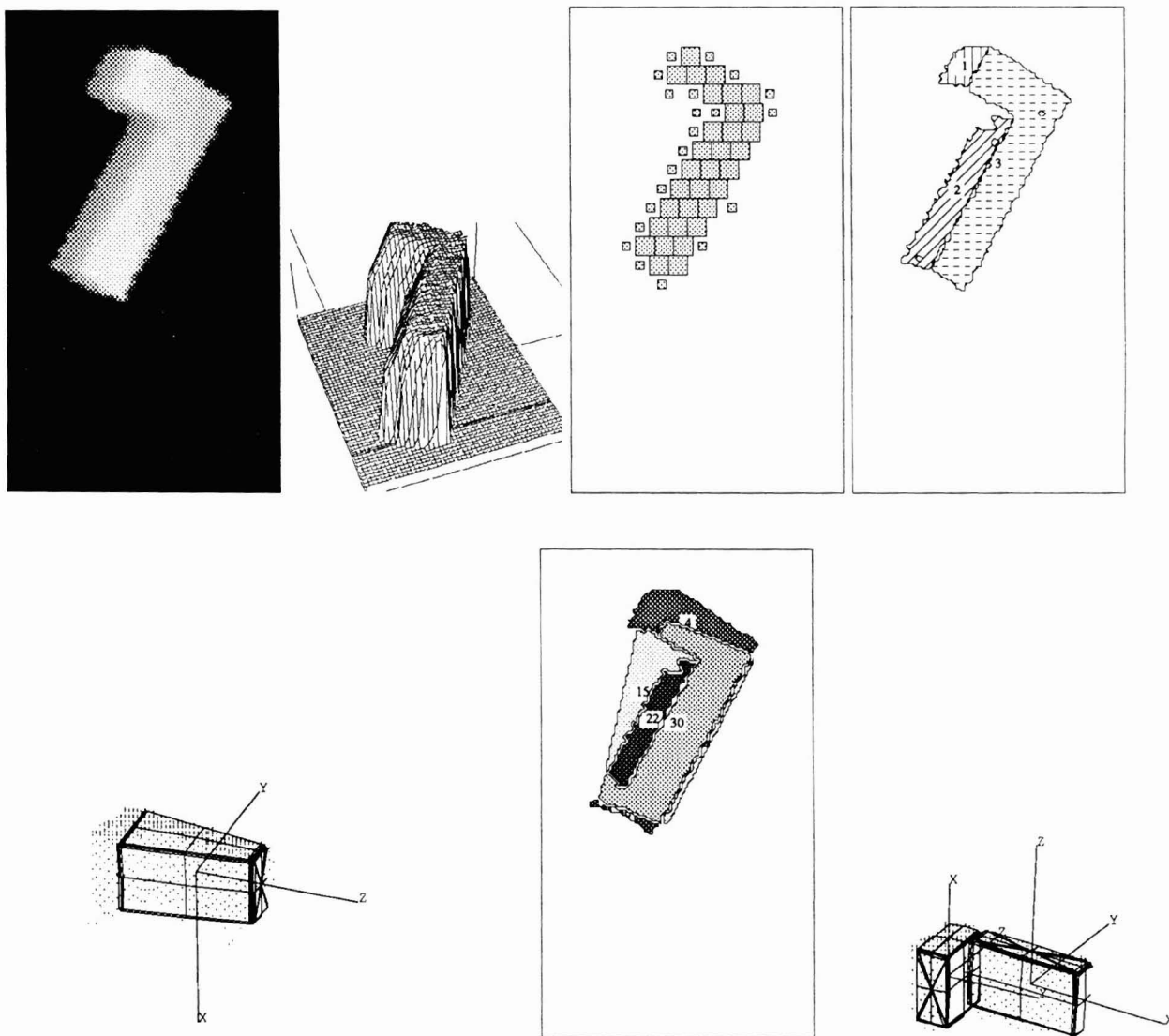


Figure 8.2: **The L1-shape:**Top: Range image, 3-D plot, bi-quadric seeds, and surface segmentation; Bottom: the global model, its Z-residuals, and volumetric segmentation.

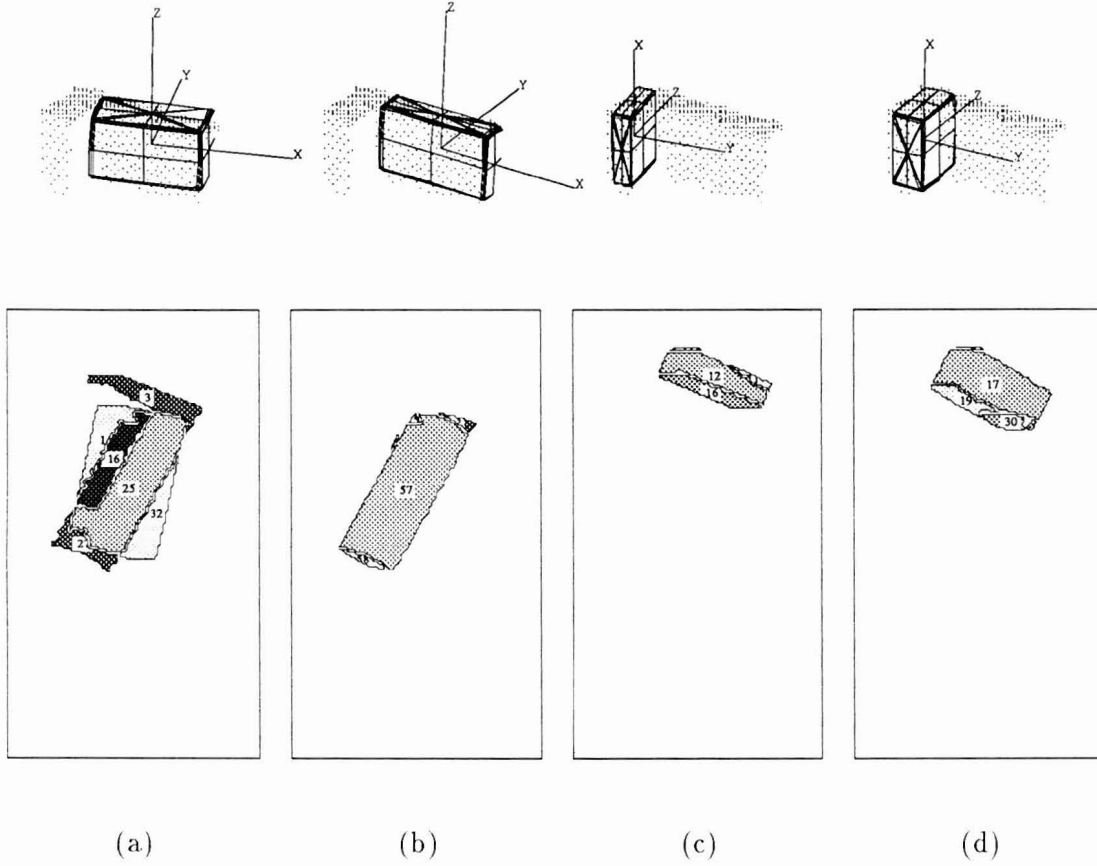


Figure 8.3: **The L1-shape: Volumetric segmentation:**(a) & (b) The first and final iterations of the global model. (c) The part (local) model starts at the **s_under** region (# 4 of global residual, figure 8.2). (d) During the second iteration, the part model continues to grow into the **s_under** region (# 3 in (a)) to form the final part model. The bottom row shows the residuals of the corresponding superquadric models. The residuals of the final part-model consist of **exact** (# 17), **c_over** (# 19), and **occ_ok** (# 30) regions.

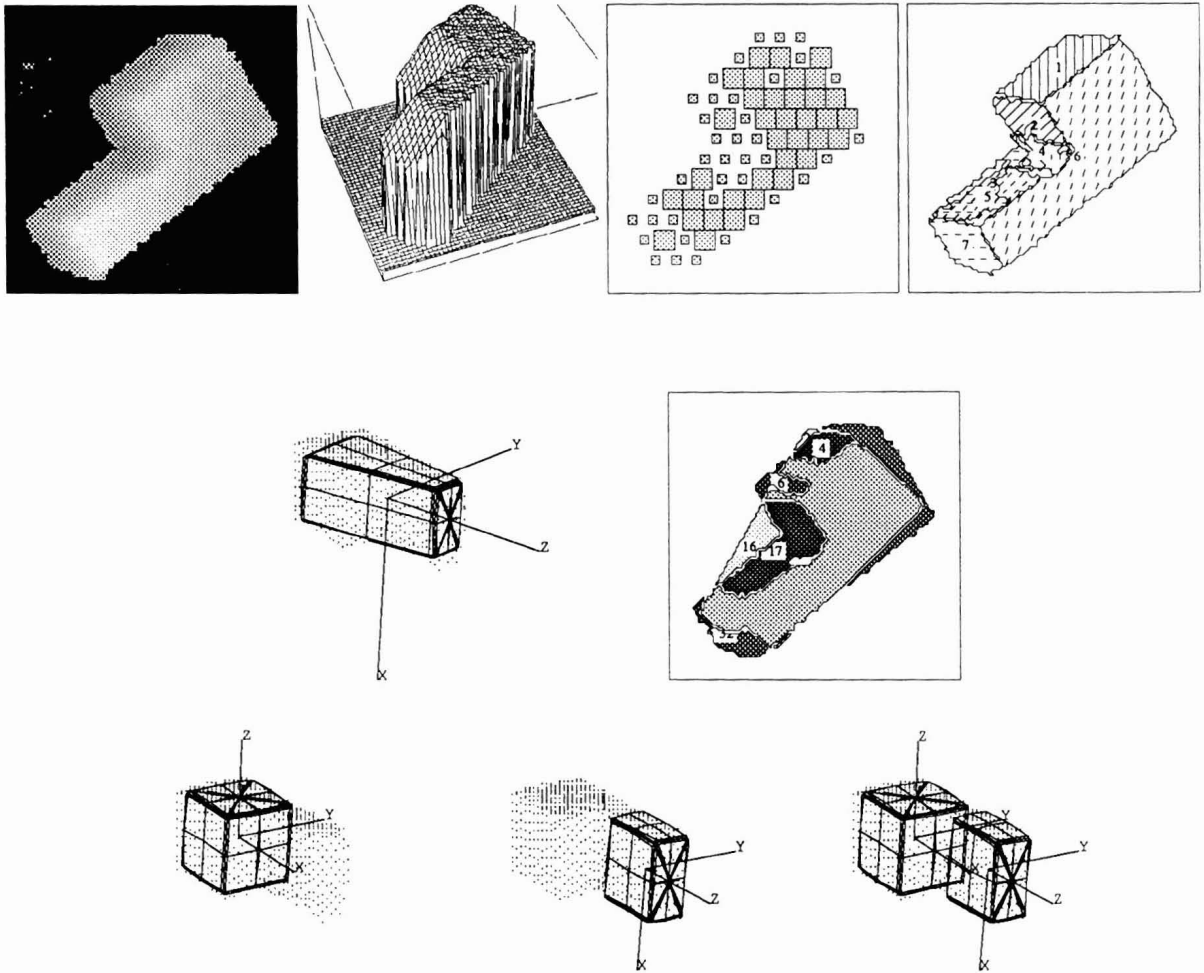


Figure 8.4: **The L2-shape:** Top: Range Image, 3-D plot, bi-quadric seeds, bi-quadric segmentation. Center: Global fit, Z-residuals of the global fit, and Bottom: Refined global model, grown local model, and the final volumetric description.

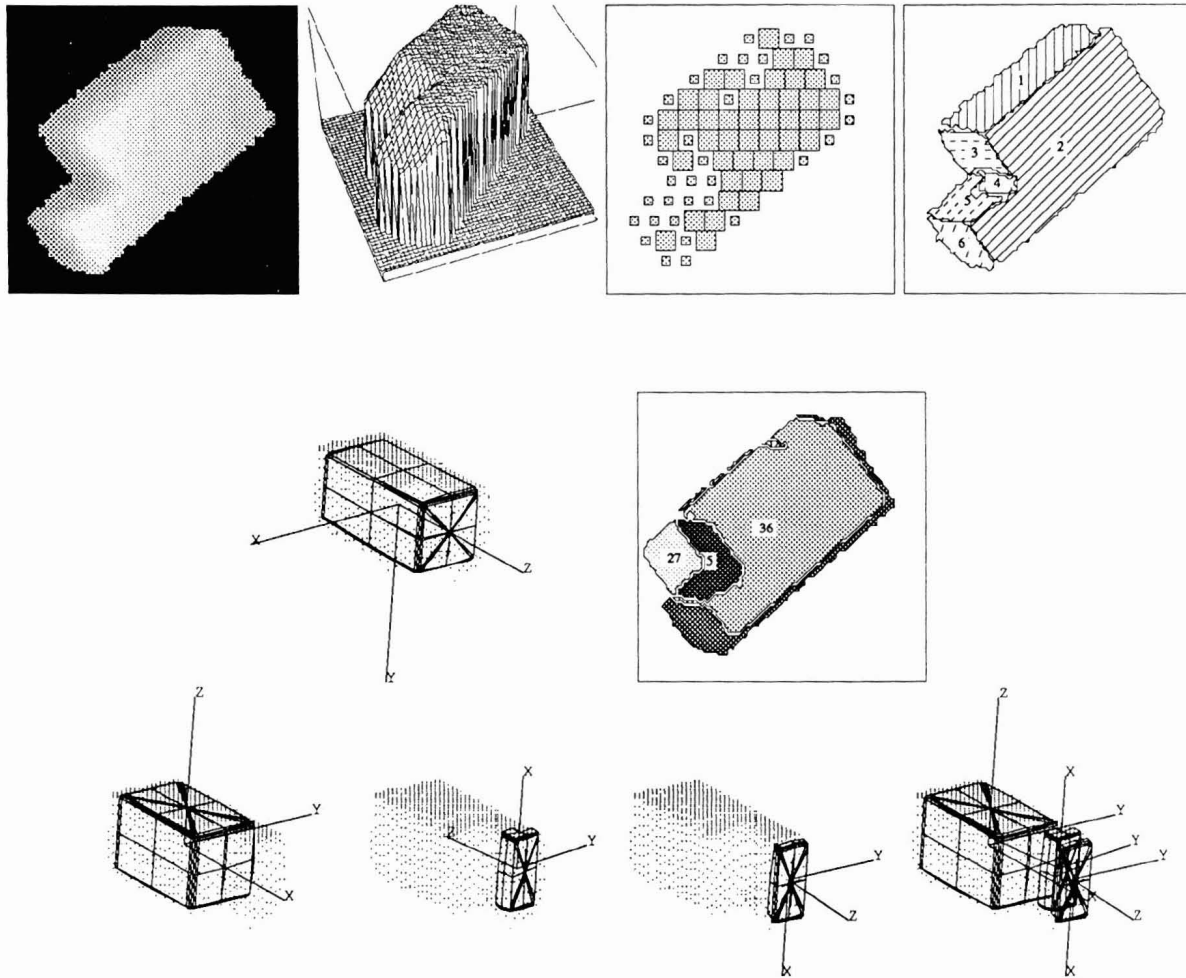


Figure 8.5: **The L3-shape:** Top: Range Image, 3-D plot, bi-quadric seeds, bi-quadric segmentation. Center: Global fit, Z-residuals of the global fit, and Bottom: Refined global model, the two part models, and the final volumetric description.

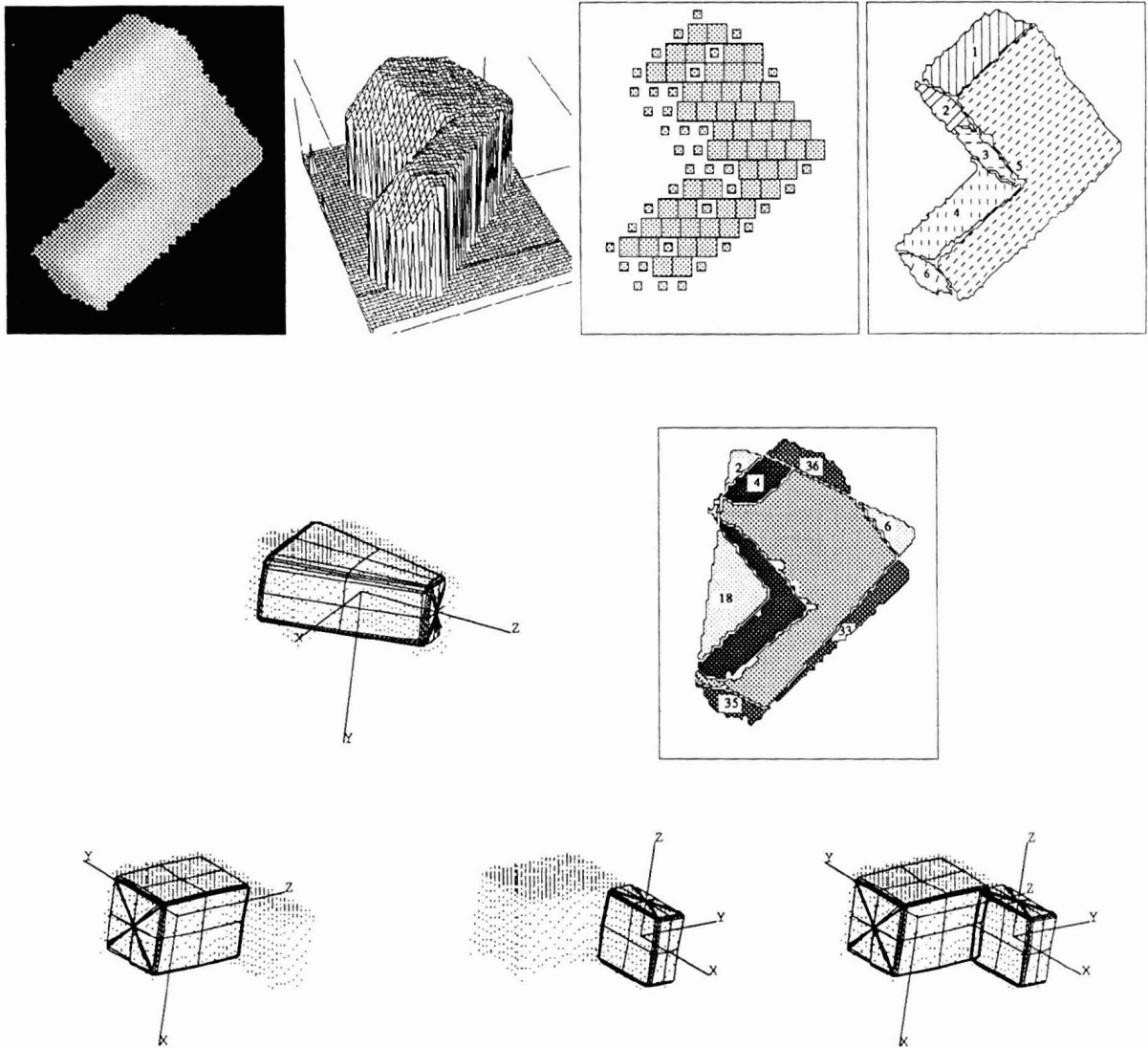


Figure 8.6: **The L4-shape:** Top: Range Image, 3-D plot, bi-quadric seeds, bi-quadric segmentation. Center: Global fit, Z-residuals of the global fit, and Bottom: Refined global model, grown local model, and volumetric description.

8.2.2 Bent Objects

Results are presented for two bent objects, with Bent2 object bending more than Bent1.

Bent1 object: The bi-quadric segmentation gives a piecewise curved description (surfaces 2 and 4) of the curved surface (figure 8.7, top right). Surfaces 1 and 3 are planar. Clearly, a convex combination of patches is not sufficient (because surface 3 needs to be segmented further in the volumetric sense), and the general global-to-local strategy is required. The presence of significant amount of residuals indicates the need for segmentation.

Beginning with a global model (shown in figure 8.7), the **s_under** region (# 3) is selected as the domain for the part-model, which grows (shown in figure 8.8), while the global model shrinks in 4 iterations to an acceptable model. The acceptance criterion used was that at least 80% of the points should be correctly described in the Z-residual map, with the tolerance along the Z-direction being 3 pixels. Part model is acceptable during all iterations, while the global model strives for an acceptable description. After four iterations of the global model shrinking and part model growing, the procedure terminates when the global model becomes acceptable.

Figure 8.9 displays the variation of goodness-of-fit values and the residuals of the global and part models during the segmentation procedure. The qualitative residuals are given as percentage of data points relative to the given data. At the end of the fourth iteration, the global model has 80% of points falling in **exact** category, which terminates the further shrinking of the global model. During the iterations, the percentage of data points contributing to residuals decreases, with the lowest values after the fourth iteration. The GOF (the fit error) for the part model increases, but remains at an acceptable level. The residuals of the part model remain within the acceptance criteria throughout the iterative process (bottom right in figure 8.9), with more than 90% of points lying on the model at the end of the final iteration.

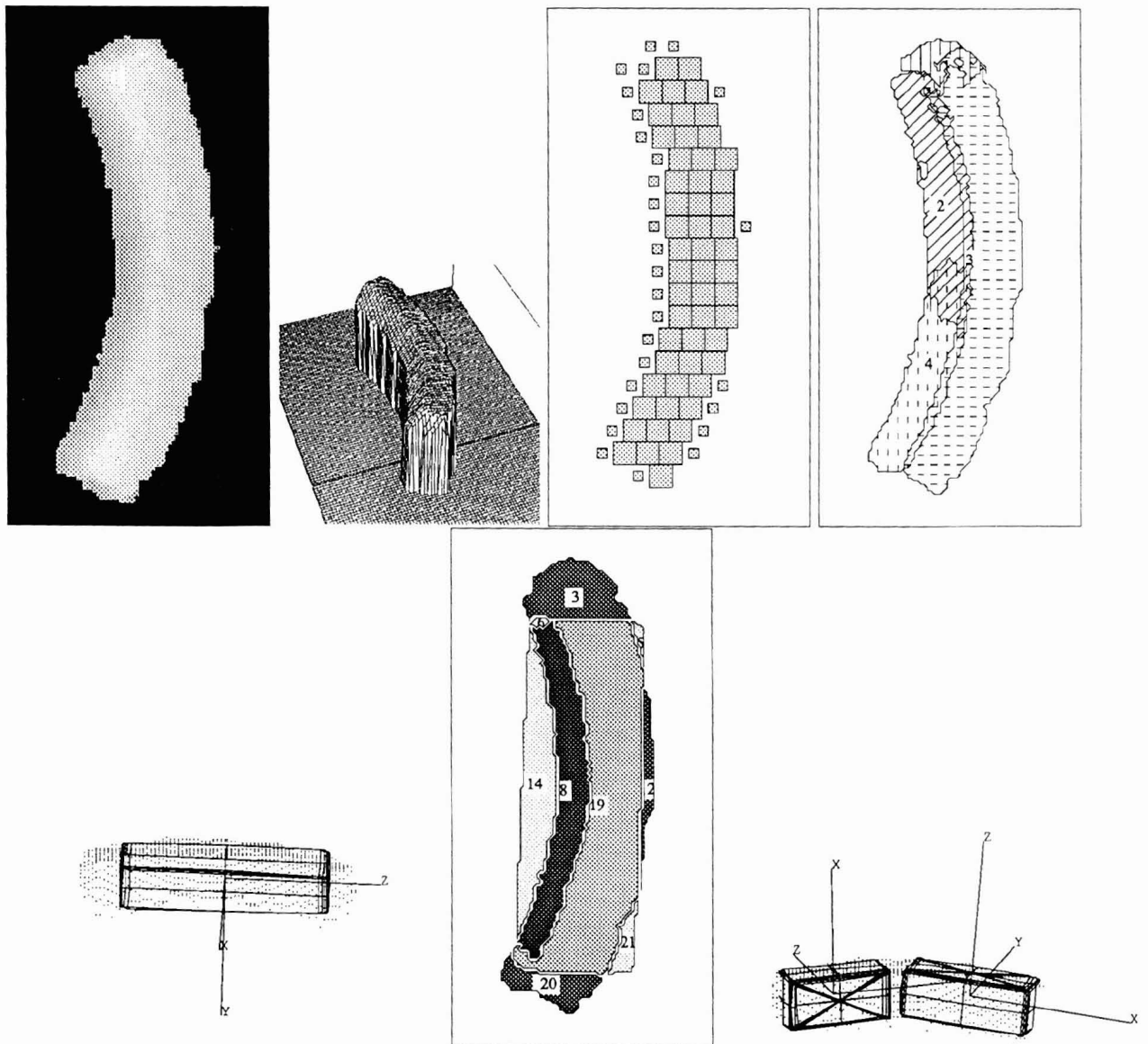


Figure 8.7: **The Bent1 shape:** Top: Range image, its 3-D plot, bi-quadric seed regions, and the bi-quadric segmentation. Bottom: The global model, its Z-residuals and the final volumetric description.

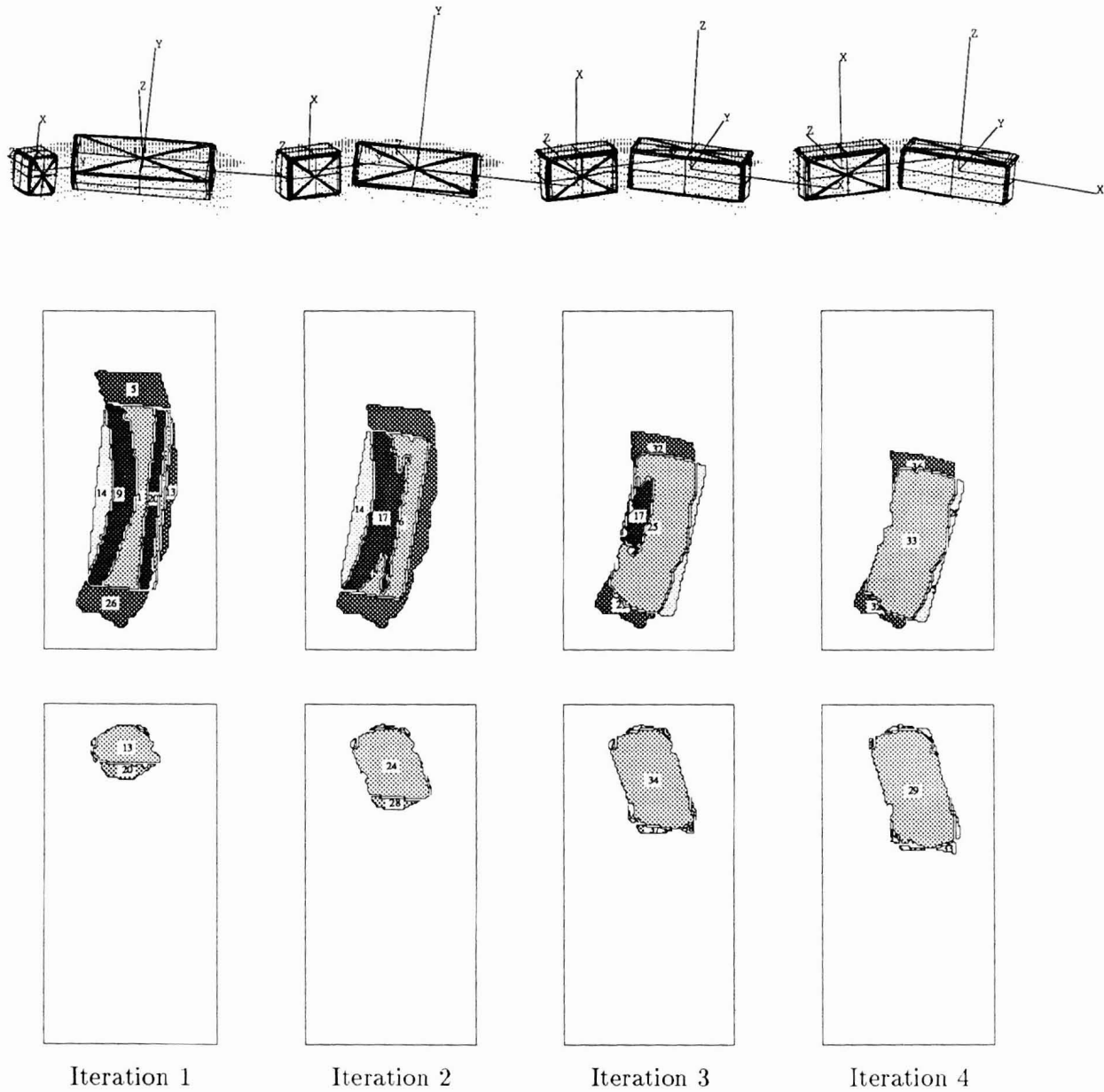


Figure 8.8: **The Bent1 shape: Volumetric segmentation:** The four iterations taken by the system to satisfy the acceptance criteria. Each column shows the volumetric description, residuals of the global model, and the residuals of the part model. The part model grows by accepting data in the **s_under** regions only beyond the extent of the global model along one of the coordinate direction, and into the complete region.

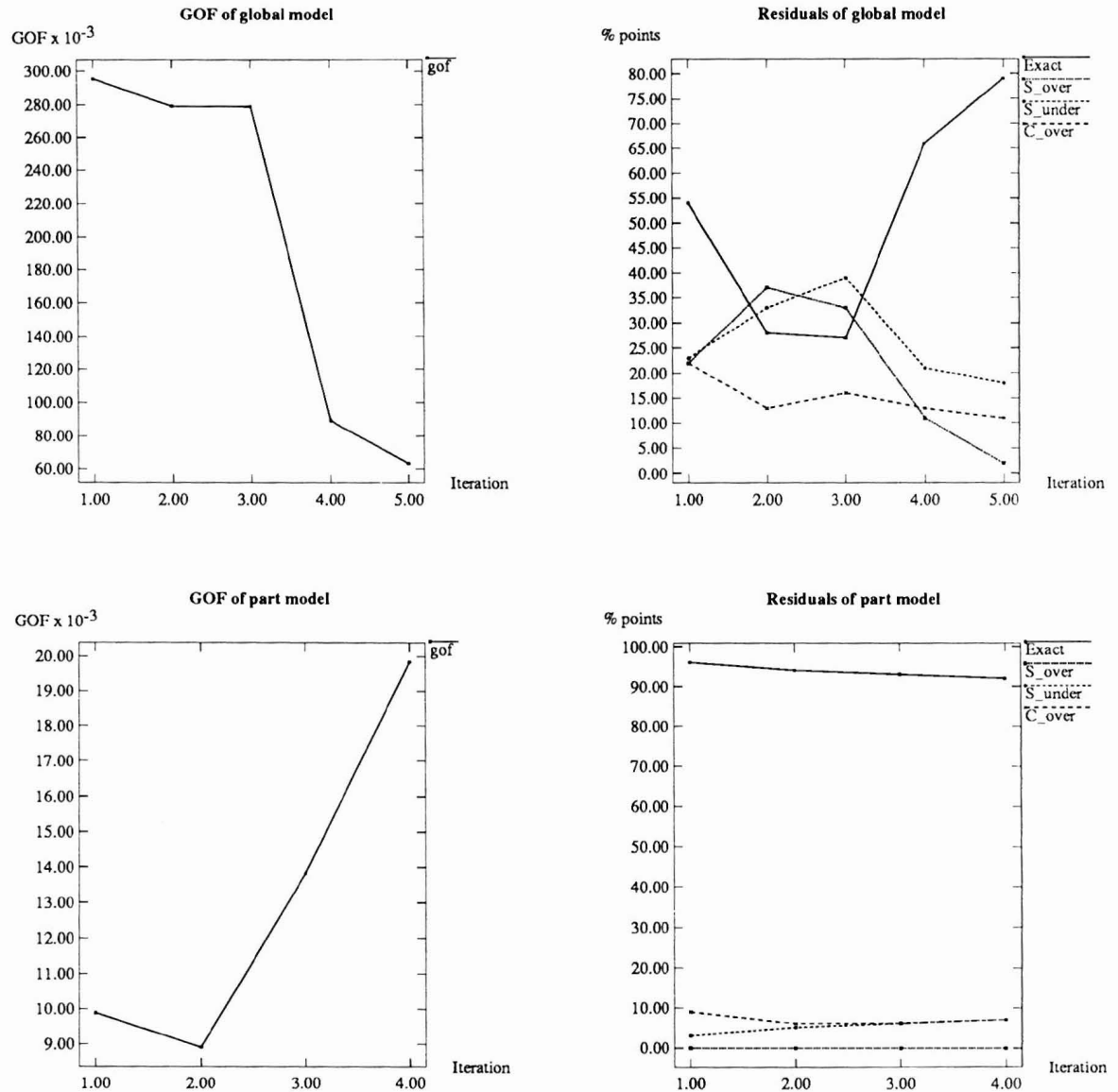


Figure 8.9: **The Bent1 shape: Residuals:** Top: For the 4 iterations (2 to 5) of the global model. The goodness-of-fit values (left), and the residuals of the global fit as percentage of data points. Bottom: For the 4 iterations of the part-model. The goodness-of-fit values (left), and the residuals of the model.

Bent2 object: The Bent2 object bends more than the Bent1 object. The object and the final results are shown in (figure 8.10). Bi-quadric segmentation describes the curved surface as a piecewise combination of three curved patches (2, 4, and 5). As in the case of Bent1 object, the convex combination hypothesis is rejected and the residual-guided global-to-local analysis of the scene is performed by the control module. The part model, placed on `s_under` region 12 grows into a model which reaches the limit of the acceptance criteria and can accept no more points. Since the global model is not final at that point, a new part-model is started which terminates when the global model satisfies the acceptance criteria.

These examples demonstrate the capability of the control structure to generate piecewise descriptions of the objects which do not have a model in the superquadric vocabulary. Since the procedure is guided by acceptance criteria (which in turn measures the deviation between the model and data in terms of tolerances), it is possible to obtain a scale-space of volumetric descriptions by simply relaxing the tolerances or by making them stricter. Thus, a stricter Z-tolerance will result in more parts for the Bent2 object, while a relaxed criteria will accept a description with fewer parts.

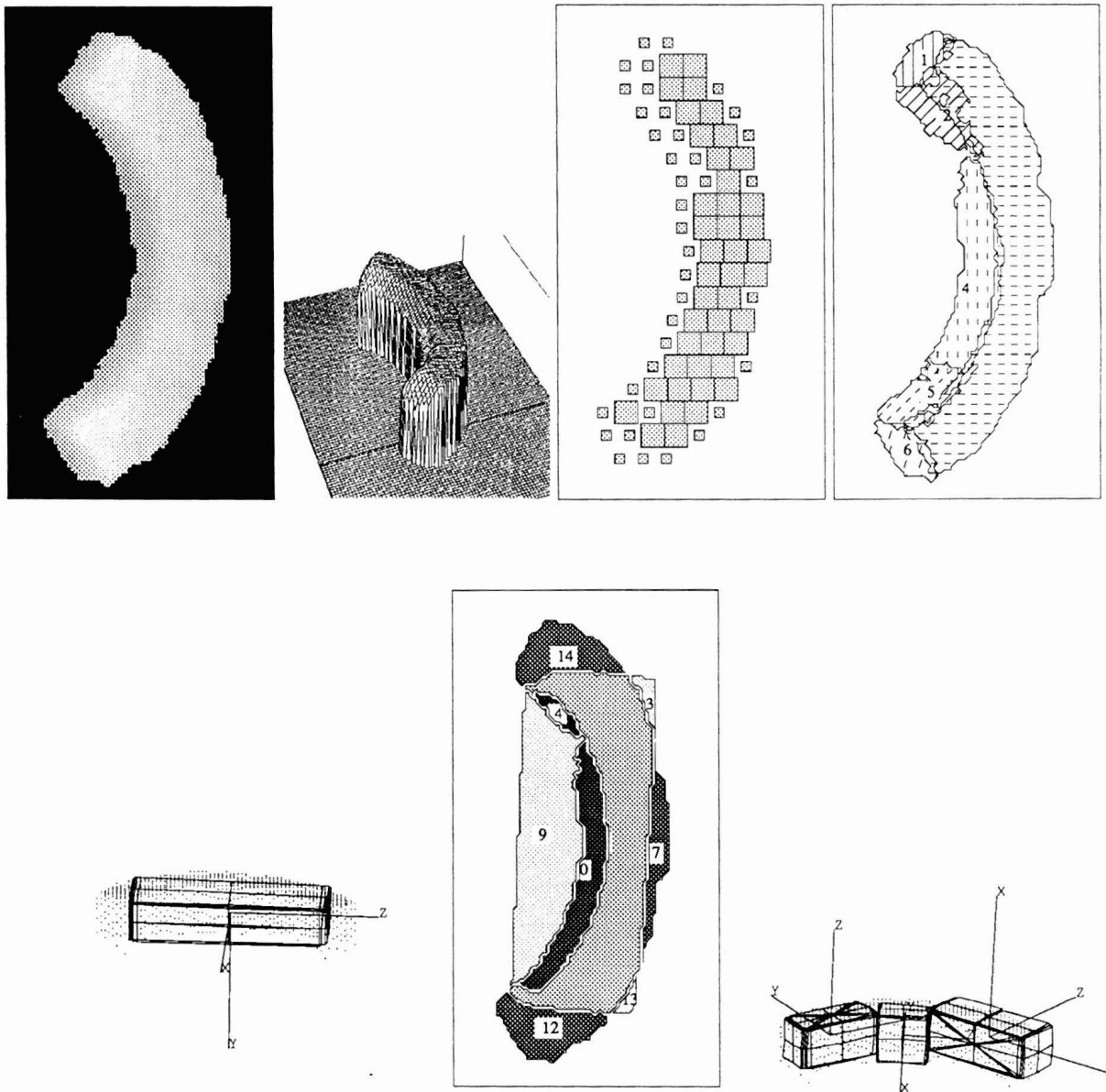


Figure 8.10: **The Bent2 shape:** Top: Range image, its 3-D plot, bi-quadric seed regions, and bi-quadric segmentation. Bottom: The global model, its Z-residuals and the final volumetric description consisting of three models

8.2.3 The Coffee Mug

The coffee mug is an example of an object with a cavity and a hole, and both convex and concave surfaces. The biquadric segmentation (figure 8.11) gives one surface each for the convex and concave patches along with the orientation information which is used to orient the local models. The convex part of the main body is separated as a convex elliptic paraboloid.

The surface description isolates the concave piece (surface 1) from the remainder of the data due to the presence of the step edge between them. The connected-component analysis of the Z-residuals therefore starts two global models at each component (figure 8.12). The concave part is modeled by a cylinder which is accepted as a negative cylinder. The main body of the cup, along with the handle constitute the remaining data, which is modeled by a box. The box is further decomposed into the convex part, due to the presence of concave ramp edges between the handle and the convex part. The cylindrical description of the convex part is accepted, so are the surface level superquadrics for the handle decomposing it into 4 parts. The Z-residuals of the concave part of the cup consist of the `s_over` (# 27) and `occ_ok` (#31) regions. This information coupled with the bi-quadric analysis establishes the fact that a negative volume description is needed.

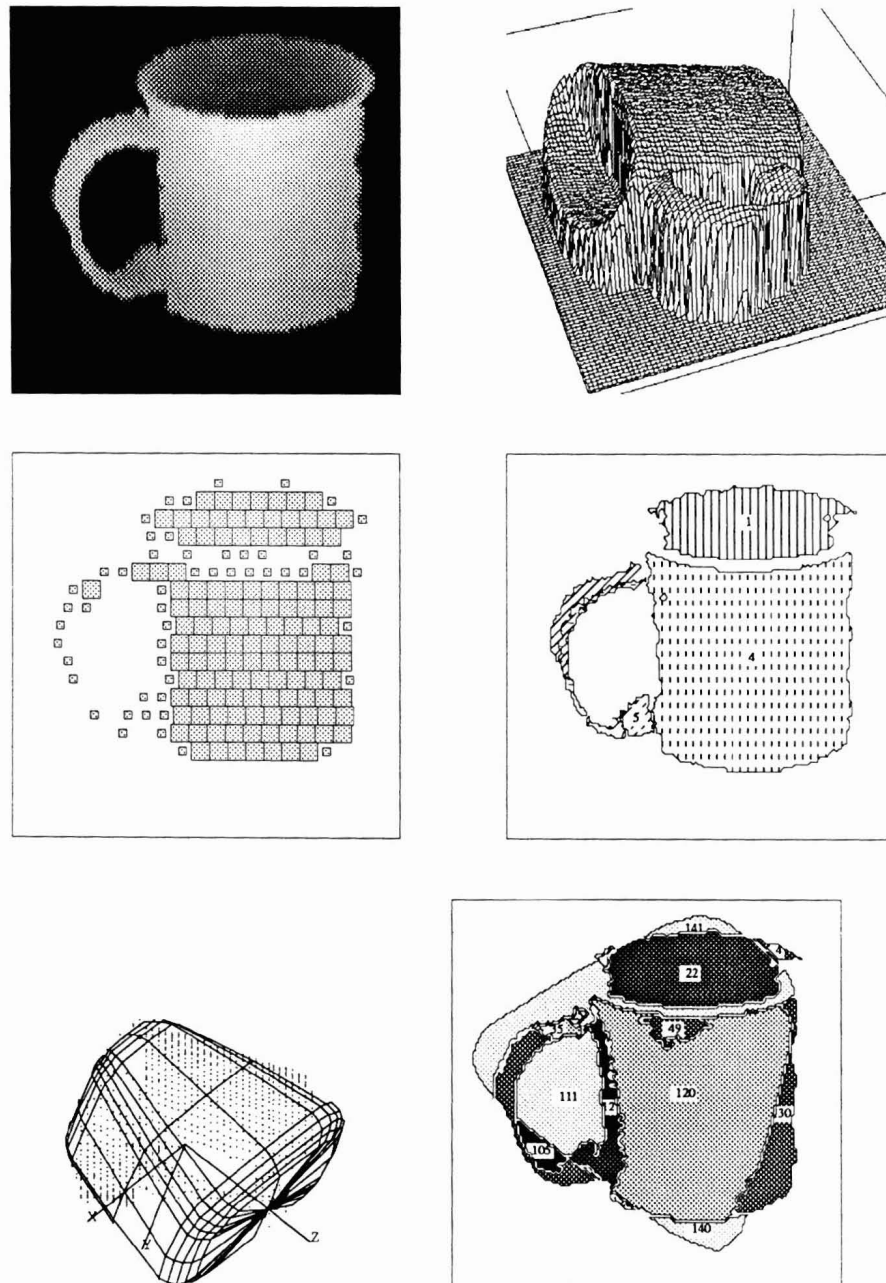


Figure 8.11: **The Coffee Mug**: Top: Range image and its 3-D plot; Center: biquadric seed regions, and surface segmentation; Bottom: The global model and its Z-residual map.

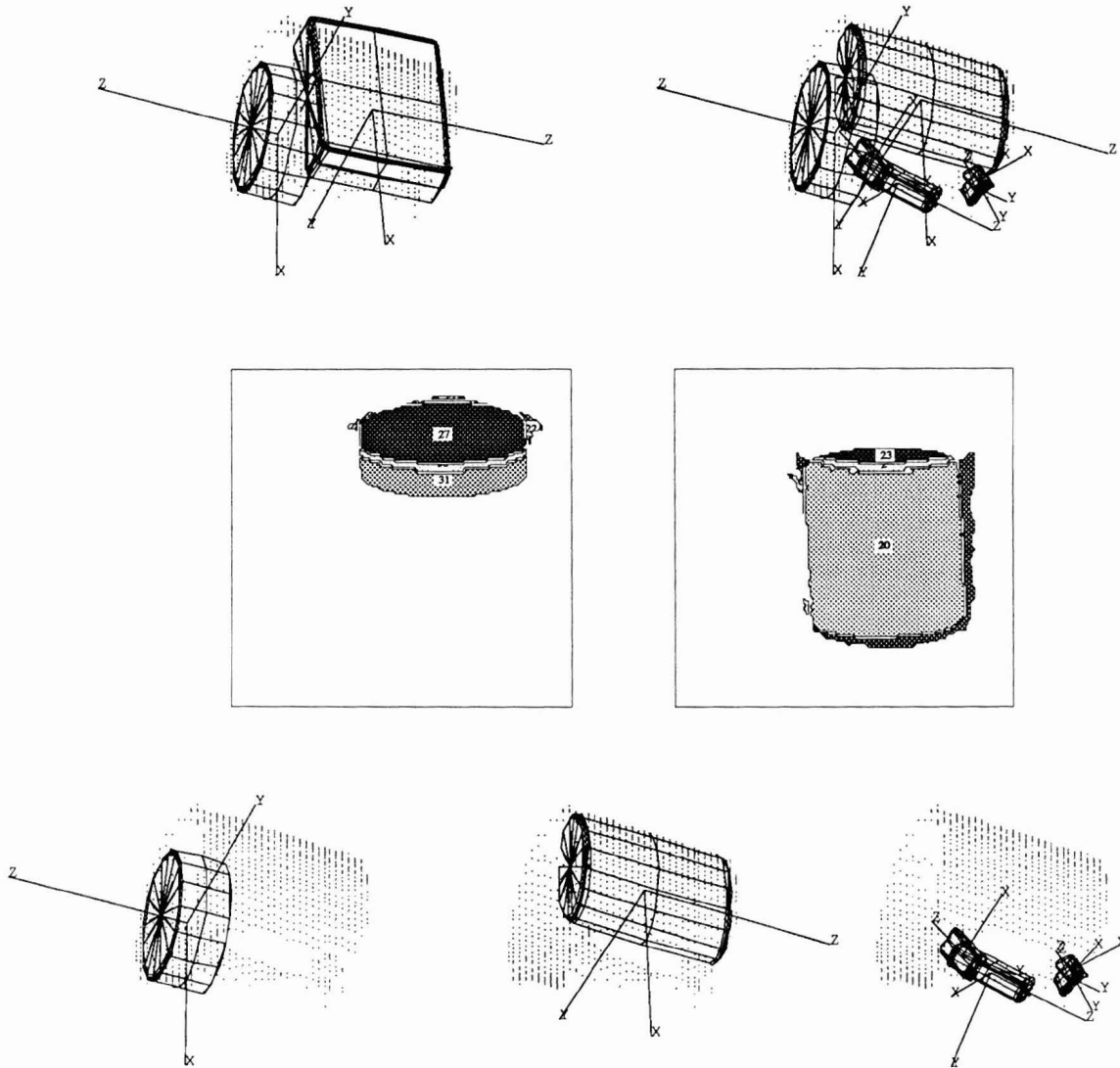


Figure 8.12: **The coffee mug:** Top: Description after the first component separation (left), and the final description (right). Center: The Z-residual regions of the concave and convex parts of the cup. Bottom: final descriptions of the main body of the cup and the handle.

8.2.4 Scene3: Multiple Clusters

Scene3 was formed by merging three views of the scene, and is noisier than other images due to the correspondence problem in our experimental setup. The scene consists of three clusters that can be trivially segmented from the background.

The cluster formed by the box and the cylinder is separated into two bi-quadric patches (9 and 12 in figure 8.14). The two patches have a step edge between them, and hence are spatially separated. Since the Z-residual analysis is performed on the data with surface support only, the patches 9 and 12 appear as separate clusters, resulting in a total of 4 independent data clusters as evident in the residual adjacency graph shown in figure 8.13. The left part of the phone hand-set is segmented into four patches (8, 10, 11, and 13) instead of two, because of the correspondence problem between two views forming to that part. The right part is correctly segmented into two surface patches (2 and 3). The handle of the phone is segmented into a planar patch (4) and an overlapping second-order patch (5).

Connected component analysis of the Z-residual map results in four clusters, which define individual connected components described by individual global models in the next iteration as shown in figure 8.15 (a). The description for the two boxes and the cylindrical object is final, while the phone requires further segmentation. The axis of the cylindrical object is correctly oriented using the bi-quadric information, even though the diameter of the cylinder is greater than its height. The surface level segmentation of the curved surfaces on the phone (figure 8.14) forms convex clusters describing the two curved parts and the handle of the phone, shown in figure 8.15 (b).

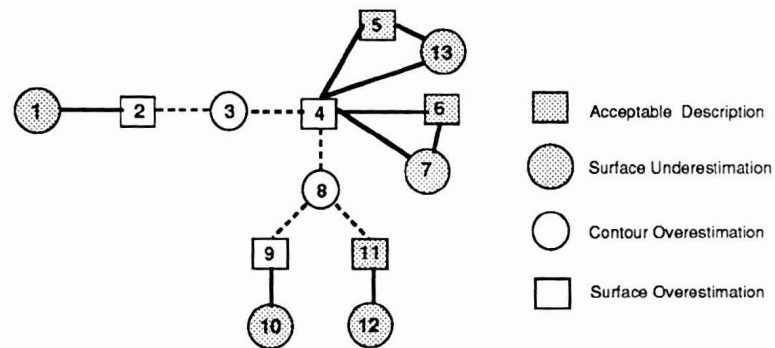


Figure 8.13: **Residual adjacency graph (RAG) for Scene 3:** The connected component analysis gives four independent clusters for further processing.

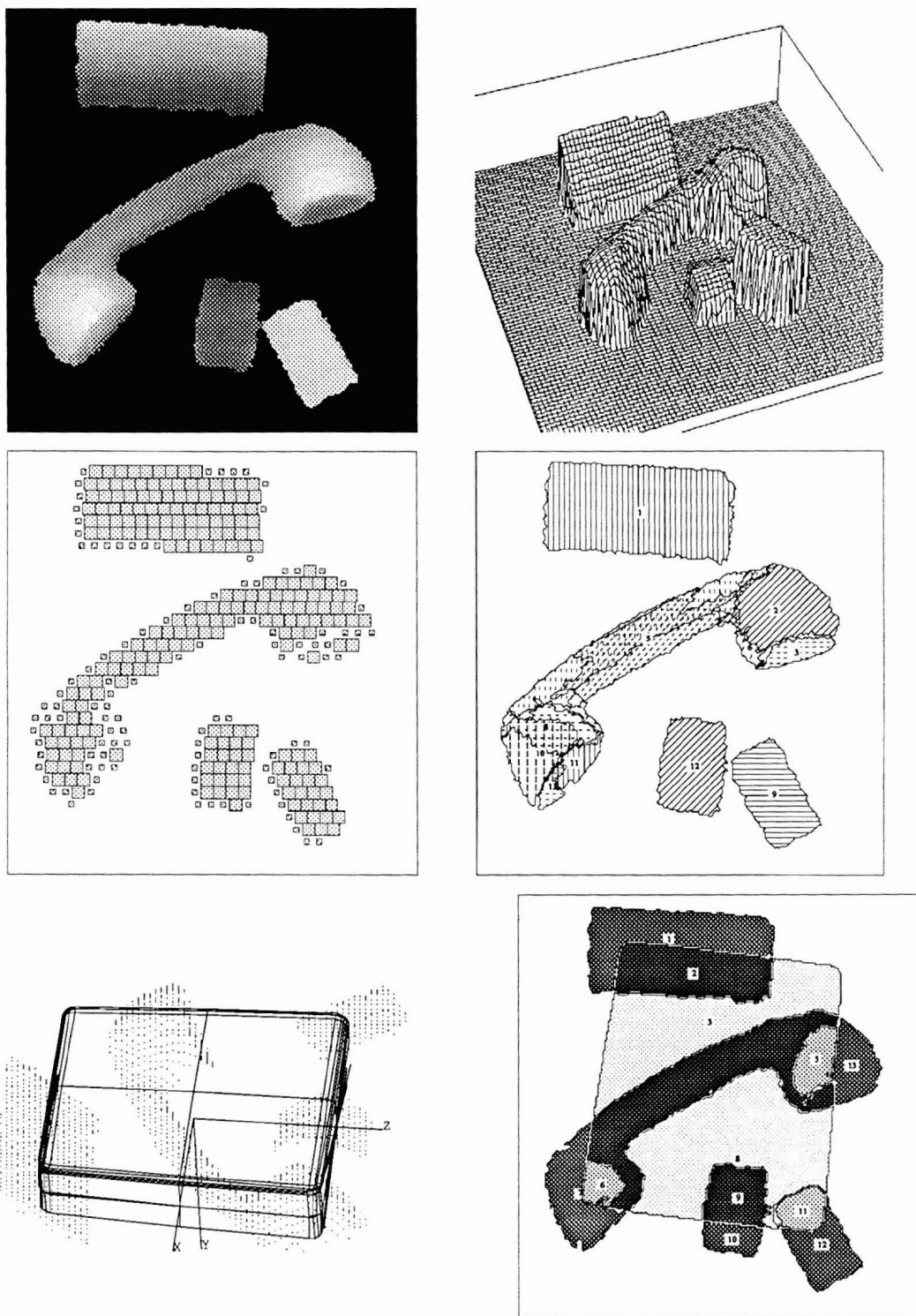
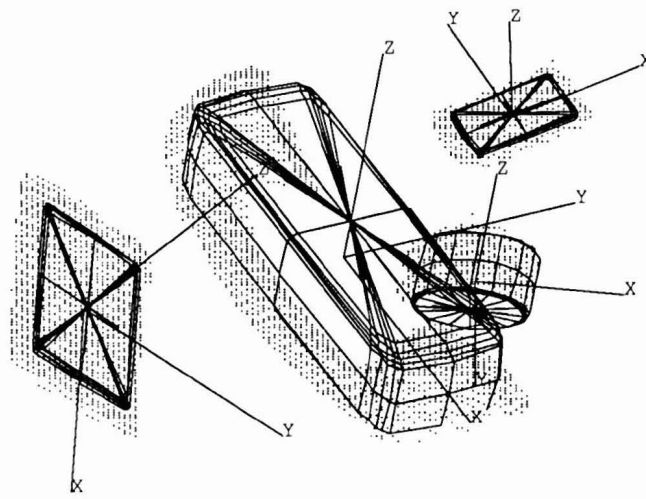
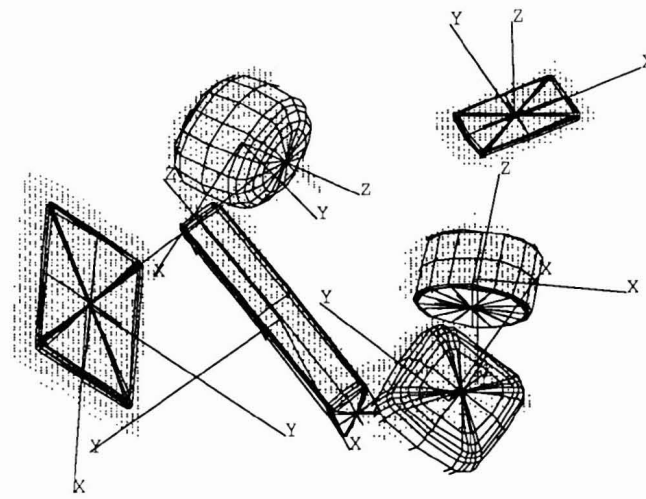


Figure 8.14: **Scene 3: Multiple clusters:** Top: Range image, and its 3-D plot; Center: bi-quadric seed regions, and bi-quadric segmentation; Bottom: The global model and its residuals.



(a)



(b)

Figure 8.15: **Scene 3: Volumetric segmentation:** (a) Models for individual clusters (first iteration). All the models except the phone are final at this stage. (b) Final description of the scene (second iteration).

8.2.5 The Wrench

The wrench shown in figure 8.16 has a definite part-structure, although the surface segmentation is inadequate to capture it. In fact, the two surfaces also have considerable overlap (figure 8.16) due to the first-order surface smoothly merging into the curved handle of the wrench. In addition, the surface constituting the head of the wrench is viewed in such a way that only its planar part is visible, with no information about the depth. Thus, the surface segmentation is of no help in obtaining the volumetric segmentation.

Starting with the global model, the SUPERSEG procedure places the part-models at **s_under** regions labeled 12 and 13 (bottom row, figure 8.16), and begins to shrink the global model. The four iterations are shown in figure 8.17, where the head of the wrench is segmented into 3 parts, and neck is separated from the handle of the wrench. At the end of the first iteration there are two active part models and a global model. The second iteration results in terminating the growth of the two part models, and initiating a third part model to complete the description of the head of the wrench. During the third iteration, a part model is placed for the neck which terminates during the fourth iteration. The procedure terminates after the fourth iteration because of the global model (now describing the curved part of the handle) satisfying the acceptance criteria.

Figure 8.18 shows the plots of the qualitative residuals and the goodness-of-fit values (with and without the volume criterion), as also the size of global model during the four iterations. The model reduces size along the Z dimension, which is expected since the residuals of the global model lie in that direction. As the global model shrinks, the goodness-of-fit values drop (indicating a better fit), so do the qualitative residuals evaluating the fit as percentage of over/underestimated points.

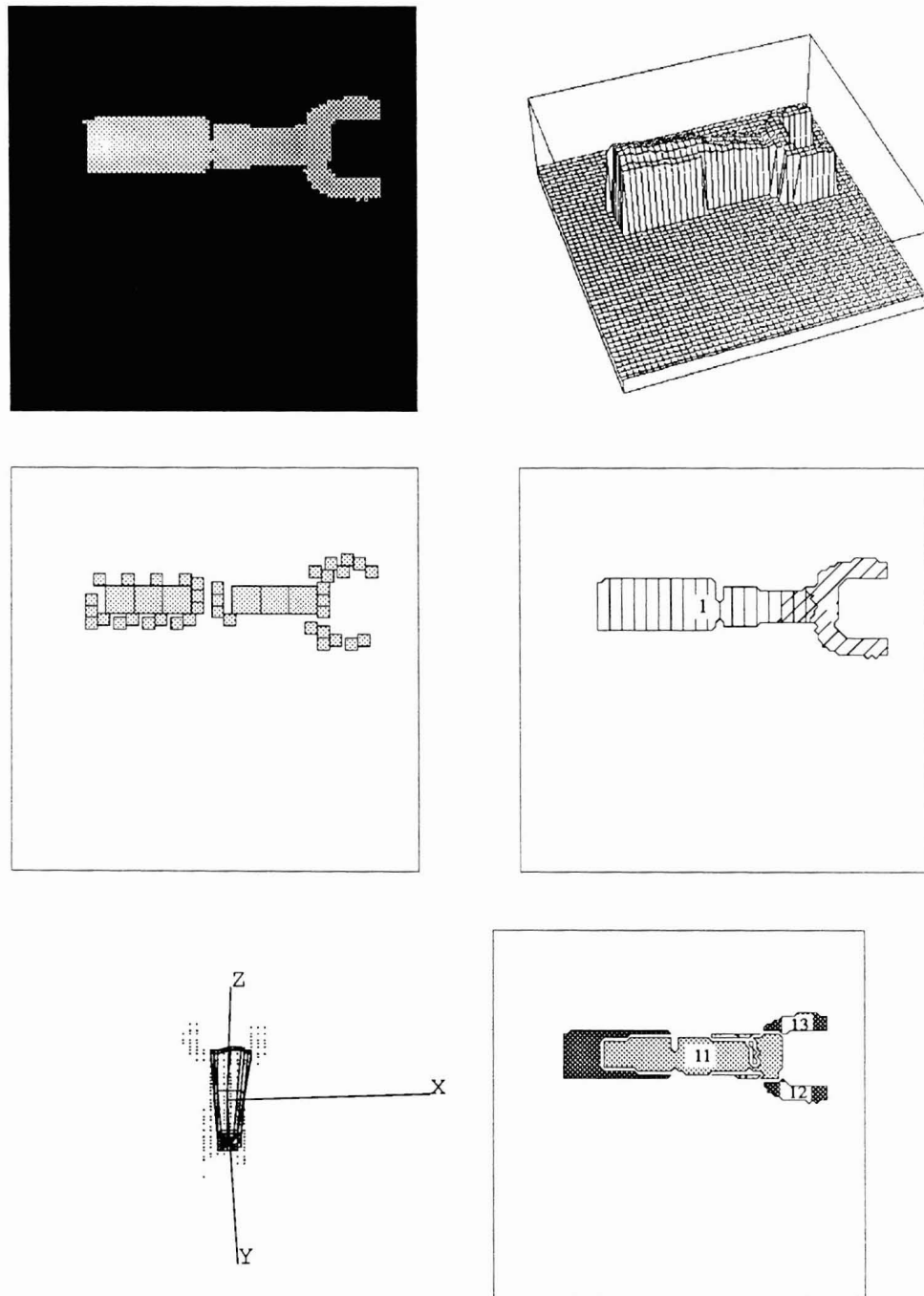


Figure 8.16: **The Wrench:** Top: Range image and its 3-D plot. Center: Bi-quadric seed regions and bi-quadric segmentation. Bottom: Global model and its residuals.

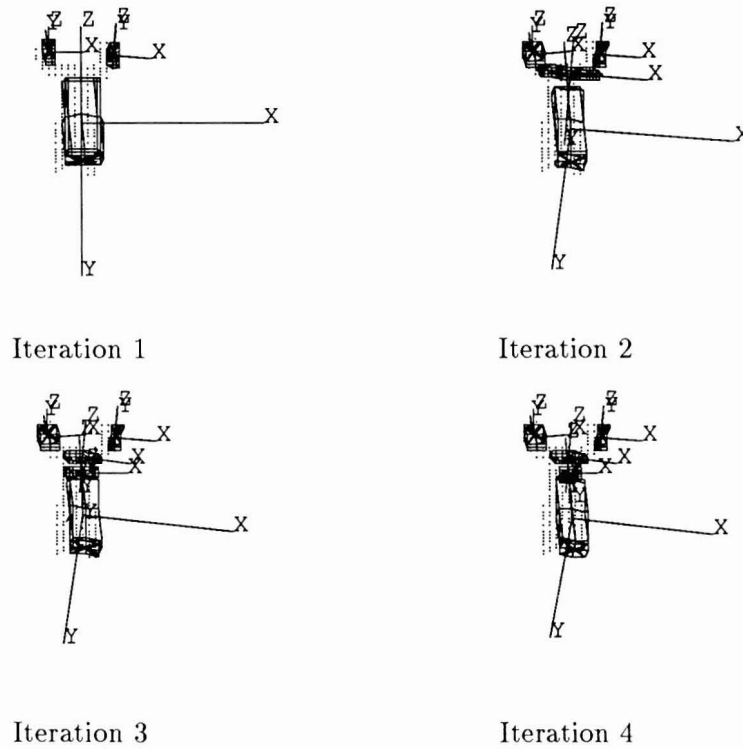


Figure 8.17: **The Wrench: Volumetric segmentation:** Iterations 1 through 4, segmenting the part-structure. Top: Iterations 1 and 2. Bottom: Iterations 3 and 4.

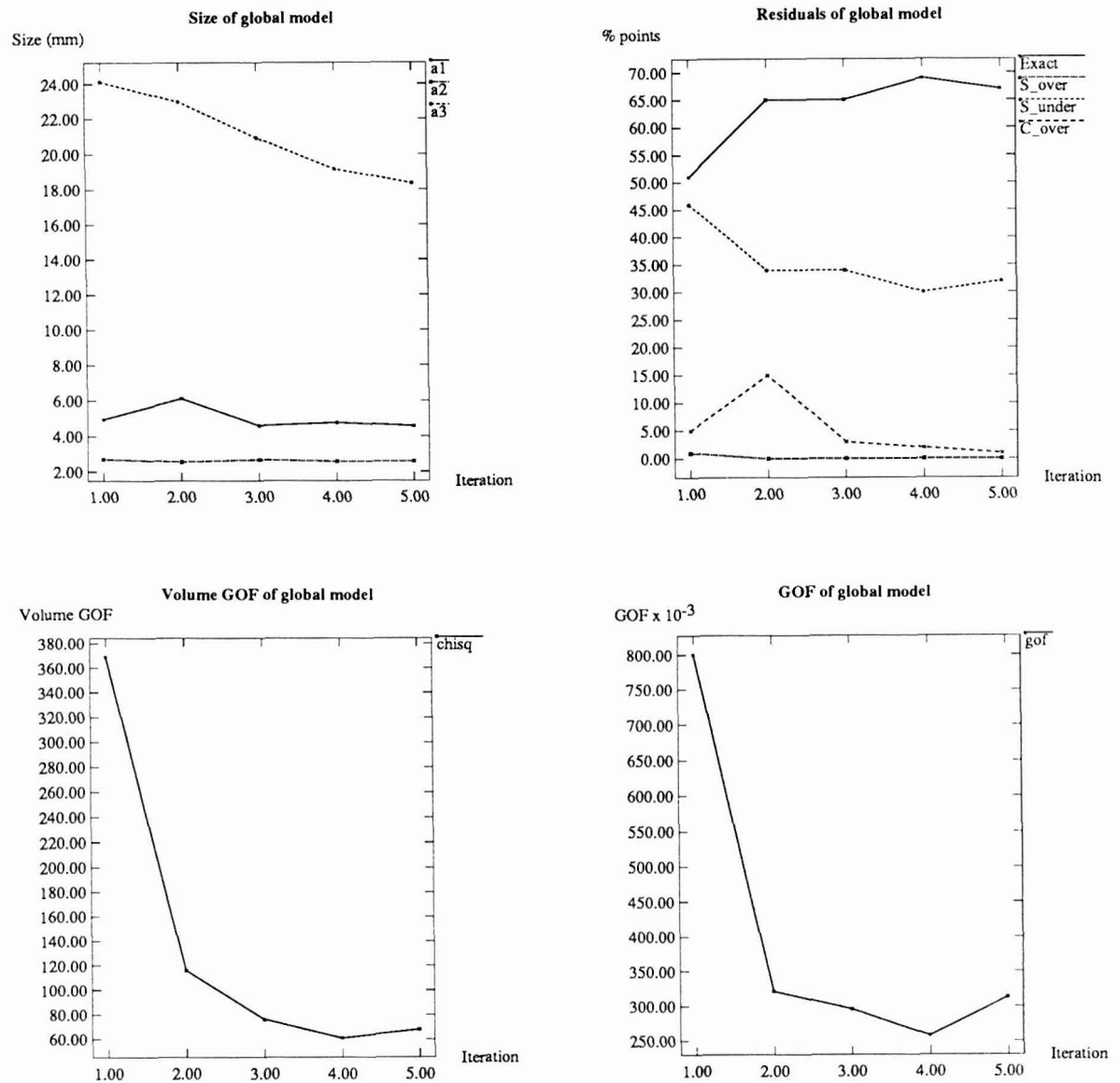


Figure 8.18: **Wrench: Global model:** Variation of parameters and residuals during 4 iterations. Top: Size of the model, and qualitative residuals. Bottom: Volume-based error of the fitting function, and the non-volume-based error.

8.2.6 The NIST Object

The object (figure 8.20) has holes and a definite part structure. The bi-quadric surface description correctly segments the surface into planar patches (2, 4, 5, 6, and 7) and second-order patches (1 and 3). following the procedure described in chapter 4, the internal edges (c_1 discontinuities) are localized and marked as convex or concave (figure 4.6), and the surface adjacency graph (sag) is constructed (figure 8.19). the approach for convex combination will work for this object, but we show the results for the general approach.

The global model shows substantial residuals, and starts the part segmentation by combining the base of the object in convex sense. here the control structure recognizes the convex component and opts for the convex combination of the two patches rather than going for the elaborate model growing process. the global model now describes the data corresponding to the other two parts (top left in figure 8.21), resulting in two disconnected components in the Z-residuals. Thus, the global model bifurcates into two global models, and the two components are individually processed to recover the part-models corresponding to the individual parts. This example shows how the control structure integrates surface information with the global-to-local surface approach and dynamically segments the data clusters. The residuals for the parts show the need for negative volume descriptions for the holes, which can be done by a secondary process.

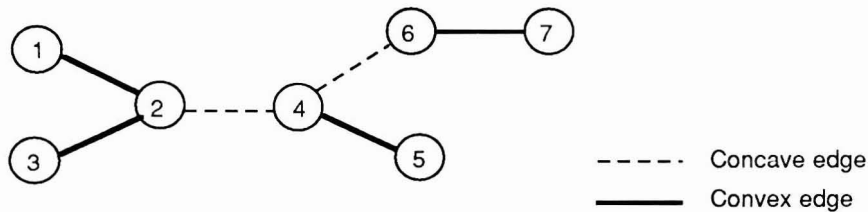


Figure 8.19: **Surface adjacency graph (SAG) for the NIST object:** The removal of concave edges splits the graph into three connected components, corresponding to the three parts in the object.

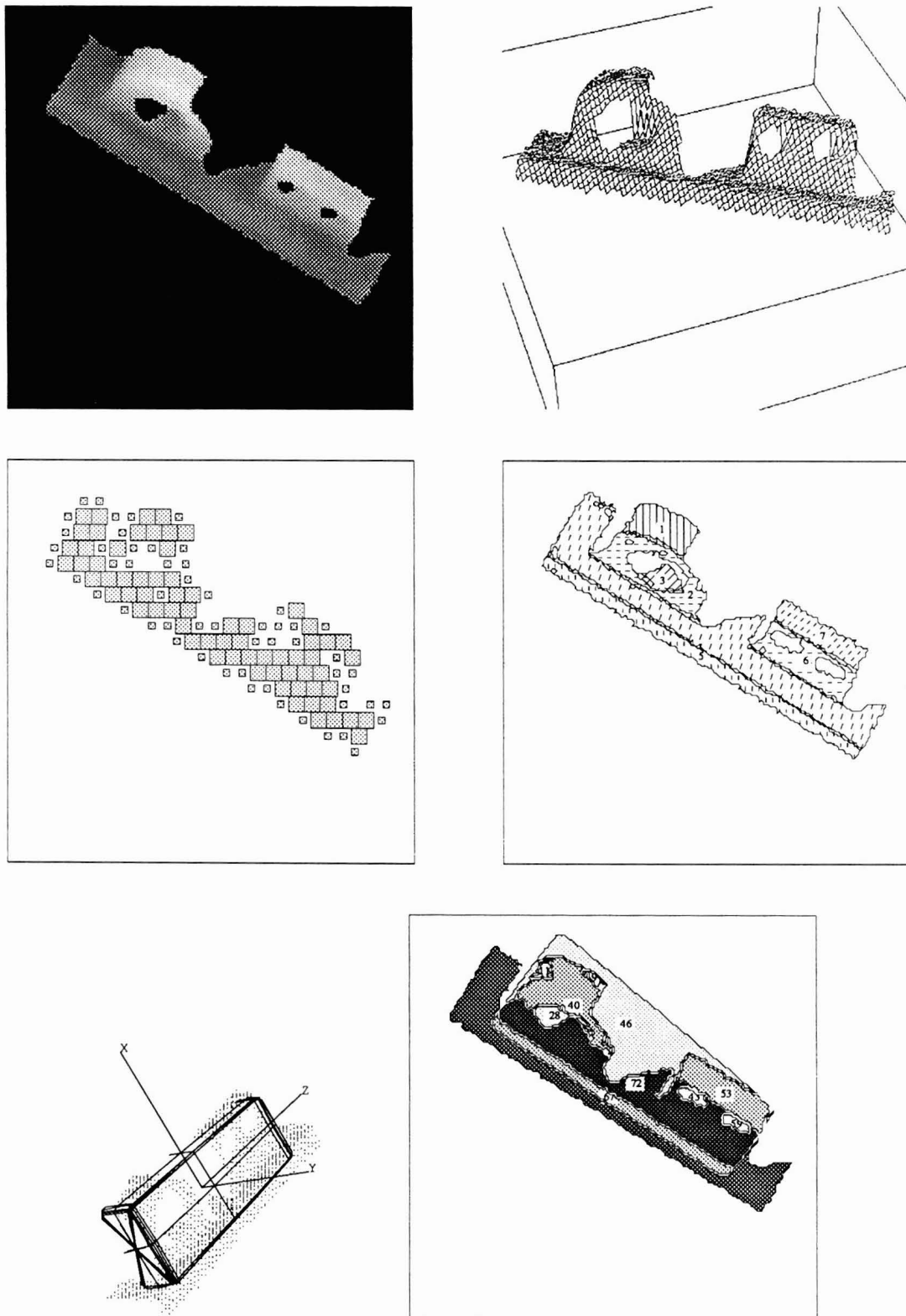


Figure 8.20: **The NIST object:** Top: The range image and its 3-D plot. Center: bi-quadric seed regions and the bi-quadric segmentation. Bottom: The global model and its Z-residuals.

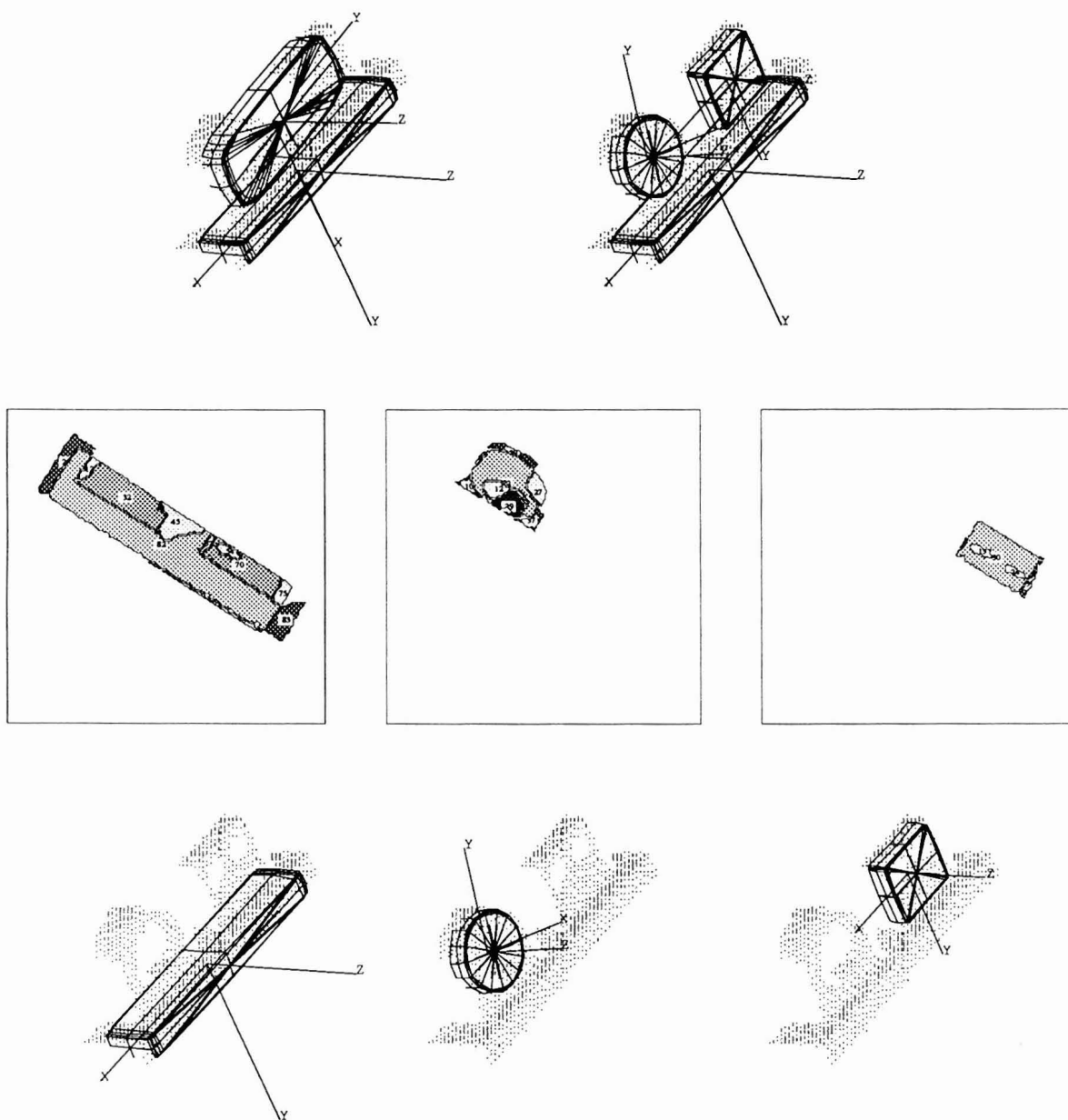


Figure 8.21: **The NIST Object: Volumetric segmentation:** Top: (left) Description after the first component separation, and (right) the final description. Center: The Z-residual regions of the three parts of the object. Bottom: Final descriptions of the three parts

8.3 Implementation Aspects

The SUPERSEG system was implemented in a modular manner with the integration package as the central core. The surface segmentation package and the global superquadric module recovery are individual packages while the integration module performs all the remaining tasks. The system is composed of more than 15,000 lines of documented C code, and runs on SUNs and IBM 6000 machines, with an X display interface for real-time visualization, and a POSTSCRIPT interface for output generation.

The range data is obtained from a structured lighting scanner. The range images have considerable shadows due to the geometry of the scanner, and the depth information is noisy at the step boundaries. Median filtering of data along the step boundaries is done to bring the noisy edge points to conform to the neighboring surfaces, without loss of the spatial extent of the object. Median filtering works better than eroding the data along the step edges. The range image has different spatial resolution along X and Y axes. The image is scaled uniformly along X and Y axes to 1mm/pixel using lagrangian interpolation method. The depth resolution of the scanner is 1.5mm.

8.3.1 System Parameters

The thresholds are sensor dependent, and reflect the effect of noise due to sensor measurements and quantization, and the tolerance of the deviation of a model from the data. All the values were determined empirically.

Surface Segmentation:

1. Acceptable global chi-square error for placement of seeds = 1mm^2 .
2. Compatibility constraint, $C = \pm 3 \text{ mm}$.
3. Acceptable global chi-square error of a patch = 3mm^2 .
4. If during an iteration, a model does not grow at least 1%, or if a model whose order was updated (say from 1 to 2) does not grow at least 10%, then the model growing is terminated. In the latter case, the order of the model is reverted back to the previous order.
5. Model selection parameters: $K_1 = 1$, $K_2 = 5$, and $K_3 = 5$. K_2 and K_3 are normalized with respect to K_1 . A value of 5 accepts small regions in the final description by making them more cost effective than the pixel based description.

Superquadric Recovery and Segmentation:

1. Maximum number of iterations (m in Algorithm 5.1) for model recovery: 15.
2. Tolerance for measuring deviation from model: 0.1 for the inside-outside function, and 3mm for the Z-residuals.
3. Acceptable non-volume goodness-of-fit = 0.1.
4. Acceptance criteria for a superquadric model: At least 80% points **exact** and, less than 10% each of other residuals.

8.4 Discussion

The results presented in this chapter demonstrate the various capabilities of the general volumetric segmentation method. The control procedure is able to adapt to the complexity of the input and generate suitable descriptions. The procedure effectively integrates the global-to-local strategy of the volumetric segmentation with the descriptions obtained by the local-to-global clustering of the surface segmentation. The control module can be adapted to prefer certain descriptions to other, as also to give coarse or fine descriptions based on scale considerations, which determine the thresholds in the residual analysis. Since relative error measures are used for evaluation, the method is not sensitive to the absolute differences in size.

The descriptions are generated in object-centered coordinate systems with respect to a known world coordinate system. The superquadric descriptions are hierarchical in nature, depicting the coarse-to-fine segmentation of data into convex volumes. As observed in the case of bent objects, the “scale” of the description is defined by tolerance in measuring the deviation of data from the model. Thus, the global model (for the entire data) provides the description at the coarsest scale. The first level of refinement involves satisfying the acceptance criteria within the specified tolerances. For example, for the same acceptance criteria, the description obtained for a tolerance of ± 3 mm in Z-residuals will be coarser than what obtained for a tolerance of ± 2 mm. A typical acceptance criteria may require that 80% of the data should be accounted for by the model within the specified tolerance, and that the model should not have contour overestimations of more than 20%. Special handling is needed for the parts that appear flat due to the degenerate viewing position, where a relaxed acceptance criteria is applied. The reason being that the volumetric model, in the absence of the depth information tends to underestimate the data, resulting in significant

s_under regions. For example, the head of the wrench is segmented into “volumes” even though the depth information is not available. This is possible because the system is able to recognize the fact that the part being modeled is flat, and alter the acceptance criteria to accommodate the part dimensionality.

Bi-quadric description is considered final after refining the patches along the intersection curves. Bi-quadric surfaces can be further segmented by the volumetric segmentation procedure, as in the case of L-shaped objects. Superquadric models are object-centered with known transformation from the world (image) coordinate system, as shown in figure 5.8. Bi-quadric models are reduced to their standard form with known transformation from the world coordinate system.

The superquadric models are scale, position, and orientation invariant as long as the viewpoint for the parts is not degenerate. They are not sensitive to occlusion due to other parts if the available information can constrain all the model parameters. Curved bi-quadric models, on the other hand, are sensitive to orientation since the Z-axis is fixed along the viewing direction. Planar patches are invariant to orientation to a great extent, except when the plane tilts significantly with respect to the viewing direction. The least-squares method of minimizing error along the viewing direction can result in a curved patch instead of a planar one. This is not a matter of concern for us since the planarity check mentioned in chapter 7 corrects this problem.

At the level of surfaces, planar patches are insensitive to viewpoint changes, while the piece-wise curved description can vary with the changes in the viewpoint. The variation at the volumetric level is due to the inherent ambiguities in the definition of part-structure using volumetric primitives. The volumetric segmentation is completely data-driven and does not have a priori knowledge about the objects or object domain. For this reason, segmented descriptions of an object viewed from different angles may differ in part-structure. However, the number of parts, and their relationships will still be qualitatively similar. For example, an L-shaped object will be described as consisting of 2 parts, even though the individual parts may vary in quantitative details. While this is undesirable if precise object-recognition is the goal, such descriptions are adequate if the goal is qualitative description or classification of the object. These descriptions are also adequate for designing grasping strategies for object manipulation, and for navigation or path planning applications. The invariance demanded by object recognition tasks can be imposed by using the high-level knowledge (in the form object models) to guide the segmentation of the object. At the very least, these descriptions can reduce the complexity of the search by narrowing down the

potential candidates during the database search.

Due to the Z-depth ($2\frac{1}{2}$ -D) nature of the range data, the volumetric models (which are 3-D models) may not be fully constrained for curved objects. Superquadric models impose symmetry constraints, and predict data on the hidden side that satisfies the surface and volume constraint for the data on the visible side. This will result in a perceptually acceptable box-shaped description for two perpendicular planes, while an elliptical cylinder may be obtained for a circular cylinder. Whaite and Ferrie [1990] have studied this problem of inherent lack of uniqueness in predicting unobserved data using volumetric models. They define an “ellipsoid of confidence”, derived from the covariance matrix of the fitting procedure, within which a number of superquadric models fit the data equally well. Although their procedure does not commit a model, it still imposes the symmetry constraint to predict the invisible data, which suggests the possible viewing direction to minimize the ambiguity. Model recovery can also be constrained by using the physical constraints of support and stability, as investigated in [Gupta et al. 1989b]. However, in cases where these constraints cannot be imposed (either due to the formulation of the problem or because of their unavailability), the former approach seems most plausible since it can guide the additional acquisition of data in the spirit of active perception.

In the next chapter we will summarize our approach and discuss the various applications where the structured descriptions generated by the program can prove to be useful.

CHAPTER 9

Summary and Conclusions

We developed a control structure for segmenting dense range data of complex 3-D objects into their constituent parts in terms of surface and volumetric primitives. The procedure is completely data-driven, with shape constraints imposed by the geometric models. The descriptions are *recovered* without *a priori* domain knowledge or stored models. *Bi-quadric* models for surface representation and *superquadric* models for object-centered volumetric representation are used to obtain a hierarchical shape description.

The surface segmentation uses a novel approach of *searching* for the best piecewise description of the image in terms of bi-quadric ($z = f(x, y)$) models. It is used to generate the region adjacency graphs, to localize surface discontinuities, and to derive global shape properties like embedding of the surface in space, and the orientation of the standard form of the surface. A superquadric model is recovered for the entire data set and residuals are computed to evaluate the fit. The goodness-of-fit value based on the inside-outside function, and the mean-squared distance of data from the model provide quantitative evaluation of the model. The qualitative evaluation criteria check the local consistency of the model in the form of residual maps of *overestimated* and *underestimated* data regions.

The control structure invokes the models in a systematic manner, evaluates the intermediate descriptions, and integrates them to achieve final segmentation. Superquadric and bi-quadric models are recovered in parallel to incorporate the best of the coarse-to-fine and fine to coarse segmentation strategies. The model evaluation criteria determine the dimensionality of the scene, and decide whether to terminate the procedure, or selectively refine the segmentation by following a global-to-local part segmentation approach. The control module generates hypotheses about superquadric models at clusters of underestimated data and performs controlled extrapolation of the part-model by shrinking the global model. As

the global model shrinks and the local models grow, they are evaluated and tested for termination or further segmentation.

We presented results on real range images of scenes of varying complexity, including objects with occluding parts, and scenes where surface segmentation is not sufficient to guide the volumetric segmentation. We analyzed the issue of segmentation of complex scenes thoroughly by studying the effect of missing data on volumetric model recovery, generating object-centered descriptions, and presenting a complete set of criteria for the evaluation of the superquadric models.

We now discuss the applications of our approach in data reduction, 3-D object recognition, geometric modeling, automatic model generation, object manipulation, and active vision.

9.1 Applications and Future Directions

The volume and surface descriptions generated by our approach have applications in various middle and high level vision tasks.

The segmentation process can be viewed as a data reduction mechanism where symbols are attached to the raw data to obtain structured symbolic representations. Such descriptions can be used by high-level vision tasks like object recognition and classification, and database matching, which require canonical descriptions of the part-structure of an object.

This dissertation dealt with the data-driven bottom-up approach to scene segmentation. Our objective was to study the utility of the geometric models in participating in the segmentation process in the absence of any domain knowledge. Thus, the segmented descriptions correspond to the geometric models and not necessarily to the notion of objects as they exist in the real world. Additionally, the descriptions may suffer from ambiguities and non-invariance to the orientation of the objects. Clearly, domain knowledge in the form of *a priori* constraints and object models can be and should be incorporated to make the descriptions meaningful in the given context. The method has the possibility of incorporating domain specific constraints in order to make it prefer certain types of descriptions, and to resolve ambiguities in the part description. If object-models are available, they can constrain the parts to lie in a known configuration, and bring in object-specific constraints.

The procedure currently uses dense sparse data to exploit the neighborhood information provided by the Z-depth representation. A general method that uses sparse depth information obtained from reflectance images will tremendously broaden the scope of the

paradigm.

The parametric descriptions have the advantage of compactly describing the shape, size, and the pose of the object, making them useful for object manipulation and path planning tasks. The descriptions have already been used to segment objects in a cluttered scene, and generate grasping strategies.

The analysis of $2\frac{1}{2}$ -D data in terms of 3-D primitives also adds the possibility of supplementing more information or integrating it as it becomes available in a temporal fashion. Thus, it is possible to update the description, instead of computing it afresh. Additionally, superquadrics predict data by imposing symmetry constraints, which require additional data for the verification.

The surface and volumetric descriptions form the basis of 3-D object models. It is possible to use the surface segmentation as a clustering mechanism for an automatic model generation system or it can be further refined by an operator-assisted model generation system in the spirit of reverse engineering. The segmentation provided at surface level into planar and curved patches has been shown to reduce the complexity of model building process significantly.

As mentioned earlier, shape models with local deformations offer better modeling capabilities, but complicate the already difficult problem of segmentation. The description generated by the SUPERSEG system can be used as the first approximation for such models, which can then refine the segmentation by modeling the data more closely than that is possible by rigid models.

9.2 Contributions of the Dissertation

We developed a paradigm for shape description of single viewpoint range data, demonstrating that the issues of representation and segmentation are not separable. The part segmentation problem was formulated with the definition of the geometric primitives, which represent the parts. We proposed a fine-to-coarse surface segmentation strategy and a coarse-to-fine volume segmentation strategy, and a control structure that integrates these descriptions in a systematic manner.

A new paradigm for surface segmentation was developed which searches for the best description of image in terms of primitives by dynamically combining model recovery and model selection. The paradigm is general and has a variety of applications. The surface segmentation obtained from this was used to provide initial segmentation for the object

centered superquadric part-models. The method automatically determines the domain of applicability, as also implicitly detects and labels the surface discontinuities.

We analyzed shape vocabulary and recovery methods for superquadrics. We provided a novel interpretation for the modified inside/outside function that allows us to use it for superquadric model evaluation. The interpretation formulated as superquadric expansion and contraction also gives a close initial approximation for the procedure computing the Euclidean distance of a point from a superquadric model. We derived a closed-form solution for tracing the occluding contours of superquadrics. Additionally, we developed algorithms for tracing occluding contours of deformed superquadrics, for rendering superquadric model along a viewing direction, for computing and representing superquadric edges, etc.

A complete set of criteria was developed to evaluate the superquadric models by comparing it against the given data, and to guide the segmentation procedure. We developed a control structure that recognizes the dimensionality of the scene and segments data by recovering the relevant models on the appropriate pieces of data. We discussed the issues involving segmentation of static scenes using volumetric models. The concept of global-to-local segmentation by growing the part models and shrinking the global models was introduced, and shown to be effective in extracting the part-structure.

APPENDIX A

Linear Least Squares for Surface Fitting

A linearly parameterizable surface patch $\mathcal{S}(r, \mathbf{a}, \mathbf{x})$ can be written as :

$$\mathcal{S}(r, \mathbf{a}, \mathbf{x}) = \{(\mathbf{x}, z) \in \mathcal{I} \times \mathcal{Z} \mid z = \hat{f}(r, \mathbf{a}, \mathbf{x})\} \quad (\text{A.1})$$

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

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

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

$$\chi^2(r, \mathbf{a}, \mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} d^2(r, \mathbf{a}, \mathbf{x}) \quad (\text{A.3})$$

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 SSE function $\chi^2(r, \mathbf{a}, \mathcal{D})$. Using least-squares regression we get :

$$\chi^2(r, \hat{\mathbf{a}}, \mathcal{D}) = \min_{\mathbf{a} \in \mathcal{A}} \chi^2(r, \mathbf{a}, \mathcal{D}) \quad (\text{A.4})$$

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

The solution is given by :

$$\mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^T \mathbf{Y} \quad (\text{A.5})$$

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

$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (\text{A.6})$$

$$\text{where } \hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_p \end{bmatrix}_{p \times 1}, \quad \mathbf{Y} = \begin{bmatrix} g(\mathbf{x}_1) \\ \vdots \\ g(\mathbf{x}_n) \end{bmatrix}_{n \times 1}, \quad \mathbf{X} = \begin{bmatrix} f_1(\mathbf{x}_1) & \dots & f_p(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ f_1(\mathbf{x}_n) & \dots & f_p(\mathbf{x}_n) \end{bmatrix}_{n \times p} \quad (\text{A.7})$$

and $f_l(\mathbf{x})$ are the basis functions. The solution depends on the points in \mathcal{D} .

A.1 Solving the Least Squares Problem

In the step 2 of the segmentation algorithm 3.1 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.

$$\rho^{(0)} = \chi^2(\mathcal{D}^{(0)}) \quad (\text{A.8})$$

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:

$$\mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^T \mathbf{Y} \quad (\text{A.9})$$

Here we describe an efficient way for solving the normal equations [Seber 1977].

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

$$\mathbf{A} = \begin{pmatrix} \mathbf{X}^T \mathbf{X} & \mathbf{X}^T \mathbf{Y} \\ \mathbf{Y}^T \mathbf{X} & \mathbf{Y}^T \mathbf{Y} \end{pmatrix}_{(p+1) \times (p+1)} \quad (\text{A.10})$$

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

$$\begin{pmatrix} \mathbf{V} & \mathbf{d} \\ 0 & \rho^{(0)} \end{pmatrix}_{(p+1) \times (p+1)} \quad (\text{A.11})$$

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,

$$\rho^{(0)} = \mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (\text{A.12})$$

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.

APPENDIX B

Deriving the Standard Forms of Bi-quadrics

The general form of a bi-quadric is given by:

$$z = ax^2 + by^2 + cxy + dx + ey + f \quad (\text{B.1})$$

There are three types of curved bi-quadric patches, viz. elliptic paraboloid, hyperbolic paraboloid, and the elliptic cylinder, as shown in figure 4.3. Since every bi-quadric has a standard form without the cross-terms, it is possible to transform the general form into the standard form and obtain the orientation information as a result. The standard forms of the three bi-quadric surfaces are given by:

$$\begin{aligned} \text{Elliptic Paraboloid: } z &= \frac{x^2}{a'^2} + \frac{y^2}{b'^2} \\ \text{Hyperbolic Paraboloid: } z &= \frac{x^2}{a'^2} - \frac{y^2}{b'^2} \\ \text{Parabolic Cylinder: } z &= \frac{x^2}{a'^2} \end{aligned} \quad (\text{B.2})$$

The transformation is accomplished by translating and orienting the patch to remove the linear and cross-multiple terms in the equation. Since we are interested in only the orientation of the surface, we will deal with only the second-order terms. The standard form of a bi-quadric surface is given by [Korn and Korn 1961, Hall et al. 1982]:

$$z = a'x'^2 + b'y'^2 + c \quad (\text{B.3})$$

Equation B.1 can be represented in matrix form as

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & c/2 \\ c/2 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d & e & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + f = 0 \quad (\text{B.4})$$

or

$$\mathbf{X}^T \mathbf{D} \mathbf{X} + \mathbf{F} \mathbf{X} + f = 0. \quad (\text{B.5})$$

We want to diagonalize \mathbf{D} so that the xy term disappears, and remove the linear terms to get:

$$\begin{bmatrix} x' & y' \end{bmatrix} \begin{bmatrix} a' & 0 \\ 0 & b' \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} - z = 0 \quad (\text{B.6})$$

or

$$\hat{\mathbf{X}}^T \hat{\mathbf{D}} \hat{\mathbf{X}} - z = 0; \quad (\text{B.7})$$

The transformation from one orthonormal basis to another can be performed by

$$\hat{\mathbf{X}} = \mathbf{P} \mathbf{X} \quad \text{or} \quad \mathbf{X} = \mathbf{P}^{-1} \hat{\mathbf{X}}, \quad (\text{B.8})$$

where \mathbf{P} is a linear transformation matrix, composed of rotation information of the form:

$$\begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (\text{B.9})$$

encoding the rotation of x and y axes about the z -axis by the angle α . If \mathbf{P} is the transition matrix from a basis $\hat{\mathbf{F}}$ for a finite dimensional vector space, then \mathbf{P} is invertible [Hall et al. 1982], and $\mathbf{P}^{-1} = \mathbf{P}^T$. We have to find a matrix \mathbf{P} that will diagonalize matrix \mathbf{D} to form matrix $\hat{\mathbf{D}}$:

$$\mathbf{P}^{-1} \mathbf{D} \mathbf{P} = \hat{\mathbf{D}} \quad (\text{B.10})$$

By substituting equation B.8 into equation B.7, we get

$$(\mathbf{P}^{-1} \hat{\mathbf{X}})^T \mathbf{D} (\mathbf{P}^{-1} \hat{\mathbf{X}}) - z = 0. \quad (\text{B.11})$$

Since $(\mathbf{P}^{-1} \hat{\mathbf{X}})^T = \hat{\mathbf{X}}^T \mathbf{P}^{-1}$, we get

$$\hat{\mathbf{X}}^T \mathbf{P}^{-1T} \mathbf{D} (\mathbf{P}^{-1} \hat{\mathbf{X}}) - z = 0.$$

Which gives

$$\hat{\mathbf{X}}^T(\mathbf{PDP}^T)\hat{\mathbf{X}} - z = 0. \quad (\text{B.12})$$

Comparing the quadratic forms of equations B.12 and B.7 gives,

$$\mathbf{PDP}^T = \hat{\mathbf{D}} \Rightarrow \mathbf{DP}^T = \mathbf{P}^T\hat{\mathbf{D}} \Rightarrow \mathbf{DP}^T = \mathbf{\Lambda P}^T. \quad (\text{B.13})$$

Therefore, \mathbf{P} is an eigenvector matrix, specifying the rotation of the bi-quadric in the general form to reduce it to its standard form. Every bi-quadric has two principal planes and the directions of the normals to the principal planes are directed along the eigenvectors associated with matrix \mathbf{D} . Thus, the column vectors of \mathbf{P} specify the orientation of the standard bi-quadric with respect to the image coordinate system.

APPENDIX C

Superquadric Surface Normals

Given a superquadric surface:

$$\mathbf{x}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\varepsilon_1}(\eta) \cos^{\varepsilon_2}(\omega) \\ a_2 \cos^{\varepsilon_1}(\eta) \sin^{\varepsilon_2}(\omega) \\ a_3 \sin^{\varepsilon_1}(\eta) \end{bmatrix} \quad \begin{matrix} -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{matrix} \quad (\text{C.1})$$

the surface normal vector $N(\eta, \omega) = N(x, y, z)$ at $(\eta, \omega) = (x, y, z)$ can be derived as:

$$N(\eta, \omega) = \frac{\partial \mathbf{x}}{\partial \eta} \times \frac{\partial \mathbf{x}}{\partial \omega}. \quad (\text{C.2})$$

Which gives:

$$N(\eta, \omega) = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{1}{x} \cos^2(\eta) \cos^2(\omega) \\ \frac{1}{y} \cos^2(\eta) \sin^2(\omega) \\ \frac{1}{z} \sin^2(\eta) \end{bmatrix} \quad (\text{C.3})$$

Eliminating (x, y, z) gives the normals in terms of (η, ω) :

$$N(\eta, \omega) = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{1}{a_1} \cos^{(2-\varepsilon_1)}(\eta) \cos^{(2-\varepsilon_2)}(\omega) \\ \frac{1}{a_2} \cos^{(2-\varepsilon_1)}(\eta) \sin^{(2-\varepsilon_2)}(\omega) \\ \frac{1}{a_3} \sin^{(2-\varepsilon_1)}(\eta) \end{bmatrix} \quad (\text{C.4})$$

Eliminating η and ω and simplifying the equations gives the superquadric normals:

$$\begin{aligned}
 n_x &= \left[1 - \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}\right] \frac{x^{\left(\frac{2}{\epsilon_2}-1\right)} a_2^{\frac{2}{\epsilon_2}}}{(y a_1^{\frac{2}{\epsilon_2}}) + (x a_2^{\frac{2}{\epsilon_2}})} \\
 n_y &= \left[1 - \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}}\right] \frac{y^{\left(\frac{2}{\epsilon_2}-1\right)} a_1^{\frac{2}{\epsilon_2}}}{(y a_1^{\frac{2}{\epsilon_2}}) + (x a_2^{\frac{2}{\epsilon_2}})} \\
 n_z &= \frac{z^{\left(\frac{2}{\epsilon_1}-1\right)}}{a_3^{\frac{2}{\epsilon_1}}}
 \end{aligned} \tag{C.5}$$

Bibliography

- [Agin and Binford 1973] Agin, G. J. and T.O. Binford, Computer description of curved objects, in *Proceedings of International Joint Conference of Artificial Intelligence*, pp. 629–640, 1973.
- [Arnheim 1974] Arnheim, R., *Art and Visual Perception*, University of California Press, 1974.
- [Asada and Brady 1986] Asada, H. and Michael Brady, The curvature primal sketch, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 2–14, January 1986.
- [Bajcsy and Gupta 1989] Bajcsy, Ruzena and Alok Gupta, Part description and segmentation of range images by integration of contour, surface, and volumetric primitives, in *Proceedings of the Topical Meeting on Image Understanding and Machine Vision*, pp. 100–103, Optical Society of America, June 1989.
- [Bajcsy and Solina 1987] Bajcsy, R. and F. Solina, Three dimensional shape representation revisited, in *First International Conference on Computer Vision*, pp. 231–240, London, 1987.
- [Bajcsy et al. 1986] Bajcsy, R., E. Krotkov, and M. Mintz, *Models of Errors and Mistakes in Machine Perception*, Technical Report MS-CIS-86-26, University of Pennsylvania, 1986.
- [Bajcsy et al. 1990a] Bajcsy, R., F. Solina, and A. Gupta, Segmentation versus object representation – are they separable?, in *Analysis and Interpretation of Range Images*, edited by R. Jain and A. Jain, Springer-Verlag, 1990a.
- [Bajcsy et al. 1990b] Bajcsy, Ruzena, Kwangyeon Wohn, Franc Solina, Alok Gupta, Gareth Funka-Lea, Celina Imielinska, Pramath Sinha, and Constantine Tsikos, *Final report on Advanced Research in Range Image Interpretation for Automated Mail Handling*, Technical Report MS-CIS-90-14, CIS Department, University of Pennsylvania, 1990b.

- [Ballard and Brown 1982] Ballard, Dana H. and Christopher M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [Barr 1981] Barr, Alan H., Superquadrics and angle-preserving transformations, *IEEE Computer Graphics and Applications*, 1, , January 1981.
- [Barr 1984] Barr, Alan H., Global and local deformations of solid primitives, *Computer Graphics*, 18, 21–30, July 1984.
- [Bennett and Hoffman 1987] Bennett, B. M. and D. D. Hoffman, Shape decompositions for visual recognition: the role of transversality, in *Image Understanding 1985-86*, edited by W. Richards and S. Ullman, Ablex Publishing Corporation, Norwood, NJ, 1987.
- [Besl 1988] Besl, P. J., *Surfaces in Range Image Understanding*, Spriger-Verlag, 1988.
- [Besl and Jain 1985] Besl, Paul J. and Ramesh C. Jain, Three-dimensional object recognition, *ACM Computing Surveys*, 17, March 1985.
- [Besl and Jain 1986] Besl, P. J. and Ramesh C. Jain, Invariant surface characteristics for three dimensional object recognition in range images, *Computer Vision, Graphics, and Image Processing*, 33, 33–88, 1986.
- [Besl and Jain 1988] Besl, Paul J. and Ramesh C. Jain, Segmentation through variable-order surface fitting, *IEEE Trans. Pattern Anal. Machine Intelligence*, 10, March 1988.
- [Biederman 1985] Biederman, I., Human image understanding: recent research and theory, *Computer Vision, Graphics, and Image Processing*, 32, 29–73, 1985.
- [Biederman 1987] Biederman, I., Matching image edges to object memory, in *Proc. of International Conference on Computer Vision*, pp. 384–392, 1987.
- [Binford 1971] Binford, T. O., Visual perception by computer, in *IEEE Conference on Systems and Control*, Miami, 1971.
- [Binford 1982] Binford, Thomas O., Survey of model-based image analysis systems, *International Journal of Robotics Research*, 1, 18–64, Spring 1982.
- [Blake and Zisserman 1987] Blake, A. and A. Zisserman, *Visual Reconstruction*, The MIT Press, 1987.

- [Bolle and Vemuri 1991] Bolle, Ruud M. and Baba C. Vemuri, On three-dimensional surface reconstruction methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, Jan 1991.
- [Bolles and Fischler 1981] Bolles, R. C. and M. A. Fischler, A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, 24, 381–395, June 1981.
- [Boult and Gross 1987] Boult, T.E. and A. D. Gross, Recovery of superquadrics from depth information, in *Proceedings of Spatial Reasoning and Multi-Sensor fusion workshop*, pp. 128–137, St. Charles, IL, 1987.
- [Brady 1983] Brady, M., Criteria for representation of shape, in *Human and Machine Vision*, edited by J. Beck, B. Hope, and A. Rosenfeld, Academic Press, Orlando, FL, 1983.
- [Brady et al. 1985] Brady, M., J. Ponce, A. Yuille, and H. Asada, Describing surfaces, *Computer Vision, Graphics, and Image Processing*, 32, 1–28, 1985.
- [Brooks 1983] Brooks, Rodney, Model-based 3-d interpretations of 2-d images, *IEEE Pattern Analysis and Machine Intelligence*, 5, 140–150, 1983.
- [Chen 1989] Chen, D. S., A data-driven intermediate level feature extraction algorithm, *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 11, 749–758, July 1989.
- [Comba 1968] Comba, Paul G., A procedure for detecting intersections of three-dimensional objects, *Journal of the ACM*, 15, 354–366, July 1968.
- [Darrell et al. 1990] Darrell, Trevor, Stan Sclaroff, and Alex Pentland, Segmentation by minimal description, in *Proceedings of the Third International Conference on Computer Vision*, pp. 112–116, IEEE, Osaka, Japan, December 1990.
- [Fan 1988] Fan, T. J., *Describing and Recognizing 3-D Objects Using Surface Properties*, PhD thesis, University of Southern California, August 1988, Technical report IRIS 237.
- [Faugeras and Hebert 1986] Faugeras, O.D. and M. Hebert, The representation, recognition, and locating of 3-d objects, *International Journal of Robotics Research*, 5, 27–52, 1986.
- [Ferrie et al. 1989] Ferrie, F. P., J. Lagarde, and P. Whaite, Darboux frames, snakes, and super-quadrics: geometry from the bottom-up, in *Proceedings of the IEEE Workshop on Interpretation of 3D Scenes*, pp. 170–176, Austin, TX, November 1989.

- [Godin and Levine 1989] Godin, G. D. and M. D. Levine, Structured edge map of curved objects in a range image, in *Proceedings of Computer Vision and Pattern Recognition Conference*, 1989.
- [Gombrich 1972] Gombrich, E. H., *Art and Illusion*, Phaidon Press, London, 1972.
- [Gupta 1988] Gupta, A., *Range Image Segmentation for 3-D Object Recognition*, Master's thesis, University of Pennsylvania, Philadelphia, 1988, Technical Report : MS-CIS-88-32.
- [Gupta 1989] Gupta, Alok, *Part Description and Segmentation Using Contour, Surface, and Volumetric Primitives.*, Technical Report MS-CIS-89-33, CIS Department, University of Pennsylvania, 1989.
- [Gupta and Bajcsy 1990] Gupta, Alok and Ruzena Bajcsy, Part description and segmentation using contour, surface and volumetric primitives, in *Proceedings of the Conference on Sensing and Reconstruction of 3D Objects and Scenes*, pp. 203–214, SPIE, Santa Clara, CA. Feb 1990.
- [Gupta et al. 1989a] Gupta, Alok, Luca Bogoni, and Ruzena Bajcsy, Quantitative and qualitative measures for the evaluation of the superquadric models, in *Proceedings of the IEEE Workshop on Interpretation of 3D Scenes*, pp. 162–169, Austin, TX, November 1989a.
- [Gupta et al. 1989b] Gupta, Alok, Gareth Funka-Lea, and Kwangyeon Wohn, Segmentation, modeling and classification of the compact objects in a pile, in *Proceedings of the Conference on Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, pp. 98–108, Philadelphia, November 1989b.
- [Gupta et al. To appear] Gupta, Alok, Gareth Funka-Lea, and Kwangyeon Wohn, Segmentation, modeling and classification of the compact objects in a pile, *International Journal of Research and Engineering: Postal Applications*, To appear.
- [Hall et al. 1982] Hall, E. L., J. B. K. Tio, C. A. McPherson, and Firooz A. Sadjadi, Measuring curved surfaces for robot vision, *Computer*, 42–54, December 1982.
- [Hoffman and Jain 1987] Hoffman, R. and A. K. Jain, Segmentation and classification of range images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9, 1987.
- [Hoffman and Richards 1985] Hoffman, D.D. and W.A. Richards, Parts of recognition, *Cognition*, 18, 65–96, 1985.

- [Horikoshi and Kasahara 1990] Horikoshi, T. and H. Kasahara, 3-d shape indexing language, in *Proceedings of the International Conference on Computers and Communications*, IEEE, Scottsdale, AZ, March 21-23 1990.
- [Horowitz and Pavlidis 1974] Horowitz, S. L. and T. Pavlidis, Picture segmentation by a directed split-and-merge procedure, in *Proceedings of 2nd International Joint Conference in Pattern Recognition*, pp. 424-433, 1974.
- [Kapur and Mundy 1989] Kapur, Deepak and Joseph L. Mundy, *Geometric Reasoning*, MIT Press, 1989.
- [Klingenberg 1978] Klingenberg, W., *A course in Differential Geometry, Graduate Texts in Mathematics*, Springer-Verlag, 1978.
- [Koenderink and vanDoorn 1979] Koenderink, J. J. and A. J. van Doorn, The internal representation of solid shape with respect to vision, *Biological Cybernetics*, 32, 211-216, 1979.
- [Koenderink and vanDoorn 1982] Koenderink, J. and A. van Doorn, The shape of objects and the way contours end, *Perception*, 11, 129-137, 1982.
- [Korn and Korn 1961] Korn, G. A. and T. M. Korn, *Mathematical Handbook for Scientists and Engineers*, McGraw Hill, New York, 1961, Chapter: Solid Analytical Geometry.
- [Kuan and Drazovich 1985] Kuan, D. and R. Drazovich, Model-based interpretation of range imagery. in *Proceedings of AAAI Conference*, pp. 210-215, 1985.
- [Leclerc 1989] Leclerc, Y. G., Constructing simple stable descriptions for image partitioning, *International Journal of Computer Vision*, 3, 73-102, 1989.
- [Leonardis et al. 1990a] Leonardis, Ales, Alok Gupta, and Ruzena Bajcsy, Segmentation as the search for the best description of the image in terms of primitives, in *Proceedings of the Third International Conference on Computer Vision*, pp. 121-125, IEEE, Osaka, Japan, December 1990a.
- [Leonardis et al. 1990b] Leonardis, Ales, Alok Gupta, and Ruzena Bajcsy, *Segmentation of Range Images as the Search for the Best Description of the Scene in Terms of Geometric Primitives*, Technical Report MS-CIS-90-30, CIS Department, University of Pennsylvania, 1990b.

-
- [Levin 1976] Levin, Joshua, A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces, *Communications of the ACM*, 19, 555–563, October 1976.
- [Levin 1979] Levin, Joshua Zev, Mathematical models for determining the intersections of quadric surfaces, *Computer Graphics and Image Processing*, 11, 73–87, 1979.
- [Liang and Todhunter 1990] Liang, P. and J. S. Todhunter, Representation and recognition of surface shapes in range images: a differential geometry approach, *Computer Vision, Graphics, and Image Processing*, 52, October 1990.
- [Luh and Krolak 1965] Luh, J. Y. S. and R. J. Krolak, A mathematical model for mechanical part description, *Communications of the ACM*, 8, 125–129, February 1965.
- [Mahl 1972] Mahl, Robert, Visible surface algorithms for quadric patches, *IEEE Transactions on Computers*, C-21, 1–4, January 1972.
- [Marr 1982] Marr, D., *Vision*, Freeman, San Francisco, 1982.
- [Mumford and Shah 1985] Mumford, D. and J. Shah, Boundary detection by minimizing functionals, i., in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22–26, San Francisco, CA, 1985.
- [Nackman 1984] Nackman, Lee R., Two-dimensional critical point configuration graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6, 442–449, July 1984.
- [Nackman and Pizer 1985] Nackman, L. R. and S. M. Pizer, Three-dimensional shape description using the symmetric axis transform i: theory, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7, 187–202, March 1985.
- [Nevatia and Binford 1977] Nevatia, R. and T.O. Binford, Description and recognition of complex-curved objects, *Artificial Intelligence*, 8, 77–98, 1977.
- [Pavlidis 1977] Pavlidis, Theo, *Structural Pattern Recognition*, Springer-Verlag, Heidelberg, 1977.
- [Pentland 1986] Pentland, A.P., Perceptual organization and the representation of natural form, *Artificial Intelligence*, 28, 293–331, Feb 1986.

- [Pentland 1987a] Pentland, Alex, *The Parts of Perception*, Technical Report CSLI-87-77, Center for the Study of Language and Information, CSLI/Stanford, Ventura Hall, Stanford, CA 94305, 1987a.
- [Pentland 1987b] Pentland, Alex P., Recognition by parts, in *International Conference on Computer Vision*, pp. 612-620, London, 1987b.
- [Pentland 1988] Pentland, Alex, *Automatic Extraction of Deformable Part Models*, Technical Report 104, MIT Media Lab, July 1988.
- [Pentland 1990] Pentland, A. P., Automatic extraction of deformable part models, *International Journal of Computer Vision*, 4, 107-126, 1990.
- [Poggio et al. 1985] Poggio, T., V. Torre, and C. Koch, Computational vision and regularization theory, *Nature*, 317, 1985.
- [Ponce and Brady 1984] Ponce, J. and Michael Brady, Toward a surface primal sketch, in *Proc. of Conf. on Robotics and Automation*, pp. 420-425, March 1984.
- [Press et al. 1988] Press, William H., Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.
- [Rao 1988] Rao, Kashipati, *Shape Description from Sparse and Imperfect Data*, PhD thesis, University of Southern California, December 1988, Technical report IRIS 250.
- [Requicha 1980] Requicha, Aristides A. G., Representations for rigid solids: theory, methods, and systems, *ACM Computing Surveys*, 12, Dec 1980.
- [Rosch 1978] Rosch, E., Principles of categorization, in *Cognition and Categorization*, edited by E. Rosch and B. Lloyd. Erlbaum, Hillsdale, NJ, 1978.
- [Sander and Zucker 1988] Sander, P. T. 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.
- [Seber 1977] Seber, G. A. F., *Linear Regression Analysis*, Wiley, 1977, Chapter 11.
- [Shapiro 1980] Shapiro, L. G., A structural model of shape, *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-2, 111-126, March 1980.

- [Smith and Kanade 1985] Smith, D. R. and Takeo Kanade, Autonomous scene description with range imagery, *Computer Vision, Graphics and Image Processing*, 31, 322–334, 1985.
- [Solina 1987] Solina, Franc, *Shape Recovery and Segmentation with Deformable Part Models*, PhD thesis, University of Pennsylvania, 1987, Technical Report MS-CIS-87-111.
- [Solina and Bajcsy 1990] Solina, F. and R. Bajcsy, Recovery of parametric models from range images: the case for superquadrics with global deformations, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12, 131–147, February 1990.
- [Soroka and Bajcsy 1978] Soroka, B. and R. Bajcsy, A program for describing complex three-dimensional objects using generalized cylinders as primitives, in *Proc. of the IEEE Conference on Pattern Recognition and Image Processing*, pp. 331–339, 1978.
- [Terzopoulos 1986] Terzopoulos, D., Regularization of inverse visual problems involving discontinuities, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8, 1986.
- [Terzopoulos and Metexas 1990] Terzopoulos, Demetri and Dimitri Metexas, Dynamic 3d models with local and global deformations: deformable superquadrics, in *Proceedings of the Third International Conference on Computer Vision*, pp. 606–615, IEEE, Osaka, Japan, December 1990.
- [Terzopoulos et al. 1988] Terzopoulos, D., A. Witkin, and M. Kass, Constraints on deformable models: recovering 3d shape and nonrigid motion, *Artificial Intelligence*, 91–123, August 1988.
- [Tversky and Hemenway 1984] Tversky, B. and K. Hemenway, Objects, parts, and categories, *Journal of Experimental Psychology:General*, 113, 169–193, 1984.
- [Whaite and Ferrie 1990] Whaite, P. and F.P. Ferrie, From uncertainty to visual exploration, in *Proceedings of the Third International Conference on Computer Vision*, pp. 690–697, IEEE, Osaka, Japan, December 1990.
- [Witkin and Tenenbaum 1983] Witkin, Andrew P and Jay M. Tenenbaum, On the role of structure in vision, in *Human and Machine Vision*, edited by J. Beck, B. Hope, and A. Rosenfeld, Academic Press, Orlando, FL, 1983.
- [Woodwark 1989] Woodwark, J., *Geometric Reasoning*, Oxford Science Publications, 1989.

-
- [Zucker 1976] Zucker, S. W., Region growing: childhood and adolescence, *Computer Graphics and Image Processing*, 5, 382–399, 1976.