



This is to certify that the

dissertation entitled

OBTAINING GENERIC PARTS FROM RANGE IMAGES USING A MULTI-VIEW REPRESENTATION

presented by

Narayan Sriranga Raja

has been accepted towards fulfillment of the requirements for

Ph. D. degree in Computer Science

And kumar Jos

Major professor

Date______ 14, 1992

-

MSU is an Affirmative Action/Equal Opportunity Institution



PLACE IN RETURN BOX to remove this checkout from your record. TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
	·	
·		

MSU Is An Affirmative Action/Equal Opportunity Institution ctckc/datedue.pm3-p.1

OBTAINING GENERIC PARTS FROM RANGE IMAGES USING A MULTI-VIEW REPRESENTATION

By

Narayan Sriranga Raja

A DISSERTATION

Submitted to Michigan State University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Computer Science Department

1992

ABSTRACT

OBTAINING GENERIC PARTS FROM RANGE IMAGES USING A MULTI-VIEW REPRESENTATION

By

Narayan Sriranga Raja

We describe a system for obtaining a "generic" parts-based 3-D object representation. We use range image data as the input, obtaining a 3-D object representation based on twelve geon-like 3-D part primitives as the output. The 3-D parts-based representation consists of parts detected in the image, and their identities. Unlike previous work, we do not make simplifying assumptions such as the availability of perfect line drawings, perfect segmentation, or manual segmentation.

We propose a novel method of specifying "generic" 3-D parts, i.e., by means of Surface Adjacency Graphs (SAGs). Using the SAGs, we derive an extremely compact multi-view representation of the part primitives, consisting of a total of only 74 views for all 12 primitives. Based on the multi-view representation of parts, we present a method of performing part segmentation from range images, given a good surface segmentation. This method for part segmentation is more general than common approaches based on Hoffman and Richards' "principle of transversality." We present two approaches for identifying the parts as one of the 12 3-D part primitives. The first approach applies statistical pattern classification methods using parameters estimated by superquadric fitting. Five features derived from the estimated superquadric parameters are used to distinguish between the 12 part primitives. Classification error rates are estimated for k-nearest-neighbor and binary tree classifiers, for real as well as synthetic range images. The second approach for part identification draws inferences from the distribution of angles between surface normals and the principal axis of a part.

We show that intensity data can be used to recover from some misclassifications yielded by the purely range-based methods of part identification. A simple test is applied to check the concavity or convexity of the part silhouette in the intensity image. This serves as a reliable test of whether the part axis is straight or curved.

Results of part segmentation and identification are presented for real range images of several multi-part objects. Our system successfully performs part segmentation and identifies the parts. Copyright © by

Narayan Sriranga Raja

1992

To my Parents, Grandparents, and Great-aunt, and to Min-Sun and Meera

ACKNOWLEDGMENTS

Many people contributed in many ways to make my Ph.D. thesis possible. Although such a brief mention is inadequate, I gratefully acknowledge the contributions of the following people.

Above all, I am grateful to my adviser, Dr. Anil Jain: for being patient when I was choosing a problem to work on, but also for being impatient when it was time for me to move on. He provided useful references and insightful suggestions almost every week, supported me as a Research Assistant for 4 years, and by appointing me a PRIP manager, made it possible for me to enjoy unlimited amounts of computer hacking.

I am grateful to my parents, Smt. Lakshmi Narayan and Shri S. Narayan, to my grandparents, and to my great-aunt Smt. Janakiammal, for bringing me up in such a way that I was eventually interested and able to embark upon a Ph.D. program. For the same reason, I am also grateful to my teachers at school and college in India.

Min-Sun provided invaluable moral support and sympathy for several years, and my sister Meera boosted my confidence by taking it for granted—mistakenly, at times—that I was making steady progress and would soon be a Dr. Raja. My uncle, Shri S. Krishnan contributed to my intellectual growth by keeping up a regular supply of stimulating books.

Besides Dr. Jain, I would also like to acknowledge the contributions of my other committee members—Dr. Koul, Dr. Stockman, Dr. Tüceryan, and my favourite faculty member, Dr. Enbody. It was Dr. Tüceryan, with his AI class, who got me interested in Computer Vision in the first place. I owe him special thanks. I am grateful to Dr. Stockman for conducting an excellent course on 3-D object recognition.

Every past and present member of the Pattern Recognition and Image Processing Group (PRIP) deserves my gratitude. Each in their own way, they made the PRIP lab an ideal place to work. M. D. Ramaswamy encouraged me to join the PRIP group. Pat Flynn, Joe Miller, Debby Trytten, Sally Howden, and Farshid Farrokhnia tolerated my initial bumbling and taught me how to use the lab equipment. Our full-time managers, Dave Marks and John Lees, and all the various student managers ensured that the show would go on. Jinlong Chen, Jon Courtney, Chitra Dorai, Marie-Pierre Dubuisson, Hans Dulimarta, Qian Huang, Tim Newman, Philippe Ohanian, Philippe Ballard, Sushil Bhattacharjee, Steve Walsh, Sharat Pankanti, Jian-Chang Mao, and Yao Chen were helpful friends and knowledgeable colleagues.

Greg Lee, Satisha Nadabar, Debby Trytten, and I, having taken our comprehensive examinations together, came to be known as the Gang-of-Four. Ours will be a lifelong friendship. Besides occasional technical chat, we kept one another sane by listening sympathetically to each other's problems at our weekly Gang-of-Four lunches. Towards the end, our conversations regularly turned to topics such as suicide, murder, and cannibalism! Nevertheless, we bounced back soon after the Ph.D. defence and are already planning the First International Gang-of-Four Conference.

In my work, I used several pieces of software developed by others. Pat Flynn wrote dozens of useful HIPS and range image utilities, including a surface segmentation procedure based on earlier work by Rick Hoffman. Honda Shing developed the MSU Ph.D. thesis style file in IAT_EX . I am very grateful to Prof. Ruzena Bajcsy, Franc Solina, Alok Gupta, Helen Anderson, Luca Bogoni, and others from the GRASP Lab at the University of Pennsylvania, for making available their software for superquadric fitting. Farshid Farrokhnia and Suryana Setiawan adapted and improved the software for fitting a polygonal approximation, originally written by myself and Farshid.

Carol Case, Cathy Davison, and Lora Mae Higbee took care of all kinds of necessary administrative work. I am grateful to them for helping me on innumerable occasions.

If I have forgotten to include anyone who should have been mentioned, I sincerely apologize. I hereby acknowledge their help.

TABLE OF CONTENTS

LI	LIST OF TABLES xi		
LI	ST C	OF FIGURES	xii
1	Intr	oduction	1
	1.1	Object Recognition	2
	1.2	Object Representation and Matching	3
	1.3	3-D Parts-based Representation	6
	1.4	Range Images	11
	1.5	Contributions of this Thesis	12
	1.6	Organization of this Thesis	14
	1.7	Summary	15
2	3-D	Object Representation	16
	2.1	Introduction	16
	2.2	Viewpoint-independent vs. Viewpoint-dependent Representations	19
	2.3	Object-intrinsic vs. Parts-based Representations	20
	2.4	Examples of Object-intrinsic Representations	23
		2.4.1 Feature-based Approaches	23
		2.4.2 Aspect Graphs	24
	2.5	Examples of Parts-based Representations	28
		2.5.1 Generalized Cylinders or Generalized Cones	29
		2.5.2 "Geons" and Recognition By Components	32
		2.5.3 Superquadrics	37
		2.5.4 Other Part Primitives	42
	2.6	Summary	43
3	Par	ts and Part Segmentation	44
	3.1	Choice of Part Primitives	44
		3.1.1 Notation	46
	3.2	Surface Adjacency Graphs	47

	3.2.1	Surface Classification by Curvatures	0
	3.2.2	SAGs of Part Primitives 5	j1
3.3	Multi-v	view Representation of Parts 5	i7
	3.3.1	Catalog of Part Views 5	57
3.4	Part Se	egmentation from Surface Segmentation 6	i 8
	3.4.1	Surface segmentation and classification	'1
	3.4.2	Using the Catalog of Part Views	'2
	3.4.3	Part Identification: Two Approaches	6
3.5	Summa	ary	7
Geo	ons from	n Superquadrics 7	'8
4.1	Motiva	ution	19
4.2	Relatin	ng Superquadrics and Geons	31
4.3	Experi	ments in Shape Discrimination	33
	4.3.1	Choice of Features	33
	4.3.2	Design of Experiments	34
4.4	Experi	mental Results)2
	4.4.1	Parameter Estimation)2
	4.4.2	Part Discrimination)7
	4.4.3	Confusion Matrices)2
4.5	Summa	ary)4
Dir	ect Tes	ts of Shape Attributes 10	8
5.1	Variati	on of Cross-section Size)9
	5.1.1	Principal Axis Orientation	4
5.2	Axis –	straight or bent? 11	6
5.3	Experi	mental Results	22
5.4	Summa	ary	25
Obt	aining	Parts-based Representation 12	:7
6.1	Test In	$nage 1 \dots $	29
	6.1.1	Part Segmentation	29
	6.1.2	Part Identification	29
6.2	Test In	nage $2 \dots $	31
	6.2.1	Part Segmentation	31
	6.2.2	Part Identification	32
6.3	Test In	nage $3 \ldots 13$	32
6.4	Test In	$nage 4 \ldots 13$	<u>.</u>
	 3.3 3.4 3.5 Geo 4.1 4.2 4.3 4.4 4.5 Dir 5.1 5.2 5.3 5.4 Obt 6.1 6.2 6.3 6.4 	3.2.1 3.2.2 3.3 Multi-variation 3.3.1 3.4 Part Social 3.4.1 3.4.2 3.4.3 3.5 Summary Geoms from 4.1 Motiva 4.2 Relatin 4.3 Experi 4.3.1 4.3.2 4.4 Experi 4.4.1 4.4.2 4.4.1 4.4.2 4.4.1 4.4.2 4.4.3 4.5 Summary Direct Tes 5.1 Variation 5.1.1 5.2 5.2 Axis - 5.3 Experi 5.4 Summary Obtaining 6.1 6.1 Test Ir 6.2 Test Ir 6.2 Test Ir 6.4 Test Ir	3.2.1 Surface Classification by Curvatures 5 3.2.2 SAGs of Part Primitives 5 3.3 Multi-view Representation of Parts 5 3.3.1 Catalog of Part Views 5 3.4 Part Segmentation from Surface Segmentation 6 3.4.1 Surface segmentation and classification 7 3.4.2 Using the Catalog of Part Views 7 3.4.3 Part Identification: Two Approaches 7 3.4.3 Part Identification: Two Approaches 7 3.5 Summary 7 Geons from Superquadrics 7 4.1 Motivation 7 4.2 Relating Superquadrics and Geons 8 4.3 Experiments in Shape Discrimination 8 4.3.1 Choice of Features 8 4.3.2 Design of Experiments 8 4.4 Experimental Results 9 4.4.2 Part Discrimination 9 4.4.3 Confusion Matrices 10 5.1 Variation of Cross-section Size 10 5.1 Variation of Cross-section Size

	6.5	Test Image 5	134
	6.6	Test Image 6	134
	6.7	Test Image 7	138
	6.8	Test Image 8	138
	6.9	Test Image 9	138
	6.10	Test Image 10	142
	6.11	Test Image 11	142
	6.12	Test Image 12	1 42
	6.13	Summary	146
7	Con	clusions and Future Research	148
	7.1	Contributions of the Thesis	149
	7.2	Shortcomings	1 50
	7.3	Recommendations for Further Work	151
A	Para	ameters	154
	A.1	Surface Classification	154
	A.2	Constructing the SAG	154
	A.3	Tree Classifier	155
	A.4	Direct test for Cross Section size	155
	A.5	Polygonal Approximation	155
	A.6	Direct test for Axis shape	155
BI	BLI	OGRAPHY	157

LIST OF TABLES

3.1	Notation for specifying the 12 parts	47
3.2	Curvature-based classification into 5 surface types	51
3.3	Some inferences of part shape based on surface type	72
4.1	Mean and standard deviation of estimates of ϵ_1 for the 12 parts \ldots	96
4.2	Mean and standard deviation of estimates of ϵ_2 for the 12 parts \ldots	97
4.3	Mean and standard deviation of estimates of k_x for the 12 parts	98
4.4	Mean and standard deviation of estimates of k_y for the 12 parts	9 8
4.5	Mean and standard deviation of estimates of r/a_z for the 12	
	parts	99
4.6	Overall classification error rates (in %) for synthetic range data	100
4.7	Classification error rates (in $\%$) by range image spatial resolution	100
4.8	Classification error rates (in %) by model "elongation"	101
4.9	12-class error rates (in $\%$) for tree classifier	101
4.10	Overall classification error rates (in %) for real range data from "rough"	
	models	102
4.11	Overall classification error rates (in %) for real range data from	
	"smooth" models	102
4.12	Confusion table for the synthetic images	103
4.13	Confusion table for real images ("rough" models)	105
4.14	Confusion table for real images ("smooth" models)	106
5.1	Confusion table for real images ("smooth" models)	124

LIST OF FIGURES

1.1	Robust system for computing a 3-D parts-based representation	8
1.2	One module of the robust 3-D parts-based representation system	9
1.3	Proposed system for obtaining a 3-D parts-based representation	10
2.1	Aspect graph of a Tetrahedron	25
2.2	Four attributes yield 36 different geons	34
2.3	Non-accidental properties indicate geon type	34
2.4	Complex objects represented using geons	36
2.5	Effect of the parameter ϵ on superquadric shape $\ldots \ldots \ldots \ldots$	39
3.1	Parts with straight edges and constant cross-section	47
3.2	Parts with straight edges and increasing cross-section	48
3.3	Parts with straight edges and incrand-decr. cross-section	48
3.4	Parts with curved edges and constant cross-section	48
3.5	Parts with curved edges and increasing cross-section	48
3.6	Parts with curved edges and incrand-decr. cross-section	49
3.7	Coding scheme for SAG edge attribute	52
3.8	SAG for the part s-s-co	53
3.9	SAG for the part b -s-co	53
3.10	SAG for the part s - s - t	54
3.11	SAG for the part b -s-t	54
3.12	SAG for the part s-s-id	55
3.13	SAG for the part b-s-id	55
3.14	SAG for the part s-c-co	55
3 .15	SAG for the part b - c - co	56
3.16	SAG for the part s - c - t	56
3.17	SAG for the part b - c - t	56
3 .18	SAG for the part s-c-id	56
3 .19	SAG for the part b-c-id	57
3.20	S-aspects of a rectangular block	61

3.21	S-aspects of an increasing-decreasing rectangular block	62
3.22	S-aspects of a tapering rectangular block	63
3.23	S-aspects of a cylinder	63
3 .24	S-aspects of an increasing-decreasing cylinder	64
3 .25	S-aspects of a tapering cylinder	64
3.26	S-aspects of a curved-axis rectangular block	65
3.27	S-aspects of a curved-axis, increasing-decreasing rectangular block	66
3.28	S-aspects of a curved-axis, tapering rectangular block	67
3.29	S-aspects of a curved-axis cylinder	68
3.30	S-aspects of a curved-axis, increasing-decreasing cylinder	69
3.31	S-aspects of a curved-axis, tapering cylinder	70
3.32	Two parts which meet along a convex crease	74
3.33	Two parts which meet along a "mixed" crease	74
3.34	L-shaped object for which part segmentation fails	75
3.35	Another view of the L-shaped object	76
4 1		06
4.1	Synthetic range images: snapes with straight axes	80 07
4.2	Synthetic range images: snapes with bent axes	87
4.3	Hand-carved geons with "rough" surface finish	90
4.4	Real range images of "rough" hand-carved geons	91
4.3	Real range images of some "smooth" geons	93
5.1	Directions of principal axis and surface normals	110
5.2	Part with constant cross-section	111
5.3	Part with increasing cross-section	112
5.4	Part with increasing-and-decreasing cross-section	113
5.5	Intensity image and silhouette of a straight-axis part	118
5.6	Polygonal approximation of the straight-axis part	119
5.7	Intensity image and silhouette of a bent-axis part	120
5.8	Polygonal approximation of the bent-axis part	121
5.9	Complete system for obtaining parts and their identities	126
6.1	Correspondence between colors and alphabetical labels	128
6.2	Object made up of a cylinder and a rectangular block	130
6.3	Complicated object made up of 5 cylindrical parts	133
6.4	Scene with occlusion, containing three objects	135
6.5	Object made up of two cylindrical parts	136
6.6	Everyday object: a bulb	137
6.7	Funnel made up of a cone and a cylinder	130
		100

6.8	Object made up of a cylinder and a rectangular block	140
6.9	Another view of the funnel	141
6.10	Complicated object made up of 5 cylindrical parts: second view	143
6.11	Complicated object made up of 5 cylindrical parts: third view	144
6.12	View of an object resembling a table-lamp	145
6.13	View of a coffee-mug	147

CHAPTER 1

Introduction

The basic goal of computer vision, posed in its most obvious form, is scene understanding. That is, given a sensed 2-D array of picture elements of different brightnesses of a 3-D scene, infer the *presence*, *identity*, *position*, and *orientation* of various "objects" in it. Animal vision is an existence proof that the vision problem can be solved; as Lowe [60] remarks, "without this proof of feasibility, it is hard to imagine that anyone would even think of attempting to interpret the array of light intensities projected from a scene onto a two-dimensional screen".

Ullman points out [91] that visual object recognition by humans is based on several distinct kinds of reasoning. Small dark blobs flying in an erratic, but characteristic way are recognized as "flies" on the basis of their motion. A door-knob, even one of a completely novel shape, can be recognized from its context and from "world knowledge". Trees can often be distinguished from one another on the basis of their coloring and textural appearance. Nevertheless, like most of the computer vision community, we will restrict ourselves to shape-based 3-D object recognition.

Computer vision research shares some fundamental assumptions with the rest of Artificial Intelligence. First, it assumes that like other perceptual and cognitive capabilities, the visual faculty consists of "computational" processes, in some sense, or can at least be simulated by such processes. Second, it assumes that corresponding computational processes can equally be implemented on non-biological physical media (for example, using cameras and electronic hardware instead of eyes and nervous tissue). Computer vision, in its present form, is doomed to failure (in duplicating human visual capabilities) if these assumptions are invalid.

The overall paradigm which is dominant among students of object recognition, both by humans and by computers, postulates that object recognition is a two-stage process. Object recognition is assumed to proceed by first producing some internal representation of the visual input, and then matching it against representations of models known to the viewer. This is taken for granted by virtually all researchers. Consequently, we can distinguish between, or classify, object recognition systems according to (a) their representation scheme, and (b) their matching scheme.

1.1 Object Recognition

The problem of object recognition is generally equated to the process of finding that representation (of a model) in memory that "best" matches the representation of an object in the scene. Clearly, this can be viewed as an instance of a more general problem, i.e., finding a solution from a solution space. Given the solution space, i.e., the set of representations of all known models, the best solution has to be located. (Under good viewing conditions, humans can identify objects with very little uncertainty so much so, that it makes sense to talk of a "correct" solution versus "incorrect" solutions, rather than just the "best" solution from a continuum of possibilities).

Many other problems in AI have also been formulated in terms of searching or traversing a solution space. Examples are game-playing (e.g., find a "good" next move when playing chess) and problem-solving (e.g., given the symptoms, diagnose the patient's illness(es)). These problems often deal with huge solution spaces and/or search trees. Not surprisingly, therefore, they are hard problems if thus formulated. While we do not know of a better alternative formulation of the object recognition problem at this point, it is worth remembering that the problem of adding 2 and 2 might also be difficult if our addition algorithm searched for a solution in the set of all possible integers.

Algorithms to solve the general problem of "finding a solution from a solution space" can be classified into several broad categories. Since the "matching" stage of object recognition can be viewed as an instance of this more general problem, a natural taxonomy of different "matching strategies" can be obtained by relating them to these broad categories of algorithms.

1.2 Object Representation and Matching

An attempt to roughly quantify the problem of object recognition by search is very useful in pinpointing the exact reason why it is so difficult. At present, computerbased 3-D object recognition systems can cope with a few tens of models at most. For the purpose of argument, suppose there are 1,000 possible models, and they can only appear singly or in pairs in an image. Further, suppose that occlusion is not allowed. The total number of possible combinations of models is, therefore, $C_1^{1000} + C_2^{1000}$, which is approximately 500,000. Object recognition requires searching for the correct solution from among these possibilities. But searching a mere 500,000 candidates sounds rather trivial! For example, even a PC can sequentially search through 500,000 numbers in a few seconds. Why, then, should the object recognition task posed above (1,000 models, unoccluded, either singly or in pairs) be far beyond any object recognition system built so far? Why is searching a set of numbers so easy, and searching a set of model objects so difficult?

There are two reasons for this, and they are closely related. Firstly, consider the task of checking whether a particular candidate solution is correct or incorrect.

If we are searching a set of numbers to find the first instance of, say, the number 7, we can easily determine whether a particular candidate is the correct solution or not. Here, the criterion for "correctness" of a solution is very clearly defined—equality of two numbers—and can be tested very quickly. By contrast, the analogous problem in object recognition, i.e., "verification," is neither easy to define nor to perform. Given some object, suppose we are searching through the solution space of possible models. **Testing** whether a particular candidate is correct or not is itself a difficult task. That is, given some object, the question "Is it model #5, i.e., the cobra head?" itself is a laborious one to answer given current techniques.

The reason is that answering this question could itself involve examining all the possibilities in another search tree! This is the *interpretation tree* [39], which essentially takes each component (e.g., 2-D or 3-D contours, or surface patches) of the object and explores matching it against every component of the model being considered. (Various heuristics are used to prune the interpretation tree, but it is still not a trivial operation). Testing whether a particular candidate solution is correct would be much simpler if every object and model were bar-coded! The computer system in a supermarket has no difficulty in performing its own form of visual object recognition, given the laser-readable unique bar-codes attached to objects.

We cannot expect every object in the universe to be conveniently labeled in such a manner. Nevertheless, this aspect of the object recognition problem has not been viewed before from this perspective. In our opinion, it would be very interesting to investigate and improve methods for checking candidate solutions. Ideally, what is desired is a module which, when asked, say, "is this object a cobra head?" could answer "yes" or "no" very rapidly. If such a module were available, searching the solution space of models for object recognition would be as easy as searching a set of numbers.

Secondly, there is the problem of how to organize the solution space itself.

Undergraduate courses in data structures and algorithms teach us that sorting and *indexing* are invaluable for searching a large set. Searching a set of 500,000 numbers is easy, because a key and an ordering are easily defined. The set of numbers can be organized as a sorted array or as a search tree. Binary search can then explore this data structure in logarithmic time. As usual, unfortunately, we do not have comparably effective methods to organize the set of known models in object recognition.

To recapitulate, virtually all vision researchers assume that object recognition involves producing an internal representation of the scene, and then matching it against internal representations of known model objects. This formulation of object recognition can be seen as an instance of "finding a solution from a solution space."

Now, consider the standard methodologies for the general problem of finding a solution from a solution space. They include exhaustive search (with various flavors or refinements like depth-first search, breadth-first search, backtracking, alpha-beta pruning, branch-and-bound, or dynamic programming), binary search of sorted data, indexing methods, hashing schemes, the use of content-addressable memories, and various probabilistic algorithms. As expected, the common matching schemes for object recognition can be related to these general solution methodologies. Due to the lack of effective sorting and indexing schemes, analogous matching strategies for object recognition are not well-developed. Hypothesize-and-test and interpretation tree methods [40, 51] are versions of exhaustive search. The generalized Hough transform, pose clustering, and geometric hashing [18, 19, 45, 56, 85] can be viewed as a type of parameter-based hashing, combined with voting. Artificial neural networks for recognition are a kind of content-addressable associative memory. With the possible exception of the RANSAC paradigm [32], the use of probabilistic algorithms in machine vision has not been explored.

1.3 3-D Parts-based Representation

We presented a general framework within which various matching strategies for object recognition fit quite naturally. While doing so, we emphasized two major difficulties encountered during the matching step of object recognition, i.e., excessive number of hypotheses which need to be verified, and the difficulty of verification itself. This motivates an intelligent choice of the underlying representation scheme so that these two difficulties can be minimized.

A 3-D parts-based representation is one which models objects in terms of a set of 3-D part primitives together with the relative positions and orientations of the parts. The appeal of 3-D parts-based representations arises in large measure because they offer more sophisticated, yet natural ways to organize and index into the set of object models. *Geons* are a set of qualitatively defined 3-D part primitives [8]. We believe that geon-like volumetric part primitives are an excellent choice for 3-D object representation because they are useful for indexing and have considerable shape discrimination capability. Geons are defined and discussed extensively in Chapter 2.

Although geons were originally proposed in the context of line drawings, their utility for object recognition is not limited to one particular kind of input data. The geon theory can be equally useful for object recognition from range images. Hence the use of dense range image data to detect and identify geons is quite justifiable.

Assuming the use of a 3-D parts-based representation, computing this representation is the crucial first stage of object recognition. Over the last three decades, there has been an enormous amount of research in scene understanding by computer. There has been much success in solving specific problems under highly limiting assumptions. However, almost every system reported in the literature is fragile, i.e., performance degrades sharply if the very specific, often unrealistic, underlying assumptions are violated. In our view, a robust system for computing a 3-D parts-based representation, capable of operating in a general setting (noisy, cluttered, and with possible occlusion), can be reliably achieved only by integrating different kinds of information. This information would include intensity data, imperfect line drawings, "intrinsic array" data such as depth and orientation, as well as hypotheses about line labelings, surface segmentation, part segmentation and part identities.

At any given time, each of these kinds of information might not be complete, but only partial, imperfect, or hypothesized. The robust system (Figure 1.1) might consist of a number of co-operating modules, each of which, given the current hypotheses about all the above kinds of information, would output an improved version or better hypothesis about one specific kind of information. (In Figures 1.1 and 1.2, rectangular boxes represent modules for performing some computation, whereas round and oval boxes represent information).

For example, the part segmentation module might take as input the intensity and range data, current versions of the imperfect line drawing, hypothesized line labeling, hypothesized segmentation into surfaces, hypothesized segmentation into parts, and hypothesized identity of various parts. It would produce as output an improved version of the part segmentation (Figure 1.2). Iterative refinement of the hypotheses would continue till an acceptable 3-D part representation (i.e., segmentation into parts and their identification as specific part primitives) is obtained.

To put this dissertation in perspective, the work described here can be viewed as a partial implementation of two modules—part segmentation and part identification of the idealized, robust system described above for computing a 3-D parts-based representation. Range image data are the only input information used for part segmentation and part identification. There is no ongoing interaction between the two modules, nor are the hypothesized part segmentations or part identifications improved iteratively as in the design proposed in Figures 1.1 and 1.2.

7



Figure 1.1. Robust system for computing a 3-D parts-based representation



Figure 1.2. One module of the robust 3-D parts-based representation system



Figure 1.3. Proposed system for obtaining a 3-D parts-based representation

A block diagram of our work is shown in Figure 1.3. Surface segmentation is performed by the method of Flynn [33], based on the earlier work of Hoffman and Jain [50]. We have designed and implemented our own methods for part segmentation and part identification. These are discussed in subsequent chapters.

1.4 Range Images

Most researchers in computer vision have used and continue to use intensity images as the input. However, since the early '80s there has been an increasing use of range images [13]. With intensity images as the input, considerable research effort has been devoted to various Shape-from-X computations such as shape from shading, shape from texture, shape from stereo, shape from motion, etc. These efforts were aimed at computing various kinds of 2.5 D information, especially the depth or range value at each pixel.

With the development of various range sensors, there is an increasing tendency to assume the availability of this 2.5 D data directly in the form of range images. The advantage of using accurate range data is that it enables researchers to bypass the difficult "lower-level" computations of 3-D depth and orientation values from intensity values, and work directly on 3-D representation and later stages of 3-D object recognition. Disadvantages are that sensors for acquiring high-quality range images tend to be slower and much more expensive than the ordinary CCD cameras used to acquire intensity images.

Another recent development is the use of *fused* range and intensity data for various aspects of computer vision research [57, 65]. *Registration*, i.e., establishing the correspondence of points in two separate images, is the major problem in fusing separately acquired range and intensity data.

Flynn [33] describes 3-D sensing in more detail, while Brady et al. [13] give an

overview of recent research using range imagery. We will give a brief description of our range sensor, i.e., the 100X White Scanner from Technical Arts Corporation [88].

The 100X scanner is a structured-light range sensor. It projects a plane of laser light which illuminates objects placed on the work stage. The shape of the resulting light stripe is determined by the 3-D contour of the object onto which the laser light has been projected. The light stripe is imaged by a CCD camera with a 240 \times 240 sampling array. The stage can be moved past the plane of light by stepper motors for the X and Y axes. An object of known 3-D dimensions is used to calibrate the scanner so that for each point on the stripe of laser light, the X, Y, and Z values can be computed.

Range data acquired by the 100X scanner are estimated to have a noise standard deviation of less than 0.01 inch in the Z value. A range image takes 2 to 3 minutes to acquire, and may be as large as 240 rows by 240 columns. Data are in the form of X, Y, and Z values at each pixel.

For two reasons, we have chosen to use range images as the input to our system for computing a 3-D parts-based representation. First, the problem of obtaining a 3-D parts-based representation in terms of a distinct, fairly large set of geon-like parts from range images has not been explored previously. Second, previous work on obtaining a geon-based 3-D representation has made certain simplifying assumptions such as the use of perfect line drawings, perfect part segmentation, and/or manual segmentation. We hope that the use of range images may enable us to avoid these unrealistic assumptions.

1.5 Contributions of this Thesis

This dissertation describes research on obtaining a "generic" parts-based 3-D object representation. By "generic' we mean that the part primitives are defined by qualita-

tive shape properties and not by exact numerical measures. We use range image data as the input, obtaining a 3-D object representation based on a small number (12) of geon-like 3-D part primitives as the output. The 3-D parts-based representation consists of parts detected in the image, and their identities. Previous work on obtaining a 3-D object representation in terms of geons or geon-like parts [8, 4, 3, 21, 22] assumed perfect line drawings as the input, and/or performed segmentation manually. We do not make any such idealized assumptions. Real range images are used as the input, and a segmentation is computed automatically from the image.

Multiple views of the 3-D part primitives are stored in order to help segment objects into component parts and identify those parts as being specific part primitives. Large, "meaningful" surfaces of the part primitives determine the views to be stored in the *multi-view representation*. Two approaches are described for identifying the parts as one of the 12 3-D part primitives. The first approach applies *statistical pattern classification* methods using parameters estimated by *superquadric* fitting. The second approach draws inferences from the distribution of angles between surface normals and the *principal axis* of a part. *Intensity image data* are used to recover from some shortcomings of the method based only on range data.

The specific contributions of this dissertation are as follows:

- We describe the first attempt (to our knowledge) to obtain, from range data, a
 3-D parts-based object representation in terms of a specific, but fairly large set of 12 geon-like 3-D part primitives.
- We propose a novel method of *specifying* "generic" 3-D parts, i.e., by means of Surface Adjacency Graphs (SAGs). Using the SAGs, we derive an extremely compact multi-view representation of the part primitives.
- Based on the multi-view representation of parts, we present a method of performing part segmentation from range images, given a good surface segmen-

tation. This method for part segmentation is more general than common approaches based on Hoffman and Richards' "principle of transversality" [47], i.e., the observation that parts usually come together with a concave crease at the joint.

• We describe two methods to perform the part identification of isolated parts. In doing so, we attempt to relate two influential families of 3-D part primitives, superquadrics and geons. We test the shape discrimination capability of superquadric parameters under varying conditions of image resolution, model elongation, and noise. Lastly, we show that intensity data can be used to recover from some shortcomings of the purely range-image based method of part identification.

1.6 Organization of this Thesis

In order to motivate and justify our design choices, it is necessary to review the previous research on object representation and related issues. The following section describes the organization of this report.

Chapter 2 classifies and discusses object representations. Chapter 3 describes our choice of 3-D part primitives. It sets forth a very compact multi-view representation of the parts, and explains how this can be used for part segmentation. Chapter 4 presents our first approach towards identification of the parts previously isolated by the segmentation algorithm. It also tests the shape discrimination capability of superquadric parameters under varying conditions of image resolution, model elongation, and noise. Chapter 5 presents a second approach towards part identification, using the multi-view representation as well as the distribution of angles of surface normals with respect to the principal axis of a part. A method for using intensity data to recover from some shortcomings of the purely range-based approach to part identification is also described. Chapter 6 gives the results of part segmentation and part identification for several multi-part objects. Finally Chapter 7 concludes, identifying some unresolved issues and offering suggestions for future research.

1.7 Summary

Object recognition is usually posed as a two-stage problem, i.e., representation followed by matching. Object recognition systems can therefore be meaningfully classified and compared by identifying their representation schemes and matching strategies within an coherent overall framework. Excessive number of hypotheses to be verified, and the complexity of verification itself, are the main difficulties encountered during the matching stage of object recognition. This motivates the choice of powerful geonlike volumetric part primitives for 3-D object representation, in order to minimize these two difficulties.

The research described in this dissertation is aimed at obtaining a parts-based 3-D object representation, using range data as the input. We derive an extremely compact multi-view representation of the part primitives, based on their large, "important" surfaces. We use the multi-view representation to perform part segmentation, given a good surface segmentation. Further, we develop two methods to perform the part identification of isolated parts. We also attempt to relate two influential families of 3-D part primitives, superquadrics and geons. We test the shape discrimination capability of superquadric parameters under varying conditions of image resolution, model elongation, and noise. Lastly, we show that intensity data can be used to recover from some shortcomings of the purely range-based method of part identification.

CHAPTER 2

3-D Object Representation

Besl and Jain [6], Binford [10], Chin and Dyer [17], and Stockman [86] give broad surveys of object recognition systems. Brady et al. [13] discuss the recent work in object recognition using range data. This chapter briefly surveys some influential schemes for 3-D object representation.

2.1 Introduction

Representation schemes can be classified in many different ways, of which two seem most fundamental. First, they may be either viewpoint-independent or viewpointdependent. Second, they may adopt either an object-intrinsic representation, or one which is parts-based. Other important differences are whether a scheme involves geometric modeling of the object, or instead uses some symbolic representation. In the case of representations performing geometric modeling of 3-D objects, one can also distinguish between volumetric (i.e., 3-D) and surface-based (i.e., 2.5-D) shape primitives.

A wide variety of representation schemes is available for 3-D object recognition. The underlying primitives or features for representation include 2-D or 3-D points, such as polyhedral vertices or the center of a sphere; 2-D or 3-D contours such as intensity edges or 3-D jump edges or crease edges; 2.5 D features such as object wings; surface patches of various types such as planar, concave, convex, etc; parametrized surfaces such as quadrics or splines; and ultimately 3-D volumetric primitives such as generalized cylinders, geons, superquadrics, and deformable models.

Assuming that objects have "internal representations", the nature of the mapping between objects and representations is one point of divergence. Viewpointindependent schemes may lead to a one-to-one mapping. Viewpoint-dependent representations can result in a one-to-many relationship between objects and their internal representations. A many-to-one mapping could be produced when individual objects are lumped together into a single category. For instance, we often ignore the unique shape of individual lampshades or shampoo bottles. Indeed, objects like these are often recognized by properties other than shape, such as context or function. The same glass object which looks like a lampshade when attached to the ceiling may appear like a fruit-bowl or flower-vase if placed on a table. Such categories might be represented by a generic, prototypical model of a "typical-shaped" lampshade, shampoo bottle, etc.

It would be quite reasonable to classify representation schemes according to, say, their memory requirements or their degree of psychological plausibility. Two classifications, however, seem most fundamental: whether the representation is viewpointindependent or not, and whether it is "object-intrinsic" or parts-based. These distinctions have immediate implications for other properties such as memory requirements.

The ultimate test of a representation scheme, of course, is its usefulness and effectiveness for object recognition. This being understood, having classified various representation schemes, how can we compare them? Marr and Nishihara [62] propose three criteria for doing so. Firstly, the representation's *accessibility*, i.e., can the desired description be computed from an image, and can it be done relatively inexpensively? The second criterion is the representation's *scope and uniqueness*. What class of shapes can the representation handle, and do the shapes in that class have unique descriptions? For example, a representation intended for polyhedral objects would have difficulty with curved surfaces. Also, to avoid the problem of deciding whether two different representations describe the same shape, it is important that shapes have unique descriptions as far as possible. Thirdly, there is the *stability and sensitivity* of a representation. To be useful for recognition, the similarity between two shapes must be reflected in their descriptions. At the same time it should be possible to represent even subtle differences. For instance, when using stick figures to represent animal shapes, *stability* might be increased by using larger sticks, while smaller sticks would improve *sensitivity* by revealing finer details.

Dickinson ([20], pg. 4) also proposes several criteria along which representation schemes differ. Firstly, there is the *primitive complexity*, ranging from the simplest 2-D points, through 3-D points and contours, to complex 3-D volumetric primitives. Next, *model complexity* generally decreases as the primitive complexity increases, since the same object can either be described using many simple features or a few complex features. Thirdly, the large number of simple features may lead to higher *search complexity*, i.e., the number of hypothesized matches between features in the model and scene representations. Fourthly, simple features have less discriminatory power and may increase the recognition procedure's *reliance on verification*. *Model flexibility*, i.e., sensitivity to changes in shape, is another important criterion. Lastly, there is the issue of *ease of recovery*, i.e., the difficulty or otherwise of computing the representation from an input image.

2.2 Viewpoint-independent vs.

Viewpoint-dependent Representations

Marr and Nishihara [62] list some important choices to be made in the design of a representation scheme. One of these, the choice of viewer-centered or object-centered co-ordinate system, is rather similar to our distinction between viewpoint-independent and viewpoint-dependent schemes. However, the latter is more general because it does not assume that a representation must necessarily include a co-ordinate system.

For instance, some representation schemes may not perform geometric modeling at all. An aspect graph representation cannot be said to use viewer-centered or objectcentered co-ordinates, but still captures a viewpoint-independent topological property. In spite of this distinction, we will sometimes use terms like "viewer-centered" and "object-centered" interchangeably with "viewpoint-dependent" and "viewpointindependent".

Representations which require geometric modeling of objects must have an associated co-ordinate system. Such schemes can be classified as viewer-centered if locations are specified relative to the viewer, or as object-centered if locations are specified in a co-ordinate system defined by the viewed object [62].

Since viewpoint-dependent properties can be obtained directly from an input image, viewer-centered representations may be easier to compute. The advantage is that, unlike in an object-centered representation, one does not have to compensate for the vantage point. However, different views of the same object may have to be treated essentially as distinct objects. This may require a very large number of views of each object—potentially, every possible view of every possible object.

The other major alternative is to use an object-centered representation. Ideally, just one canonical description, independent of the vantage point, would have to be stored for each object. This is the approach favored by Marr and Nishihara [62] and
many others. The disadvantage is that the representation can be difficult to compute from the image. Not only does a unique co-ordinate system have to be **defined** for each object, but this has to be identified from the image before constructing the description. This can be a problem when the object has several plausible natural axes. Further, the effects of perspective projection have to be corrected for.

Other shape representation schemes have also been explored by cognitive psychologists [46]. For instance, objects might have a single viewpoint-dependent representation corresponding to a "canonical upright" position. A "mental rotation" operation would transform an input shape into the canonical orientation before memory and input are compared. Experiments suggest that shapes may be represented by a limited number of viewer-centered descriptions, corresponding to a variety of orientations at which the object is likely to be observed. Work by Hinton and Parsons [46] indicates that people may use *scene-based* representations in some shape comparison tasks. These are descriptions relative to some larger salient object in the scene, such as a room, table-top, blackboard, or page. When subjects were asked to decide whether two objects (at different lines of sight from the viewer) were identical or mirror images, they often physically rotated one of the objects till it had the same relationship to the table-top (and room) as the other object.

2.3 Object-intrinsic vs. Parts-based Representations

This distinction has to do with the "unitary-ness" and organization of the information in a representation scheme. It is closely related to the distinction between *statistical* and *structural* approaches in pattern recognition. Ullman [91] makes a very similar distinction between *invariant properties* methods and *parts-decomposition* methods for object recognition. An *object-intrinsic* representation consists of some overall invariant features or properties characteristic of the object as a whole. It does not model an object as a compound entity consisting of its "parts" arranged in some spatial configuration. Examples of object-intrinsic representations range from simplistic schemes like template-matching, through spatial Fourier-domain descriptions or representation by a suitable set of features. We would even classify a fairly recent and sophisticated approach like aspect graphs as an object-intrinsic representation scheme.

A "parts-based" representation, on the other hand, considers an object as having one or more meaningful parts which are put together in a particular spatial configuration, and which may possibly have further detailed descriptions of their own. This is quite similar to structural description theories of shape recognition. A structural description is a data structure which can be thought of as a list of propositions, whose arguments correspond to parts of the object, and whose predicates correspond to properties of the parts and spatial relationships among them [63, 73]. A structural description is a more general extension of a parts-based description, since it allows for representation of non-visual and non-spatial information as well.

Marr and Nishihara [62] list the organization of shape information about an object as one of the important design choices for a representation. In the simplest case, all the elements in a description have the same status, and no organization is imposed. Alternatively, primitive elements of the description could be organized into modules, in order to distinguish certain subgroupings of the primitives. Thus they can be referenced efficiently, and properties can also be associated with them. A modular organization might be particularly useful for recognition because it can make "stability" and "sensitivity" differences explicit, by arranging that all constituents of a given module lie at roughly the same level of stability and sensitivity [62].

The explicit representation of spatial relationships of parts, i.e., geometric modeling in some sense, is a key aspect of parts-based descriptions. This distinguishes them from "unitary" feature-based descriptions which may very well be based on geometrical parts (e.g., holes, corners), without modeling the spatial relationships between those features themselves. Thus, it appears that parts-based descriptions must necessarily have (one or more) associated co-ordinate systems.

By representing different parts of an object as separate modules, we may be able to break up the recognition process into simpler subprocesses [73]. We may, for instance, merely need to be able to visually recognize a small number of part and attribute primitives. Additionally, if logical and spatial predicates can coexist in the representation, structural descriptions can differentiate between parts that must be present in an object, parts that may be present with some probability, and parts that must not be present. Another advantage is that some visual information in a structural description may be useful for non-visual reasoning. Conversely, non-visual information about objects or parts (i.e., their uses, typical situations where they are found) may be used to aid segmentation and recognition in a top-down manner.

It is, indeed, generally agreed that a parts-based approach seems more natural for realistic, complicated scenes. One of the main problems with this representation, however, is the lack of a firm basis on which to decide a set of primitives to be used, and to justify why they are necessary, sufficient, or appropriate. This is even more true for the predicates (properties) than for the part primitives. The very generality of this representation lends it a certain vagueness. Another crucial issue is the **accessibility** of all the primitives (in the sense mentioned by Marr and Nishihara [62]) from the input image. Problems of uniqueness of the description, and of **part segmentation** also naturally arise.

2.4 Examples of Object-intrinsic Representations

Among well-known examples of object-intrinsic representations, *Templates* and *Fourier representations* are traditional 2-D shape representations described in cognitive psychology textbooks [73]. *Feature-based approaches* are more commonly used for 2-D than for 3-D object representation. *Aspect graphs* are a relatively new concept and have attracted considerable interest over the last few years.

2.4.1 Feature-based Approaches

An object is represented by a list or vector of numerical and symbolic values, the *features* [73, 86]. Typical numeric features might be the object's perimeter, moments of area, or the number of holes or right-angled corners in it. One can also use symbolic-valued features like the color of the object, or test for the presence of specific "salient" features (say, a 60° corner formed by a circular arc and a straight line segment). The representation is simply an aggregate of such features. There is no attempt to describe the spatial relationships (e.g., relative locations) between different geometric features.

Given the feature representation of an object, a variety of standard statistical pattern recognition methods [23, 35] are available to classify it into a known category. For instance, one might use discriminant functions, a k-nearest-neighbor (knn) classifier, or a hierarchical (tree) classifier. Discriminant functions define decision surfaces, bounding different regions in the feature space. A knn classifier makes its decision based on the categories of the unknown object's k nearest neighbors in the feature space. The object is assumed to belong to the same category as the majority of its k nearest neighbors. With an infinite number of training samples, it is known that the knn rule approaches the optimal Bayes decision rule as the number k of nearest neighbors to be considered tends to infinity. A hierarchical classifier is basically a decision tree. It may be able to make decisions and arrive at a classification, based

on just a subset of the features. This could be particularly useful when some of the features are difficult or impossible to compute (e.g., due to occlusion).

The advantages of a feature representation are its extreme compactness as compared to other representations, and the existence of the large body of statistical pattern recognition theory which can be used to perform the classification and estimate the classification error rate. Further, the classifier can be "trained" from examples. It may even be possible to use clustering techniques and discover distinct categories of objects which are naturally separated in the feature space [52]. A "reject" option can easily be provided to take care of ambiguous cases.

The disadvantages of this representation are, firstly, that the object cannot be recreated or modeled accurately from the features. Secondly, it is difficult to establish a suitable set of features for general 3-D objects. Thirdly, it may be difficult to compute the feature representation of an object from a single 3-D view, or when multiple objects with occlusion are present. (A Viewsphere or property sphere approach can extend the applicability of feature representations to 3-D objects. A large number (e.g., 200 or more) of covering viewpoints is selected, and the feature vector is computed for the projection onto 2-D from each viewpoint [24, 54]). Fourthly, an object might be confused with a totally different object which has the same features, but arranged differently—e.g., an unsymmetrical object and its mirror image. Lastly, a feature representation achieves compactness at the cost of discarding a lot of potentially useful information about the object.

2.4.2 Aspect Graphs

Aspect graphs are an example of the multiple-view approach to 3-D object representation. The aspect graph concept was first proposed by Koenderink and van Doorn [53] as a possible mechanism involved in human vision. They described a graph structure called the "visual potential" of an object. Each node of this graph represents a



Figure 2.1. Aspect graph of a Tetrahedron (based on [53])

different *aspect* of the object. An aspect is a "stable view" of the object as seen from some maximal connected region or cell of the viewpoint space. (It could be thought of as a specific collection of faces and edges visible from any viewpoint within that cell). Nodes are connected by arcs indicating "visual events" where the observer transits from one such cell to another and the aspect of the object changes. "Thus the visual potential represents in a concise way any visual information an observer can obtain by looking at the object when traversing any orbit through space" [53], (pp. 214). For example, Figure 2.1 shows the aspect graph of a tetrahedron.

Callahan and Weiss [15] have proposed an object representation called the *viewing* data or viewing graph, which is closely related to the aspect graph. The viewing data of an object is the partition of the viewing sphere (corresponding to different aspects), together with the view from each region, edge, and vertex of the partitioned sphere.

The aspect graph of an object could very well, for example, be combined with an exact 3-D model of the object, thus getting the completeness of a viewpointindependent representation as well as the simplicity of a viewpoint-dependent one. It seems, therefore, that aspect graphs are potentially very useful for 3-D object recognition. A major problem is that complicated objects can have aspect graphs with millions of nodes. Appropriate heuristics have to be established to arrive at reasonably "stable" aspect graphs of manageable size. Following the original paper by Koenderink and van Doorn [53], aspect graphs have in fact been the subject of very active research [12, 15, 16, 27, 29, 37, 38, 44, 51, 54, 55, 74, 82, 83, 84, 93].

Computing the aspect graph for an arbitrary object is itself an unsolved problem. Much work has been devoted to solving this problem for progressively more general classes of objects, starting with convex polyhedra. These efforts can be classified according to the way they partition the viewpoint space.

A basic distinction in computing an aspect graph is whether an orthographic or perspective projection is assumed. If orthographic projection is assumed, then the "cell" of viewing space associated with a node of the aspect graph corresponds to a surface area on a unit sphere [27, 37, 38, 44, 51, 54, 55, 74, 82]. The advantage is that an approximate aspect graph can be computed by uniformly tessellating the viewing sphere and then merging facets which see the same aspect. However, this orthographic projection aspect graph only captures changes in the aspect due to changes in the viewing orientation. Relatively less progress has been made in computing the more general perspective projection aspect graph [26, 74, 83, 84, 93], in which cells correspond to volumes of the viewing space. It should be noted that some of these algorithms for finding aspect graphs have never been implemented because of computational difficulties.

Gigus et al. [37, 38] have developed an algorithm to find the so-called "viewing data" (closely related to the orthographic projection aspect graph) for line drawings of arbitrary polyhedral objects. They also provide a full catalog of the visual events that can occur for such objects. Eggert and Bowyer [27] and Kriegman and Ponce [55] have independently developed algorithms to find the orthographic projection aspect graph for solids of revolution.

Rieger [78] cataloged some of the visual events which can occur for an object consisting of smooth surfaces which meet in smooth curves (i.e., piecewise smooth, curved objects). Sripradisvarakul and Jain [82] have extended this catalog and presented an algorithm to find the aspect graph under orthographic projection for arbitrary piecewise smooth, curved objects.

Stewman and Bowyer [84], Watts [93], and Edelsbrunner et al. [26] independently developed methods to compute the perspective projection aspect graph for convex polyhedral objects. Stewman and Bowyer [83] further extended this for arbitrary polyhedra.

A number of researchers have used methods based on or similar to aspect graphs to build object recognition systems. Perhaps the earliest such work is that of Dudani et al. [25], who store rotation- and scale-invariant moment features from several different viewpoints to perform identification of aircraft types. Chakravarty and Freeman [16] use characteristic views to represent objects. They use heuristic constraints (on the orientation of the object with respect to the camera) to select a subset of views as the representation. The aspects and the partition of viewing space are done manually. Ikeuchi [51] also uses aspect graphs, with aspects being defined by the faces detectable by photometric stereo. Hebert and Kanade [44] use the aspect graph approach for recognizing polyhedral objects in range images. Aspects are defined by the set of occluding edges in the image. Both Ikeuchi [51] and Hebert and Kanade [44] approximate the exact partition of the viewing sphere by performing a uniform tessellation and then merging neighboring regions which view the same aspect. Fekete and Davis [29] propose the use of "property spheres" which encode "geometrical and topological properties of orthographic images as a function of viewpoint". Viewpoints are obtained by approximating a spherical surface by planar patches and using the center of each patch as a viewpoint. Korn and Dyer [54] tessellate the unit sphere in a manner similar to Fekete and Davis. Facets with a similar view are then merged into larger regions. Bowyer et al. [12] have also implemented an object recognition system for convex polyhedral objects. According to them, aspect graphs can be useful in generating initial estimates and avoiding local minima of rotation and translation parameters in systems which use non-linear optimization as a strategy for recognition.

2.5 Examples of Parts-based Representations

In this section we discuss the best-known schemes for parts-based representation or recognition. Much vision research has been devoted to obtaining 2.5 D information, i.e., local surface characteristics (shape from shading, texture, stereo, etc.) in the form of image arrays. However, reliable and general procedures to compute such image arrays from the local image information are not available. In fact, according to Pentland, "these recovery techniques have been sufficiently unimpressive that many researchers now believe that it is simply not possible to compute accurate, objectcentered representations from arrays of locally derived estimates of surface properties, as was originally envisioned" ([69], pg. 143). Pentland also points out that even if depth maps and other maps of intrinsic properties could be reliably and densely computed, it is still merely an array of numbers which by itself suffices only for simple applications like obstacle avoidance. It must be segmented and interpreted if it is to be used for more sophisticated tasks ([69], pg. 143). This motivates the idea that the initial task of perception is not to produce local descriptions of images, surfaces or volumes, but rather to recognize instances of simple, generic real-world modeling primitives.

If we accept this view, it remains to discover such a set of generic primitives, describe how they combine to form objects, recognize these primitives and their combinations from the image, and use this description to interpret or recognize the image content. Hoffman and Richards propose that the visual system segments objects into constituent parts by detecting part boundaries rather than the parts themselves [48]. This decomposition depends on a regularity of nature called *transversality*, i.e., the observation that when two arbitrarily shaped surfaces interpenetrate, they usually meet in a contour of concave discontinuity of their tangent planes. In other words, part boundaries are marked by a concave crease at the joint. Hoffman and Richards argue that people use this part decomposition rule, along with spatial relations and part descriptions, to perform the first indexing into the database of known objects.

Generalized cylinders or cones, geons, and superquadrics are the best known modeling primitives for 3-D shape description or recognition. They are discussed in the following subsections.

2.5.1 Generalized Cylinders or Generalized Cones

Generalized cones were introduced by Binford [9] and have been used in the ACRONYM system [14] for recognizing airplanes from aerial photographs. Generalized cylinders were proposed by Marr and Nishihara [62] as the modeling primitives in their theory of object-centered hierarchical representation and recognition. We will discuss both these methods in subsequent sections.

Marr and Nishihara's Theory of Recognition

Marr and Nishihara's theory of 3-D object representation, together with Marr's ideas about the stages by which such a representation is computed, is definitely among the most influential body of ideas in vision research [61, 62]. Marr and Nishihara point out that a representation does not have to reproduce a shape's surface in order to describe it adequately for recognition. For instance, animal shapes can be portrayed quite effectively by "stick figures" (figures which show the arrangement and relative sizes of a small number of sticks).

The simplicity of this stick figure representation is due to the correspondence between the sticks and the natural or "canonical" axes of the shapes described. Marr and Nishihara propose an object-centered co-ordinate system together with the use of volumetric primitives (generalized cylinders) of various sizes, and a modular, hierarchical representation.

The notion of stick-figure-like representation is not original to Marr and Nishihara. For instance, Blum [11] studied a similar classification scheme for 2-D silhouettes based on a *brushfire* technique, while Binford [9] proposed the use of "generalized cones" for 3-D objects. The difference is in the modular organization of information in Marr and Nishihara's scheme. In Blum's or Binford's scheme, for example, each part of a human arm would be represented by a single primitive. In Marr and Nishihara's scheme, it is possible to have both a single stick corresponding to the whole arm, as well as three smaller sticks corresponding to the upper arm, forearm, and hand. The hand, in turn, can be represented at a finer level of detail with individual sticks for each of the fingers.

This overall representation method is reasonably effective for some classes of objects, i.e., those which have a canonical co-ordinate system (since the description is object-centered) and a natural decomposition into component axes. However, Marr and Nishihara did not develop a generally applicable way of deriving this 3-D model description from an image. They used points of deep concavity on an object contour as segmentation points. Heuristics were used to connect them with other segmentation points, giving the decomposition into parts.

The overall recognition process functions by first selecting a model from the catalogue, based on the distribution of components along the length of the principal axis. This model then provides relative orientation constraints that help to determine the absolute orientations (relative to the viewer) of the component axes in the image. This can be used to compute their relative lengths. This new information can then be used to disambiguate shapes at the next level of specificity.

Generalized Cones and the ACRONYM System

Generalized cones [9] describe 3-D volumes. A generalized cone is specified by a curve through space, called the *spine*, along which a 2-D shape, called the *cross-section*, is swept. The cross-section is kept at a constant angle to the tangent of the spine, and may be deformed according to a deformation function called the *sweeping rule*.

The particular domain of the ACRONYM system [14] is aerial photographs of airplanes. In ACRONYM, the generalized cones are restricted to having straight line segments or circular arcs as spines, cross-sections bounded by straight lines or circular arcs, and sweeping rules which are linear magnification functions about the spine, or linear in two orthogonal directions about the spine. The input to ACRONYM consists of aerial photographs with a number of airplanes in view.

Given certain classes of geometric models (of airplanes), the goal is to identify their instances in the image, along with their location and orientation. ACRONYM must also make any possible subclass identifications, determine 3-D parameters of the objects, and determine the location and orientation of the camera, if not known *a priori*.

A description of the image is extracted in terms of *ribbons* and *ellipses*. Ellipses are described by the lengths of their major and minor axes. Ribbons (2-D analog of generalized cones) are suitable for describing images generated by the generalized cones themselves, while ellipses are suitable for describing the shapes generated by the ends of generalized cones.

Even with noisy data, ACRONYM produces fairly accurate interpretations in terms of 3-D models, by the combined use of geometric and algebraic constraints. However, the top-down approach of ACRONYM, which predicts the projected appearance in the image of the parts of a specific model using complex quantitative constraints, make it unsuitable for general 3-D object recognition with a large number of models which may or may not be present in a scene.

2.5.2 "Geons" and Recognition By Components

The fundamental assumption in Biederman's theory of "Recognition By Components" (RBC) [8] is that there exists a small number (i.e., 36) of fundamental part primitives, whose combinations can represent more complicated objects for the purposes of "primal access".

An analogy is drawn with speech perception, in which only about 55 phonemes are needed to represent virtually all the millions of words in all of human speech. Further, Biederman proposes that these 36 basic "geometric ions" or geons can be identified by using non-accidental properties of edges in an image, i.e., curvature, collinearity, symmetry, parallelism, and co-termination. Recognition occurs by detecting the identity and spatial arrangement of an object's component geons.

Like Marr and Nishihara's theory [62], Biederman's scheme also does not do detailed geometrical modeling of object surfaces. Biederman's scheme avoids the problem of computing 2.5 D information by working directly from 2-D edges. The 3-D representation is obtained from easily observable non-accidental features in the line drawing. Thus, Biederman's theory provides an elegant link between perceptual organization (recognition of "low-level" structure) and 3-D object representation (description in terms of "high-level" structure). Detection of low-level structure helps to identify the geons, and geons with their combination identify the object.

RBC is a theory of so-called "primal access" in humans, i.e., "the first contact of a perceptual input from an isolated, unanticipated object to a representation in memory" ([8], pg. 178). Typically, humans recognize the object quickly, even when it is viewed from novel orientations, with moderate levels of visual noise, when partially occluded, or when it is a new exemplar of a category. This requires a quick procedure which does not need very precise quantitative computations. RBC assumes that an object is recognized through the following subprocesses: first an edge extraction stage provides a line drawing of the object. From this, non-accidental properties of edges are detected. The object is parsed into parts, primarily at concave regions. The nonaccidental properties are used to constrain the identity of the components. (Stages up to and including the identification of components are assumed to be bottomup). Finally, the arrangement of components is matched against representations in memory.

The primitive components or geons hypothesized in the RBC theory are simple generalized cones without sharp concavities. They are classified into 36 basic types on the basis of 4 attributes (Figure 2.2). Three of the attributes describe characteristics of the cross section: its shape (straight or curved), symmetry (asymmetrical, reflection, or reflection and rotation), and size as it is swept along the axis (constant, increasing, or increasing and decreasing). The fourth attribute describes the shape of the axis (straight or curved).

The values of these four generalized cone attributes can be directly detected as differences in non-accidental properties, e.g., collinearity, curvilinearity, symmetry, parallelism, and co-termination (Figure 2.3). Further, these non-accidental properties are rapidly identifiable from a line drawing, and are quite invariant to viewpoint and noise.

A set of geon relations is also proposed. The relations between any pair of geons includes verticality, i.e., whether the first geon is above, below, or to the side of the second one; relative size, i.e., whether the first geon is much larger than, much smaller than, or approximately the same size as the second; centering, i.e., whether the point of attachment on a geon's surface is centered or off-centered; and surface size at join, i.e., whether a geon is joined at a large surface or a small one. Altogether, there are



Figure 2.2. Four attributes yield 36 different geons (based on [8])



Figure 2.3. Non-accidental properties indicate geon type (based on [8])

almost 60 possible combinations of relations that can hold for a pair of geons.

Biederman argues that 36 geons apparently have sufficient representational power to express humans' capacity for basic visual categorization. Figure 2.4 shows some multi-part 3-D objects which can supposedly be represented using geons. Note that although the decomposition of some objects into geons seems obvious (e.g., wateringcan, table-lamp, flashlight, and knife), others are not very obvious (e.g., telephone handset, head of the elephant). Biederman states that there are less than 3,000 common basic-level categories such as "chairs" and "elephants" in a typical human language (e.g., English). Liberally assuming an average number of 10 distinguishable exemplars for each category, this gives an estimate of 30,000 readily discriminable objects. By contrast, the combination of 2 geons can give over 70,000 possible objects (36×36 possible geon combinations, multiplied by about 60 geon relations for each combination). Similarly, 3 geons can be combined in over 150 million ways.

Biederman's experiments indicated that line drawings of objects with only 2 or 3 of their components could be accurately identified from a single 100-ms exposure. These simple line drawings were identified almost as rapidly as full-color, detailed, textured slides showing all the components. It appears that neither the full complement of an object's geons, nor its color, texture, or full bounding contour need be present for rapid identification.

Bergevin and Levine [4, 3] have built a computer vision system based on geons, called PARVO (Primal Access Recognition of Visual Objects). PARVO introduces a new intermediate representation not present in the original RBC theory, i.e., faces of parts. First, the parts are identified as belonging to one of 11 generalized solid types, based on the first three attributes of geons (curved or straight edges, curved or straight axis, and constant, increasing, or increasing-and-decreasing sweeping function). The symmetry attribute is inferred later, giving the complete identity of the geon.

The system was tested with a variety of complete as well as imperfect line drawings



Figure 2.4. Complex objects represented using geons (taken from [1])

of common man-made objects. PARVO was usually able to segment the line drawings into component parts, identify each part as a geon, compute a coarse spatial structural description of the object, and match it to coarse models. Disadvantages include the use of perfect line drawings and the assumption that pairs of matched segmentation points (i.e., points of concave discontinuity) can always be found.

Recently, Dickinson et al. [21, 20, 22] have built a system called OPTICA (Object recognition using Probabilistic Three-dimensional Interpretation of Component Aspects). OPTICA computes a 3-D object representation based on 10 geon-like parts. The most significant contribution of Dickinson et al. is the idea of combining the "parts-based" and "viewpoint-dependent" ideas. They put forward the excellent idea of using a multi-view representation only for the limited number of 3-D part primitives rather than for entire complex 3-D objects composed of these parts.

Storing characteristic views of only the part primitives has two advantages. Firstly, the number of views is much smaller. Secondly, even with a different set of part primitives, it may be possible to use the same scheme for recognition by using the multi-view representation of the new set of primitives. Constructing the multi-view representation of the chosen generic part primitives is an off-line process that has to be performed only once.

OPTICA has been tested with perfect line drawings of 3-D objects. Segmentation into parts is performed manually. Methods have been outlined for computing the line drawing from an input intensity image and for performing segmentation, as well as for recovering from segmentation errors. However, these are not completely implemented.

2.5.3 Superquadrics

Superquadrics are an extension of basic quadric surfaces. They are a family of parametric shapes discovered by the Danish designer Piet Hein [36]. Superquadrics have been used or proposed for use as shape representation primitives for computer graphics [2], and recently, for computer vision [71]. Depending on the parameter values, superquadrics can take different shapes (e.g., ellipsoids, cylinders, parallelepipeds, pyramids, cones, as well as round-edged shapes intermediate between these standard shapes). Thus, they are versatile part primitives, and can be deformed and "glued" together into realistic-looking models as in Pentland's *Supersketch* graphics system [71].

Superquadrics are described by the following equation (using the notation,

$$\cos(\eta) = C_{\eta}, \sin(\omega) = S_{\omega}):$$

$$\vec{X}(\eta, \omega) = \begin{bmatrix} a_x \cdot C_{\eta}^{\epsilon_1} \cdot C_{\omega}^{\epsilon_2} \\ a_y \cdot C_{\eta}^{\epsilon_1} \cdot S_{\omega}^{\epsilon_2} \\ a_z \cdot S_{\eta}^{\epsilon_1} \end{bmatrix}, \quad -\frac{\pi}{2} \le \eta \le \frac{\pi}{2}, \ -\pi \le \omega < \pi$$

where $\vec{X}(\eta, \omega)$ is a 3-D vector that sweeps out a surface parametrized in latitude η and longitude ω in an object-centered co-ordinate system. The parameters a_x, a_y , and a_x affect the size of the superquadric along the x, y, and z axes, respectively. The relative shape parameters ϵ_1 and ϵ_2 affect the relative shape of the superquadric in the latitudinal (xz) and longitudinal (xy) directions, respectively. When all these five parameters are unity, the shape of the superquadric is the unit sphere. When a relative shape parameter $(\epsilon_1 \text{ or } \epsilon_2)$ is equal to 1, the cross-section along the corresponding (i.e., latitudinal or longitudinal) direction is an ellipse. As ϵ approaches 0, the shape becomes progressively more rectangular; as ϵ approaches 2, the cross-section changes from an ellipse to a diamond shape. When ϵ is greater than 2, the shape becomes "pinched", approaching a cross as ϵ approaches infinity. The effect of ϵ on shape is shown in Figure 2.5, in which one of the ϵ parameters is held constant at 1.0, while the other is varied between 0.0 and 3.0.

After Pentland's paper [71] which first proposed the use of superquadrics as part representation primitives for computer vision, most of the subsequent literature on





 $\epsilon = 0.0$





 $\epsilon = 0.7$

 $\epsilon = 1.0$



Figure 2.5. Effect of the parameter ϵ on superquadric shape

superquadrics has dealt with estimation methods and error-of-fit measures for recovering the parameters from real or synthetic range data [1, 41, 42, 72, 81]. Issues of segmentation are avoided and range data is taken from the isolated, unoccluded superquadric whose parameters are to be estimated. Only Ferrie et al. [30] report some results from range image segmentation and the actual use of superquadrics as part primitives for complex multi-part objects.

Besides the three extent parameters (a_x, a_y, a_z) and two relative shape parameters (ϵ_1, ϵ_2) , an additional three parameters are needed to represent displacement of the superquadric from the origin, and another three more for its orientation. Hence, a total of 11 parameters have to be estimated for a superquadric at arbitrary location and orientation.

Pentland [71] originally suggested an analytic solution using the parametric equations of the position vector and normal vector at each point on the surface. Using linear regression, and starting with information from 2-D contours and shading, a best-fit estimate was to be computed. This approach turned out to be too complicated and was not fully implemented. Pentland's second approach [72] was based on search through the entire superquadric parameter space. For each 3-D range image point, the parameter space is searched for the best fit to the immediately surrounding area. This method is too expensive computationally, unless perhaps implemented on a parallel architecture.

Bajcsy and Solina [1] suggested a faster (iterative) solution technique, i.e., minimizing an error-of-fit function by the Levenberg-Marquardt method [75] for non-linear least squares. Their error measure favors the superquadric of smallest possible volume that closely fits the data points. Poisson "jitter" is introduced to escape local minima in the space of 11 parameters. Gross and Boult [41] compare the effectiveness of different error-of-fit measures for Bajcsy and Solina's technique. According to their experiments (with superquadrics centered at the origin and aligned along the co-ordinate axes), an error measure based on Euclidean distance between each data point and the surface of the estimated superquadric performs better than the one proposed by Bajcsy and Solina. In particular, use of the Euclidean distance error measure leads to faster convergence and yields better estimates from sparse range data. It also gives better estimates when only partial range data are available (e.g., from only 1, 2, or 4 octants instead of all 8). However, Ferrie et al. [30] report that while Gross and Boult's Euclidean distance error measure generally fit the range data well, the volume of the estimated superquadric tended to become very large, especially when the surface patch was flat. According to them, the minimum-volume error measure motivated by Bajcsy and Solina [1] produced "more intuitive" results.

Solina and Bajcsy [81] have proposed a formalism for expressing deformations of superquadrics (specific kinds of tapering, bending, and cavities). They are able to estimate all the parameters by incorporating appropriate extensions in their original technique. Gupta, Bogoni and Bajcsy [42] argue that both qualitative and quantitative measures should be used to evaluate the fit of an estimated superquadric. They propose two global, quantitative measures (Bajcsy and Solina's original goodness-of-fit measure [1], and a mean-distance criterion derived from Gross and Boult's Euclidean distance measure [41]). They also introduce three local, qualitative evaluation criteria. The Min-distance map is produced by mapping the magnitude of the shortest Euclidean distance of individual points from the model surface. The Contour-difference map compares the apparent contour formed by the model in the viewing direction, with the object's occluding contour. Lastly, the Z-distance map shows the distance between the points in the range image and the model surface in the viewing direction. These three measures complement the earlier two, and, in conjunction with them, can suggest improvements in object segmentation or model deformation.

2.5.4 Other Part Primitives

Yet another part representation for man-made objects (e.g., chairs), using sticks (long, thin objects), plates (flat, wide objects), and blobs (objects with 3 significant dimensions) has been proposed by Shapiro et al. [80].

Terzopoulos et al. [90] have proposed a 3-D part model similar to generalized cylinders, with additional deformation parameters to control the elasticity of the main axis and walls of the cylinder. With human interaction to set constraints on parameters, they are able to recover the models from 2-D silhouettes. Later work by Terzopoulos and Metaxas [89] proposes a representation which combines the local degrees of freedom of splines (which makes them useful for surface reconstruction) with the global shape parameters of a superquadric (which makes them useful for shape recognition). Results of fitting are provided for both intensity and range data. Initial segmentation into parts is done manually by the user.

Pentland and Sclaroff [68] propose a very interesting 3-D deformable model representation based on the *Finite Element Method* (FEM) and the FEM technique of *Modal Analysis*. The model consists of a large number of finite elements, with properties such as mass, stiffness, and damping which govern their dynamic behavior when the model is deformed. The input data are treated as constraints forcing the deformable model to assume a shape that closely fits the data. The complex deformation undergone by the model is decomposed into the superposition of various simple deformations corresponding to the *free vibration modes* of the FEM equilibrium equation. This is similar to a Fourier representation where a complicated waveform can be represented by the combination of independent simple (sinusoidal) waveforms of different frequencies.

The vector of coefficients corresponding to the lowest order vibration modes (say the 30 lowest order vibration modes) are used to represent the fitted shape. Two shapes can be matched simply by finding the normalized dot product of the corresponding modal representations. The use of this representation for tracking motion is also described by Pentland and Horowitz [67].

2.6 Summary

We presented a brief survey of 3-D object representation schemes. These were classified as either viewpoint-dependent or viewpoint-independent, and either objectintrinsic or parts-based. A parts-based approach has the advantages of flexibility and modularity. Selection of an appropriate set of part primitives, and part segmentation (decomposition of an image corresponding to the separate parts) are the crucial issues in a parts-based approach.

We are primarily interested in somewhat "meaningful" part primitives, i.e., volumetric models with a considerable degree of capability for shape discrimination. As mentioned earlier, this choice simplifies later steps in the recognition process (representation of a 3-D object is more compact and "meaningful," fewer hypotheses need be generated for matching, and verification is much less essential). The price to be paid is that segmenting and identifying these parts from an image is more difficult.

For our 3-D object representation scheme, we intend to use a small set of 3-D volumetric parts based on geons. The choice of these primitives, as well as a method for part segmentation (i.e., decomposition into separate parts), are discussed in the following chapter.

CHAPTER 3

Parts and Part Segmentation

This chapter is organized as follows: first, we explain our choice of parts for the 3-D parts-based object representation. Next, we put forward a novel method of *specifying* "generic" 3-D parts, i.e., by using *Surface Adjacency Graphs* (SAGs). Then we derive an extremely compact multi-view representation of the part primitives, based on their SAGs. Finally, we present a method of performing part segmentation from range images, given a good surface segmentation. Some of the work described here has been presented earlier in [76].

3.1 Choice of Part Primitives

Many of the 3-D object recognition systems described in the literature share some common problems: firstly, they cannot cope with a large number of known objects, say even 100 models. This problem is particularly acute when models are not stored hierarchically, but are put all together in a "flat" knowledge base. This makes efficient indexing much more difficult, which in turn makes the recognition strategies impractical when the number of models is large. Secondly, most of them use features or modeling primitives which are too "weak," such as line segments, angles, and surface patches. This means that every object yields a large number of such features or primitives, making for an unconvincing representation scheme which leads to too many hypotheses and too much "branchiness" of the search tree if the matching scheme is based on search. Chapter 1 discussed this point in considerable detail.

In order to resolve these issues, a good object representation scheme should have the following features:

- It should use more powerful modeling primitives, yielding a compact and "meaningful" representation scheme. Specifically, we would like to use a 3-D partsbased representation.
- The representation scheme should lend itself naturally to efficient indexing when matching object and model descriptions, so that a model-based recognition system built on top of it would be able to cope with a large number of models.
- It should facilitate efficient segmentation/matching routines, by using viewpoint-dependent representations. But the number of such views should be small and the choice of views should be "natural" and justifiable at least to some extent.

We are interested in powerful part primitives, i.e., volumetric models with a considerable degree of capability for shape discrimination. As we saw, this choice simplifies later steps in the recognition process. It makes the representation of a 3-D object more compact, fewer hypotheses are generated for matching, and verification is much less essential.

Many 3-D parts-based representations have been proposed in the object recognition literature. The most influential 3-D part primitives are generalized cylinders, superquadrics, and geons. Geons, since they are capable of satisfying the criteria listed above, and due to the evidence (from visual perception literature) for their plausibility as a shape indexing mechanism, are an attractive choice as 3-D part models. Of the four geometrical properties that characterize the 36 different geons, some are more useful than others as important shape attributes. Straightness or curvedness of the axis, as well as nature of the "sweeping function" of the cross-section along the axis, seem to be important attributes. Straightness or curvedness of the crosssectional edge is an attribute which can be difficult to extract from images: firstly, there may be objects which do not show distinct "faces" where the cross-sectional edges are easily observable (e.g., a cucumber or a rectangular block with smoothened edges). Secondly, this distinction becomes blurred in the case of, say, an octagonal prism approximating a cylinder.

Cross-section symmetry is somewhat problematic. It is harder to compute, as well as less "obviously important" than the others as a shape attribute useful for object recognition. It is worth noting that much other RBC-inspired research also ignores cross-section symmetry, and uses a smaller set of parts either as the complete set of primitives, [20, 64] or as an intermediate step towards computing the entire set of 36 geons [4].

Ignoring the symmetry distinctions leads to a reduced set of only 12 geon-like 3-D parts. We select these 12 geon-like parts as the 3-D modeling primitives for 3-D object representation.

3.1.1 Notation

We will use the following notation to specify the 12 geon-like 3-D parts: axis shape is either bent (b) or straight (s); cross-section edges are either curved (c) or straight (s); cross-section size is either constant (co), increasing-&-decreasing (id), or tapered (t). Any of the 12 parts can therefore be specified by a triplet for axis shape, crosssection edge, and cross-section size. For example, b-c-co indicates a part with bent axis, curved cross-section edges, and constant cross-section. s-s-id indicates a part with straight axis, straight cross-section edges, and increasing-decreasing cross-section

Notation	Axis	Cross-sectional edges	C.S. size along axis
b-c-co	bent	curved	constant
b-c-id	b ent	<i>c</i> urved	incr&-decr.
b-c-t	bent	<i>c</i> urved	tapering
b-s-co	bent	straight	constant
b-s-id	bent	straight	incr&-decr.
b-s-t	bent	straight	tapering
8-C-CO	straight	<i>c</i> urved	constant
s-c-id	straight	<i>c</i> urved	incr&-decr.
s-c-t	straight	<i>c</i> urved	tapering
8-8-CO	straight	straight	constant
s-s-id	straight	straight	incr&-decr.
s-s-t	straight	straight	tapering

Table 3.1. Notation for specifying the 12 parts



Figure 3.1. Parts with straight edges and constant cross-section

size.

Table 3.1 lists the 12 parts using this notation, and the parts themselves are shown in Figures 3.1 through 3.6.

3.2 Surface Adjacency Graphs

Previous work on geons is based on line drawings, i.e., it uses edges derived from intensity images as the input. Consequently, methods for identifying the geons have also used perceptual properties of edges in the line drawing.

We pointed out earlier that ours is the first attempt to recover a specific, reasonably



Figure 3.2. Parts with straight edges and increasing cross-section



Figure 3.3. Parts with straight edges and incr.-and-decr. cross-section



Figure 3.4. Parts with curved edges and constant cross-section



Figure 3.5. Parts with curved edges and increasing cross-section



Figure 3.6. Parts with curved edges and incr.-and-decr. cross-section

rich set of geon-like 3-D parts from range data. Generally speaking, when working with range data it is natural to deal with surfaces and their properties rather than with edges. Hence, we would like to specify the part primitives by using their surface properties. Also, this has to be done in a way that captures the "generic" nature of the parts, i.e, the fact that they are defined by qualitative shape properties and not by specific numerical shape measures.

We use the Surface Adjacency Graph (SAG) as a fundamental representation throughout this chapter. This representation has many advantages. It can be used to precisely define the generic parts in terms of their surfaces. Next, it yields a multiview representation of the parts in a very natural manner. Given a good method for surface segmentation, the SAG can be easily obtained from range data. The same representation can be used for part segmentation by comparing the SAG of the scene with our multi-view representation of parts. Gupta [43] has independently proposed the use of SAGs, and their use for segmenting a data set by breaking concave edges of the SAG. Our segmentation method is more general.

49

3.2.1 Surface Classification by Curvatures

Given a surface S, at each point P on the surface there is a tangent direction along which the surface curves most, as well as a tangent direction along which the surface curves least. Except in degenerate cases, these directions are orthogonal. They are called *principal directions*; the corresponding maximum curvature and minimum curvature are together referred to as the *principal curvatures*. For example, on a cylindrical surface, the curvature is zero along the tangent direction parallel to the axis of the cylinder; this is the minimum curvature. The curvature is greatest (in absolute value) along the surface tangent direction perpendicular to the axis of the cylinder; this is the maximum curvature. Planes and spheres are special cases because the curvature on a planar or spherical surface is independent of the direction of travel along the surface. Curvature of a plane is zero in all directions. For a sphere, the curvature is the inverse of the radius and is also constant in all directions.

The actual sign of a curvature depends on the way the surface curves with respect to the positive Z axis. For our range images, the positive X axis corresponds to the direction of increasing column numbers, the positive Y axis is in the direction of decreasing row numbers, and the positive Z axis points out of the XY image plane. Hence, the range image of a convex spherical cap would show a surface curving downwards with respect to the positive Z axis, and so it would have negative principal curvatures. However, in our terminology we reverse the signs and refer to a "convex" curvature as positive. Similarly, the concave "in-side" of a sphere would be referred to having a negative curvature.

Mean curvature (H) and Gaussian curvature (K) are defined as the average and the product, respectively, of the principal curvatures. Based on the signs (positive, zero, or negative) of H and K, we can perform a qualitative classification of surfaces into 8 possible surface types [5]. We use a qualitative classification based on the

Max. curvature	Min. curvature
+	+
+	0
+	-
0	0
_	0

Table 3.2. Curvature-based classification into 5 surface types

principal curvatures. Restricting ourselves to a situation where the "inside" surfaces of part primitives are never visible, some surface types never occur. Based on the signs of principal curvatures, we can then distinguish between five kinds of surfaces, as shown in Table 3.2.

We compute curvature by the method implemented by Flynn [33]. At each point on a surface, a bicubic surface is fitted to the point and its neighbors. Principal curvatures and directions are then computed from the coefficients of the fitted bicubic surface. A threshold of 0.2 is used for qualitative classification of principal curvatures as positive, zero, or negative. A curvature measure for a surface is taken to be positive if its median value is greater than 0.2. It is taken to be negative if the median value is less than -0.2. If the median value is between -0.2 and 0.2, the curvature measure is taken to be zero. (A curvature value of 0.2 is the same as that of a sphere of radius 1/0.2, i.e, 5 inches).

3.2.2 SAGs of Part Primitives

The Surface Adjacency Graph (SAG) is a graph in which the nodes represent surfaces, while an edge between two nodes indicates that the corresponding surfaces are adjacent. Nodes as well as edges may have attributes. Node attributes can represent any property of the corresponding surface, say its type (planar, convex, concave, etc.) or area. In our case a node has only one attribute, i.e., the "type" of the corresponding surface as determined from its principal curvatures. An edge has only two attributes,



Figure 3.7. Coding scheme for SAG edge attribute

both concerning the angle between the surfaces corresponding to the two nodes connected by the edge. The first attribute describes whether the angle is concave or convex. The second attribute describes whether the angle is acute, right, obtuse, or zero. The node attribute (i.e., surface type) as well as edge attributes (i.e., convex or concave adjacency, acute/right/obtuse/zero angle) can also be "don't-care," indicating that the attribute may take any of the possible values.

Figures 3.8 through 3.19 show the SAG representations of our 12 geon-like parts. Figures 3.8 through 3.13 show SAGs for parts with straight-edged cross-sections, while Figures 3.14 through 3.19 shows SAGs for parts with curved-edged cross-sections. Each node is labeled according to the signs of the principal curvatures of the corresponding surface. For our 12 parts, adjacencies between surfaces are always convex, so the convex/concave attribute is not shown for any of the edges. The other SAG edge attribute, i.e., acute/right/obtuse/zero angle, is coded as in Figure 3.7.



Figure 3.8. SAG for the part s-s-co



Figure 3.9. SAG for the part b-s-co



Figure 3.10. SAG for the part s-s-t



Figure 3.11. SAG for the part b-s-t



Figure 3.12. SAG for the part s-s-id



Figure 3.13. SAG for the part b-s-id



Figure 3.14. SAG for the part s-c-co


Figure 3.15. SAG for the part b-c-co



Figure 3.16. SAG for the part s-c-t



Figure 3.17. SAG for the part b-c-t



Figure 3.18. SAG for the part s-c-id



Figure 3.19. SAG for the part b-c-id

3.3 Multi-view Representation of Parts

We stated our intention to use viewpoint-dependent representations for part segmentation and as an aid towards part identification. In the last few years, under the name of "aspect graphs" or "characteristic views," there has been a flurry of research activity on viewpoint-dependent representations. This was briefly reviewed in the previous chapter. We will now explain the details of our own multi-view representation.

3.3.1 Catalog of Part Views

Recently, the geon-inspired literature has also dealt with the issue of a multi-view, viewpoint-dependent representation. A viewpoint-dependent representation for general 3-D objects would cause two major difficulties: firstly, for a general, complicated 3-D object, no analytical method is known for computing the aspect graph. Secondly, the number of aspects would be too large (the literature on aspect graphs finds hundreds of aspects even for simple objects).

Dickinson et al. [21, 20, 22] overcome these difficulties by combining the "partsbased" and "viewpoint-dependent" ideas. That is, there is a small number of 3-D part models. Dickinson et al. put forward the idea of using a multi-view representation only for this limited number of 3-D parts rather than for entire complex 3-D objects composed of these parts. By using an empirical viewsphere approach and making use of part symmetries, they obtain 688 views for their entire set of 10 part primitives.

As mentioned earlier, storing characteristic views of only the part primitives has two advantages. Firstly, the number of views is much smaller. Secondly, even with a different set of part primitives, it may be possible to use the same scheme for recognition by using the multi-view representation of the new set of primitives. Constructing the multi-view representation of the chosen generic part primitives is an off-line process that has to be performed only once.

Following the idea proposed by Dickinson et al., we will use multi-view representations of the part primitives, and not of entire multi-part objects. The expectation is that a multi-view representation of parts will be useful in segmenting objects into component parts, as well as towards identifying those parts as instances of specific part primitives.

What views are "natural" to use for the viewpoint-dependent representation of the 3-D part primitives? When we look at simple objects (such as our 12 3-D part primitives), we can generally distinguish a few large, "meaningful" surfaces oriented in a particular way with respect to one another. The surfaces usually make sense, like "the top of the coke can," "the side of the monitor," etc. Usually, if we want to draw the object, we can do so unambiguously by drawing its important surfaces.

We may be able to extract large, "meaningful" surfaces from an image by looking at perceptually important 2-D features such as long curves, co-terminating lines, and parallel lines. These might have been obtained from an intensity image. But these features can also be found from range images. The point is that this multi-view SAG representation (in terms of large, "meaningful" surfaces) can be easily obtained for the part primitives. It need only be computed once, and can in fact even be done manually and off-line. On the other hand, for a scene image, the SAG representation can easily be computed if a good surface segmentation is available.

In our case, the catalog of possible views of parts is based only on surface configurations. We are concerned only with the types of visible surfaces, their adjacencies, and angles between them. We are not interested in every possible aspect, but only those in which the surface types, adjacencies, and/or angles differ. Unlike the usual definition of an aspect, which takes into account qualitative changes in the vertices, edges and surface identities, we define a simpler kind of characteristic view, the surface aspect (s-aspect). The s-aspect of an object is the set of all possible views for which the surface types, adjacencies, and angle attributes are the same. Our multiview representation will contain not the aspects (in the usual sense), but rather all possible s-aspects of the part primitives. So our multi-view representation of the part primitives, i.e., our catalog of views, turns out to be much smaller and simpler than the usual "aspect graph."

There are a number of ways to construct the catalog of views. Given the complete aspect graph of a part, we could obtain the multi-view representation by combining all those aspects in which the number, types, adjacencies, and angles of the surfaces is the same. A simpler alternative would be to make use of the fact that for our purposes, a part is completely characterized or defined by its attributed SAG. Hence, all possible views of a given part can be determined by finding all possible connected subgraphs of the part's attributed SAG, and removing duplicates and impossible views.

This method results in a total of 161 views for all 12 parts. However, many of these views can never be obtained in reality, e.g., a view of a rectangular block in which all 6 surfaces are visible.

It is easy to show that any subgraph that includes two oppositely oriented planar surfaces corresponds to a configuration of surfaces that can never be seen from any viewpoint. (When we say that a configuration of surfaces can be seen from a point, we mean that every surface in that configuration can be seen from that point).

Surfaces of part primitives are considered to have two sides, i.e., an "inner" side and an "outer side." Surface normals on the *outer* side of a part surface point *out* of the volume occupied by the part. For a point P on a part surface to be visible from a point Q in space, the outward-pointing surface normal at P must make a positive projection on the vector from P to Q. Given two "oppositely oriented" planar surfaces S_1 and S_2 (i.e., planar surfaces whose outward-pointing surface normals are oppositely oriented), obviously there exists no vector in space such that a pair of outward-pointing surface normals, one from S_1 and one from S_2 , both have a positive projection on it. Hence, there is no point Q in space such that from Q, the outer sides of both S_1 and S_2 are visible.

We showed that if a subgraph includes two oppositely oriented planar surfaces, then it corresponds to a configuration that can never be seen from any viewpoint in space. However, the converse is not true. There are many surface configurations without two oppositely oriented planar surfaces, but which still cannot be seen from any viewpoint in space. For example, no two surfaces of a tetrahedron are oppositely oriented. Nevertheless, there is no viewpoint from which the "outer" sides of all four surfaces of a tetrahedron are simultaneously visible.

We obtained the multi-view part representation empirically, because, as mentioned earlier, a large number of "impossible" views are included in the catalog found by the exact approach (based on enumerating all distinct connected subgraphs of the attributed SAG of each part). We constructed CAD models corresponding to the 12 shape classes using *GEOMOD*, a geometric CAD solid modeling package. GEOMOD is one component of *I-DEAS* (Integrated Design and Engineering Analysis System) [87]. The 12 CAD models are shown in Figures 4.1 and 4.2.

We then viewed the CAD model of each of the part primitives from a large number (100) of random viewpoints on the surface of a viewsphere and noted the different



Figure 3.20. S-aspects of a rectangular block

aspects. Figure 3.20 illustrates the simplicity of our multi-view representation of a rectangular block (one of the 12 basic geons), which has just 3 s-aspects. Note that the true aspect graph of even a simple cube would have 26 aspects. Figure 3.29 shows the multi-view representation of a bent cylinder, another of the 12 basic geons. This part has only 7 s-aspects. In fact, the entire set of 12 geon-like shapes (as defined by the SAGs shown earlier) has only 74 different views or s-aspects!

The surfaces in Figures 3.20 through 3.31 are labeled according to the signs of the maximum and minimum curvatures, which may be positive (+), zero (0), or negative (-).

The multi-view representation of parts contains the following information: for each characteristic view of each of the chosen generic parts, it describes the surface types, their adjacencies, and the angle attribute (concave or convex) between adjacent surfaces. This information is stored in the form of attributed SAGs, one for each view in the multi-view representation. At present, we do not take into account whether the angle between two surfaces is acute/right/obtuse.



Figure 3.21. S-aspects of an increasing-decreasing rectangular block



Figure 3.22. S-aspects of a tapering rectangular block



Figure 3.23. S-aspects of a cylinder



Figure 3.24. S-aspects of an increasing-decreasing cylinder



Figure 3.25. S-aspects of a tapering cylinder



Figure 3.26. S-aspects of a curved-axis rectangular block



Figure 3.27. S-aspects of a curved-axis, increasing-decreasing rectangular block



Figure 3.28. S-aspects of a curved-axis, tapering rectangular block



Figure 3.29. S-aspects of a curved-axis cylinder

For our 12 geon-like parts, the angles between surfaces are always convex. Hence, in this case the multi-view representation of parts simply amounts to a catalog of adjacencies of attributed surfaces, with surface adjacencies always being convex. Note that if we selected a different set of generic part primitives, for which some surface adjacencies were concave (say "L" or "T" shaped primitives), then concave angles would appear in the catalog of part views.

3.4 Part Segmentation from Surface Segmentation

Object recognition and surface classification methods based on differential geometric measures such as curvatures are common in the literature [5, 7, 28, 34, 92]. Curvature

68



Figure 3.30. S-aspects of a curved-axis, increasing-decreasing cylinder



Figure 3.31. S-aspects of a curved-axis, tapering cylinder

measures (such as principal curvatures, or mean and Gaussian curvatures) have been used to distinguish between spheres, cones, cylinders and cubes [7, 59]. Hence it seems natural to propose the use of similar measures to aid in part segmentation and identification for our somewhat larger set of part primitives. Since curvature estimates are noisy, we use only the signs (positive, zero, or negative) of the median values of principal curvatures on each surface.

3.4.1 Surface segmentation and classification

First we segment the range image into surfaces of different types. Surface segmentation is performed by the method of Flynn [33]. Flynn's algorithm in turn begins with a preliminary segmentation obtained by Hoffman and Jain's "split-and-merge" method [50]. Hoffman and Jain apply the CLUSTER program to get an initial surface segmentation. A squared-error clustering procedure finds up to 16 clusters in the 6-dimensional space consisting of 6 local surface features, i.e., the three components of the normal vector at each point on the surface, along with the co-ordinates of that point. The result is usually oversegmented. Hoffman and Jain then obtain their final surface segmentation by merging adjacent surface patches which meet a "similarity" criterion. Using this as the preliminary segmentation, Flynn classifies individual surfaces into the following types: planar, spherical, cylindrical, and unknown. Appropriate geometric parameters are estimated for each surface. Finally, adjacent surface patches are merged if they are of the same type and have similar parameters. In most cases, this gives a good surface segmentation.

After surface segmentation, we begin construction of the SAG by classifying each surface according to the distributions of curvature measures on the surface. From the signs of the maximum and minimum curvatures (within appropriate thresholds), we can distinguish between 5 surface types, as shown in the first two columns of Table 3.3, and draw some inferences about the part primitive to which they belong.

Max. curv.	Min. curv.	Inference
+	+	Curved edges & bent axis, OR
		Curved edges & c.s. size incr-decr
+	0	Curved edges & straight axis, OR
		Straight edges & curved axis
+	-	Curved edges & bent axis
0	0	End face of any part, OR
		Straight edges & straight axis
-	0	Straight edges & bent axis

Table 3.3. Some inferences of part shape based on surface type

3.4.2 Using the Catalog of Part Views

The main problem of parts-based representation is part segmentation, i.e., how to segment an arbitrary object into instances of the part primitives. We will use comparison of the surface types and adjacencies in the image with those stored in the multi-view representation (catalog of aspects or views) of the part primitives for part segmentation as well as an aid towards part identification.

Good surface segmentation is crucial to our method for part segmentation. After surface segmentation and classification, the image consists of a set of surfaces, with their adjacencies, types, and angles between adjacent surfaces. Like the views in the catalog, this is also represented as an attributed SAG. Surfaces of insignificantly small area are ignored after taking their connectivity into account, i.e., if such a surface connects two larger surfaces, it is removed from the graph after ensuring that the nodes corresponding to the two larger surfaces are connected to one another.

It is assumed that a single surface in the SAG belongs to exactly one part. Every edge in the SAG, therefore, is potentially an adjacency between surfaces belonging to two different parts. Hence, each edge in the SAG is examined to see whether the multiview representation can account for two adjacent surfaces of those particular types, being adjacent with those particular angle attributes. (Currently, concave/convex is the only angle attribute used). If not, the surfaces must belong to different parts, so the edge is removed. Note that we could have chosen more complicated generic part primitives (for example a "T" or "L"), in which some adjacent surfaces meet along a concave crease even though they belong to the same part.

Every edge of the graph is examined in this manner to get a set of hypothesized part segmentations. The graph is thus divided into connected, attributed subgraphs, each consisting of surfaces hypothesized to belong to the same part. Time complexity of this part segmentation method is proportional to the number of edges in the SAG.

Each connected subgraph is then examined to verify whether it can be accounted for by one or more part views from the multi-view representation, i.e., whether there is some view of some part, such that the subgraph partially matches the corresponding SAG.

This is worth doing because of the current fragility of surface segmentation and curvature-based surface classification. By comparing the subgraphs with the catalog of part views, in some cases it is possible to determine immediately that there have been errors in surface segmentation and classification, because the subgraphs do not partially match any view of any part in the catalog. Each subgraph for which we are able to find one or more partial matches in the catalog corresponds to a hypothesized part and one or more possible interpretations of what the part identity might be.

This method of part segmentation is in some ways more general than common approaches which, in one way or another, make use of Hoffman and Richards' "principle of transversality" [47]. This principle states that when two arbitrarily shaped surfaces interpenetrate, they usually meet in a contour of concave discontinuity of their tangent planes. However, Figures 3.32 and 3.33 show intersecting parts which do not meet along concave creases. Part segmentation techniques which depend solely on concavity would fail in these cases. However, our method would still work because irrespective of the angle between the surfaces, our catalog of part views cannot account for two adjacent cylindrical surfaces belonging to the same part.



Figure 3.32. Two parts which meet along a convex crease



Figure 3.33. Two parts which meet along a "mixed" crease



Figure 3.34. L-shaped object for which part segmentation fails

Our part segmentation method also has some major limitations. Firstly, it is totally dependent on reliable surface segmentation and reliable curvature-based surface classification. But in reality, surface segmentation as well as curvature-based surface classification are often incorrect.

Secondly, the method does not use any information about shape of the object, relying instead solely on surface properties. Hence, it fails when two parts come together in such a way that the adjacency of the corresponding surfaces is not directly visible, even though in many such cases the intensity or occluding contour data clearly indicates the joining of two parts. For example, consider a view of an L-shaped object in which only a single surface, i.e., the L-shaped surface, is visible (Figure 3.34). The occluding contour clearly suggests the presence of two parts. Nevertheless, our method cannot segment this object into two parts from such a viewpoint. The SAG consists of a single node, and there are no SAG links to be broken.

Additionally, for this example (the L-shaped object), even if two surfaces belonging to different parts had been visible (Figure 3.35), part segmentation would have been incomplete. The reason is that the single L-shaped surface is itself produced by the coming together of two parts, and our part segmentation method has no way to



Figure 3.35. Another view of the L-shaped object

break up a single surface into pieces belonging to different objects. Many other such examples can be found.

3.4.3 Part Identification: Two Approaches

We used the catalog of part views to perform part segmentation by breaking SAG links of the given scene image. The following chapters discuss two methods for identification of the isolated parts thus obtained.

The first method does not make any further use of the catalog of part views. The isolated parts are classified based on features derived from estimated parameters of the best-fitting superquadric. This approach is described in Chapter 4.

The second method uses the catalog of part views to hypothesize which parts could possibly account for each of the subgraphs isolated by part segmentation. A direct test of one geon attribute (cross-section size) is then applied to eliminate some of the possibilities and determine the part identity. A direct test of another geon attribute (axis shape) using intensity image data is also developed in order to partially recover from some shortcomings of the method. Further details are given in Chapter 5.

3.5 Summary

In this chapter, we first explained our choice of parts for the 3-D parts-based object representation. A set of 12 geon-like "generic" parts was arrived at by ignoring the cross-sectional symmetry attribute of geons. Next, we introduced the notion of Surface Adjacency Graphs (SAGs), i.e., attributed graphs in which nodes represent attributed surfaces and edges represent attributed surface adjacencies. We put forward a novel method of *specifying* "generic" 3-D parts, i.e., by specifying their SAGs. After a brief overview of aspect graph (i.e., viewpoint-dependent representation) research, we derived an extremely compact multi-view representation of the part primitives, based on their SAGs. We presented a method of performing part segmentation from range images, given a good surface segmentation. Lastly, we pointed out that the catalog of part views can also be used to detect errors in surface segmentation and curvature-based surface classification.

After part segmentation, the next step is part identification, i.e., the parts that have been isolated must be identified as instances of the 12 part primitives. The following two chapters examine two approaches towards part identification.

CHAPTER 4

Geons from Superquadrics

In the previous chapter, we presented a method for part segmentation, given surface segmentation. The next step is to classify the isolated parts into one of the 12 geonlike shape classes. The research described below has two goals. Firstly, it attempts to perform this part identification of isolated parts. Secondly, in doing so, it tests the shape discrimination capability of superquadric parameters under varying conditions of image resolution, model elongation, and noise.

We will now attempt to relate two influential families of 3-D shape primitives, superquadrics and geons. As mentioned previously, superquadric surfaces have attracted considerable interest recently as part models for representation of 3-D objects. Here, we explore their utility for a different purpose, i.e., for shape discrimination. We investigate how well the estimated superquadric parameters can reflect certain intuitive geometric attributes of elongated objects, such as axis shape (straight or bent), type of cross-sectional edges (straight or curved), and variation of cross-section size along the axis (constant, tapered, or increasing-and-decreasing). Parameters of the bestfitting superquadric are estimated for real as well as synthetic range images, obtained from a large number of viewpoints, of models belonging to 12 shape classes based on the above geometric attributes.

Five features derived from the estimated superquadric parameters are used to

distinguish between these 12 shape classes. Classification error rates are estimated for k-nearest-neighbor and binary tree classifiers. The effects of varying the range image resolution, noise level, and the model elongation are also investigated. The results indicate that existing methods of superquadric parameter estimation are quite sensitive to noise in range values, and also do not work well for "rough," coarsesurfaced objects. However, for low-noise real range images of objects with "smooth" surfaces, numerical features derived from estimated superquadric parameters can be used with a binary tree classifier to infer qualitative shape properties with almost 80 % reliability. Some of the work described here has earlier been presented in [77].

4.1 Motivation

As mentioned earlier, *superquadric* surfaces, as well as *geons*, are recent additions to the set of 3-D part primitives proposed for object modeling. Aside from this fact, however, the two differ in almost every possible way. For instance, superquadrics are precisely defined parametric surfaces, whereas geons are specified by qualitative geometric properties; superquadrics are usually estimated from range images, while geons are obtained from line drawings; superquadrics are "fitted" by minimizing some error-of-fit criteria, whereas geons are recognized from their non-accidental geometric properties, and so on. Nevertheless, superquadrics and geons have strengths which mutually complement one another, making it advantageous to relate the two representations.

While it seems clear that superquadrics provide a compact, versatile representation for modeling and visualization (CAD/graphics), their use for 3-D object recognition is almost unexplored. Since obtaining reasonable estimates of superquadric parameters is computationally laborious, using the superquadric representation seems worthwhile only if it can offer significant advantages for object recognition (e.g., by serving as a good basis for indexing into the object database).

The above reference to indexing suggests a natural connection with another influential class of shape primitives, i.e., geons, which are supposed to discriminate between classes of 3-D shapes. It seems, therefore, that a good test of superquadrics for shape discrimination (as opposed to just representation) would be to see whether they can distinguish between geons.

Establishing such a link would be valuable because the superquadric and geon representations have some mutually complementary strengths. For example, the superquadric representation by itself provides no advantage from the point of view of indexing into a large database of objects. However, if superquadrics could be classified into shape categories similar to geons, they would have all the strengths of the geon representation in indexing into a database. Similarly, a weakness of the "pure" geon representation is that it does not perform precise geometric modeling, e.g., does not store size information which would allow it to distinguish between, say, a human being and a doll. On the other hand, a superquadric representation would capture these "numerical" shape differences and produce a precise geometric model.

The superquadric representation by itself cannot distinguish between all the symmetry attributes relevant to geons. This is because undeformed superquadrics have a cross-section with at least reflection symmetry, and deformations such as tapering or bending do not distort the cross-section. Hence, we have tried to at least discriminate between a set of only 12 geon classes (ignoring symmetry). This leaves 3 geometric properties to be detected: edge shape, axis shape, and type of cross-section sweeping function.

This chapter does not deal with part segmentation, which is a separate issue [31, 70], and has been discussed in Chapter 3. Rather, it focuses on a possible subsequent step, i.e., classifying the individual parts into one of 12 geon-like shape classes. We are concerned with the *sensitivity to shape* property of estimated superquadric

parameters, i.e., their capability to reliably discriminate between shape properties of the isolated parts.

The existing literature on superquadrics does not address this issue, and it is not at all obvious that such shape discrimination can in fact be done. For instance, estimation methods may yield superquadric parameters which minimize some error criterion (e.g., least squares), and which may appear even visually to be a "good" overall fit. Yet, the sensitivity of one or more of the estimated parameters to the qualitative shape properties may be too low (or too high) for their successful use as features for reliable shape classification. Besides attempting shape classification based on the estimated parameters, this is also the first study which performs an analysis of superquadric parameter estimation over a large number of range images taken from many different viewpoints, while varying three other relevant factors (noise level, object elongation, and image spatial resolution).

The remaining parts of this chapter are organized as follows: Section 4.2 discusses the relationship between superquadrics and geons. Section 4.3 justifies the choice of features to be used by the statistical classifiers, and explains the design and methodology of experiments performed. Section 4.4 presents the results of our classification experiments, and Section 4.5 discusses conclusions which can be drawn from the experimental results and summarizes this chapter.

4.2 Relating Superquadrics and Geons

As argued earlier, it would be advantageous to relate superquadrics and geons. But, do the superquadric parameters contain enough information, at least theoretically, to distinguish between the shape attributes relevant to geons? The answer, in general, is negative, because not all geons can be modeled by superquadrics (with possible bending and tapering). All elongated, convex superquadrics are geons, but not all geons are superquadrics. (Non-convex superquadrics, with shape parameters $\epsilon > 2$, would more naturally be modeled as a Boolean combination of simpler geons). Both superquadrics and geons are classes of generalized cylinders, but the specification of a geon is less restrictive. Superquadrics are precisely defined mathematical shapes, whereas the description of a geon, by intention, specifies only 4 qualitative (nonnumerical) shape attributes and is indifferent to other details.

For example, a geon with unsymmetrical cross-section cannot be modeled by a superquadric. Cross-sections of undeformed superquadrics are necessarily symmetric at least with respect to reflection. If their x and y dimensions $(a_x \text{ and } a_y)$ are equal, they are also symmetric with respect to 90° rotation. Bending and linear tapering are two deformations commonly applied to superquadrics, which affect the symmetry attributes of the object as a whole. However, these operations do not affect the symmetry attributes of the cross-section. We could try to formalize more complex deformations, such as warping, shearing, etc. in order to distort the cross-sections themselves, and thus to model a larger variety of geons. But this would increase the number of parameters which need to be estimated from range data to fully specify the deformed superquadric.

Since every elongated, convex superquadric is a geon, whereas not every geon can be modeled by a superquadric, it is clear that elongated, convex superquadrics are a *subset* of geons. Consequently, in general, the superquadric parameters will not suffice to model or discriminate between all geons. Nevertheless, deformed superquadrics can model most of the geons that are commonly encountered in industrial and household objects. For this reason, it is still worth exploring the utility of superquadric parameters for shape discrimination.

4.3 Experiments in Shape Discrimination

The objective in these experiments is, primarily, to evaluate the utility of the superquadric representation for discriminating the 12 classes of 3-D parts shown in Figures 3.1 through 3.6. We attempted to do this by first estimating parameters of superquadrics which best fit the object in the given range image, deriving a set of features from the estimated parameters, and then using standard statistical pattern recognition methods (k-nearest-neighbor and binary tree classifiers) for classifying the given shape into one of the known categories. We also tested the effects of (a) using range images of different resolutions to estimate the superquadric parameters, and (b)using objects which are "elongated" to different extents. Our initial experiments with real range images used hand-carved objects with rough, "noisy" surfaces. In order to understand/explain the results of these experiments, we also studied the effect of (c)adding varying levels of noise to the range images.

4.3.1 Choice of Features

Following the formalism of Solina and Bajcsy [81], 15 parameters must be estimated from a range image of a bent, tapered superquadric at arbitrary location and orientation. Of these, only 8 are relevant for the purposes of shape discrimination: the dimensions along x, y, and z axes, a_x, a_y , and a_z ; the shape parameters ϵ_1 and ϵ_2 ; coefficients for linear tapering, k_x and k_y ; and the radius of bending, r. Of the remaining 7 parameters, 6 specify the estimated position and orientation of the superquadric, which is irrelevant to its shape. The seventh unused parameter estimates the angle of bend in case of superquadrics deformed by bending. However, we can detect bending using other parameters, as explained below.

By convention, the object is oriented with its longest dimension along the z-axis. It is clear that the ratio r/a_z indicates the extent to which the superquadric is bent. The parameter ϵ_2 determines whether the cross-sectional edges are straight or curved. The nature of the cross-section sweeping function is more complicated, depending on ϵ_1 , k_x , and k_y . In the absence of tapering, the object has almost "constant" cross-section size if ϵ_1 is close to zero. As ϵ_1 increases, the cross-section size becomes "increasingand-decreasing". The cross-section size is "increasing" (or "decreasing") if ϵ_1 is zero and at least one of the tapering coefficients is non-zero. Hence, a set of 5 features is selected from the superquadric parameters for our 12-class shape discrimination problem: r/a_z , ϵ_1 , ϵ_2 , k_x , and k_y .

The superquadric representation is non-unique in the sense that the same shape may be represented by more than one set of parameters. For example, a "square" cross-section ($\epsilon = 0$) can also be interpreted as a "diamond-shaped" cross-section ($\epsilon = 2$), rotated by 45°. Such ambiguities do not arise here because the estimation procedure always yields ϵ parameters within the range [0, 1].

4.3.2 Design of Experiments

Experiments were performed using synthetic as well as real range images. The synthetic images are useful because they are taken from objects designed with a CAD modeling system, allowing exact control over the elongation and other shape attributes. Ignoring the symmetry attribute, there are 12 shape classes corresponding to the "collapsed" set of geons.

Each classification experiment was repeated with 1-nearest neighbor (1-nn), 3nn, 5-nn, and binary tree classifiers. K-nearest-neighbor classifiers determine the ktraining patterns that lie closest in the feature space to a test pattern, and assign it to the category with the largest number of representatives among those neighbors [23]. The binary tree classifier is based on the method proposed by Sethi and Sarvarayudu [79]. It hierarchically partitions the feature space, at each stage using hyperplanes parallel to the feature axes. A single feature is used at each nonterminal node of the decision tree. The feature to be used at each node, and the corresponding threshold, are determined by maximizing an information-theoretic measure, the "average mutual information gain."

Error rates for the k-nearest-neighbor as well as binary tree classifiers were obtained using the "holdout" method [23], in which part of the available patterns (90% in our case) are used for training, and the remaining part is held back for testing the classifier. This method is known to yield a pessimistic estimate of classification error.

Experiments with Synthetic Range Images

We constructed models corresponding to the 12 shape classes using GEOMOD, a geometric CAD solid modeling package. GEOMOD is one component of I-DEAS (Integrated Design and Engineering Analysis System) [87], which, besides object modeling, also has finite element analysis, database manipulation, and drafting capabilities. Figures 4.1 and 4.2 show synthetic range images of the 3-D shapes modeled using IDEAS. For ease of viewing, the range images are rendered as pseudo-intensity images.

Three models of different "elongations" were constructed for each of these 12 shape classes. The "axis bent" shapes are actually piecewise linear, as seen in Figure 4.2. By "elongation", we mean the ratio of the longest axial dimension (by convention, oriented along the z-axis) to the larger dimension along either of the other two axes (x or y) at the mid-point. The three models for each shape class have elongations of 1:2, 1:4, and 1:6. 90 synthetic range images are generated, corresponding to viewing each of these models from 30 different viewpoints. (For each viewpoint, range images are taken at resolutions of 64×64 , 128×128 , and 256×256). This gives a total of 270 viewpoints for each shape class, 90 from each of the 3 models. One can also categorize this data as 90 images from each of the three resolutions.

Given these 270 synthetic range images for each of the 12 shape classes, we first estimated the superquadric parameters for each image using Solina and Bajcsy's



Figure 4.1. Synthetic range images: shapes with straight axes



Figure 4.2. Synthetic range images: shapes with bent axes

method. In practice, this procedure does not always quickly converge. In such cases, we used the current estimate after 100 iterations. We then performed classification experiments as follows:

- 1. Overall shape discrimination: Using all the 270 pattern vectors for the 12 shape classes, attempt to discriminate between:
 - (a) parts with straight and curved axes
 - (b) parts with straight and curved cross-sectional edges
 - (c) parts with constant, increasing, and increasing-decreasing cross-sections
 - (d) all twelve parts
- 2. Effect of range image resolution: Perform the same four experiments as in (1), separately for the patterns derived from 64×64 , 128×128 , and 256×256 range images (90 patterns for each resolution are available for each of the 12 parts).
- Effect of object elongation: Perform the same four experiments as in (1), separately for the patterns derived from parts with elongations of 1:2, 1:4, and 1:6.
 (90 patterns for each elongation are available for each of the 12 parts).

Preliminary experiments with *real* range images led us to realize that surface imperfections ("roughness") have a very significant effect on the quality of superquadric parameter estimates. Consequently, we attempted to study the effect of surface imperfections by modeling them as "noise." The experiments described earlier all made use of noise-free synthetic images. We then studied the mean and standard deviation of the estimated parameters with two different levels of zero-mean additive Gaussian noise. Noise was added to the z-component of the synthetic images. The standard deviations of the two levels of noise were 2.5 and 5.0 (about 1.0% and 2.0%, respectively, on a scale of 255. Range values in the images were quantized between 0 and 255). These experiments with noisy images were done only for the "best" condition of the other factors, i.e., for objects of elongation 1:6 and image resolution of 256 \times 256.

Experiments with Real Range Images

Each of the 12 parts was constructed using easily carvable material such as potatoes and sweet potatoes (Figure 4.3). No attempt was made to achieve a very smooth and even surface finish, since geons are not supposed to be sensitive to such details.

Range images were then obtained by viewing each of these 12 parts from 20 arbitrary viewpoints using a Technical Arts 100X White scanner [88]. Figure 4.4 shows typical range images of these parts, obtained with the White scanner. Since the White scanner obtains depth maps by the principle of triangulation, it cannot compute the depth at points where the laser stripe is invisible to the camera, whether due to occlusion or due to the intensity of the reflected laser stripe being insufficient at some surface orientations. Such points on the object are displayed as black pixels in Figure 4.4. From some viewpoints, entire surfaces of the models may thus be missing, reducing the amount of depth data available for estimating the superquadric parameters.

These real range images are of resolution 240×240 . Parameters were estimated as before for the best-fitting superquadrics, and classification experiments were performed to discriminate between:

- 1. parts with straight and curved axes
- 2. parts with straight and curved cross-sectional edges
- 3. parts with constant, increasing, and increasing-decreasing cross-sections
- 4. all 12 parts



Figure 4.3. Hand-carved geons with "rough" surface finish



Figure 4.4. Real range images of "rough" hand-carved geons
As mentioned earlier, surfaces of these hand-carved objects were not of smooth finish. Further, over the 2 days when imaging was in progress, the surface quality became somewhat rougher due to drying out of the vegetable material, resulting in uneven surface deformations. Based on the poor results of these initial experiments with real range data, we surmised that surface coarseness or roughness, as well as "missing" data (due to the reflected laser stripe being insufficiently intense at some orientations) substantially reduced the accuracy of estimated superquadric parameters.

Hence, the above experiments were also repeated with real range images taken from models with relatively smooth, even surface quality. Some of these were "manufactured" objects, while others were carved by hand from soap bars. The advantage of using this material was that surfaces could be made smooth and even by washing the finished model in water. Range images were then obtained by viewing each of these 12 models from 10 arbitrary viewpoints using the White scanner. Figure 4.5 shows typical range images of these models.

4.4 Experimental Results

We report the results of superquadric parameter estimation, as well as shape classification based on features extracted from the estimated parameters.

4.4.1 Parameter Estimation

As described earlier, 5 features were chosen with the expectation that their values would reflect the three shape attributes of interest, i.e., axis shape, type of crosssection edges, and variation of cross-section size along the principal axis of a part. In this subsection, for each of the 12 parts, we will discuss the "ideal" values of the 5 features, and present the values actually obtained by superquadric parameter



Figure 4.5. Real range images of some "smooth" geons

estimation.

"Proper" Values for Features

It would be useful to indicate the "proper" values for each of the five features which have been derived from superquadric parameter estimates. The parameter ϵ_1 governs the shape of the superquadric along longitudinal planes, i.e., for planes passing through the principal axis. The primary effect of this parameter, therefore, is on the variation of cross-section size along the principal axis. For constant cross-section objects, ϵ_1 should be close to 0.0, indicating a rectangular profile in longitudinal planes. At the same time, k_x and k_y should also be close to 0.0. For objects with tapering cross-section, ϵ_1 should be close to 0.0, but either k_x , or k_y , or both, should be significantly higher than 0.0. By experience, we have observed that estimates of ϵ_1 which are less than 0.3 can be considered "close to 0.0," i.e., they correspond to a more or less rectangular profile in longitudinal planes. "High" estimates of ϵ_1 (closer to 1.0) indicate an increasing-and-decreasing profile in longitudinal planes, corresponding to an increasing-and-decreasing cross-section size along the principal axis. Estimates of k_x and k_y less than about 0.25 or 0.3 can be considered "close to 0.0," i.e., they correspond to non-tapering longitudinal profiles.

The parameter ϵ_2 governs the shape of the superquadric along latitudinal planes, i.e., for "cross-sectional" planes perpendicular to the principal axis. This parameter primarily affects the nature of the cross-sectional edges, i.e., straight-edged or curvededged cross-section. Again by experience, we have observed that estimates of ϵ_2 which are less than 0.3 can be considered "close to 0.0," i.e., they correspond to a more or less straight-edged profiles in cross-sectional planes.

The feature r/a_z governs the straightness of the principal axis. By experience, we have observed that estimates of r/a_z less than about 6.5 correspond to bent-axis objects.

Actual Values of Estimated Features

For each of the 12 parts, listed using the above notation, Tables 4.1 through 4.5 show the mean and standard deviation for estimates of the five features ϵ_1 , ϵ_2 , k_x , k_y , and r/a_x , respectively. These statistics are presented for real range images as well as for synthetic range images with varying levels of noise. It is useful to view these tables in conjunction with the confusion matrices presented in Tables 4.12 through 4.14, which indicate the nature of misclassifications for each of the 12 parts. "Unexpectedly" low or high values of the parameter estimates (shown in Tables 4.1 through 4.5) explain several of the striking kinds of misclassifications.

Table 4.1 shows the mean and standard deviation for estimates of the superquadric parameter ϵ_1 . The first three columns give statistics from synthetic range images with additive Gaussian noise of standard deviations (σ) 0.0, 2.5, and 5.0, respectively. The last two columns show statistics from real range images of the hand-carved and "smooth"-surfaced objects, respectively. Tables 4.2 to 4.5 show similar statistics for the other four features, i.e., ϵ_2 , k_x , k_y , and r/a_z .

In general, superquadric parameter estimates for synthetic images with low values of noise σ (0.0 and 2.5) are similar, and close to the values we would expect. The standard deviations are quite large, indicating that parameter estimates vary considerably depending on the viewpoint. For example, in column 2 of Table 4.1 (σ =0.0), standard deviations for the shape classes b-c-id and b-s-id are quite large. This is because from certain viewpoints, these shapes (with increasing-and-decreasing crosssection size) are mistaken for shapes with a bent axis and constant cross-section size. On the other hand, in the same column, the standard deviation is almost zero for objects with a straight axis.

The superquadric parameter estimates are poorer for synthetic images with a higher noise σ (5.0), as well as for real range images from the "rough" objects. This

Shape class	Synthetic da	ta with differ	ent noise levels	Real	data
	$\sigma = 0.0$	$\sigma = 2.5$	$\sigma = 5.0$	Hand-carved	"Smooth"
	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.
b-c-co	0.10, 0.02	0.13, 0.07	0.19, 0.20	0.16, 0.08	0.12, 0.03
b-c-id	0.49, 0.42	0.66, 0.43	0.23, 0.29	0.33, 0.23	0.86, 0.10
b-c-t	0.17, 0.12	0.17, 0.11	0.28, 0.30	0.14, 0.13	0.20, 0.03
b-s-co	0.11, 0.04	0.10, 0.00	0.16, 0.19	0.17, 0.13	0.10, 0.00
b-s-id	0.35, 0.26	0.23, 0.17	0.27, 0.29	0.49, 0.33	0.32, 0.03
b-s-t	0.19, 0.11	0.16, 0.14	0.26, 0.29	0.34, 0.29	0.21, 0.04
8-C-CO	0.10, 0.00	0.11, 0.02	0.13, 0.05	0.21, 0.20	0.10, 0.00
s-c-id	1.00, 0.00	0.97, 0.07	0.21, 0.22	0.77, 0.25	1.00, 0.00
s-c-t	0.10, 0.02	0.12, 0.10	0.14, 0.07	0.39, 0.28	0.27, 0.17
8-8-CO	0.11, 0.07	0.10, 0.01	0.12, 0.05	0.23, 0.24	0.10, 0.00
s-s-id	1.00, 0.00	0.96, 0.16	0.28, 0.35	0.70, 0.27	0.33, 0.28
s-s-t	0.11, 0.04	0.13, 0.11	0.12, 0.04	0.29, 0.27	0.13, 0.05

Table 4.1. Mean and standard deviation of estimates of ϵ_1 for the 12 parts

gives a forewarning that part identification based on these parameter estimates will be poor for the higher-noise synthetic images as well as for the real images of hand-carved objects with "rough" surfaces. In fact, classification error rates for the higher-noise $(\sigma = 5.0)$ synthetic range images are similar to those for real range images of the "rough" surfaced objects.

For most of the parts, parameter estimates from real range images of their "smooth" models are generally similar to those from the low-noise synthetic images.

Ideally, the parameter ϵ_1 should be close to zero for objects whose cross-section size is constant or tapering along the axis, and larger for objects with increasing-anddecreasing cross-section size. The mean values of ϵ_1 for low-noise synthetic images generally conform to this expectation, as can be seen from Table 4.1. Empirically, if $\epsilon_1 > 0.3$, the superquadric cross-section has a distinctly "increasing-and-decreasing" appearance.

The parameter ϵ_2 should ideally be close to zero for objects with "straight" crosssection edges, and close to 1.0 for objects with "curved" cross-section edges. As can

Shape class	Synthetic da	ta with different	ent noise levels	Real	data
	$\sigma = 0.0$	$\sigma = 2.5$	$\sigma = 5.0$	Hand-carved	"Smooth"
	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.
b-c-co	0.72, 0.39	0.71, 0.36	0.33, 0.18	0.20, 0.18	0.28, 0.36
b-c-id	0.91, 0.16	0.84, 0.08	0.37, 0.21	0.45, 0.35	0.95, 0.05
b-c-t	0.57, 0.39	0.68, 0.31	0.31, 0.20	0.50, 0.28	0.94, 0.05
b-s-co	0.10, 0.00	0.10, 0.00	0.19, 0.16	0.21, 0.21	0.10, 0.00
b-s-id	0.10, 0.00	0.10, 0.00	0.17, 0.17	0.47, 0.40	0.14, 0.02
b-s-t	0.11, 0.03	0.13, 0.10	0.16, 0.11	0.27, 0.28	0.19, 0.09
s-c-co	1.00, 0.00	0.92, 0.07	0.42, 0.15	0.84, 0.14	0.98, 0.03
s-c-id	0.99, 0.05	0.83, 0.10	0.37, 0.21	0.83, 0.13	0.84, 0.08
s-c-t	0.98, 0.06	0.85, 0.18	0.39, 0.17	0.78, 0.18	0.93, 0.07
8-8-CO	0.10, 0.00	0.10, 0.00	0.26, 0.30	0.28, 0.27	0.10, 0.00
s-s-id	0.10, 0.00	0.10, 0.00	0.21, 0.24	0.60, 0.39	0.15, 0.03
s-s-t	0.13, 0.16	0.10, 0.00	0.26, 0.32	0.30, 0.35	0.18, 0.07

Table 4.2. Mean and standard deviation of estimates of ϵ_2 for the 12 parts

be seen from columns 2 and 3 of Table 4.2, this expectation is borne out quite well for synthetic images with low or no noise.

Estimates for k_x and k_y should ideally be close to zero for non-tapering parts, and distinctly higher for tapering parts. The deterioration in estimates of these two parameters is particularly striking as the noise level increases (columns 4 and 5 of Tables 4.3 and 4.4).

For the feature r/a_z , low values indicate a bent-axis part, while high values indicate a straight axis. While estimates of this feature are best for the low noise images, it is in fact estimated quite well even for the higher noise synthetic images and real images of the hand-carved models.

4.4.2 Part Discrimination

Using all 5 features together, the 12-category classification results using binary tree classifiers and k-nearest-neighbor (k-nn) classifiers were strikingly different. In general, the binary tree classifier performed much better. This is expected, since it

Shape class	Synthetic da	ta with different	ent noise levels	Real	data
	$\sigma = 0.0$	$\sigma = 2.5$	$\sigma = 5.0$	Hand-carved	"Smooth"
	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.
b-c-co	0.07, 0.09	0.06, 0.08	0.27, 0.16	0.32, 0.35	0.17, 0.14
b-c-id	0.13, 0.11	0.11, 0.11	0.38, 0.30	0.26, 0.20	0.06, 0.04
b-c-t	0.18, 0.09	0.21, 0.11	0.30, 0.21	0.28, 0.21	0.29, 0.13
b-s-co	0.04, 0.07	0.09, 0.13	0.22, 0.18	0.25, 0.32	0.04, 0.03
b-s-id	0.08, 0.08	0.10, 0.08	0.25, 0.19	0.43, 0.31	0.06, 0.06
b-s-t	0.23, 0.26	0.33, 0.25	0.35, 0.29	0.36, 0.36	0.35, 0.24
8-C-CO	0.07, 0.06	0.04, 0.07	0.24, 0.20	0.24, 0.22	0.29, 0.17
s-c-id	0.30, 0.21	0.22, 0.17	0.60, 0.34	0.27, 0.10	0.55, 0.27
s-c-t	0.51, 0.05	0.39, 0.10	0.31, 0.20	0.45, 0.21	0.32, 0.09
8-8-CO	0.03, 0.05	0.06, 0.15	0.30, 0.20	0.30, 0.33	0.05, 0.02
s-s-id	0.30, 0.25	0.26, 0.21	0.60, 0.28	0.41, 0.26	0.15, 0.08
s-s-t	0.46, 0.09	0.37, 0.18	0.32, 0.21	0.49, 0.29	0.23, 0.24

Table 4.3. Mean and standard deviation of estimates of k_x for the 12 parts

Table 4.4. Mean and standard deviation of estimates of k_y for the 12 parts

Shape class	Synthetic da	ta with different	ent noise levels	Real	data
	$\sigma = 0.0$	$\sigma = 2.5$	$\sigma = 5.0$	Hand-carved	"Smooth"
	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.	Mean, S.D.
b-c-co	0.07, 0.11	0.09, 0.12	0.36, 0.21	0.35, 0.28	0.19, 0.15
b-c-id	0.13, 0.10	0.11, 0.11	0.35, 0.24	0.22, 0.25	0.15, 0.08
b-c-t	0.20, 0.10	0.20, 0.10	0.26, 0.20	0.37, 0.30	0.27, 0.13
b-s-co	0.05, 0.08	0.07, 0.11	0.25, 0.20	0.35, 0.34	0.02, 0.01
b-s-id	0.10, 0.12	0.09, 0.09	0.31, 0.24	0.36, 0.30	0.14, 0.10
b-s-t	0.35, 0.33	0.27, 0.27	0.40, 0.36	0.30, 0.30	0.17, 0.17
8-C-CO	0.12, 0.08	0.04, 0.05	0.39, 0.19	0.26, 0.22	0.21, 0.19
s-c-id	0.33, 0.19	0.24, 0.18	0.58, 0.25	0.15, 0.10	0.46, 0.31
s-c-t	0.49, 0.06	0.34, 0.10	0.29, 0.18	0.39, 0.23	0.27, 0.13
8-8-CO	0.05, 0.13	0.02, 0.02	0.33, 0.19	0.15, 0.21	0.06, 0.04
s-s-id	0.28, 0.24	0.21, 0.21	0.49, 0.35	0.30, 0.24	0.24, 0.23
s-s-t	0.45, 0.09	0.42, 0.20	0.29, 0.16	0.27, 0.26	0.41, 0.21

2 parts
the 1
for
r/a,
ų
estimates o
f
deviation
standard
and
Mean
Table 4.5.

$\sigma = 0.0$ $\sigma = 2.5$ $\sigma = 5.0$ Hand-carvedb-c-co $2.10, 0.95$ $2.00, 0.94$ $2.21, 2.56$ $2.66, 1.72$ b-c-id $1.55, 0.31$ $1.44, 0.24$ $2.21, 2.56$ $2.66, 1.72$ b-c-id $1.55, 0.31$ $1.44, 0.24$ $2.29, 2.36$ $2.14, 1.07$ b-c-id $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.14, 1.07$ b-s-id $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.25, 1.21$ b-s-id $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.25, 1.21$ b-s-id $1.22, 0.52$ 1.72 3.74 $8.78, 24.03$ b-s-id $1.22, 0.52$ 1.72 3.59 $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ 1.72 3.59 $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.865, 12.06$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.65$ s-c-id $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-id $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-id $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $156.89, 311.10$ s-s-id $31.81, 39.01$ $26.11, 21.38$ $6.66, 11.87$ $12.54, 24.43$ s-s-id $31.81, 39.01$ $26.11, 21.38$ $6.66, 11.87$ $12.54, 24.43$ s-s-id $31.81, 39.01$ $26.11, 21.38$ $240, 30, 35.40, 376, 37.34$ $257.44, 376, 37.34$	Real da	lata
Mean, S.D.Mean, S.D.Mean, S.D.Mean, S.D.b-c-co $2.10, 0.95$ $2.00, 0.94$ $2.21, 2.56$ $2.66, 1.72$ b-c-id $1.55, 0.31$ $1.44, 0.24$ $2.29, 2.36$ $2.14, 1.07$ b-c-t $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.97, 2.12$ b-s-co $1.83, 1.00$ $1.87, 1.27$ $2.07, 1.95$ $2.97, 2.12$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.865, 12.06$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.66$ s-c-co $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-t $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-t $2218.27, 6015.25$ $329.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-td $31.81, 39.01$ $26.11, 21.38$ $118.60, 91.30$ $35.40, 38.69$	nd-carved	"Smooth"
b-c-co $2.10, 0.95$ $2.00, 0.94$ $2.21, 2.56$ $2.66, 1.72$ b-c-id $1.55, 0.31$ $1.44, 0.24$ $2.29, 2.36$ $2.14, 1.07$ b-c-t $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.97, 2.12$ b-s-co $1.83, 1.00$ $1.87, 1.27$ $2.07, 1.95$ $2.97, 2.12$ b-s-td $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-td $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-td $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-td $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-t $9.58, 36.11$ $9.83, 31.44$ $10.65, 27.34$ $6.06, 12.53$ b-s-t $1443.08, 1252.69$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.65$ s-c-co $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-td $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-s-cd $2218.27, 6015.25$ $329.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-td $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$ s-s-t $171.38, 11.86, 0.40, 30.30$ $254.03, 38.69$	ean, S.D.	Mean, S.D.
b-c-id $1.55, 0.31$ $1.44, 0.24$ $2.29, 2.36$ $2.14, 1.07$ b-c-t $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.25, 1.21$ b-s-to $1.83, 1.00$ $1.87, 1.27$ $2.07, 1.95$ $2.97, 2.12$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-to $1.443.08, 1252.69$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.6i$ s-c-co $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-ct $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-t $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-s-c-t $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-s-c-t $1965.48, 2534.28$ $229.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-c-t $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$ s-s-tid $31.81, 39.01$ $26.11, 21.80$ $56.60, 11.87$ $12.54, 24.43$.66, 1.72	3.76, 1.75
b-c-t $1.94, 0.91$ $1.98, 1.83$ $2.40, 2.83$ $2.25, 1.21$ b-s-co $1.83, 1.00$ $1.87, 1.27$ $2.07, 1.95$ $2.97, 2.12$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-t $9.58, 36.11$ $9.83, 31.44$ $10.65, 27.34$ $6.06, 12.53$ b-s-t $9.58, 36.11$ $9.83, 31.44$ $10.65, 27.34$ $6.06, 12.53$ b-s-t $1443.08, 1252.69$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.66$ s-c-id $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-id $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-t $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-s-to $2218.27, 6015.25$ $329.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-tid $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$ s-s-tid $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$.14, 1.07	2.68, 0.64
b-s-co $1.83, 1.00$ $1.87, 1.27$ $2.07, 1.95$ $2.97, 2.12$ b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-t $9.58, 36.11$ $9.83, 31.44$ $10.65, 27.34$ $6.06, 12.53$ s-c-co $1443.08, 1252.69$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.66$ s-c-co $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-t $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-t $2218.27, 6015.25$ $329.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-t $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$ s-s-t $18577, 015.25$ $329.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$.25, 1.21	3.51, 0.50
b-s-id $1.22, 0.52$ $1.72, 3.59$ $2.67, 3.42$ $8.78, 24.03$ b-s-t $9.58, 36.11$ $9.83, 31.44$ $10.65, 27.34$ $6.06, 12.53$ s-c-co $1443.08, 1252.69$ $739.84, 1912.69$ $51.35, 72.97$ $318.57, 1184.66$ s-c-id $18.65, 12.06$ $27.53, 32.20$ $6.24, 8.75$ $12.24, 21.18$ s-c-id $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-t $1965.48, 2534.28$ $429.46, 876.45$ $54.23, 87.23$ $57.84, 75.25$ s-c-t $1965.48, 2534.28$ $229.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-t $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$ s-s-t $1857.24, 277, 6015.25$ $329.55, 411.21$ $123.22, 156.83$ $156.89, 311.10$ s-s-td $31.81, 39.01$ $26.11, 21.80$ $6.66, 11.87$ $12.54, 24.43$.97, 2.12	5.12, 0.79
b-s-t 9.58, 36.11 9.83, 31.44 10.65, 27.34 6.06, 12.53 s-c-co 1443.08, 1252.69 739.84, 1912.69 51.35, 72.97 318.57, 1184.66 s-c-id 18.65, 12.06 27.53, 32.20 6.24, 8.75 12.24, 21.18 s-c-id 18.65, 12.06 27.53, 32.20 6.24, 8.75 12.24, 21.18 s-c-id 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-c-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-s-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-s-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-s-t 1965.48, 276.015.25 329.55, 411.21 123.22, 156.83 156.89, 311.10 s-s-td 31.81, 39.01 26.11, 21.80 6.66, 11.87 12.54, 24.43 s-s-t 1857.24 736.17 195.96 71.38 118.69 316.93	78, 24.03	2.45, 0.12
s-c-co 1443.08, 1252.69 739.84, 1912.69 51.35, 72.97 318.57, 1184.68 s-c-id 18.65, 12.06 27.53, 32.20 6.24, 8.75 12.24, 21.18 s-c-id 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-c-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-s-co 2218.27, 6015.25 329.55, 411.21 123.22, 156.83 156.89, 311.10 s-s-t 31.81, 39.01 26.11, 21.80 6.66, 11.87 12.54, 24.43 s-s-t 1857.24 7736.17 195.96 31.130	06, 12.53	6.44, 0.78
s-c-id 18.65, 12.06 27.53, 32.20 6.24, 8.75 12.24, 21.18 s-c-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-c-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-s-co 2218.27, 6015.25 329.55, 411.21 123.22, 156.83 156.89, 311.10 s-s-tid 31.81, 39.01 26.11, 21.80 6.66, 11.87 12.54, 24.43 s-s-t 1857.24.776.17 195.96 171.38 118.60 24.036	57, 1184.68 1	183.22, 143.82
s-c-t 1965.48, 2534.28 429.46, 876.45 54.23, 87.23 57.84, 75.25 s-s-co 2218.27, 6015.25 329.55, 411.21 123.22, 156.83 156.89, 311.10 s-s-id 31.81, 39.01 26.11, 21.80 6.66, 11.87 12.54, 24.43 s-s-t 1857 24 2736.17 195 26 171.38 118.69 249.30 35.49 38.69	.24, 21.18	2.24, 0.47
s-s-co 2218.27, 6015.25 329.55, 411.21 123.22, 156.83 156.89, 311.10 s-s-id 31.81, 39.01 26.11, 21.80 6.66, 11.87 12.54, 24.43 s-s-t 1857 24 2736.17 195 26 171 38 118.69 249 30 35.49 38.69	.84, 75.25	16.91, 8.07
s-s-id 31.81, 39.01 26.11, 21.80 6.66, 11.87 12.54, 24.43 s-s-t 1857 24 2736 17 105 26 171 38 118 60 240 30 35 40 38 62	.89, 311.10	55.11, 29.22
s-s-t 1857 24 2736 17 195 26 171 38 118 60 240 30 35 40 38 62	.54, 24.43	2.45, 1.40
	.49, 38.62 2	206.32, 294.80

Experiment	l-nn	3-nn	Tree
Axis - straight or curved?	7.1	6.8	6.8
C.S. edges - straight or curved?	23.1	24.4	7.4
C.S. size - const., tapered, or incr&-decr.?	33.6	38.0	10.0
12-class	49.1	53.1	13.8

Table 4.6. Overall classification error rates (in %) for synthetic range data

Table 4.7. Classification error rates (in %) by range image spatial resolution

Experiment	64×64			1	128×128			256×256		
	1-nn	3-nn	Tree	l-nn	3-nn	Tree	l-nn	3-nn	Tree	
Axis	17.6	13.9	7.9	4.6	5.6	7.6	1.9	1.9	5.1	
CS edges	29.6	28.7	8.8	19.4	17.6	7.5	20.4	22.2	4.3	
CS sweep	39.8	43.5	11.6	41.7	38.9	9.5	38.9	44.4	10.6	
12-class	60.2	66.7	15.9	50.0	55.6	13.2	50.9	50.9	13.7	

discriminates the 12 classes hierarchically instead of performing a one-shot classification like the k-nn classifiers. Classification error rates for synthetic data (models constructed using *IDEAS*) were estimated using the holdout method. Error rates were higher (or comparable) for 5-nn classifier, and are therefore not reported. Table 4.6 shows the *overall* error rates using all the 270 patterns for each of the 12 parts. It can be seen that axis shape (straight or curved) is by far the most reliably discriminated shape attribute. 12-category classification using the tree classifier is quite good (86.2% correct), and is far better than the performance of k-nn classifiers.

The effect of range image spatial "resolution" on classification error is shown in Table 4.7. We would expect the features derived from lower-resolution images to have a higher error rate, both because less range data are available for estimating the bestfitting superquadric, as well as because the range values themselves are less precise. The results do generally conform to this expectation. Estimates of the parameters k_x and k_y seem to be particularly sensitive to the image spatial resolution. Overall, the results are not very different for images of resolution 128 × 128 and 256 × 256. Both

Experiment	1:2				1:4			1:6		
	1-nn	3-nn	Tree	1-nn	3-nn	Tree	1-nn	3-nn	Tree	
Axis	13.9	8.3	7.5	6.5	5.6	6.2	6.5	0.9	3.5	
CS edges	19.4	19.4	7.2	32.4	28.7	6.8	25.9	27.8	8.1	
CS size	30.6	29.6	10.3	43.5	46.3	10.9	38.0	34.3	9.8	
12-class	48.1	46.3	14.8	60.2	62.0	13.5	51.9	52.8	13.3	

Table 4.8. Classification error rates (in %) by model "elongation"

Table 4.9. 12-class error rates (in %) for tree classifier

	128×128	256×256
Elongation 1:4	12.5	12.8
Elongation 1:6	12.8	12.2

of these cases are noticeably better than for resolution 64×64 .

As the elongation of the parts increases, we would expect to obtain a better fit of superquadric surfaces to the "increasing-and-decreasing" class of parts. Hence, error rates should decrease with increasing elongation. Further, the axis shape (straight or bent) should also be more reliably estimated. These predictions are borne out, on the whole, by the data in Table 4.8. The results are quite similar for models with elongations of 4 and 6 units.

It is clear that the tree classifier performs much better than the k-nn classifiers, and that error rates decrease with increasing image resolution and increasing elongation of the models. Hence, Table 4.9 shows only the overall (12-category classification) results for the tree classifier, using only the data from images of resolutions 128×128 and 256×256 , and model elongations 1:4 and 1:6. Even with our rather simple choice of features, correct classification of about 87% can be achieved for the 12-class problem.

For the *real* range images, Table 4.10 shows the error rates for classification of the rough-surfaced objects. Table 4.11 gives the corresponding results for the "smooth-

Experiment	l-nn	3-nn	Tree
Axis - straight or curved?	15.4	12.5	12.5
C.S. edges - straight or curved?	33.8	32.1	33.3
C.S. size - const., tapered, or incr&-decr.?	53.3	58.3	50.0
12-class	64.2	67.1	70.8

Table 4.10. Overall classification error rates (in %) for real range data from "rough" models

Table 4.11. Overall classification error rates (in %) for real range data from "smooth" models

Experiment	l-nn	3-nn	Tree
Axis - straight or curved?	8.3	13.3	15.0
C.S. edges - straight or curved?	28.3	36.7	11.7
C.S. size - const., tapered, or incr&-decr.?	28.3	31.7	20.0
12-class	43.3	55.0	23.3

surfaced" objects. The real data itself was used both for training and testing, and the error rate is estimated by the "holdout" method, as before. The error rate is much higher for the rough-surfaced objects than for the synthetic data. Classification error is considerably less for the smooth-surfaced objects, though still higher than for the synthetic images.

4.4.3 Confusion Matrices

Table 4.12 gives the confusion matrix for 12-category classification of the synthetic images. For each row in the confusion matrix, the leftmost column shows the true part identity. The remaining 12 columns show the percentage of cases where that part was classified as each of the 12 parts. The most striking fact is that classification is much better for parts with straight axis. In fact, we get 100% correct classification for 4 of the 6 straight-axis objects. For the bent-axis parts, most of the misclassifications are with respect to variation of cross-section size along the axis. The other two shape attributes are much less often misclassified.

8-s-t	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	0.0	10.0	0.0	90.0
s-s-id	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0
S-S-CO	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	0.0	83.3	0.0	3.3
s-c-t	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
s-c-id	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	3.3
s-c-co	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
b-s-t	6.7	0.0	10.0	10.0	13.3	70.0	0.0	0.0	0.0	6.7	0.0	0.0
b-s-id	0.0	3.3	0.0	0.0	76.7	6.7	0.0	0.0	0.0	0.0	0.0	0.0
b-s-co	6.7	0.0	3.3	83.3	10.0	6.7	0.0	0.0	0.0	0.0	0.0	3.3
b-c-t	6.7	13.3	56.7	6.7	0.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0
b-c-id	16.7	60.0	13.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
b-c-co	63.3	23.3	16.7	0.0	0.0	3.3	0.0	0.0	0.0	0.0	0.0	0.0
Shape	b-c-co	b-c-id	b-c-t	b-s-co	b-s-id	b-s-t	s-c-co	s-c-id	s-c-t	S-S-CO	s-s-id	s-s-t

Table 4.12. Confusion table for the synthetic images

Table 4.13 shows the confusion matrix for real range images of the rough-surfaced models. As in the case of synthetic images, classification is distinctly better for the straight-axis objects. Much of the misclassification still tends to be with respect to variation of cross-section size, though this effect is not so pronounced as for the synthetic images.

Table 4.14 shows the same statistics for range images of the "smooth-surfaced" models. Some strange effects can be noticed here. The *b-c-co* part is very poorly classified; most of the misclassification is in the straight-edged/curved-edged cross-section attribute. For this part, whose visual appearance is unmistakably curved-edged, Table 4.2 does indeed show a very low estimate for the parameter ϵ_2 . In fact, the *b-c-co* part accounts for most of the error in the entire 12-category classification for all the images. If we eliminate this part, 11-category classification with real data can be achieved with about 89% accuracy.

The other striking fact is that two straight-axis parts, *s-c-id* and *s-s-id*, tend to be mistaken for their curved-axis counterparts. We can see from Table 4.5 that mean values of the parameter r/a_z are very low for these two parts, explaining the misclassification.

4.5 Summary

The research presented in this chapter had two goals: firstly, to develop a method to identify the parts previously isolated by performing part segmentation; and secondly, to test the utility of superquadric parameters for part *discrimination*. Superquadrics, as well as geons, have recently been proposed as 3-D part primitives for object representation. In the case of superquadrics, most of the literature has been devoted to formalizing various deformations, estimating the parameters reliably, and formulating measures of goodness-of-fit.

8-8-t	0.0	5.0	0.0	0.0	5.0	5.0	5.0	0.0	0.0	35.0	0.0	40.0
s-s-id	0.0	5.0	5.0	0.0	10.0	0.0	0.0	10.0	5.0	5.0	40.0	0.0
8-8-CO	0.0	0.0	0.0	5.0	0.0	15.0	0.0	0.0	5.0	35.0	10.0	30.0
s-c-t	0.0	0.0	0.0	0.0	5.0	5.0	35.0	0.0	45.0	5.0	0.0	0.0
s-c-id	5.0	0.0	5.0	0.0	10.0	0.0	0.0	70.0	10.0	0.0	20.0	0.0
8-C-C0	0.0	0.0	0.0	5.0	0.0	0.0	60.0	5.0	30.0	10.0	0.0	5.0
b-s-t	20.0	5.0	15.0	5.0	5.0	30.0	0.0	0.0	5.0	0.0	5.0	10.0
b-s-id	5.0	15.0	0.0	5.0	25.0	10.0	0.0	0.0	0.0	5.0	20.0	15.0
b-s-co	35.0	5.0	5.0	55.0	10.0	5.0	0.0	5.0	0.0	5.0	0.0	0.0
b-c-t	0.0	20.0	50.0	5.0	10.0	5.0	0.0	10.0	0.0	0.0	5.0	0.0
b-c-id	15.0	15.0	10.0	10.0	10.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0
b-c-co	20.0	30.0	10.0	10.0	10.0	15.0	0.0	0.0	0.0	0.0	0.0	0.0
Shape	b-c-co	b-c-id	b-c-t	b-s-co	b-s-id	b-s-t	s-c-co	s-c-id	s-c-t	S-S-CO	s-s-id	s-s-t

Table 4.13. Confusion table for real images ("rough" models)

ape	b-c-co	b-c-id	b-c-t	b-s-co	b-s-id	b-s-t	s-c-co	s-c-id	s-c-t	8-8-CO	s-s-id	s-s-t
r	0.0	0.0	30.0	50.0	0.0	20.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	20.0	0.0	0.0	0.0	20.0	60.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
	0.0	20.0	0.0	0.0	0.0	0.0	0.0	80.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0
	0.0	0.0	0.0	0.0	20.0	20.0	0.0	0.0	0.0	0.0	60.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20.0	0.0	80.0

The amount of computational effort required to estimate the superquadric parameters leads us to desire that they should be useful not merely for the accurate and concise representation of 3-D objects, but also towards their recognition. As a possible step in this direction, we have tried to use superquadrics for shape discrimination. Specifically, we represented 3-D parts by the best-fitting superquadrics, and attempted to use the superquadric parameters to classify the part into one of 12 broad classes of shapes similar to geons. Relating superquadrics to geons is advantageous because of the complementary strengths of these two representations.

The experimental results indicate that qualitative shape attributes can be inferred from superquadric parameters quite reliably (about 87% accuracy for synthetic images; about 77% for real images) using a hierarchical (binary tree) classifier, even with a rather simple choice of features. In fact, 11-category classification of the "smoothsurfaced" real models (with only 11 shape types) could be done with about 89% accuracy! The worst-case time complexity of this classifier is a constant, depending on the depth of the decision tree constructed using the training patterns. Estimates of superquadric parameters are highly dependent on the viewpoint. Estimates may be very poor for parts with "rough" surfaces, or if the range values contain even a small amount of noise.

The future of superquadrics as part primitives for 3-D object representation depends on their effectiveness in facilitating object recognition or indexing. We have suggested a possible method of using superquadric parameters for this purpose. Building a system which performs generic classification of objects using a parts-based superquadric representation would be an interesting goal for future research. In the next chapter we will present another method for performing part identification.

CHAPTER 5

Direct Tests of Shape Attributes

Chapter 3 described a method to perform part segmentation by using the catalog of part views, and Chapter 4 presented one approach for classifying isolated parts as one of the 12 part primitives. Here we will examine an alternative method for part identification.

Having constructed the attributed SAG of a sensed image and performed part segmentation by referring to the catalog of part views, we are left with a number of hypothesized parts as well as hypotheses about their identity. Hypotheses about part identity are obtained when a subgraph of the scene SAG partially matches any view of any part in the catalog. We can eliminate the incorrect hypotheses about part identity by directly testing shape attributes of the part.

It would therefore be very useful to develop tests to directly check the shape attributes which define our 12 geon-like generic parts. We saw that features derived from estimated superquadric parameters can be used with fairly high accuracy to determine the three shape attributes of our 12 geon-like parts. Nevertheless, the confusion matrices in Chapter 4 indicate some consistent types of misclassification of part identity. For example, the *b-c-co* part is consistently misclassified as *b-s-co*. Similarly, the *s-s-id* and *s-c-id* parts tend to be misclassified as their curved-axis counterparts. For this reason we would like to develop tests for directly testing shape attributes, which are independent or at least partially independent of the superquadric-based method.

We have developed two such tests which directly check two of the three shape attributes which define our set of 12 geon-like parts, i.e., axis shape and cross-section size along the axis. The third shape attribute, i.e., type of cross-sectional edges, is obtained implicitly by determining the other two shape attributes and accordingly ruling out hypotheses about part identity.

5.1 Variation of Cross-section Size

To test cross-section size variation along the axis, we compute the histogram of the angles of the surface normals with respect to the axis direction. The angle values range from 0° to 180° .

In the case of constant cross-section, this distribution should be concentrated near 90°, since normal directions should be mostly perpendicular to the axis. In the case of objects with increasing cross-section, the angles should be concentrated some distance away from 90° (either greater than or less than 90°). Finally, for objects of increasing-decreasing cross-section, the angles should be spread along both sides of 90°). The rationale behind this heuristic is illustrated in Figure 5.1.

Figures 5.2 through 5.4 show three parts with constant, tapering, and increasing decreasing cross-sections, respectively. From Figure 5.2, it is visually apparent that normal directions on the side surfaces of this constant cross-section part are close to 90° with respect to the principal axis (direction along which the part is elongated). Similarly, the tapering part in Figure 5.3 shows one entire side surface for which normal directions are clearly at a non-perpendicular angle with respect to the principal axis. Lastly, Figure 5.4 shows that the part with increasing-and-decreasing crosssection has one entire side along which the normal directions spread from less than



Figure 5.1. Directions of principal axis and surface normals



(4)



Figure 5.2. Part with constant cross-section. (a) Range image of the part; (b) Normal directions on surfaces of the part.

90° to greater than 90° with respect to the principal axis.

These observations about angle directions are not valid for the "end" surfaces of parts. Hence, it is important that only the side surfaces (not the end surfaces) should be considered for the histogramming of surface normal directions described above. Hence we use a heuristic which ignores normal directions less than 45° or greater than 135° with respect to the axis, since they may belong to end surfaces.

The following thresholds are used: if at least 80% of the angles are either between 45° and 90°, or between 90° and 135° (i.e, they are concentrated on one side, either less than or greater than 90°), the cross-section is assumed to be increasing. Else, if at least 80% of the angles lie within 80° and 100° (i.e., they are concentrated near 90°, the cross-section is assumed to be constant. Else, the cross-section is assumed to



(a)



Figure 5.3. Part with increasing cross-section. (a) Range image of the part; (b) Normal directions on surfaces of the part.



(a)



Figure 5.4. Part with increasing-and-decreasing cross-section. (a) Range image of the part; (b) Normal directions on surfaces of the part.

be increasing-and-decreasing. The percentages mentioned are over only those points whose normals are not ignored due to the heuristic.

5.1.1 Principal Axis Orientation

As described above, our direct test of cross-section size variation requires first identifying the principal axis of the part. We find the principal axis of a part by estimating the best-fitting superquadric model for it. The advantage of finding the axis by this method is that it can be performed for bent-axis as well as straight-axis parts. Hence, the principal axis can be estimated even if it is not straight. The disadvantage is that a superquadric fit is usually not very reliable if only a few small surface patches are present. A reliable superquadric fit requires general viewpoint as well as the visibility of a large fraction of the object being fitted.

Once the best-fitting superquadric is estimated, orientation of the principal axis can be found from the estimated orientation parameters, i.e., the Euler angles ϕ , ω , and ψ . The superquadric estimation software uses Z-Y-Z Euler angles. The Z-Y-Z Euler angles relating two co-ordinate frames A and B with the same origin are defined as follows: we start with the two co-ordinate frames A and B coincident. Then, we first rotate B about Z_B by the angle ϕ , then rotate the new B about the new Y_B by the angle ω , and finally rotate the new B about Z_B by the angle ψ . Using the notation

$$cos(\phi) = C_{\phi}, sin(\phi) = S_{\phi}$$

 $cos(\omega) = C_{\omega}, sin(\omega) = S_{\omega}$
 $cos(\psi) = C_{\psi}, sin(\psi) = S_{\psi},$

the co-ordinates of a point in frame **B**, P_B , can be related to the co-ordinates of that point in frame **A**, P_A , by the transformation $P_B = T \cdot P_A$, where T is defined by:

$$C_{\phi} \cdot C_{\omega} \cdot C_{\psi} - S_{\phi} \cdot S_{\psi} \qquad S_{\phi} \cdot C_{\omega} \cdot C_{\psi} + C_{\phi} \cdot S_{\psi} \qquad -S_{\omega} \cdot C_{\psi}$$
$$-C_{\phi} \cdot C_{\omega} \cdot S_{\psi} - S_{\phi} \cdot C_{\psi} \qquad -S_{\phi} \cdot C_{\omega} \cdot S_{\psi} + C_{\phi} \cdot C_{\psi} \qquad S_{\omega} \cdot S_{\psi}$$
$$C_{\phi} \cdot S_{\omega} \qquad \qquad S_{\phi} \cdot S_{\omega} \qquad \qquad C_{\omega}$$

The superquadric estimation software aligns the principal axis with the Z direction. Hence, the principal axis is parallel to the new Z axis. We are not concerned with its exact location in space but only with its orientation, since we use it only to compute the angle between surface normals and the principal axis.

The cross-section variation attribute of the part primitives (cross-section size along the axis—constant, increasing, or increasing-decreasing) can then be determined from the angle of the normal direction at each point on the side surfaces, with respect to the axis direction.

For straight-axis parts, this angle can be directly computed by taking the dotproduct of a unit normal vector with a unit vector oriented along the principal axis. However, for bent-axis parts the axis direction is different at different points along the axis. Having computed the normal direction at a given point on the surface, we would need to know the principal axis direction at that location along the axis. This is difficult to compute. Instead, we first "straighten" the bent-axis parts by inverting the bending transformation [81] estimated during the superquadric fit. If the co-ordinates of a point are (X, Y, Z), its co-ordinates after applying the inverse bending transformation are given by:

$$x = X - \cos(\alpha) \cdot (R - r)$$

$$y = Y - \sin(\alpha) \cdot (R - r)$$

$$z = \gamma/k,$$

where α and k are bending parameters of the fitted superquadric, and R, r, and γ

are quantities computed from the bending parameters and the initial co-ordinates (X, Y, Z).

After applying this inverse bending transformation, the angle computation for bent-axis parts is the same as for straight-axis parts. The decision as to whether a part has a straight or bent axis is taken by applying the hierarchical tree classifier with the set of five features described in Chapter 4.

5.2 Axis – straight or bent?

We pointed out that using features derived from superquadric parameters, the s-s-id and s-c-id parts are often confused with their curved-axis counterparts. Since we use superquadric fitting to determine the principal axis, obviously the axis estimate also becomes erroneous because the fitting procedure finds "bent" axes for these two parts. We have therefore explored the use of intensity data to decide whether the axis is indeed straight or bent.

Lee and Stockman [58] have developed a method for capturing fused range and intensity images using the 100X White Scanner. A pair of range and intensity images is sensed separately and the two are brought into registration by applying a transformation matrix. The transformation is estimated by initially calibrating the system using a known object and manually entering the correspondence of several pairs of image points.

For our experiments we captured 5 pairs of range and intensity images of the "smooth" models of each of the 4 parts involved in the misclassification of axis shape, i.e., *s-s-id* and *s-c-id*, and the corresponding bent-axis parts, *b-s-id* and *b-c-id*. However, we did not apply the fusion procedure to put the range and intensity images into registration, because we do not use fused data at any time, i.e., we do not simultaneously need registered range and intensity values in the scene.

The intensity images alone are used to determine whether the part axis is straight or bent, as follows: first the intensity image is thresholded to get its silhouette. Edges are detected from the thresholded image, and line segments are fitted to the silhouette to get a polygonal approximation of the part contour. The silhouette is broken into pieces at points where the change in direction between adjacent line segments is "large" (greater than 20°). This breaks the silhouette into several components.

The following simple test for "concavity" of the silhouette is then applied: we join the end-points of each component and check whether the mid-point of the line joining the end-points lies inside or outside the object silhouette. If the mid-point lies outside the silhouette, that component is taken to be concave. If more than 10% of the total length of the silhouette is thus classified as concave, the entire silhouette is classified as concave, i.e., the part has a bent axis. Otherwise, the part is assumed to have a straight axis.

Figure 5.5 shows a straight-axis part and the extracted silhouette. Figure 5.6 shows the polygonal approximation of the silhouette. Similarly, Figure 5.7 shows a bent-axis part and the extracted silhouette, while Figure 5.8 shows the corresponding polygonal approximation.

We now briefly explain the polygonal approximation algorithm described by Pavlidis [66], which we adapted in our implementation. This suboptimal algorithm divides the data points into groups and approximates each group by a side of a polygon. For simplicity, the algorithm draws the side of the polygon between the endpoints of each group rather than searching for the optimal approximation. The algorithm is essentially a split-and-merge type algorithm. Each time the algorithm examines a small group of points of size k_0 , called the *hop distance*. The algorithm keeps track of a current line L_1 , and a new line L_2 , fitted to two groups of points. If the angle between L_1 and L_2 is "small," the two groups are merged and a line is fitted to the new larger group. On the other hand, a group of points is divided (split) into



(a)



Figure 5.5. Intensity image and silhouette of a straight-axis part. (a) Intensity image of the part. (b) Extracted silhouette of the part.



Figure 5.6. Polygonal approximation of the straight-axis part.



(a)



Figure 5.7. Intensity image and silhouette of a bent-axis part. (a) Intensity image of the part. (b) Extracted silhouette of the part.



Figure 5.8. Polygonal approximation of the bent-axis part.

smaller groups if the maximum deviation, measured by distance from the fitted line divided by the length of the line, is "large." The tolerances for angle and deviation are specified by parameters $\Delta \theta$ and Δd , respectively.

We used $k_0 = 7$, $\Delta \theta = 20^\circ$, and $\Delta d = 0.2$ in our implementation. These values were determined by trying different values for the three parameters and visually evaluating the polygonal approximations for selected images.

Each part description consists of a set of line segments with following attributes:

(x, y, a, l),

where x and y are the coordinates of the midpoint, a is the angle with respect to positive x-axis, and l is the length of the segment. We break the contour into separate components at points where the direction change between successive line segments is greater than 20°.

5.3 Experimental Results

Initial hypotheses about part identity arise when subgraphs of the scene SAG partially match part views in the catalog of views. Incorrect hypotheses are then eliminated by directly testing the variation of cross-section size of our geon-like parts. We have tested our method using 10 real range images of each of the "smooth"-surfaced parts described in the previous chapter.

To get the initial hypotheses about part identity, we need to construct the attributed SAG for each image of the "smooth" parts. However, we already pointed out that surface segmentation and curvature-based surface classification are not robust. In many cases, the surface segmentation is so poor as to be totally unusable. Hence, in order to test our method of part identification, we assumed perfect surface segmentation and surface classification of the smooth models, giving a set of hypothesized part identities for each image. This is assumed as the input data for the experiments. Thereafter we tested the cross-section variation as described in Section 5.1.

Table 5.1 shows the confusion matrix for 10 range images of each of the 12 parts. In general, except for the *s*-*c*-*id* and *s*-*s*-*id* parts, the overall classification performance is comparable to the superquadric-based approach described in Chapter 4. However, parts with increasing-and-decreasing cross-section size are sometimes mistaken for tapering parts. This is probably because of self-occlusion or shadowing, where only some portion of the side surfaces is visible. A striking improvement over the hierarchical tree classifier approach to part identification is that the *b*-*c*-*co* part is identified much more accurately. Actually, this is true even without assuming perfect surface segmentation and surface classification, because for this part, surface segmentation and classification are correctly performed in most cases. Identification is noticeably more accurate for straight-axis parts than for bent-axis parts.

The s-s-id and s-c-id parts have a very high misclassification rate. This is because the superquadric estimation method confuses them with their bent-axis counterparts, leading to error in estimation of the principal axis. We obtained 5 range and intensity image pairs for each of these two parts and their bent-axis counterparts. The views were chosen so that for bent-axis parts the silhouette was distinctly concave. Using the method described above, we then attempted to determine from the intensity image whether the part axis is really bent or straight. This could be done correctly in all 20 cases. While this by itself does not suffice to determine the actual part identity, it does provide an indication when the superquadric estimation routine mistakes straight-axis parts for bent-axis ones. Table 5.1. Confusion table for real images ("smooth" models)

s-s-t	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0	40.0	80.0
s-s-id	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	40.0	0.0
8-8-CO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	90.06	10.0	20.0
s- c-t	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30.0	100.0	0.0	0.0	0.0
s-c-id	0.0	0.0	0.0	0.0	0.0	0.0	0.0	40.0	0.0	0.0	0.0	0.0
s-c-co	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
b-s-t	20.0	0.0	0.0	30.0	40.0	70.0	0.0	0.0	0.0	0.0	0.0	0.0
b-s-id	0.0	0.0	0.0	0.0	60.0	30.0	0.0	0.0	0.0	0.0	10.0	0.0
b-s-co	0.0	0.0	0.0	70.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
b-c-t	20.0	30.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
b-c-id	0.0	70.0	0.0	0.0	0.0	0.0	0.0	20.0	0.0	0.0	0.0	0.0
b-c-co	60.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Shape	b-c-co	b-c-id	b-c-t	b-s-co	b-s-id	b-s-t	s-c-co	s-c-id	s-c-t	S-S-CO	s-s-id	s-s-t

5.4 Summary

We described an alternative method for determining part identities by directly testing the variation of cross-section size along the principal axis. This method succeeds in identifying 10 of the 12 parts. The main cases of misclassification occur when parts with straight axis and increasing-and-decreasing cross-section are mistaken for bent-axis parts by the superquadric fitting routine. For other 10 parts, classification performance is comparable to the method based on the use of superquadric parameters in conjunction with a hierarchical tree classifier. We are able to use intensity image data successfully to decide whether the axis is really bent or straight.

Chapters 4 and 5 presented two alternative methods of determining part identities. Both the methods were unreliable for the parts *s-s-id* and *s-c-id*, because the superquadric estimation confused these two parts with the corresponding bent-axis parts. However, the confusion matrix shown in Table 4.14 reveals that a simple heuristic can improve the reliability of the first method (which is based on the use of superquadric parameters in conjunction with a hierarchical tree classifier). If a given part is classified as *b-c-id* or *b-s-id*, we can use the intensity image to check whether the axis is really bent, as described earlier in this chapter. If in fact the axis is not bent, we can classify the part as *s-c-id* or *s-s-id*, respectively. No such simple heuristic is available for the second method. Hence, we will use the first method (which is based on the use of a hierarchical tree classifier), together with the above heuristic, as the procedure for part identification.

The flowchart of our complete system for obtaining a 3-D parts-based object representation is shown in Figure 5.5. The next chapter presents the results obtained by our system when tested with several real range images.



Figure 5.9. Complete system for obtaining parts and their identities

CHAPTER 6

Obtaining Parts-based Representation

The previous chapters presented methods for part segmentation and part identification. We pointed out that part segmentation relies on good surface segmentation, and fails if the surface segmentation is poor. It also fails in cases when two parts come together in such a way that the adjacency of the corresponding surfaces is not directly visible. Similarly, for part identification we presented confusion tables indicating some consistent types of misclassification.

This chapter shows the results of part segmentation as well as part identification for 12 real range images of various 3-D objects. The images include isolated geonlike objects as well as complicated multi-part objects with occlusion. The choice of objects was dictated by two considerations: firstly, major surfaces should be separated by distinct creases (in order to get a good surface segmentation and subsequent part segmentation). Secondly, the individual parts should be fairly elongated and not too thin (in order to obtain a good superquadric fit).

In all the examples, results of surface segmentation (from which the initial surface adjacency graph is derived) are shown in color. Nodes in the SAG are labeled "A," "B,", "C," and so on. Throughout all the examples, the above alphabetical


Figure 6.1. Correspondence between colors in the surface segmentation and alphabetical labels in the Surface Adjacency Graph.

labels always correspond to the same color shown in the surface segmentation. The correspondence between colors and the alphabetical labels is shown in Figure 6.1.

Different views of the most complicated object for which we have successfully performed part segmentation and identification are shown in Figures 6.3a, 6.10a, and 6.11a. This is a manufactured object consisting of five cylindrical parts. One major part is attached to the main body at an oblique angle, and an additional cylindrical part is attached to each of the three remaining ends of the two major parts. This object yielded three different 3-D parts-based representations, corresponding to the three different part segmentations obtained.

Part segmentation was performed by the method described in Chapter 3. Part identification was performed by two different methods. The first method used a hierarchical tree classifier with five features derived from estimated superquadric parameters, as discussed in Chapter 4. The second method applied tests for directly checking axis shape and cross-section variation along the axis, as described in Chapter

128

6.1 Test Image 1

5.

Figure 6.2a shows the range image of an object made up of a cylinder and a rectangular block.

6.1.1 Part Segmentation

Figure 6.2b shows the surface segmentation, and Figures 6.2c and 6.2d show the initial and final surface adjacency graphs, respectively. As can be seen from Figure 6.2d, the surface adjacency graph of Figure 6.2c is correctly broken into two subgraphs, corresponding to two parts, i.e., the rectangular block and the cylinder.

6.1.2 Part Identification

We fit superquadrics to these parts and extract the five statistical features. The hierarchical tree classifier correctly identifies the rectangular block as part s-s-co and the cylinder as s-c-co.

Part identification can also be performed successfully by the second method. The subgraph corresponding to the rectangular block can be accounted for (from the multi-view representation) as a straight-axis, straight-edged part of either constant or increasing cross-section. The other part can be accounted for either as a straight-axis, curved-edge part of constant or increasing cross-section, or as a straight-axis, straight-edged part of increasing-decreasing cross-section.

We observed that the subgraph corresponding to the rectangular block can be accounted for as a straight-axis, straight-edged part of either constant or increasing cross-section. The principal axis of this part is estimated by fitting a superquadric, and the histogram of angles of surface normals with respect to the axis direction is



Figure 6.2. Object made up of a cylinder and a rectangular block. (a) Range image of the object. (b) Surface segmentation showing 6 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). The small surface B has been removed, and the edge between the side of the cylinder and the top of the block (D-C) broken.

constructed. The 80% threshold is exceeded neither in the interval between 45° and 90°, nor in the interval between 90° and 135°. Hence the object is not inferred to have an increasing cross-section. Next, it is found that for 87% of the side surface points the angle between the surface normal and the axis direction is between 80° and 100°. Since the percentage is above the threshold of 80%, it is inferred that the part has a constant cross-section along the axis. From the surface types and the multi-view representation, this part was already known to have a straight axis and straight edges. Hence the part identity is known unambiguously because all three shape attributes are known.

Similarly, by fitting a superquadric to the other part (cylinder), we estimate its principal axis and construct the histogram of the angles of surface normals with respect to the axis direction. Once again, the object does not exceed the thresholds for increasing cross-section. However, 82% of the side surface points have a normal between 80° and 100° with respect to the axis. Hence this part is also inferred to have a constant cross-section. This information eliminates two of the three possibilities obtained earlier for the part identity. It must be a part with straight axis, curved edge, and constant cross-section.

6.2 Test Image 2

Figure 6.3a shows the range image of a complicated object made up of five cylindrical parts. This is the most complicated object for which we have so far performed part segmentation and part identification successfully.

6.2.1 Part Segmentation

Figure 6.3b shows the surface segmentation, and Figures 6.3c and 6.3d show the initial and final surface adjacency graphs, respectively. The surface surface segmentation is fairly good but not perfect, as can be seen from Figure 6.3b. The initial SAG has as many as four small surfaces which arise due to imperfect surface segmentation. These can be removed while retaining connectivity of the SAG. The remaining 5 "large" surfaces are all classified as having positive maximum curvature and zero negative curvature. The catalog of part views shows no example of two adjacent surfaces of this type. Hence, interestingly, the corresponding SAG links can be removed without even referring to the angles between these surfaces. This gives a correct part segmentation into 5 parts.

6.2.2 Part Identification

By fitting superquadrics to the five cylindrical parts and deriving the same 5 features, we are able to identify the parts as being of type s-c-co by using the hierarchical tree classifier.

Similarly, we are also able to identify the 5 parts correctly by using the distribution of normal directions with respect to the principal axis. Each of the parts can be accounted for (from the catalog of part views) either as a straight-axis, curved-edge part of constant or increasing cross-section, or as a straight-axis, straight-edged part of increasing-decreasing cross-section. The distribution of normal directions with respect to the principal axis eliminates the other possibilities, and we conclude that all 5 parts are of type s-c-co.

6.3 Test Image 3

Figure 6.4a is a scene with three separate objects which partially occlude one another. Surface segmentation (Figure 6.4b) results in an initial SAG with 7 nodes (Figure 6.4c). Note that the SAG is already unconnected because from the range values, the cube and cylinder are recognized as being separate non-touching objects. The



Figure 6.3. Complicated object made up of 5 cylindrical parts. (a) Range image of the object. (b) Surface segmentation showing 9 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). Four small surfaces (C,E,H,I) have been removed, and edges between cylindrical surfaces broken.

small object below the cube as well as a small surface on the cylinder are discarded since each of them is below 5% of the total surface area. This gives the final part segmentation into two objects (Figure 6.4d).

Part identification using either of the two methods is also successful. The cube is identified as s-s-co, and the cylinder as s-c-co.

6.4 Test Image 4

The object consists of two cylindrical parts. The surface segmentation of this object produces two small spurious surfaces which are discarded. Once again the part segmentation can be performed without referring to the angle between the two cylindrical surfaces, because the catalog of views has no example of two adjacent cylindrical surfaces in any single part.

Both part identification methods are also successful in determining that the parts are of type s-c-co.

6.5 Test Image 5

Figure 6.6a shows the range image from a pair of registered range and intensity images of a common household object, i.e., a bulb [58]. We simply use the range image. Surface segmentation gives just one surface (Figure 6.6b). The initial and final SAGs are identical (Figures 6.6c and 6.6d).

Both methods of part identification lead to the conclusion that that part is s-c-id.

6.6 Test Image 6

The funnel shown in Figure 6.7a is also the range image of a registered range-intensity pair. Surface segmentation (Figure 6.7b) leads to a correct part segmentation into



Figure 6.4. Scene with occlusion, containing three objects. (a) Range image of the scene, showing a cube, a cylindrical object, and (at top left) a small object used to support the cube. (b) Surface segmentation showing 7 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). Two small surfaces (C,E) have been removed.



Figure 6.5. Object made up of two cylindrical parts. (a) Range image of the object. (b) Surface segmentation showing 5 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). Two small surfaces (C,E) have been removed.



Figure 6.6. Everyday object: a bulb. (a) Range image of the object. (b) Surface segmentation showing 1 surface. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final).

A O (d)



Figure 6.7. Funnel made up of a cone and a cylinder. (a) Range image of the object. (b) Surface segmentation showing 2 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). The edge between the conical and cylindrical parts has been removed.

two parts (Figures 6.7c and 6.7d).

Part identification using either of the two methods is also performed correctly for both the parts. The cylinder is identified as *s-c-co*, and the cone as *s-c-t*.

6.7 Test Image 7

Figure 6.8 is another object made up of a rectangular block and a cylinder. Surface segmentation gives 4 surfaces (Figure 6.8b), of which one (the long, thin side surface





Figure 6.8. Object made up of a cylinder and a rectangular block. (a) Range image of the object. (b) Surface segmentation showing 4 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). The edge between the rectangular block and the cylinder has been removed.

of the rectangular block) is misclassified as a kind of saddle surface with negative maximum curvature and positive minimum curvature. Since the the catalog of part views has no such surface from any view of any part, the corresponding SAG edge is left unbroken.

Part identification is successful for both parts, using either method. The rectangular block is identified as s-s-co, and the cylinder as s-c-co.



Figure 6.9. Another view of the funnel. (a) Range image of the object. (b) Surface segmentation showing 1 surface. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final).

6.8 Test Image 8

Figure 6.9a shows another view of a funnel. In this case, surface segmentation (Figure 6.9b) completely eliminates the small cylindrical part. The part segmentation therefore finds just one part (the cone).

This part is identified successfully as s-c-t by both methods.

6.9 Test Image 9

Figure 6.10a shows a second view of the complicated object discussed in Section 6.4. Shadowing caused the non-availability of range data between the leftmost part and the main body of the object. The leftmost part therefore is treated as a separate object. Surface segmentation gives 5 surfaces as seen in Figure 6.10b, and as mentioned earlier, the initial SAG (Figure 6.10c) is itself unconnected because of the non-availability of range information. The final SAG (Figure 6.10d) shows three unconnected nodes. Note that the same object that was earlier interpreted as having 5 parts (in test example 2) is now interpreted as two different objects, one of which has 2 parts.

All three parts are correctly identified as s-c-co by either of the two methods.

6.10 Test Image 10

Yet another view of the same complicated object is shown in Figure 6.11a. Surface segmentation finds 7 surfaces (Figure 6.11b). Of these, 4 are small surfaces. Figure 6.11c shows the initial SAG, and the final SAG (Figure 6.11d) segments the object into 3 parts.

Either of the two methods is able to correctly identify each of the three parts as *s-c-co*.

6.11 Test Image 11

An object resembling a table-lamp is shown in Figure 6.12a. Surface segmentation finds 4 surfaces (Figure 6.12b). Figures 6.12c and 6.12d show the initial and final SAGs, respectively. The three parts are disjoint even in the initial SAG.

Both methods are able to correctly identify each of the three parts. The three



Figure 6.10. Complicated object made up of 5 cylindrical parts: second view. (a) Range image of the object. (b) Surface segmentation showing 5 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). One small surface has been removed, and edges between cylindrical surfaces broken.



(a)





Figure 6.11. Complicated object made up of 5 cylindrical parts: third view. (a) Range image of the object. (b) Surface segmentation showing 7 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). Four small surfaces have been removed, and edges between cylindrical surfaces broken.

parts are identified as s-s-co, s-c-co, and s-c-t.

6.12 Test Image 12

Figure 6.13a shows a view of a coffee-mug. Surface segmentation finds 2 surfaces (Figure 6.13b). Figure 6.13c shows the initial SAG. The surface corresponding to the curved handle is very small. It is removed in the final SAG, which is shown in Figure 6.13d.

Both methods are able to correctly identify the one remaining part as a cylinder (s-c-co).

6.13 Summary

Results of part segmentation and part identification for several real range images showed that our methods are quite successful in cases where occlusion by the same or a different object is not too severe.

In general, we have observed that part segmentation is successful whenever a good surface segmentation, together with a view of the adjacent surfaces where different parts join together, is available.

Part identification is the most reliable part of our system. Two methods were developed for part identification, and their performance is comparable. Assuming a "good" view where several large surfaces are visible, 10 of the 12 parts can be identified with high accuracy (close to 90% for real range images). A heuristic was also developed for recovering from misclassification of axis shape property by the binary tree classifier.

Two parts (s-c-id and s-s-id) have straight principal axes, but are often mistaken for their curved-axis counterparts by the superquadric estimation routines. We de-



Figure 6.12. View of an object resembling a table-lamp. (a) Range image of the object. (b) Surface segmentation showing 4 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). The initial and final SAGs are identical.



Figure 6.13. View of a coffee-mug. (a) Range image of the object. (b) Surface segmentation showing 2 surfaces. (c) Surface adjacency graph (initial). (d) Surface adjacency graph (final). One small surface (corresponding to the handle) has been removed.

veloped an intensity-based method for checking whether the principal axis is in fact straight or curved. This heuristic can be used to correct misclassification of the axis shape property. If the intensity-based heuristic indicates that the axis is straight, we assume that the axis is indeed straight even if the binary tree classifier decides that it is a bent-axis part.

CHAPTER 7

Conclusions and Future Research

In this dissertation, we described our work on obtaining a "generic" parts-based 3-D object representation. We used range data as the input, obtaining as the output a 3-D object representation based on 12 geon-like 3-D part primitives.

Multiple views of the 3-D part primitives were stored in order to help segment objects into component parts and identify those parts as instances of specific primitives. Major surfaces of the part primitives determined a small number of views to be stored in the multi-view representation.

The shape discrimination capability of superquadric parameters was tested under varying conditions of image resolution, model elongation, and noise. Two approaches were presented for identifying the parts subsequent to part segmentation. The first approach applied a hierarchical tree classifier using features derived from estimated superquadric parameters. The second approach drew direct inferences about shape attributes of the geon-like part primitives, using properties computed from the range image or the associated intensity image.

Results of part segmentation and identification were presented for several images. Generally, given a good surface segmentation, part segmentation is successful in cases where the adjacency of surfaces from two parts coming together is visible. Part identification is quite accurate for 10 of the 12 geon-like parts. The two remaining parts have straight axes but are often mistaken for their bent-axis counterparts. We developed a heuristic to correct such misclassifications by directly checking from the intensity image whether the axis is straight or curved.

Other research on obtaining a 3-D representation using geon-like parts [3, 22] has made simplifying assumptions such as the use of perfect line drawings, perfect part segmentation, and/or manual part segmentation. Our work, on the other hand, uses real range images as the input and computes the part segmentation. Part segmentation is often successful even if surface segmentation is imperfect; further, part identification is often successful even if part segmentation is imperfect.

One of the real objects for which we performed part segmentation and identification (the object made up of 5 cylindrical parts) is more "difficult" than any object for which Bergevin et al. [3] or Dickinson et al. [22] have presented results. Conversely, given a good view, we believe that our method would be successful for most objects dealt with by Bergevin et al. or Dickinson et al.

7.1 Contributions of the Thesis

Contributions of this thesis are as follows:

- We presented the first attempt (to our knowledge) to obtain, from range data, a 3-D parts-based object representation in terms of a *specific, but fairly large* set of geon-like 3-D parts.
- We proposed a method of *specifying* "generic" 3-D parts, i.e., by using Surface Adjacency Graphs (SAGs). Based on their SAGs, we derived a very compact multi-view representation of the part primitives. We presented a method of performing part segmentation from range images, given a good surface segmentation. This method for part segmentation has some advantages over approaches

that, in one form or another, make use solely of Hoffman and Richards' "principle of transversality" [47]. It can perform part segmentation in some cases where two parts coming together meet at a convex or "mixed" crease rather than a concave one.

• We proposed two methods to perform the part identification of isolated parts. We tested the shape discrimination capability of superquadric parameters under varying conditions of image resolution, model elongation, and noise. Lastly, we showed that the associated intensity image can be used to recover from some shortcomings of the purely range-image based method of part identification.

7.2 Shortcomings

- Part segmentation is not robust. Good surface segmentation is crucial for the success of our part segmentation method. But the surface segmentation (that of Flynn, based in turn on the earlier work of Hoffman) [33, 49] is purely datadriven, and often gives a surface segmentation which is so far from ideal that our subsequent methods fail completely. Part segmentation is indeed successful even in the presence of minor surface segmentation errors—but no mechanism exists for recovering from an unacceptably poor surface segmentation. Further, even if a perfect surface segmentation were available, part segmentation would fail in some cases—as pointed out below.
- Our part segmentation method does not use intensity data. Nor does it use information about shape of the occluding contour. Hence, it fails when two parts come together in such a way that the adjacency of the corresponding surfaces is not directly visible, even though in many such cases the intensity or occluding contour data clearly indicates the joining of two parts. In Chapter

3, we gave the example of an L-shaped object in which only a single surface, i.e., the L-shaped surface, was visible. The intensity image or occluding contour clearly suggested the presence of two parts. Nevertheless, our method cannot segment this object into two parts from such a viewpoint. Additionally, for this object, as mentioned in Chapter 3, even if two surfaces belonging to different parts had been visible, part segmentation would have been incomplete. The reason is that the single L-shaped surface is itself produced by the coming together of two parts, and our part segmentation method has no way to break up a single surface into pieces belonging to different objects. Many other such examples can be found.

• The part identification method gives only part identities and nothing more. It does not give complete details of the identified parts and their adjacency, e.g., "the *small end* of the tapered cylinder is attached to the *side surface* of the rectangular block," and so on.

7.3 Recommendations for Further Work

In our opinion, the most useful as well as most interesting problem for future work would be to develop top-down (part-model-based) methods for recovering from imperfect surface segmentations, and to successively improve the surface segmentation. Another crucial contribution would be to enhance our part segmentation algorithm to use overall shape or contour information instead of relying solely on surface adjacencies.

Flynn, [33] as well as Hoffman [49], have taken an almost entirely bottom-up, data-driven approach to surface segmentation using range data alone. However, the present availability of fused range and intensity data adds a valuable new dimension of local information. Further, the recent work on MRF-based edge detection [65] and wing detection [57], provides additional low-level model-based evidence regarding surface types and the correct locations of boundaries between surfaces. It should be interesting to combine this with the earlier, purely bottom-up approach with the aim of getting more reliable surface segmentation. An even higher-level model-based approach towards obtaining better surface segmentation might be to use the surface properties of initially detected surface patches to hypothesize about the part primitive to which they belong. We saw that surface type provides strong clues regarding part identity. This might be used to generate hypotheses about parts which may be present in the scene. Moreover, curvature values and curvature directions on surface patches give strong evidence about the likely orientations and parameters of parts, e.g., the radius of curvature and directions of the principal axes. Hypotheses about parts might in turn be used to predict which surface patches belong to the same surface of the same part, and might hence be good candidates for merging.

An essential improvement to our part segmentation method would be to make use of intensity, or at least occluding contour shape information. Range data has been useful in developing a method for part segmentation based on surface and angle attributes at surface adjacencies. However, as explained earlier, surface adjacency by itself does not suffice to detect the joining of two parts in some cases where it would be obvious from the shape, e.g., from the intensity image or occluding contour.

We presented two different methods for part identification. Tests were applied to directly test some shape attributes such as axis shape and variation of crosssection size along the part axis. Integrating these semi-independent judgements about part identity and about specific shape attributes would be very useful in improving the final reliability of part identification.

Lastly, building an object recognition system for range images using an enhanced version of this 3-D parts-based object representation module does seem practicable. The use of shape information as a criterion for part segmentation, rather than surface adjacencies alone, would be a particularly valuable enhancement.

APPENDICES

APPENDIX A

Parameters

A number of empirically-determined parameters are used in various modules and algorithms described in the thesis. We list these parameters in the order in which they are mentioned in the text.

A.1 Surface Classification

This topic is discussed in Section 3.2.1. A threshold value of 0.2 is used to determine whether the median principal curvatures are "close to zero." If the median curvature is less than -0.2, it is taken to be negative; if it is greater than 0.2 it is taken to be positive; else it is taken to be zero.

A.2 Constructing the SAG

Surfaces whose area is less than 5% of the total surface area are ignored after taking their connectivity into account.

A.3 Tree Classifier

The binary tree classifier is described in Section 4.3.2. A maximum error rate of 2.0% is used when constructing the decision tree using training data.

A.4 Direct test for Cross Section size

This material is presented in Section 5.1. If the angle between a surface normal and the principal axis is less than 45° or greater thaan 135°, it is not taken into account for finding the distribution of angles. If at least 80% of the angles are either between 45° and 90°, or between 90° and 135°, the cross-section is assumed to be increasing. Else, if at least 80% of the angles lie within 80° and 100°, the cross-section is assumed to be constant. Else, the cross-section is assumed to be increasing. The percentages mentioned are over only those points whose normals are not ignored due to the heuristic.

A.5 Polygonal Approximation

Details of the polygonal approximation algorithm are given in Section 5.2. The following parameters are used for constructing the polygonal approximation: Hop distance $k_0 = 7$, tolerance for distance deviation $\Delta d = 0.2$, tolerance for angle deviation $\Delta \theta = 20^{\circ}$.

A.6 Direct test for Axis shape

This topic is discussed in Section 5.2. The silhouette is broken into separate components at locations where the angle change between successive line segments is greater than 20°. The line joining the end-points is computed for each separate component. If the mid-point of this line lies outside the silhouette, the component is classified as concave. If more than 10% of the total length of the polygonal approximation belongs to components classified as "concave," the entire silhouette is taken to be concave, i.e., the part has a bent axis.

BIBLIOGRAPHY

BIBLIOGRAPHY

- R. Bajcsy and F. Solina. Three-Dimensional Object Representation Revisited. In Proc. First IEEE International Conference on Computer Vision, pages 231-240, London, 1987.
- [2] A. H. Barr. Superquadrics and Angle-preserving Transformations. Comput. Graphics Applicat., 1:11-23, 1981.
- [3] R. Bergevin and M. D. Levine. Extraction of Line Drawing Features for Object Recognition. Pattern Recognition, 25(3):319-334, March 1992.
- [4] Robert Bergevin and Martin D. Levine. Generic Object Recognition: Building Coarse 3D Descriptions from Line Drawings. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 68-74, Austin, November 1989.
- [5] Paul J. Besl and Ramesh C. Jain. Segmentation Through Variable-Order Surface Fitting. IEEE Trans. Pattern Analysis and Machine Intelligence, 9(2):167-192, 1988.
- [6] P.J. Besl and R.C. Jain. Three-Dimensional Object Recognition. Computing Surveys, 17:75-145, 1985.
- B. Bhanu and L. Nuttall. Recognition of 3-D Objects in Range Images Using a Butterfly Multiprocessor. Pattern Recognition, 22(1):49-64, 1989.
- [8] Irving Biederman. Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review, 94(2):115-147, 1987.
- [9] T. O. Binford. Visual Perception by Computer. In Proc. IEEE Conference on Systems and Control, Miami, 1971.
- [10] Thomas O. Binford. Survey of Model-Based Image Analysis Systems. International Journal of Robotics Research, 1(1):18-64, Spring 1982.

- [11] H. Blum. Biological Shape and Visual Science. J. Theoretical Biology, 38:205– 287, 1973.
- [12] K. Bowyer, D. Eggert, J. Stewman, and L. Stark. Developing the Aspect Graph Representation for Use in Image Understanding. In Proc. DARPA Image Understanding Workshop, pages 831-849, Palo Alto, 1989.
- [13] J.P. Brady, N. Nandhakumar, and J.K. Aggarwal. Recent progress in the recognition of objects from range data. In Proc. 9th Int. Conf. on Pattern Recognition, pages 85-92, Rome, November 1988.
- [14] Rodney A. Brooks. Model-Based Three-Dimensional Interpretations of Two-Dimensional Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(2):140-150, March 1983.
- [15] J. Callahan and R. Weiss. A Model for Describing Surface Shape. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 240-245, San Francisco, 1985.
- [16] I. Chakravarty and H. Freeman. Characteristic Views as a Basis for Three-Dimensional Object Recognition. In Proc. SPIE Conf. on Robot Vision, volume 336, pages 37-45, May 1982.
- [17] R.T. Chin and C.R. Dyer. Model-Based Recognition in Robot Vision. Computing Surveys, 18:67-108, 1986.
- [18] M.F. Costabile and G.G. Pieroni. Detecting Shape Correspondences by Using the Generalized Hough Transform. In Proc. Eighth International Conference on Pattern Recognition, pages 589-591, Paris, 1986.
- [19] E.R. Davies. Improved Localization in a Generalized Hough Scheme for the Detection of Straight Edges. Image and Vision Computing, 5:279-286, 1987.
- [20] Sven Dickinson. The Recovery and Recognition of Three-Dimensional Objects Using Part-Based Aspect Matching. Technical Report CAR-TR-572, Computer Vision Laboratory, University of Maryland, August 1991.
- [21] Sven J. Dickinson, Alex P. Pentland, and Azriel Rosenfeld. Qualitative 3-D Shape Reconstruction using Distributed Aspect Graph Matching. In Proc. 3rd IEEE International Conference on Computer Vision, Osaka, December 1990.

- [22] Sven J. Dickinson, Alex P. Pentland, and Azriel Rosenfeld. 3-D Shape Recovery Using Distributed Aspect Matching. *IEEE Transactions on Pattern Analysis* and Machine Intelligence, 14(2):174-198, February 1992.
- [23] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. Wiley, 1973.
- [24] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. Wiley, 1973.
- [25] Sahibsingh A. Dudani, Kenneth J. Breeding, and Robert B. McGhee. Aircraft Identification by Moment Invariants. *IEEE Transactions on Computers*, C-26(1):39-46, January 1977.
- [26] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing Arrangements of Lines and Hyperplanes with Applications. SIAM Journal on Computing, 15:341– 363, 1986.
- [27] D. Eggert and K. Bowyer. Computing the Orthographic Projection Aspect Graph of Solids of Revolution. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 102-108, Austin, 1989.
- [28] T.J. Fan, G. Medioni, and R. Nevatia. Surface Segmentation and Description from Curvature Features. In Proc. DARPA Image Understanding Workshop, pages 351-359, 1987.
- [29] G. Fekete and L.S. Davis. Property Spheres: a New Representation for 3-D Object Recognition. In Proc. IEEE Workshop on Computer Vision: Representation and Control, pages 192-201, Annapolis, 1984.
- [30] F.P. Ferrie, J. Lagarde, and P. Whaite. Darboux Frames, Snakes, and Super-Quadrics: Geometry From the Bottom Up. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 170-176, Austin, 1989.
- [31] F.P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and superquadrics: Geometry from the bottom-up. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 170-176, Austin, November 1989.
- [32] M. Fischler and R. Bolles. Random Consensus: A Paradigm for Model-fitting with Applications in Image Analysis and Automated Cartography. Communications of the ACM, 24:381-395, 1981.

- [33] Patrick J. Flynn. CAD-Based Computer Vision: Modeling and Recognition Strategies. PhD thesis, Department of Computer Science, Michigan State University, 1990.
- [34] Patrick J. Flynn and Anil K. Jain. On Reliable Curvature Estimation. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 110-116, San Diego, June 1989.
- [35] K. Fukunaga. Introduction to Statistical Pattern Recognitiion. Academic Press, 1972.
- [36] M. Gardner. The superellipse: A curve that lies between the ellipse and the rectangle. Scientific American, 213:222-234, 1965.
- [37] Z. Gigus, J. Canny, and R. Seidel. Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects. In Proc. Second IEEE International Conference on Computer Vision, pages 30-39, Tarpon Springs, 1988.
- [38] Z. Gigus and J. Malik. Computing the Aspect Graph for Line Drawings of Polyhedral Objects. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 654-661, Ann Arbor, 1988.
- [39] W. Eric L. Grimson and Tomás Lozano-Pérez. Model-Based Recognition and Localization from Sparse Range or Tactile Data. International Journal of Robotics Research, 3(3):3-35, Fall 1984.
- [40] W.E.L. Grimson and T. Lozano-Pérez. Localizing Overlapping Parts by Searching the Interpretation Tree. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(4):469-482, July 1987.
- [41] A.D. Gross and T.E. Boult. Error of Fit Measures for Recovering Parametric Solids. In Proc. Second IEEE International Conference on Computer Vision, pages 690-694, Tarpon Springs, 1988.
- [42] A. Gupta, L. Bogoni, and R. Bajcsy. Quantitative and Qualitative Measures for the Evaluation of the Superquadric Models. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 162-169, Austin, 1989.
- [43] Alok Gupta. Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, June 1991.

- [44] M. Hebert and T. Kanade. The 3-D Profile Method for Object Recognition. In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 458-463, San Francisco, 1985.
- [45] Yaron C. Hecker and Ruud M. Bolle. Is Geometric Hashing a Hough Transform? Technical Report RC 15202(#67767), IBM Research Division, T.J. Watson Research Center, 1989.
- [46] G. E. Hinton and L. M. Parsons. Scene-based and Viewer-centered Representations for Comparing Shapes. Cognition, 30:1-35, 1988.
- [47] D. Hoffman and W. Richards. Parts of recognition. In Stephen Pinker, editor, Visual Cognition. MIT Press, 1985.
- [48] D. Hoffman and W. Richards. Parts of Recognition. Cognition, 18:65-96, 1985.
- [49] Richard L. Hoffman. Object Recognition from Range Images. PhD thesis, Department of Computer Science, Michigan State University, 1986.
- [50] Richard L. Hoffman and Anil K. Jain. Segmentation and Classification of Range Images. *IEEE Transactions on Pattern Analysis and Machine Intelli*gence, PAMI-9(5):608-620, September 1987.
- [51] K. Ikeuchi. Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin-Picking Tasks. International Journal of Computer Vision, 1:145-165, 1987.
- [52] A.K. Jain and R. Hoffman. Evidence-Based Recognition of 3-D Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:783-802, 1988.
- [53] J. J. Koenderink and A. J. van Doorn. Internal Representation of Solid Shape with Respect to Vision. *Biological Cybernetics*, 32(4):211-216, 1979.
- [54] M.R. Korn and C.R. Dyer. 3-D Multiview Object Representations for Model-Based Object Recognition. Pattern Recognition, 20:91-103, 1987.
- [55] D.J. Kriegman and J. Ponce. Computing Exact Aspect Graphs of Curved Objects: Solids of Revolution. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 116-122, Austin, 1989.
- [56] Y. Lamdan and H.J. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. In Proc. Second IEEE International Conference on Computer Vision, pages 238-249, Tarpon Springs, 1988.
- [57] Greg C. Lee. Reconstruction of Line Drawing Graphs From Fused Range and Intensity Imagery. PhD thesis, Michigan State University, East Lansing, Michigan, 1992.
- [58] Greg C. Lee and George C. Stockman. Obtaining Registered Range and Intensity Images using the Technical Arts White Scanner. Technical report, PRIP Lab., Department of Computer Science, Michigan State University, 1991.
- [59] Wei-Chung Lin and Tsu-Wang Chen. CSG-Based Object Recognition Using Range Images. In Proc. 9th Int. Conf. on Pattern Recognition, pages 99-103, Rome, November 1988.
- [60] D.G. Lowe. Perceptual Organization and Visual Recognition. Kluwer, Boston, 1984.
- [61] D. Marr. Vision. Freeman, 1982.
- [62] D. Marr and H. K. Nishihara. Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. In Proc. Royal Society, London, ser. B, volume 200, pages 269-294, 1978.
- [63] M. Minsky. A Framework for Representing Knowledge. In P. H. Winston, editor, The Psychology of Computer Vision. McGraw-Hill, 1975.
- [64] Roger Munck-Fairwood. Recognition of Geometric Primitives Using Logic-Program and Probabilistic-Network Reasoning Methods. In SPIE Vol. 1708 Applications of AI X, pages 589–600, Orlando, April 1992.
- [65] Sateesha G. Nadabar. Markov Random Field Contextual Models in Computer Vision. PhD thesis, Michigan State University, East Lansing, Michigan, 1992.
- [66] T. Pavlidis. Algorithms for Graphics and Image Processing. Computer Science Press, 1982.
- [67] A. Pentland and B. Horowitz. Recovery of Nonrigid Motion and Structure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(7):730-742, July 1991.
- [68] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically Based Modeling and Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(7):715-729, July 1991.
- [69] A. P. Pentland. From Pixels to Predicates. Ablex, 1986.

- [70] Alex P. Pentland. Automatic extraction of deformable part models. International Journal of Computer Vision, 4:107–126, 1990.
- [71] A.P. Pentland. Perceptual Organization and the Representation of Natural Form. Artificial Intelligence, 28:293-331, 1986.
- [72] A.P. Pentland. Recognition by Parts. In Proc. First IEEE International Conference on Computer Vision, pages 612-620, London, 1987.
- [73] S. Pinker. Visual Cognition: An Introduction. In Visual Cognition. MIT Press, 1984.
- [74] Harry Plantinga and Charles R. Dyer. Visibility, Occlusion, and the Aspect Graph. Technical Report 736, Computer Science Department, University of Wisconsin-Madison, December 1987.
- [75] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William H. Vetterling. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, Cambridge, England, 1986.
- [76] Narayan S. Raja and Anil K. Jain. Obtaining Generic Parts from Range Data Using a Multi-view Representation. In SPIE Vol. 1708 Applications of AI X, pages 602-613, Orlando, April 1992.
- [77] Narayan S. Raja and Anil K. Jain. Recognizing Geons from Superquadrics fitted to Range Data. Image and Vision Computing, 10(3):179–190, special issue on Range Image Understanding, April 1992.
- [78] J.H. Rieger. On the Classification of Views of Piecewise Smooth Objects. Image and Vision Computing, 5:91-97, 1987.
- [79] I. K. Sethi and G. P. R. Sarvarayudu. Hierarchical Classifier Design using Mutual Information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 4(4):441-445, July 1982.
- [80] L.G. Shapiro, J.D. Moriarty, R.M. Haralick, and P.G. Mulgaonkar. Matching Three-Dimensional Objects Using a Relational Paradigm. *Pattern Recognition*, 17:385-405, 1984.
- [81] F. Solina and R. Bajcsy. Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(2):131-147, February 1990.

- [82] T. Sripradisvarakul and R. Jain. Generating Aspect Graphs for Curved Objects. In Proc. IEEE Workshop on Interpretation of 3D Scenes, pages 109–115, Austin, 1989.
- [83] J. Stewman and K. Bowyer. Creating the perspective projection aspect graph of polyhedral objects. In Proc. Second IEEE International Conference on Computer Vision, pages 494-500, Tarpon Springs, 1988.
- [84] J.H. Stewman and K.W. Bowyer. Aspect Graphs for Convex Planar-Face Objects. In Proc. IEEE Workshop on Computer Vision, pages 123-130, Miami Beach, 1987.
- [85] G. C. Stockman. Object Recognition and Localization Via Pose Clustering. Computer Vision, Graphics and Image Processing, 40:361-387, 1987.
- [86] George Stockman. Object Recognition. In Ramesh C. Jain and Anil K. Jain, editors, Analysis and Interpretation of Range Images, pages 225-253. Springer-Verlag, 1990.
- [87] Structural Dynamics Research Corporation. *I-DEAS User's Guide*. Milford, Ohio.
- [88] Technical Arts Corporation. 100X 3-Dimensional Scanner: User's Manual and Application Programming Guide. Redmond, Washington.
- [89] D. Terzopoulos and D. Metaxas. Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics. *IEEE Transactions on Pattern Analysis* and Machine Intelligence, 13(7):703-714, July 1991.
- [90] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking Models for 3-D Object Reconstruction. Int. J. Comput. Vision, 1(3), 1988.
- [91] S. Ullman. An Approach to Object Recognition: Aligning Pictorial Descriptions. Technical Report A.I. Memo No. 931, M.I.T. AI Laboratory, 1986.
- [92] B.C. Vemuri, A. Mitiche, and J.K. Aggarwal. Curvature-Based Representation of Objects from Range Data. Image and Vision Computing, 4(2):107-114, May 1986.
- [93] N.A. Watts. Calculating the Principal Views of a Polyhedron. In Proc. Ninth International Conference on Pattern Recognition, pages 316-322, Rome, 1988.

t ,

