May 1989

Part Description and Segmentation Using Contour, Surface and Volumetric Primitives

Alok Gupta
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-89-33.

Part Description and Segmentation Using Contour, Surface and Volumetric Primitives

Abstract

The problem of part definition, description, and decomposition is central to the shape recognition systems. The Ultimate goal of segmenting range images into meaningful parts and objects has proved to be very difficult to realize, mainly due to the isolation of the *segmentation* problem from the issue of *representation*. We propose a paradigm for part description and segmentation by integration of contour, surface, and volumetric primitives. Unlike previous approaches, we have used geometric properties derived from both boundary-based (surface contours and occluding contours), and primitive-based (quadric patches and superquadric models) representations to *define* and *recover* part-whole relationships, without *a priori* knowledge about the objects or object domain. The object shape is described at three levels of complexity, each contributing to the overall shape. Our approach can be summarized as answering the following question : Given that we have all three different modules for extracting volume, surface and boundary properties, how should they be *invoked*, *evaluated* and *integrated*? Volume and boundary fitting, and surface description are performed in parallel to incorporate the best of the coarse to fine and fine to coarse segmentation strategy. The process involves feedback between the segmentor (the Control Module) and individual shape description modules. The control module *evaluates* the intermediate descriptions and formulates hypotheses about parts. Hypotheses are further tested by the segmentor and the descriptors. The descriptions thus obtained are independent of position, orientation, scale, domain and domain properties, and are based purely on geometric considerations. They are extremely useful for the high level domain dependent symbolic reasoning processes, which need not deal with tremendous amount of data, but only with a rich description of data in terms of primitives recovered at various levels of complexity.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-89-33.

Part Description and Segmentation Using Contour, Surface and Volumetric Primitives
(Dissertation Proposal)

MS-CIS-89-33
GRASP LAB 180

Alok Gupta

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

May 1989

ACKNOWLEDGEMENTS:
This research was supported in part by Airforce grant AFOSR 88-0244, AfOSR 88-0966, NASA NAG5-1045, ONR SB-35923-0, NIH 1-RO1-NS-23636-01, NSF grants INT85-14199, MCS-8219196-CER, IRI84-10413-AO2 DARPA grant N00014-85-K-0018, US Army grants DAA29-84-K-0061, DAA29-84-9-0027, ARPA N0014-88-K-0630, NATO grant No.0224/85, DuPont Corp. by Sandia 75-1055, Post Office, IBM Corp. and LORD Corp.

Part Description and Segmentation Using Contour, Surface and Volumetric Primitives

Alok Gupta
GRASP Laboratory
Computer and Information Science Department
University of Pennsylvania

Dissertation Proposal
Supervised by Dr. Ruzena Bajcsy

May 1, 1989

Abstract

The problem of part definition, description, and decomposition is central to the shape recognition systems. The Ultimate goal of segmenting range images into meaningful parts and objects has proved to be very difficult to realize, mainly due to the isolation of the *segmentation* problem from the issue of *representation.* We propose a paradigm for part description and segmentation by integration of contour, surface, and volumetric primitives. Unlike previous approaches, we have used geometric properties derived from both boundary-based (surface contours and occluding contours), and primitive-based (quadric patches and superquadric models) representations to *define* and *recover* part-whole relationships, without *a priori* knowledge about the objects or object domain. The object shape is described at three levels of complexity, each contributing to the overall shape. Our approach can be summarized as answering the following question : Given that we have all three different modules for extracting volume, surface and boundary properties, how should they be *invoked*, *evaluated* and *integrated*? Volume and boundary fitting, and surface description are performed in parallel to incorporate the best of the coarse to fine and fine to coarse segmentation strategy. The process involves feedback between the segmentor (the Control Module) and individual shape description modules. The control module *evaluates* the intermediate descriptions and formulates hypotheses about parts. Hypotheses are further tested by the segmentor and the descriptors. The descriptions thus obtained are independent of position, orientation, scale, domain and domain properties, and are based purely on geometric considerations. They are extremely useful for the high level domain dependent symbolic reasoning processes, which need not deal with tremendous amount of data, but only with a rich description of data in terms of primitives recovered at various levels of complexity.

Contents

Chapter 1

Introduction

For visual discrimination, shape plays a very important role. Human beings exhibit remarkable abilities to simplify the visual input without bringing in domain knowledge or functionality into consideration. A robot using vision for navigation or recognizing objects, has to similarly simplify the visual input to the level that is required for the specific task. To simplify means to partition images into entities that correspond to individual regions, objects and parts in the real world and to describe those entities only in detail sufficient for performing a required task. Usually the first level of simplification entails obtaining part descriptions based on the properties that are independent of the position, orientation, scale and the work domain. Physical shape of an object is an important characteristic that allows us to discriminate between two otherwise identical objects, for example a ball from cube of same color and texture. Shape is the outward appearance or form of an object defined by its boundaries and surfaces. It is therefore possible to define an object's physical shape by geometric primitives. From the perspective of shape, objects in the real world represent a complex conglomeration of primitive shapes. The primary objective of a shape recognition system is to derive a structured description of complex objects in terms of primitive shapes. The resulting decomposition into parts is very useful for the high level symbolic reasoning object-recognition processes, which can attach domain specific labels to the parts, and reason at a level where the visual input is structured in terms of primitives, rather than cope with the difficulties of low level vision and huge pile of unstructured data.

The proposal is organized in the following manner. In this chapter, we formally define the shape recognition problem, and give a philosophical overview of the problem. Shape primitives and segmentation are discussed in detail in chapter 2 and individual shape primitives are discussed in chapter 3,4, and 5. Chapter 6 describes our proposed method of shape description.

1.1 Problem Statement

The goal of this research is to obtain structured shape descriptions of complex three-dimensional objects in range images in terms of significant parts defined by a set of primitives without *a priori* knowledge about the object or the object domain. By "significant" we mean that the part boundaries are of physical, perceptual or differential geometric significance and that part decomposition is *natural.*

This brings in the vital issues of part definition, description and decomposition, each of which addresses the very basis of our research. At the outset, it is important to note that the problem of shape description and decomposition has proved to be extremely difficult mainly because the researchers have either tackled each of the components separately or limited their description to one primitive. We present arguments that the issue of part description and part segmentation[1] are related and *have* to be considered together. This observation leads us to propose three primitives for shape representation, that describe shape at three levels of complexity and participate actively in the segmentation procedure. After providing motivation for the choice of primitives, we propose to integrate them to produce the final description.

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

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

The first five questions constitute the problem analysis phase, where we attempt to formalize the problem in the most general sense. The last two questions involve important computational and integration issues that will determine the eventual robustness of the system. In this chapter

[1] We will use the terms segmentation and decomposition interchangeably.

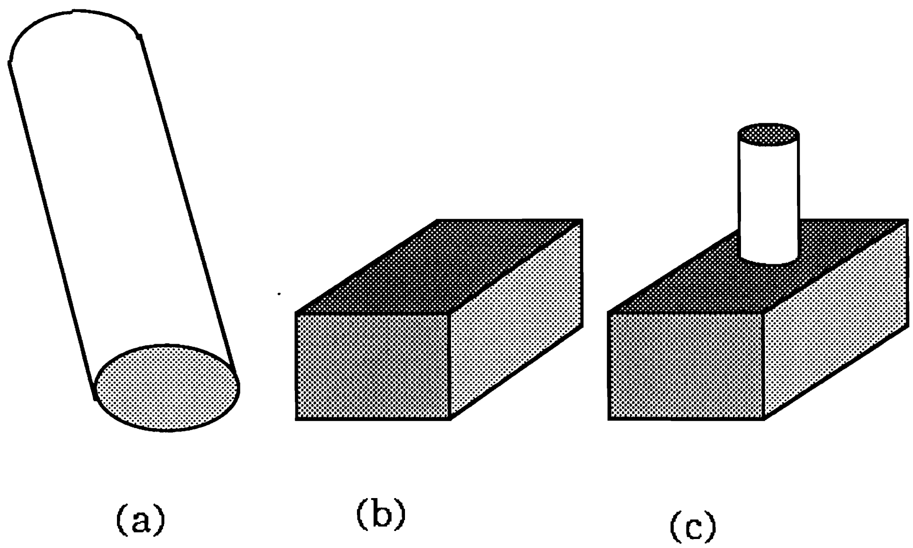


Figure 1.1: 3-D Parts : A cylinder (a)is a single volumetric part consisting of two surface patches. The Box (b) is perceived as a single volumetric part, while three planar patches are seen at surface level. The composite object (c) has two distinct volumetric parts, separated by a concavity at the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

we lay the foundation of our proposed work by giving a more general definition of the problem. Other issues will be dealt with in the subsequent chapters.

1.2 What are Parts?

Webster's dictionary defines a part as one of the portions into which something is or is regarded as divided and which together constitute the whole. Arnheim [Arn74] notes that in a quantitative sense, any section of whole can be a part. But this definition does not preserve *structure*. Partitioning by ignoring structure is not of much use in vision [WT83, HR85, Pen87, Arn74].

Part definition ultimately depends on the reliability, versatility and computability constraints imposed by the task of shape recognition and may not be unique [HR85]. It is therefore difficult to give a general definition of *part* in the context of shape recognition. However, a working definition would define a part as an easily describable and recognizable portion of a complex shape that is invariant to minor changes in viewpoint (figure 1.1). It brings the notion of description into part definition, emphasizing the fact that two are interrelated. The idea of partitioning a complex object into describable parts is not new in computer vision. It differs in the choice of primitives and the way segmentation is carried out. Traditionally [BH87, NB77, HR85] part definition has been

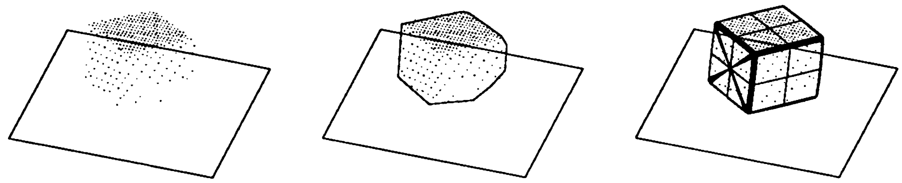
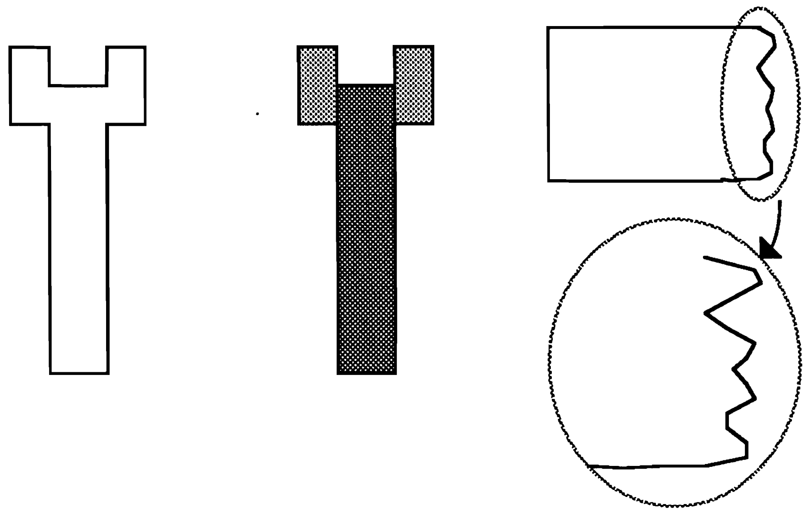


Figure 1.2: Edge and contour models are of lower granularity : It is difficult to conclude from occluding contour model that the object is roughly in a shape of cube. Volumetric models are less sensitive to missing information.

either *primitive*-based or *boundary*-based. In the literature, primitive-based approaches [AB73, NB77, SB78] have defined objects by cylindrical, polyhedral, conical or spherical shapes. The objective of such systems is to fit parts of complex objects with models in the shape vocabulary. Boundary-based approaches [HR85, BH87, KvD82, Bie85] define parts by outlining the boundaries on surfaces. Beiderman[Bie85] has emphasized the perceptual basis for part decomposition based on Gestalt principles (nonaccidental properties of 2D projection of 3D objects). Parts should be defined by *continuity*[Bin82] and *uniformity* [HR85]. In shape decomposition, one tries to follow the principle of orderliness, which means - partitioning things in the simplest possible way. Such partitioning normally reflects the structure of the physical world quite well due to the principle of parsimony [Arn74].

Bennett and Hoffman [BH87] have argued that a primitive based part definition confuses the problem of part definition with the separate problem of part description. We are considering them to be interdependent, parts are defined the way they are described by shape primitives. By including surfaces as primitives, we automatically include the boundary-based approach. In fact, we go a step further, by asserting that *primitive-description* has to go hand-in-hand with the *boundary-description.* However, it might not *always* be possible to obtain complete primitive-based description of arbitrary objects for *all* the parts. Surface primitives ensure that we obtain a part description at a level lower and less global than volumetric primitive. Volumetric primitives being global and shape dependent *do not* account for all the boundaries on the surface. Thus the part structure captured at surface level is more detailed but of lower granularity than that captured at volumetric level. Similarly the part description at occluding contour level is of even lower granularity (figure 1.2).

An important issue related to the part-whole relationships is the issue of *part* versus *detail* . That a portion of the whole merits an independent description as a part or can be considered a mere detail is a matter of scale in the bottom-up approach we are adopting. In figure 1.3 object



(a) (b) (c)

Figure 1.3: **Part versus detail** :Perception of parts depends on scale of the part with respect to the whole. The spanner shape (a) needs decomposition into parts (b). While the jagged boundary on one side of the object (c) can be ignored as a detail. However, at a finer scale, details become parts.

1 appears to have parts while the wiggles on one side of the object 2 appear to be details that do not need part level description. However by increasing the scale of the wiggliness with respect to the length of whole we get them as significant parts.

1.3 Segmentation Versus Representation

Decomposition into parts, units or primitives is the basis of scientific methodology. Because of the limits on how much information we can process at a time, we have to simplify and view the world at various levels of abstraction. We are proposing to decompose complex objects into the constituent parts based on the shape. Many reasons have been advanced in favor of such a decomposition. A recognition-by-parts approach is not sensitive to occlusion and is extremely powerful in handling countless configurations of articulated objects. A description in terms of basic shape primitives is more efficient, parsimonious in space consumption, and facilitates structured description of the world. These arguments are supported by the principles of perceptual organization [Bie85].

In computer vision literature the partitioning of images and description of individual parts is called segmentation and shape representation. We have presented arguments in [BSG88] that the problem of segmentation and representation are related and *have* to be treated simultaneously.

Solving any one of those two problems separately is very difficult. On the other hand, if any one of the two problems is solved first, the other one becomes much easier. For example, if the image is correctly divided into parts, the subsequent shape description of those parts gets easier. The opposite is also true when the shapes of parts are known, the partitioning of the image gets simpler. Since neither of them can be easily solved in isolation, at least not on the first try, we argue that they should interact to guide and correct each other. Hence, segmentation and shape recovery should not be studied separately. The complete visual interpretation problem is even more complex because the initial data acquisition process cannot be separated from the later segmentation and shape representation. How data acquisition can interact with the interpretation stage is investigated in computer vision under the heading of active vision [Baj89].

1.4 Shape Primitives

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

1. **Volumetric level** : Primitives capable of modeling parts in three dimensions are needed to describe global shape of parts.

2. **Surface level** : Surface primitives describe *internal surface boundaries* and *surface patches* which are difficult to model by volumetric primitives, but are vital source of information about recovering part structure.

3. **Occluding Contour level** : The Occluding contour encodes the 3-D shape of parts projected on the image plane.

This hierarchy of shape primitives allows one to obtain shape descriptions at volumetric, surface and occluding contour level. Since, both boundary-based and primitive-based primitives are included in our vocabulary, the representation is expressive and robust. It is clear that no one primitive will always capture all the details of shape. For example, if it is not possible to model parts with the selected volumetric primitive, an approximation at volumetric level can be obtained, with more detailed description at surface level. Thus, completeness requirement for a general representation is satisfied by obtaining hierarchical descriptions.

The criteria for selection of shape primitives have been studied extensively by vision researchers [Bra83, BA84, Mar82, Bin82, Rao88]. The shape primitives should be invariant to rotation, translation, and scale. *Accessibility*, defined as computability of the primitive is essential, since our goal is to recover the structure from the input. *Stability* of the primitive with respect to minor changes due to noise or viewpoint, with respect to scale and configuration is important to generate consistent representations. While small changes in scale should not create major changes in description, a multi-scale representation should be possible, for example, parts become detail as the scale is increased. The primitives should have local support, so that occluded parts can still be described and recognized when matching is performed against stored descriptions.

Low level models like contours and edges have low granularity (see figure 1.2)and are too local to capture or make use of the gross structure of the world. They are sensitive to local changes and difficult to put together in a global context. However, this characteristic allows them to capture local details of shape that would be missed or smoothed out by more global primitives. When analyzed as a whole, contour primitives have the remarkable capability of describing global shape and segmenting planar shapes into parts.

The next level of shape description is achieved by describing local and overall surface characteristics. Surfaces play important role in human perception of shape. A lot of effort in computer vision has been spent on describing complex surfaces as piecewise continuous patches. In order to arrive at a global interpretation, a surface representation scheme that combines relevant surface contours with the surface patches is needed.

Three dimensional primitives like generalized cylinders and cones, polyhedral models, 3-D Smoothed local symmetries [Bra83], and 3-D symmetric axis transform [NP85] have been used by model based vision systems. However, the power of representation varies from model to model. A model allowing deformations is likely to describe objects with fewer primitives than a rigid model which will need more instances to approximate the object. As we will see later, volumetric primitives are essential to generate compact object-centered descriptions and to define global part-structure. Superquadric models, our choice of volumetric primitives, provide object centered descriptions, thus allowing surface and contour level descriptions to *attach* to the local coordinate system, facilitating ease in representation and model-based matching.

1.5 The Segmentation Problem

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

Each of the shape primitive can independently describe the data. The occluding contour-based segmentation is widely studied in pattern recognition and computer vision as 2-D shape recognition problem [Pav77, Sha80, AB86]. Surface based approaches have been popular with model-based vision systems, as they have local support, and allow 3-D objects to be modeled as collection of surfaces. Volumetric models have proved to be most difficult to recover from image data. Some researchers have used a combination of features to *model* domain specific objects [KD98, Bro83], exploiting the robustness achieved by combining descriptions at different levels. To facilitate segmentation we believe that for a general purpose vision system one needs volumetric, surface and boundary shape primitives. Difficulty in recovering volumetric models in intensity images is experienced due to the loss of depth information. But the problem has not proved to be any easier even with the availability of depth information [NB77, KD98, Sol87, BG87, Rao88, SB78]. We are considering the input to be dense depth maps, scanned by an active range scanner from a single viewpoint. No information about scanner geometry or viewpoint is required.

Model based vision systems match the available models in the model database with hypothesized instances of models in the image data. Object models typically used in vision are built as a structured hierarchy of primitive part-models. Since we are addressing the problem at the level of shape-definition only, and not at the object-definition level, we do not have the high level models that restrict the part-models to a particular configuration. Therefore, the typical model-based vision strategy is too restrictive to be of any use for part segmentation. The essential difference between shape recognition problem and the model-based approach is that we are looking for instances of part-models and not object-models that constrain the part-models to configure in a known order.

Shape description systems based on individual primitives follow the approach outlined in figure 1.4a. The shape description is achieved in terms of surfaces or volumetric primitives. Some robust methods have employed [BJ86a] feedback between final description stage and lower levels. Our proposed approach (figure 1.4b) is to obtain shape description at the level of all the primitives, with feedback between the descriptor modules and the control module. We will discuss our approach in detail in the chapter describing the control module.

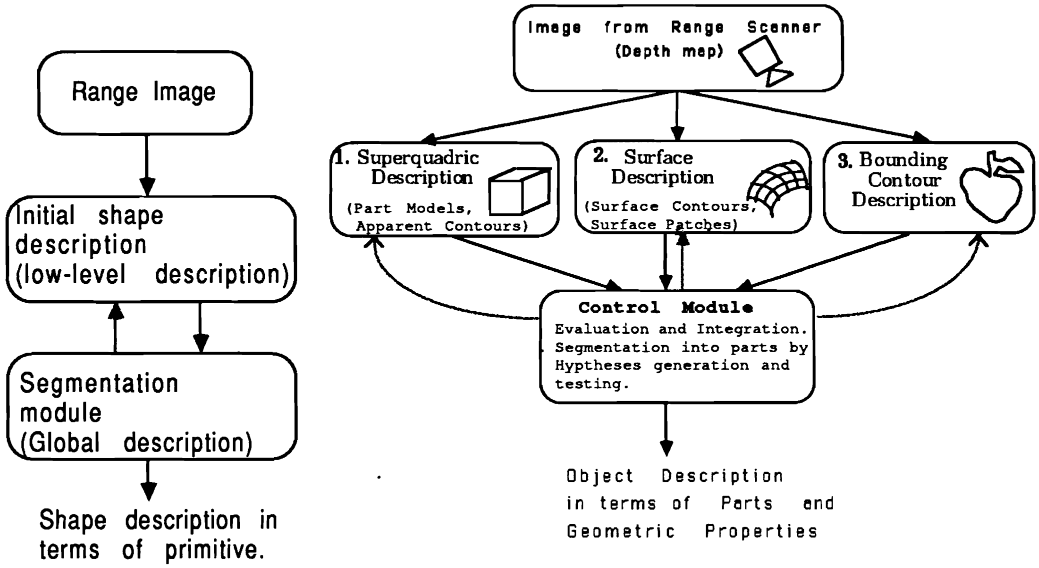


Figure 1.4: **Block diagram of a typical Shape recognition system based on single primitive (left) and our system based on primitives at different levels (right).**

1.6 The Control Structure

Given the shape primitives and the modules to recover them, a control strategy is needed to invoke, evaluate and integrate them. The control structure forms the heart of the shape recognition system. The influencing factors on the design of the control strategy are the goal of the vision system, the scene complexity and the dimensionality of the objects in the scene. Typical goals of a vision system are locating obstacles in a scene for mobile robot navigation, enabling manipulation with robot hands or identifying objects by matching recovered shape descriptions to a given data base. The complexity problem is to find out whether the scene contains a single convex object, a non-convex object consisting of parts, or more than one object. Scene classification according to its complexity can greatly simplify the control structure for interpretation. Establishing dimensionality is to find out if a scene can be interpreted only in terms of volumetric models, flat-like models or rod-like models. Global measures such as center of gravity and moments of inertia give such estimates. The importance of dimensionality parameters is that, depending on the dimensionality, different geometric primitives come into play. For example, in the case of a scene with flat-like objects only, surface primitives should be sufficient and no volumetric primitives would be required.

Since, we are dealing with objects of arbitrary complexity, a general control structure is required. The different shape description modules (figure 1.4b) have to interact with one another to evaluate the recovered description at surface and contour levels. This matching will give "difference mea-

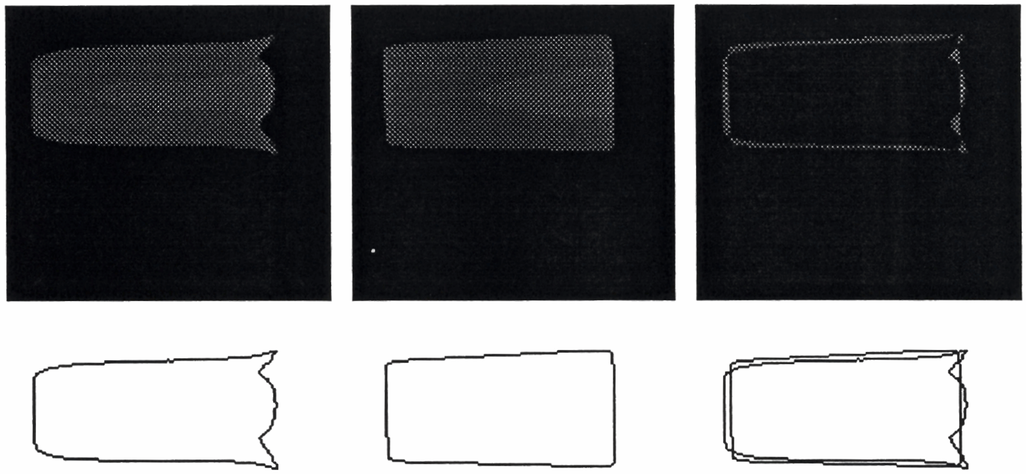


Figure 1.5: **Volumetric and Occluding contour description of a vase:** Top : Range image, projection of superquadric model on image plane, difference between the two. Bottom : Occluding contour of image, apparent contour of the superquadric model, difference between the two.

sures" of *goodness-of-description* for individual primitives. We will later see that both qualitative and quantitative measures are obtained by matching the recovered model against input data. Based on these measures, the control module will either accept the current level of description or generate hypotheses about potential "parts", for which better description can be obtained. Figure 1.5 shows the results of initial description obtained by superquadrics and bounding contour primitive. The description obtained at superquadric level can be compared at surface level and at the bounding contour level. The bounding contour of the object agree with that of the model on most of the object, except for the details, which are captured by the contour primitive only. The surface is approximated globally as cylindrical by the volumetric primitive, which when compared with the surface points indicates that the description is adequate. However, detailed surface description can only be obtained at surface level and not at volumetric level.

1.7 Input and Assumptions

We assume that a complete depth map of a scene is given. Obtaining a depth map is one of the stated goals of low level vision modules, such as stereo and shape from shading. The computation of the depth map or 2-1/2D sketch was once considered to be the harder part and that image interpretation from there on would be easy. Although dense and accurate depth maps are now available from laser range scanners, the interpretation of those images is still difficult. A depth map as the starting point, obtained either with a laser scanner or from low level image techniques on gray level images, does not simplify neither segmentation nor shape recovery to any large extent. For our research we use range images taken from a single viewpoint.

Range images are dense depth maps measuring the distance of the physical surface from a known reference plane. application. Magnetic resonance imaging systems give true 3-D images, i.e, all the points in 3-D space are specified. Structured lighting systems scan the scene with a laser stripe to obtain depth information of the visible surface in a calibrated workspace. The range images dealt with in this work are of $z(x, y)$ type, where each pixel gives the Z-depth at the coordinate x and y. Representation of range images is just like that of reflectance images. A two dimensional array of depth values specifying (x,y,z) coordinates with respect to a known coordinate frame is enough for most applications. Due to self occlusion, not all points on the surface of an object are given. Since the supporting surface is fixed, range points from the support can be easily removed at the start of scene interpretation.

Chapter 2

Shape Primitives and Segmentation

2.1 The Choice of Primitives

The choice of primitives can be guided by some general requirements such as a unique decomposition into primitives, that the primitives cannot be further decomposed or that the set of primitives is complete. Some of the shape representation criteria are designed primarily to facilitate object recognition when models recovered from images are matched to a model data base. We have outlined the different criteria for shape representation in the previous chapter. Unfortunately, all those principles have not been applied to any general shape representation scheme for 3-D objects. A review of computer vision literature which reveals the large variety of geometrical primitives that were investigated for their applicability to shape representation is a testimony to the difficulty of shape description [BJ86b]

Another discipline involved in representing shape is computer graphics, but from a synthesis (generating) point of view. Some commonly used 3-D representations in graphics are wire-frame representation, constructive solid geometry representation, spatial-occupancy representation, voxel representation, octree representation, and different surface patch representations. Splines are used for surface boundary representation. But requirements for shape primitives in computer vision are different from the ones for computer graphics. Shape primitives for computer vision must enable the analysis (decomposition) of shape. Common shape primitives for volume representation are polyhedra, spheres, generalized cylinders, and parametric representations such as superquadrics. Different orders of surface patches (planar, quadratic, cubic) are used for surface representation. For boundary description one can use linear, circular or other second order models for piecewise approximation, and higher order spline descriptions. In the rest of this section we will discuss what influences the selection of shape primitives in computer vision.

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

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

A concise model which adequately describes the data will enable partitioning or segmentation of images into right parts and ignore noise and details. Such a model will have primitive shape models capable of describing shape at both low and high levels. In everyday life, people use a default level of representation, called basic categories [Ros78]. Basic categories seem to follow natural breaks in the structure of the world which is determined by part configuration [TH84]. Shape representation on the part level is then very suitable for reasoning about the objects and their relations in a scene. For part level description in vision, a vocabulary of a limited number of qualitatively different shape primitives [Bie85] and different parametric shape models have been proposed. Parametric models describe the differences between parts by changing the internal model parameters. In computer vision, the most well known parametric models suitable for representing parts are generalized cylinders but superquadrics with global deformations seem to have some important advantages when it comes to model recovery [Pen86, BS87] It is sometimes possible to know a priori that a certain class of geometric models is sufficient to describe observed data. Another possibility is to somehow evaluate the complexity of the scene and the dimensionality of the objects in the scene. Knowing the complexity of the scene can greatly simplify the control structure for segmentation and shape recovery while knowing the dimensionality of objects simplifies the selection of shape models.

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

2.2 Our Choice of Primitives

Parametric models like generalized cylinders and their derivatives have been used as volumetric primitives by vision researchers because they give compact overconstrained estimate of overall shape. This overconstraint comes from using models defined by a few parameters to describe a large set of 3-D points. Researchers have developed rule-based systems to recover generalized cylinders from image data. In such systems monitoring of progress is difficult and a direct evaluation criteria of results is not available. Also, they can recover only a restricted subset of generalized cylinders, such as linear straight homogeneous generalized cylinders. The Volumetric primitives we are proposing to use are the *deformable superquadric part-models.* Superquadrics (figure 2.1) have been used in vision [Pen86, Pen87, Sol87] to represent natural part-structure. Pentland [Pen87] argues that superquadric part-models possess *descriptive adequacy* though they do not account for

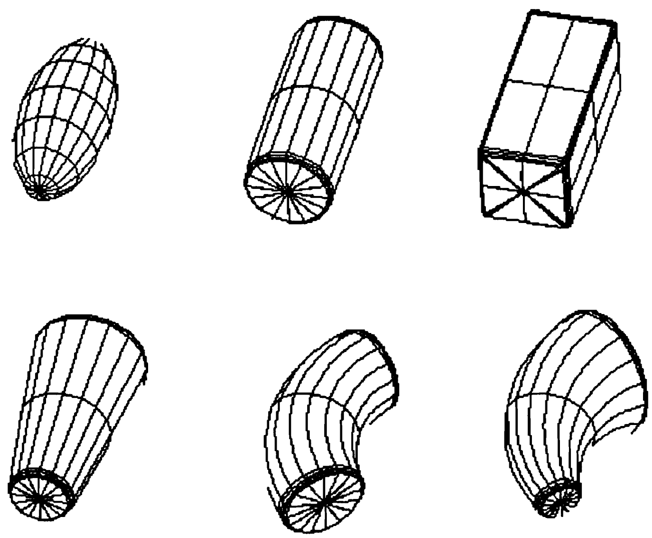
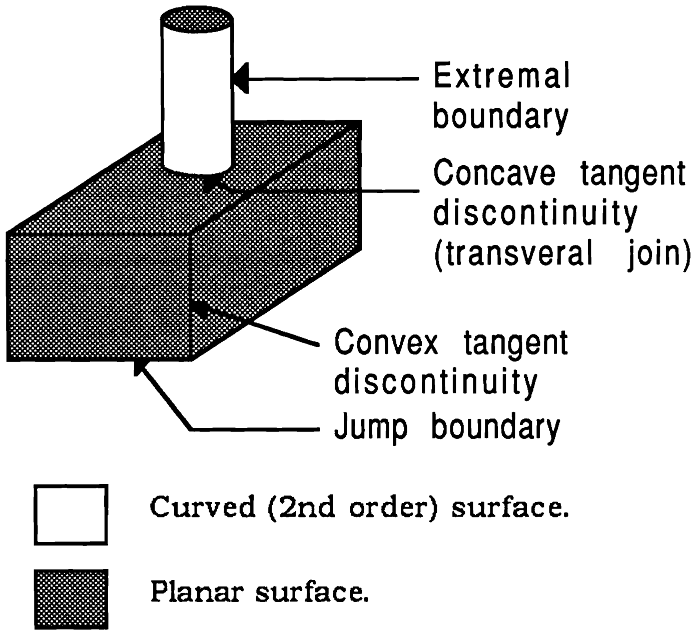


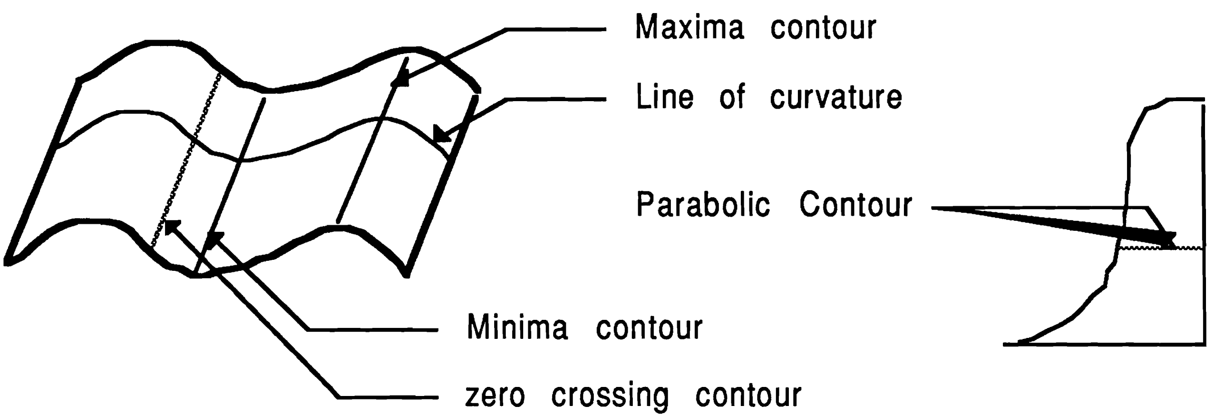
Figure 2.1: **Volumetric primitive** : Superquadrics. Clockwise from top : **ellipsoid, cylinder, box, tapered model, bent model, tapered and bent model**

every detail of the image data. Also, they are *stable* with respect to scale, noise, and configuration. Solina [Sol87] has developed a model recovery procedure to fit tapered and bent models to given data. We are proposing to use the deformed superquadric model to describe volumetric descriptions of parts.

Superquadric models use least squares minimization for recovery of their parameters. An important advantage for ease of model recovery is that the superquadric surface is defined by an analytic function, differentiable everywhere. Superquadric shapes form a subclass of shapes describable by generalized cylinders. Shape deformations like bending and tapering can be defined with global parametric deformations. Superquadrics with parametric deformations encompass a large variety of natural shapes yet are simple enough to be solved for their parameters. Due to their built-in symmetry, superquadric models predict the shape of occluded parts conforming with the principle of parsimony - among several hypotheses select the simplest [Gom72]. Except for bending, the shape vocabulary consists of convex objects. How can we model objects with *concavities, cavities* and *holes*? Cavities form when a significant chunk of volume is taken away from the object leaving a dent enclosed by the remaining object (bowl or cup). Solina [Sol87] developed a recovery procedure to identify the presence of cavities in segmented objects and model them as superquadrics. Concavities (a circular cut-out of a box) form by a similar process but they are not enclosed completely by the object, so they are visible in the 2-D projection of the object. If a model exists for a concavity



(a)



(b)

Figure 2.2: **Surface Primitives** : (a) Surface discontinuities (C_0 type) and tangent discontinuities (C_1 type), planar and second order patches. (b) Smooth boundaries of perceptual significance, are also useful as partitioning rules

or hole (like for objects with cylindrical hole), it can be modeled as negative volume. For example, the circular cut-out can be modeled as a boolean subtraction of a box and an elliptical cylinder, such that the points on the box that belong to the cylinder are not considered as part of the model. The superquadric inside-outside function presents a convenient formulation of negative volume. It should be noted that it is not necessary for the negative model to completely lie within the parent model, allowing modeling of broad categories of objects with concavities not representable by superquadric models. The choice of deformable superquadrics raises another important issue of *uniqueness* of representation. For model matching and recognition purposes it is essential that the recovered model and stored model have one-to-one mapping. The procedure restricts the parameter space to recover unique part-level models. However, when part-level models combine to form composite objects, in some cases multiple representations of composite objects are possible. We have to address this issue because the ultimate use of our system is for object recognition. Since bending deformation can model two parts joined at an articulation point (human hand for example) as a bent model for small angles, multiple representations are possible. Also, for objects as simple as an L-shaped object, there exist two representations using non-deformed superquadrics. There are two ways to handle this situation. One is to recover all the possible representations and the other is to store all possible representations in the model database. Pentland [Pen87] has adopted the latter option, since it does not burden the recovery procedure but requires model database to store all possible representations. Our procedure will identify the existence of multiple representations and recover them as needed by the model matching procedure. It is one of the "hooks" available to the high level processes which decide to prefer a particular representation.

Range images are nothing but the visible surfaces. Despite the efforts of researchers for almost a decade, finding a natural segmentation of surfaces at significant boundaries is still an open problem. Since boundaries are vital to our part segmentation paradigm we have to address the problem of reliably extracting surface discontinuities (depth discontinuities) and discontinuities in the first derivatives (tangent plane discontinuities). We feel that the issue of surface fitting and surface boundary detection are interrelated and have to be treated together. We propose to combine the two prevalent approaches of surface description: surface-patch based approach [BJ86b], and surface-boundary based approach [Fan88]. We are proposing to segment surfaces into planar and second order patches (figure 2.2), by first grouping the points based on sign of Gaussian and Mean curvature (similar to Besl and Jain's [BJ86a]), and then refining the initial segmentation by taking rough estimate of surface boundaries into account. A rough estimate of Surface boundaries can be obtained by a procedure similar to one used by Fan [Fan88]. The advantage of using multi-level primitive approach is that occluding contour and superquadrics will be involved in the process of surface contour detection. In addition to the discontinuities of surface and its first derivatives,

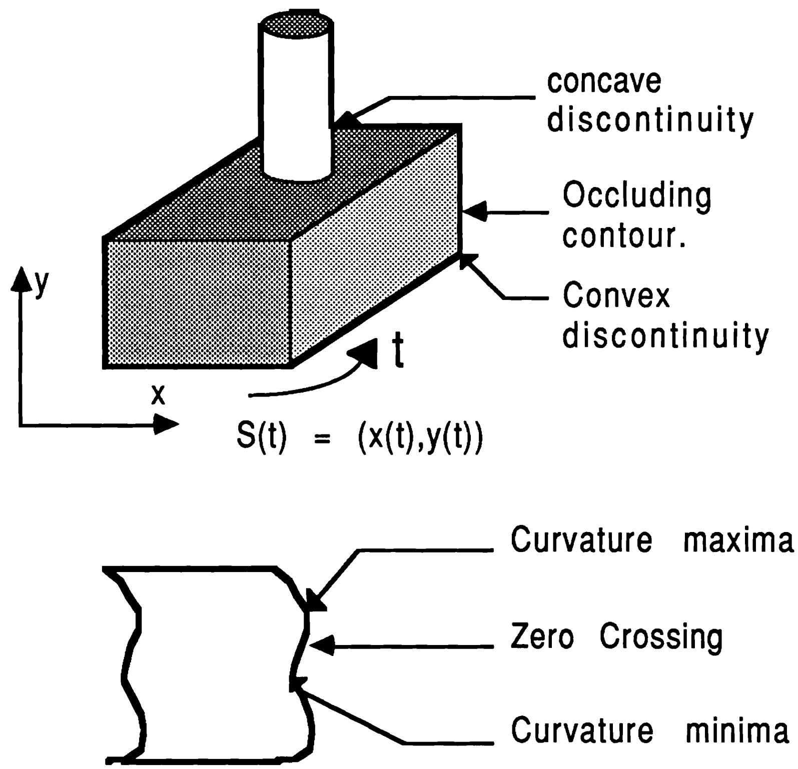


Figure 2.3: **Occluding contour primitive** : Contour representation and points of interest on contours.

smooth boundaries like minima contours [BH87, HR85], parabolic contours [KvD82], contours of zero crossings [Yui89] are of interest in generating surface level part description. Significance of these boundaries is discussed in detail in a later section.

Occluding contour (2.3) is a planar projection (orthographic in our case) of a 3-D object. Shape description at the Occluding contour level is probably the most widely studied topic in vision. Numerous representation have been suggested and successfully implemented to define two dimensional shape. Asada [AB86], Marr [Mar82], Mokhtarian [MM86], Rosenfeld [RJ73, RW75], Pavlidis [Pav80], fischler [FB86] and others have proposed various rules for contour segmentation. We have adopted the $S(t) = (x(t), y(t), z(t))$ representation parametrized by curve length. The points of interest on the curve are inflection points, minima and maxima of curvature. The $z(t)$ component is used only for detection of jump boundaries, and no attempt is made to treat the occluding contour in three-dimensional space. A major reason for this is the noise along the jump edges in $z(t)$ component due to the geometry of the range scanners. Partitioning rules commonly use minima of curvature for curve segmentation, as it has perceptual significance [HR82]. Though our primary concern will be planar occluding contours, we feel that the $z(t)$ component may give important cues for curve segmentation.

It is obvious that our primitives capture all the aspects of shape at with varying dimensionality. Since occluding contours are viewpoint dependent, they are not useful as basic primitives for invariant object recognition. However, they are extremely important to guide the segmentation process and to aid the surface primitives and superquadrics in formulating hypotheses about parts. Their role in the overall description of shape will become clear after we outline our segmentation strategy. Surface primitives are extracted from invariant properties of surfaces, and are therefore ideal for obtaining invariant shape descriptions. Superquadric primitives satisfy all the requirements for a robust volumetric primitive.

2.3 The Segmentation Process

There are two basic strategies for segmentation:

1. Proceed from coarse to fine discrimination by partitioning larger entities into smaller.
2. Start with local models and aggregate them into larger ones.

Both of these strategies have been used in the past [BB82, Pav77] The advantage of the coarse to fine strategy is that one gets first a quick estimate about the volume/boundary/surface of the object which can be further refined under control of some higher level process which determines how much details on wishes to know. The disadvantage of this approach is that the amount of detectable detail is not always sufficient without switching to a different kind of representation. For example, to describe smaller shape details one might have to go from volumetric to surface representation. This progression of looking at data at different scales is more formalized in Witkin's scale-space approach [Wit83] and in different multiresolution signal decomposition techniques [Mal88] The important idea that these methods convey is that progressive blurring of images clarifies their deep structure. Large scale structure constrains the structure at finer levels so that adding details only entails adding information and does not require changing the larger structure. Although these multiresolution techniques do not correspond to structural decomposition of objects into parts, one assumes that the same principle applies there also. When a part model must be subdivided into smaller parts to gain finer resolution it should not affect the original partitioning. In that sense, backtracking to change prior decisions would not be necessary.

The second strategy, which goes from local to global, starts with local features and incrementally builds larger representations. This can be an advantage or disadvantage at the same time. Some details could help the classification process early on by excluding any hypothesis that clearly does not include such particular details. On the other hand, keeping track of too many details at once can lead to a combinatorial explosion. As already mentioned, aggregation of low level models into models of larger granularity is difficult in presence of noise or when data is missing. It is also necessary to ignore details that cannot be represented in the next higher level of representation. Recovering from mistakes or erroneous aggregations by rearranging the low level models in new ways should be possible.

Both methods of segmentation, top-down and bottom-up, have their benefits and problems. We emphasized in the previous chapter that both methods should be used in a general vision system. Our approach to segmentation will be discussed in detail in the final chapter, for now let us see how individual primitives have been used for segmentation in computer vision.

2.3.1 Segmentation using Occluding Contours

Occluding contours being viewpoint dependent are not an ideal representation for objects with significant volume, internal boundaries and surface variations. However they constitute a very important source of perceptual information on potential segmentation sites, as they are formed by projection of parts. We should point out that we are treating occluding contours (also called *apparent contours*) separate from surface contours (discontinuities or smooth boundaries of perceptual significance, figure 2.3, reffig:surfprim). Surface contours are considered a part of surface primitives. Occluding boundaries are obtained by separating the object from the background. However, in the final analysis, both surface boundaries and occluding boundaries will have to be considered together. We have separated them in the intial phase to postulate the recognition problem in a structured fashion. Also, occluding contours are easy to extract and can be used in detecting internal boundaries, which have proved extremely difficult to detect. Occluding contours have been widely studied in psychology and computer vision, because they are seen as planar shapes rich in information content but low in raw data volume. Occluding contours play a large role in human perception. Strong spatial impressions arise from seeing only silhouettes of objects in a general orientation.

Vision Researchers have suggested various techniques for segmentation of objects into parts based on the *significant* features like extrema of curvature, maxima of curvature, and zero-crossings of the curvature. Since the methods of contour description are essentially local and sensitive to noise it is necessary to perform the analysis in scale-space. Asada and Brady's method generates detailed models of simple objects by tracing the maxima of curvature in scale-space, and fitting piecewise continuous circular splines at the knots placed at maxima of curvature. Similar scale-space based approach using zero-crossings of curvature as points significance, has been proposed by Mokhtarian [MM86]. Other methods include the method of differences given by Johnston and Rosenfeld [RJ73]. The basic idea of detecting the significant points in the curve and then generating the description of the curve locally between the knots also appeals from perceptual organization point of view, first observed by Atteneave [Att54], and experimentally demonstrated by Beiderman [Bie85].

2.3.2 Segmentation by Surface Descriptions

A large portion of computer vision literature is on different methods for surface reconstruction, representation and recognition. we are not interested in surface reconstruction techniques needed to construct dense surface maps from sparse information derived from shape from X methods. The reason for the widespread interest in surface-based object recognition is that this fits well into the prevalent bottom-up approach in vision and that surface is a much more tangible property than volume. Surface segmentation can be based either on merging similar local surface models [BJ86a] , or by defining region boundaries in terms of differential geometry [HR85, BH87]. The aggregation process begins with small local neighborhoods which are then combined if they are similar in depth values, surface normal values or some curvature measurements. The result is a scene segmented into surface regions with similar surface characteristics. While differential geometry in the *small* provides techniques for local characterization of surfaces, it is difficult to extend them to obtain a global interpretation, because very few results from the differential geometry in the *large* are useful in the context of global surface characterization. The difficulty with both surface segmentation approaches is that it is sensitive to local variations which are not important but are difficult to eliminate unless the larger context is taken into account. Since this larger context can be much easier accounted for by volumetric models, it should be here where the surface, volume and boundary segmentation could cooperate.

2.3.3 Segmentation using Superquadrics

Superquadrics are a family of parametric shapes with a rich vocabulary of part-models that encompass shapes ranging from cylinders and parallelopipeds to spheres. The representational power is further increased by introducing deformations like bending and tapering along the major axis. Superquadrics have been used as primitives for shape representation in computer vision [Pen87, Sol87, BG88].

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

$$\mathbf{x}(\eta, \omega) = \begin{bmatrix} a_1 Cos_\eta^{\varepsilon_1} Cos_\omega^{\varepsilon_2} \\ a_2 Cos_\eta^{\varepsilon_1} Sin_\omega^{\varepsilon_2} \\ a_3 Sin_\eta^{\varepsilon_1} \end{bmatrix} \qquad \begin{array}{c} \frac{-\pi}{2} \leq \eta \leq \frac{\pi}{2} \\ -\pi \leq \omega < \pi \end{array}$$

Parameters a_1, a_2, and a_3 define the superquadric size in x,y and z direction (in object centered coordinate system) respectively. ε_1 is the squareness parameter in the latitude plane and ε_2 is the squareness parameter in the longitude plane. Based on these parameter values superquadrics can model a large set of standard building blocks, like spheres, cylinders, parallelopipeds and shapes in between. If both ε_1 and ε_2 are 1, the surface defines an ellipsoid. Cylindrical shapes are obtained for $\varepsilon_1 < 1$ and $\varepsilon_2 = 1$. Parallelopipeds are obtained for both ε_1 and ε_2 are < 1. We have restricted the model recovery procedure to fit the models with $0 \leq \varepsilon_1, \varepsilon_2 \leq 1$. Since a superquadric surface can be described with an analytic function, an iterative least-squares minimization of a fitting function can be used for shape recovery. Consider a depth map of an arbitrary scene. The initial model is an ellipsoid in the right position, orientation and of the right size to cover all of the 3-D points. During the least-squares minimization, the shape of the initial model starts to change so that the given range points would lie on or close to the surface of the model. The model recovery procedure incorporates all the given points in the recovered model.

Many different methods for partitioning into volumetric primitives have been proposed in computer vision. The common problem with all the volumetric primitives is that, though they are quite rich representations, they are extremely difficult to recover from the real image data. Franc [Sol87] has described a global to local method of segmentation using superquadric recovery procedure. His goal was to decompose objects or scenes into parts which can be represented with a single superquadric model enhanced with global deformations such as tapering and bending. When several parts or objects made up of multiple parts are present, a suitable distance measure was used to decide which 3-D points should be included in a particular volumetric model and which points should be excluded. The method works on some situations, but not on an arbitrary complex object. It is only expected since it is difficult to constrain the minimization procedure to take part-structure into account. We are proposing to use superquadrics as part-models only and not attempting any segmentation at the model recovery stage. Pentland [Pen88] has described a two-part procedure to recover segmented descriptions of complex objects. His approach is first to recover part-structure by matched filtering and a maximum likelihood estimate, and then, to describe parts by superquadrics using a least squares procedure. Only Occluding boundary data is used, though he noted that surface information will be useful in extracting complete part-structure. The procedure is extremely slow on conventional machines and needs hand segmentation. Biederman [Bie87], in his theory of Recognition-By-Components has suggested an edge and volumetric primitive (generalized cylinders) based approach for describing complex objects in intensity images. He however, does not describe any procedure to recover such complex part-structure.

In the following three chapters we will discuss the three shape primitives in detail. Partitioning rules for the primitives will be defined, along with procedures to recover the primitives from the image data.

Chapter 3

Occluding Contours

A lot of research effort in last two decades has gone into analyzing object shapes in two dimensions to extract three dimensional shape, or to recognize flat objects. The methods can be classified into two categories. In the most popular category lies the shape from occluding contour (or silhouette) paradigm, that has dominated the pattern recognition and vision research, and provided working systems. The paradigm works for flat or almost flat objects that satisfy the general viewpoint constraint needed for robust recognition. These methods typically accept bounding contours, binary shapes, or silhouettes as input. These methods are also useful for generating object models from silhouettes seen from different viewpoints [CA87]. But the real world is three dimensional and reflectance images provided by the retina or a video camera are two dimensional projections. Thus the problem of extracting 3-D information from 2-D projections is underconstrained [AWB87]. Additional constraints can be provided in a variety of ways, and vision research has seen many shape from X paradigms, with the primary goal of obtaining a $2\frac{1}{2}$ sketch. Significant among them are shape-from-shading, shape-from-texture,shape-from-contour, and shape-from-motion methods. Shape-from-contour methods [BY84, Stewn, Mar82] provide constraints from surface and occluding contours that are visible or can be extracted from the image. Since our input data is three dimensional, the projections of surface contours do not concern us. We are interested in significant 3-D contours like depth (C_0) discontinuities, surface-normal (C_1) discontinuities as also the smooth surface contours like parabolic, minima, maxima, and zero crossing contours. While these contours are extremely rich in shape information and have perceptual significance for shape recognition, they have proved to be extremely difficult to detect reliably. On the other hand, depth discontinuities resulting in occluding contours provide an outline of the object, that is easy to extract and most important, have significant shape information. The occluding contour, though viewpoint-dependent, not only supplements the shape information provided by the internal boundaries of the object, but also helps us detect them. As we will observe later, occluding contours along with surfaces define partitioning rules and play an important role in evaluating the volumetric models. So we propose to include occluding contours in our study of the 3-D shape recognition problem.

Silhouettes and binary images have been used in vision research for past two decades in the disciplines of pattern recognition, computer vision and psychology with very encouraging results. The primary reason being that they are high in information content but need low volume of data for representation. Though they have been applied only to specialized tasks, they have fared better than gray level images in fostering our understanding of machine perception. Occluding contours have also been called *apparent contours* (orthographic projection of the contour-generator on the surface), *bounding contours* and *extremal contours* in literature. Since our goal is 3-D shape recognition, we have to address the contour primitive in the global context of shape :

1. The Shape properties of Planar contours. What are the significant points on the contour? How do they help in curve segmentation?

2. Contour Representation : What representation is best suited to extract the shape properties reliably? How does the representation interact with surface and volumetric representations? The representation should be invariant to scale, size, position and orientation.

Again, the problem of curve segmentation cannot be treated in isolation from the problem of curve representation. Representation is a means to achieve the segmentation requirements. Let us first describe what we mean by curve segmentation, then we will review the curve representation techniques and present some results.

3.1 Curve Partitioning

Curve segmentation is defined as partioning the curve in perceptually significant parts. As such, there are different paradigms of perceptual significance, resulting in different decompositions of the same curve. However, it is generally agreed upon that there are three types of points that can be used to partition a curve into units in a manner invariant under rotations, translations and uniform scaling :

1. **Curvature maxima :** Positive maxima of the curvature. Convex corners , where curvature is infinite are included.

2. **Curvature minima :** Negative minima of the curvature. Concave corners, where curvature is infinite are included.

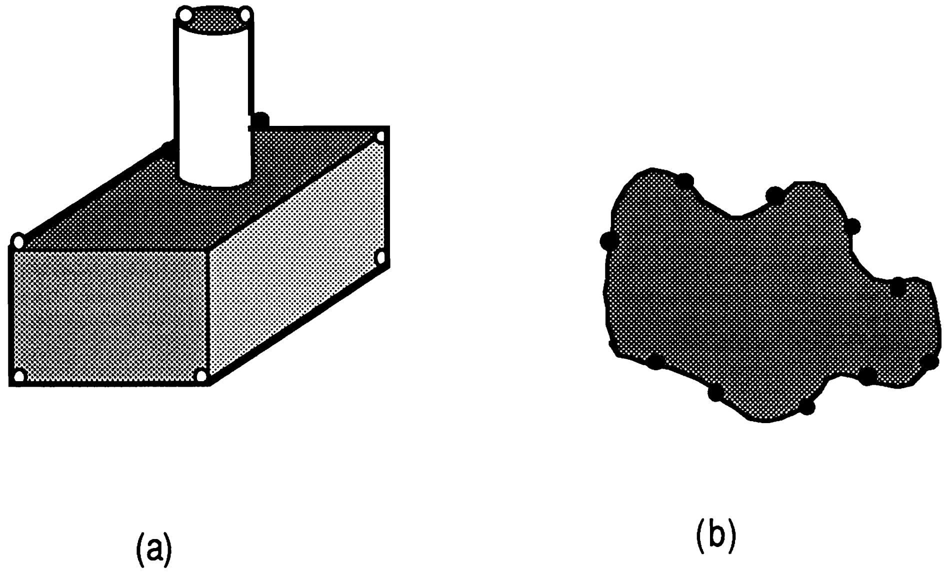


Figure 3.1: **Curve partitioning** : (a) Concavities (curvature minima, black circles) segment the contour into parts formed by projection of the cylinder and the cube. Partitioning at significant curvature changes (corners in this case, black and white circles) (c) Partitioning at inflection points.

3. **Zeros of Curvature :** Inflection points.

Curvature minima generally reflect the concavity formed by joining two subparts. This rule of traversal regularity [BH87, HR82, GP74], makes it possible to assign concave discontinuities as segmentation sites for partitioning of the contour into two segments belonging to different parts. In figure3.1, the only pair of concavities segment out the contours formed by projection of the cube and the cylinder. Hoffman and Richards [HR82] have proposed to segment the contours at curvature minima, and define the individual segments in terms of inflection points and maxima of curvature. It is important to note the distinction between minima (or maxima) of curvature and C_1 discontinuity that forms the corners used above to segment the contours. The concave (and convex) discontinuities have *infinite* negative curvature, while smooth concavities are continuous. Both concave discontinuities and smooth concavities can be used to partition a contour [HR82]. The perceptual significance of high curvature points was first noted by attneave [Att54]. He observed that such points have high shape information content. Asada and Brady [AB86] have used points of significant curvature changes like corners (C_1 discontinuity) and smooth joins (C_2 discontinuity) for curve segmentation. They do not segment at smooth curvature maxima or minima.Though this approach results in oversegmentation of the contour (figure 3.1b), it can be useful in generating the overall description of the contour. Yet another partitioning rule segments contours at their inflection points (zero crossings of curvature) [MM86, Mar82, Mil88, Fre67]. This paradigm results in convex and concave subparts of the contour (figure 3.1c). Marr[Mar82, Mar77] noted that convex and concave parts of the contour have perceptual significance. Fischler and Bolles [FB86] have critically evaluated the curve partioning schemes and have put forth the principle of *stability* which states that any perceptual decision should be stable under at least small perturbations of both the imaging conditions *and* the decision algorithm parameters. They partition the contours at curvature discontinuities.

It is clear that minima, maxima and zeros of curvature provide the *critical* points for curve segmentation. Since contour segmentation is not an end in itself, but has to complement the surface and volumetric information in segmenting 3-D shape, we have to segment the occluding contour into enclosed 2-D shapes. Thus concave discontinuities (figure 3.1, minima of negative curvature) play an important role in hypotheses generation about potential parts. However, to aid the 3-D segmentation process, we propose to generate the complete description of the contour in terms of all three critical features. It has many applications for surface boundary detection, for example, convex discontinuities in the occluding contour may correspond to creases on the surface (though not always) and inflection points on the contour may correspond to zero-crossing contour on the surface. Many of these questions have been answered in shape-from-contour paradigm, which we propose to investigate. *Holes* (figure 3.2) in the objects that are visible as occluding contours can be described as closed contours in the similar manner. However, holes do not enclose any *figure*, so segmenting at the negative curvature minima is not desirable. We have to analyze the holes as boundaries of figures, in a complementary sense. Thus, in figure3.2 the direction of traversal of hole is changed to *attach* the hole to *ground* instead of *figure*. This interpretation is more useful for us, since it provides description for the *actual parts* (the cup handle and the body) rather than for the hole.

Now that we have the partioning rules, we need a representation to describe the contours and recover the above mentioned features.

3.2 Curve Representation

Polynomial approximations to planar contours have been traditionally piecewise linear [Pav80, Pav77, Dav77]. The polygonal representation is a compact way of segmenting contours and facilitates easy matching[KK87, PH74]. However, they are not acceptable for the shapes with high curvature, for which smooth curve approximations like splines are required. Spline fitting needs knot points on the contour and a polynomial for interpolation. Circular splines [MA77, AB86] are adequate for description of tools and other objects. Based on the polygonal model, Shapiro [Sha80]

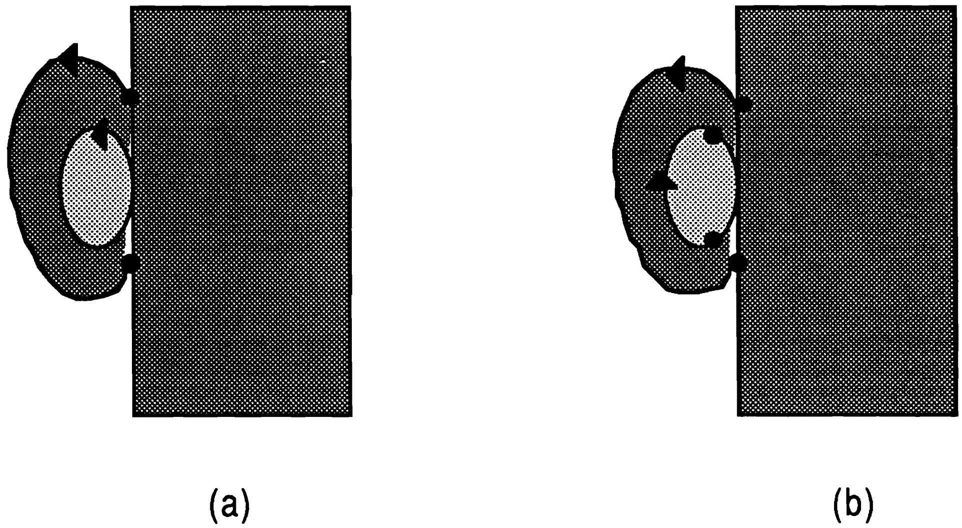


Figure 3.2: **Holes and Cavities**: (a) The hole visible as occluding contour in the outline of cup has no parts if it is considered as enclosing a figure. (b) by reversing the direction of traversal, the hole has two negative curvature minima partitioning the contour into two parts.

proposed a 2-D shape model for segmentation of 2-D shapes into parts described by a set of primitives. Her segmentation approach was based on a graph-theoretic clustering procedure. Chain coding proposed by Freeman [Fre74] has been extensively used to represent contours and extract corners and curvature properties [FD77, RJ73, RW75, Pav77, MA77]. Other approaches have taken the structural aspect and global shape in defining the representation. These are the *region-based* methods. Blum and Nagel [BN78] proposed a weighted symmetric axis transform for shape classification and description. The smoothed local symmetries (SLS) representation introduced by Brady and Asada [BA84] is both contour and region based. 2-D analogs of generalized cylinders and quadtrees are other region based representations. The main disadvantage of region-based approaches is their sensitivity to occlusion and inability to describe contour properties in detail. Horn [Hor83] has argued for a least energy curve, a curvature based representation. Kass *etal* [KWT87] have proposed energy-minimizing splines guided by external constraint forces and image forces for unifying a number of visual problems.

Parametrized curve representations have recently received a lot of attention due to their invariant properties. Parametrization based on curve length [MM86, Mok88, COCD87, Low88] has some attractive properties like computationally efficiency, invariance to rotation, uniform scaling, and the translation of the curve. This representation also affords different methods of tangent and curvature computation, curve fitting and other useful representations like $s - \theta$ representation and

$s - \rho$ representations. It also makes conversions to other representations easier. Milios [Mil88] recently proposed the Extended Circular Image representation based on a parametrization in terms of angle of the contour's tangent with respect to the x-axis. A disadvantage of this approach is that the curve segments have to be of constant curvature sign, thus segmentation is possible only along the inflection points. Dubois and Glanz have used an autoregressive model to express a polygonal approximation of 2-D object boundary as a linear combination of sequential boundary samples. Hoffman and Richards[HR82] have proposed simple primitives called *codons* that are segmented at the curvature minima. Individual codons are described by curvature zeros and maxima. Their objective was similar to ours, that of curve segmentation into parts corresponding to different parts in 3-D image.

The curve-length based parametrization appeals to us as a suitable approach for our purpose. Parametrization is done by the path length variable t along the length of the curve and expressing the curve as :

$$C = \{x(t), y(t)\}$$

where t is a linear function of the path length ranging over the closed interval $[0, 1]$. Since we are obtaining the occluding contour by tracing the boundary of a depth image, it is possible to assign z coordinate value at every boundary point.The three dimensional description extension of C can be written as a general space curve :

$$C = \{x(t), y(t), z(t)\}$$

Mokthtarian [Mok88] has proved the evolution properties of space curves. But we are *not* interested in computing the contour level description in terms of torsion and 3-D curvature, but only in making use of the C_0 (jump) discontinuities in the curve z(t). This information is available as the occluding contour is traced, and is useful in identifying parts. For the purpose of contour description at curvature level, only planar representation is necessary. From now on we deal with contour representation of the form $C = (x(t), y(t))$ only. This representation satisfies the criteria for a stable and reliable representation :

1. It is invariant under rotation, uniform scaling, and translation of the curve.

2. It admits various local continuous function approximations to the curve. For example, the curve can be locally approximated by splines or polynomials.

3. Scale-space description is possible by convolving the contour by Gaussian masks and obtaining the curvature at different scales.

4. A small change to part of the curve creates a small local change in description.

The curvature κ can be computed in terms of derivatives of fuctions $x(t)$ and $y(t)$:

$$\kappa = \frac{x_t y_{tt} - y_t x_{tt}}{(x_t^2 + y_t^2)^{3/2}}.$$

The curve $C(t)$ is convolved with the Gaussian kernel $G_\sigma(t)$ of standard deviation σ to filter out the high frequencies :

$$G_\sigma(t) = \frac{e^{-t^2/2\sigma^2}}{\sigma\sqrt{2\pi}}.$$

The convolution with the first and second order derivatives of the kernel gives the first and second derivates of $x(t)$ and $y(t)$.

$$X'(t) = G'_\sigma(t) \times x(t) \;\; and \;\; X''(t) = G''_\sigma(t) \times x(t)$$

The scale-space description of the occluding contour of vase is shown in figure 3.3. The occluding contour is obtained by thresholding the object against the background, and tracing the boundary as described in [RK82]. Note the systematic shrinking of the contour as σ increases. The source of the shrinkage is the fact that each point is being averaged with its neighbors, which in both directions curve towards the local center of curvature. This reason for the shrinkage and a method for compensating for it were recently given by Lowe [Low88].

The convolution with derivatives of Gaussian kernels gives first and second derivates of the curve without fitting a smooth function at the point. Curvature properties like minima, maxima, and zeros are easily computed using this approach (see figure 3.4). However, these need scale-space tracking before they can be reliably recovered. Other approach is to fit splines at every point, and then estimate the curvature of the spline at the point. The results obtained by fitting Akima's shape-preserving bicubic spline are shown in figure 3.5. A discrete method to compute maxima of curvature and inflection points was given by [RJ73]. Results of this method (figures 3.6 and 3.7) depend upon the scale of the contour which can change them drastically. Nevertheless, it performs very well in recovering points of maxima and inflection. It is clear that these results need to be refined to get rid of response due to local variations and noise, Scale-space tracking [Wit83, AB86, MM86] is certainly a possibility. Recently Chien and Aggarwal [CA89] proposed a modification in Rosenfeld's algorithm, which shows encouraging results.

The problem of reliably detecting tangent discontinuities (where two independent objects meet) is vital for our purpose. Bennett and Hoffman [BH87] have given a theoretical treatment for the problem of detecting transversal joins formed by smoothing the tangent discontinuity by a suitable

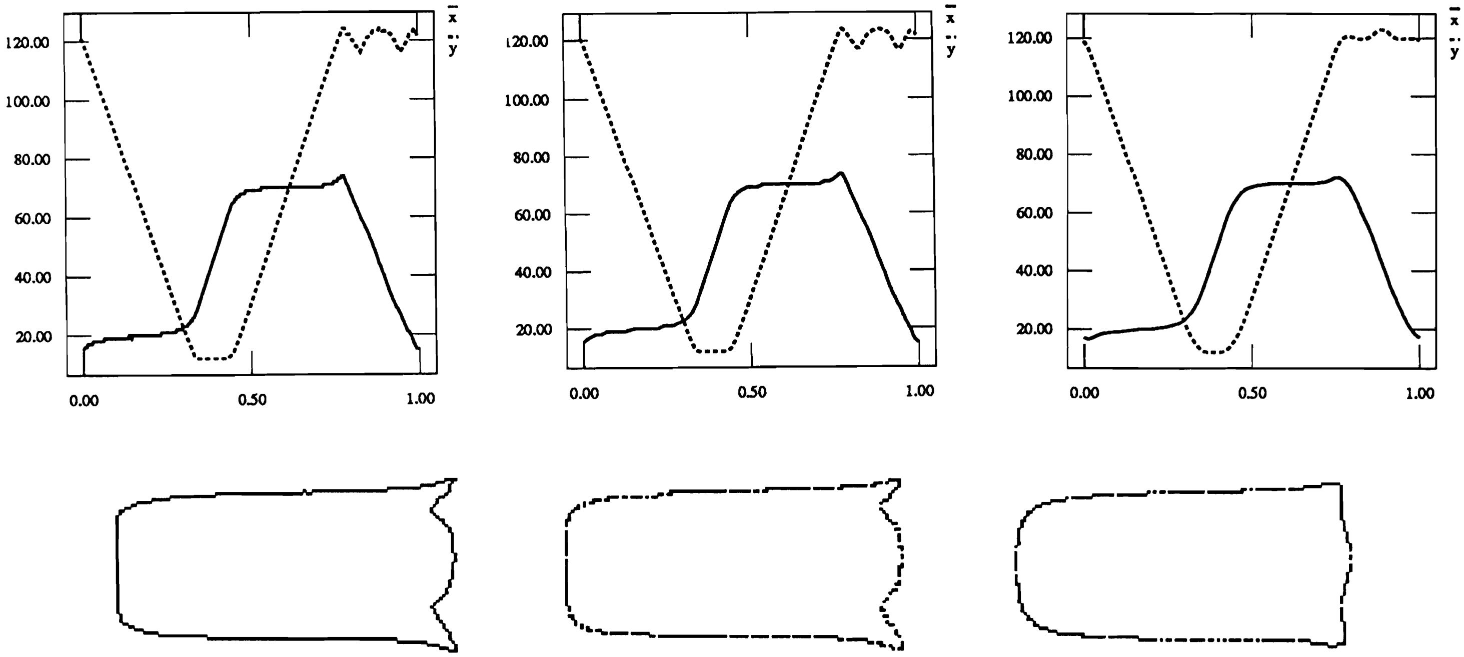


Figure 3.3: **Scale-Space smoothing of Vase contour** : Top : ($x(t)$ and $y(t)$) plotted with parameter t at $\sigma = 0.0, 2.0$ and 8.0. Bottom : Contour of the vase smoothed with the same values

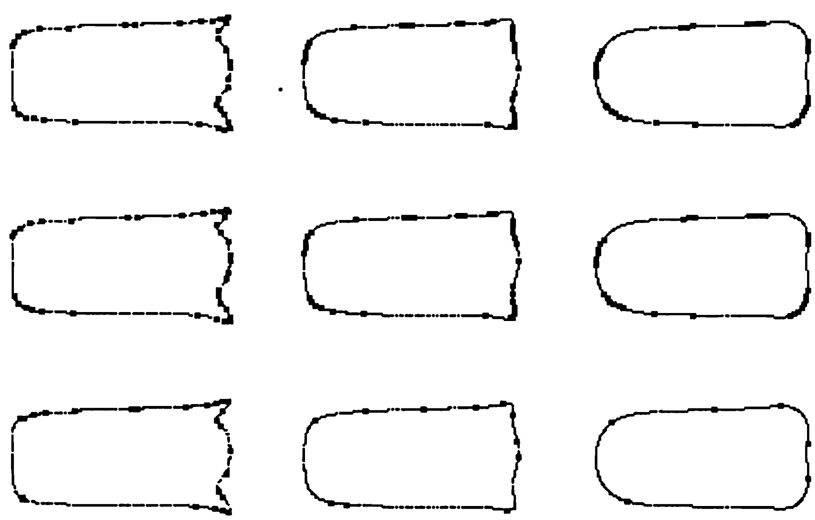


Figure 3.4: **Maxima, minima, and zeros of curvature** for $\sigma = 2.0, 8.0$ and 16.0, by convolution with the derivatives of Gaussian kernel.

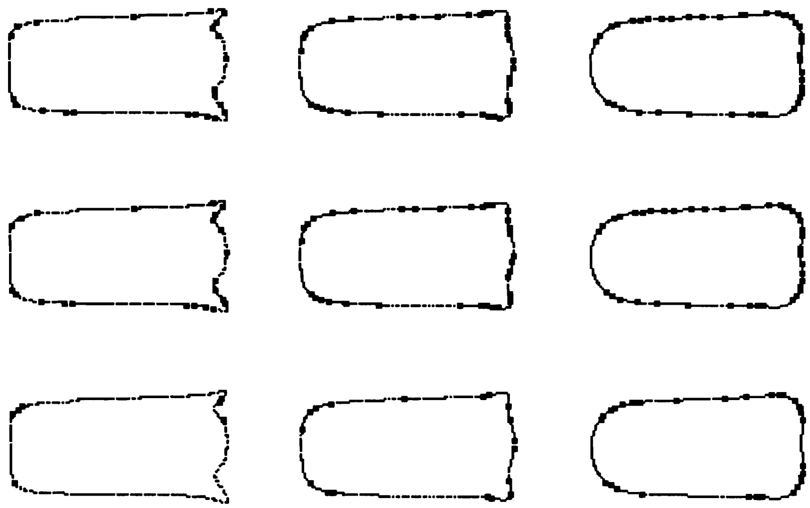


Figure 3.5: **Maxima, minima, and zeros of curvature** for $\sigma = 2.0, 8.0$ and 16.0, obtained by fitting shape-preserving akima bicubic splines.

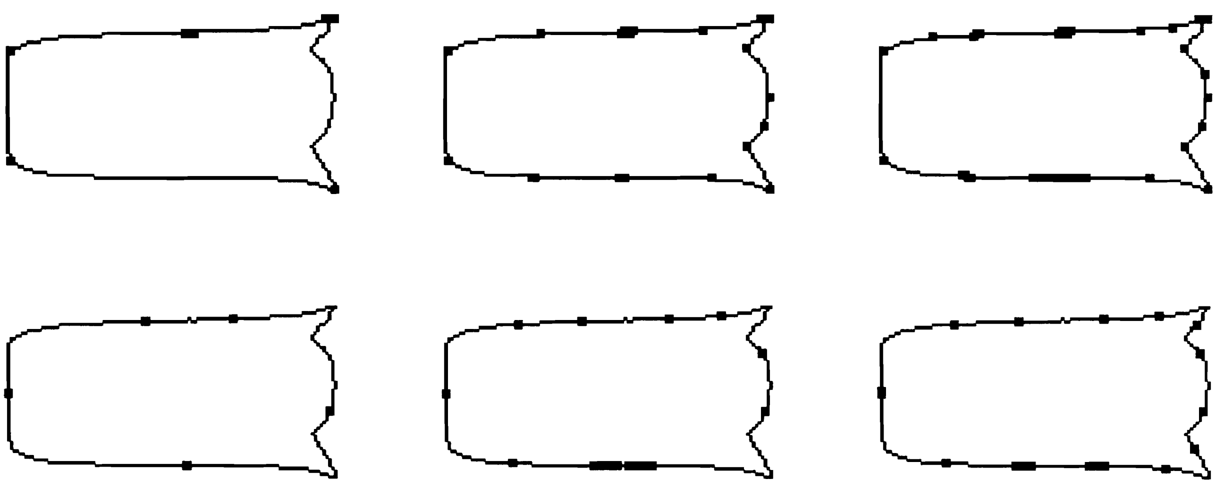


Figure 3.6: **Points of Significant curvature change (top) and inflection points (bottom)** obtained by computing k-curvature with $k = 32, 20$ and 15.

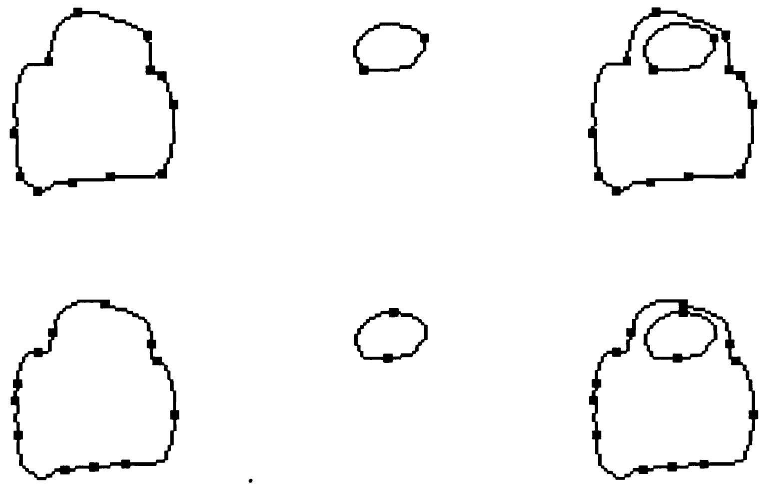


Figure 3.7: **Contour analysis of Cup (body and hole)**: Top row : Points of significant curvature change marked by k-curvature computation for $k = 15$. Bottom : Inflection points on the body and hole of the cup.

filter like Gaussian filter and then detecting minima of curvature. After smoothing, the problem translates into distinguishing between smooth minima due to a genuinely curved edge and minima due to tangent discontinuity. Brady and Asada [BA84] have cited smoothing of the join as a major hurdle in recovering "subshapes" using their powerful Smoothed Local Symmetries representation. Lowe [Low88] has suggested a curve segmentation method that will distinguish between the two cases. He has used the *third* derivative, or the rate of change of the curvature, to measure the underlying degree of smoothness of an edge. Smooth edges will have a high curvature that is changing only slowly, while the segments with high rate of change are likely to be the tangent discontinuities. We plan to investigate these approaches to obtain a reliable contour segmentation.

Chapter 4

Surface Contours and Patches

Surfaces form a very important set of primitives for shape description and recognition. Significant among them are various surface contours delineating parts based on differential geometric properties, and surface patches segmenting the surface into piecewise continuous patches. We are not interested in obtaining arbitrary surface patches that are sensitive to viewpoint and the choice of seed region during region growing process. To generate a global description of surfaces from local differential geometric description has proved to be extremely difficult. We are interested in Surface contours and piecewise continuous patches that are delineated by contours of physical, geometric or perceptual significance. Such a description is needed to decompose objects into parts based on the internal boundaries. It is therefore necessary to investigate the surface contours that partition objects into parts describable by higher level volumetric primitives or piecewise continuous patches or both. This brings in the issue of representation. What is the best representation for generating segmented descriptions? In this chapter we will discuss the *representation* and *shape description* aspects of surfaces. These aspects are defined in terms of surface properties derived from the field of differential geometry of surfaces. That is where we begin this chapter.

4.1 Local Differential Geometry of Surfaces

There are two aspects of the differential geometry of curves and surfaces [dC76]. The first one deals with the study of local properties of curves and surfaces in the immediate vicinity of a point. The second one is the global differential geometry, or the differential geometry in the large. The first and second derivative properties in the context of surface description have been described by Besl and Jain [BJ86b]. We will review the basics in this section.

Regular Surface : Parametric form of equation for a regular surface S with respect to a known coordinate system is :

$$S \subset R^3 = (x, y, z) : x = x_1(u, v), y = x_2(u, v), z = x_3(u, v), (u, v) \in U \subseteq \mathbf{R}^2$$

The surface is a locus of points in Euclidean three-space defined by the end points of the vector $\mathbf{X}(u, v)$ with $x_i(u, v)$ the components of the vector. These real functions are assumed to be defined over an open connected domain of a Cartesian u, v plane and to have continuous second partial derivatives there. In our analysis of range images we are assuming that this condition is satisfied.

The second condition for a regular surface is automatically satisfied by the Z-depth format images. It requires that the coordinate vectors $\mathbf{X}_u = \mathbf{X}_1 = \frac{\partial \mathbf{X}}{\partial u}, \mathbf{X}_v = \mathbf{X}_2 = \frac{\partial \mathbf{X}}{\partial v}$ are linearly independent :

$$\frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v} = \mathbf{X}_1 \times \mathbf{X}_2 \neq 0.$$

The surface in range images can be locally described by $z = f(x, y)$ form :

$$\mathbf{X} = (x_1, x_2, f(x_1, x_2))$$

and coordinate vectors become :

$$\mathbf{X}_1 = \left(1, 0, \frac{\partial f}{\partial x_1}\right),$$

$$\mathbf{X}_2 = \left(0, 1, \frac{\partial f}{\partial x_2}\right),$$

These vectors are linearly independent given the first condition. Also, the surface $\mathbf{X}$ is trivially orientable. It can be shown using differential geometry techniques that first and second fundamental forms(which exist only if the surface is analytic) uniquely characterize a general *smooth* surface. The first fundamental form I of a surface is defined as :

$$I(u, v, du, dv) = d\mathbf{X}.d\mathbf{X} = \begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = d\mathbf{u}^T [g] d\mathbf{u}$$

where $[g]$ matrix elements are given by :

$$g_{11} = E = \mathbf{X}_u.\mathbf{X}_u \quad g_{22} = G = \mathbf{X}_v.\mathbf{X}_v \quad g_{12} = g_{21} = F = \mathbf{X}_u.\mathbf{X}_v$$

The two tangent vectors $\mathbf{x}_u$ and $\mathbf{x}_v$ lie in the *tangent plane T(u,v)* of the surface at the point (u, v). $[g]$ matrix is symmetric for an analytic surface. The first fundamental form $I(u, v, du, dv)$

measures the small amount of movement in the parameter space (du, dv). The first fundamental form is invariant to surface parametrization changes and to translations and rotations in the surface. Therefore it depends on the surface itself and not on how it is embedded in the 3-D space. The metric functions E, F, G determine all the intrinsic properties of the surface. In addition they define the area of a surface :

$$A = \int\int_R \sqrt{EG - F^2} du\, dv$$

The second fundamental form of the surface is given by :

$$II(u, v, du, dv) = -\, d\mathbf{X}.d\mathbf{n} = \begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} = d\mathbf{u}^T [b] d\mathbf{u}$$

Where $[b]$ matrix elements are defined as :

$$b_{11} = L = \mathbf{X}_{uu}.\mathbf{n} \quad b_{22} = N = \mathbf{X}_{vv}.\mathbf{n} \quad b_{12} = b_{21} = M = \mathbf{X}_{uv}.\mathbf{n}$$

The unit normal vector at the point is given by :

$$\mathbf{n}(u, v) = \frac{\mathbf{x}_u \ X\ \mathbf{x}_v}{\mid \mathbf{x}_u \ X\ \mathbf{x}_v \mid}$$

Where the double subscript denotes second partial derivatives.

The second fundamental form measures the correlation between the change in the normal vector dn and the change in the surface position at a point (u, v) as a function of small movement (du, dv) in the parametric space. From the [g] and [b] matrices calculated above surface shape and intrinsic surface geometry can be uniquely determined.

The Gaussian curvature function K of a surface can be defined in terms of the two matrices as :

$$K = \det\left(\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1}\right) \det\left(\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & g_{22} \end{bmatrix}\right)$$

and the mean curvature of a surface is defined as :

$$H = \frac{1}{2}\mathrm{tr}\left(\begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1}\right) \det\left(\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & g_{22} \end{bmatrix}\right)$$

Gaussian and mean curvature are related to the *lines of curvature* at the point by the quadratic equation :

$$k^2 - 2Hk + K = 0$$

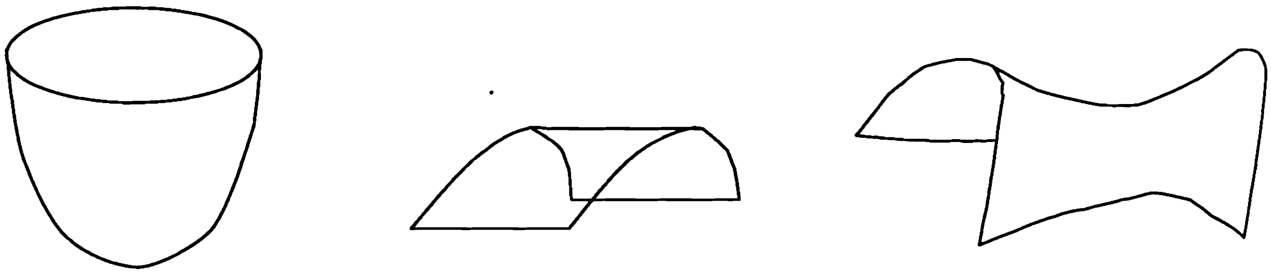


Figure 4.1: **Patches classified by sign of Gaussian curvature**: (a) elliptic ($K > 0$) (b) Parabolic ($K = 0$) (c) hyperbolic ($K < 0$)

which gives the principal curvature values :

$$k_1, k_2 = H \pm \sqrt{H^2 - K}$$

The principal directions are given by the eigen vectors of the $d\mathbf{n}$ matrix. The concept of Gaussian and mean curvature is very useful in surface characterization. The two types of curvatures are together referred to as surface curvature functions. Some of the important invariant properties of Gaussian and mean curvature are noted below [BJ86b, HC52] :

1. Gaussian curvature is an isometric invariant of a surface. It is therefore an intrinsic quantity. It is independent of the way the surface is embedded in the 3-D space. The sign of Gaussian curvature classifies a point as one of the following type (figure 4.1) :

   (a) **Elliptic point** : $K > 0$. Examples: spheres and ellipsoids.

   (b) **Hyperbolic point** : $K < 0$, a saddle point, the surface is saddle shaped in the neighborhood. Example: hyperboloid and hyperbolic paraboloid.

   (c) **Parabolic point** : $K = 0$, surface is developable in the neighborhood of the point. Example: cylinders and planes.

2. Combining the above with sign of mean curvature gives eight basic surface types.

3. Gaussian curvature function of a convex surface uniquely determines the surface.

4. Mean curvature function of a graph surface taken together with the boundary curve of a graph surface uniquely determines the graph surface from which it was computed.

5. Gaussian and mean curvature are invariant to arbitrary transformations of the (u, v) parameters of a surface as long as the Jacobian of the transformation is always non-zero.

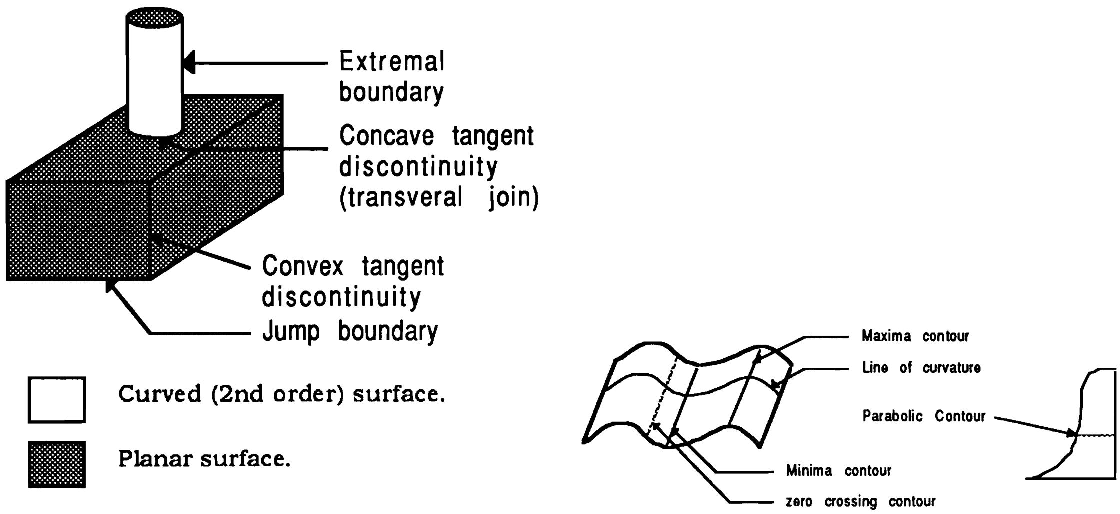


Figure 4.2: **Surface Contours** : jump boundaries (C_0 type), tangent discontinuities (C_1 type), and maxima, minima, parabolic and zero crossing contours.

6. Gaussian and mean curvatures are invariant to rotations and translations of a surface. This property enables us to obtain view-independent characteristics.

We will now make use of above invariant properties of Gaussian and mean curvature to develop our surface representation and segmentation methods.

4.2 Patches and Patch boundaries

The discussion so far is applicable only locally in a small neighborhood of every surface point. To extend this treatment to achieve a coherent global description is not trivial. What is more, the strictly theoretical results of global differential geometry are of little use for our purpose. Our objective is to obtain patches and patch boundaries to perform surface and volumetric segmentation.

As mentioned before, surface boundaries (both C_0 and C_1 discontinuities and smooth boundaries) define the *part* boundaries (see figure 4.2). While it is clear that C_0 type boundaries delineate objects, the presence of C_1 boundaries signal termination of a smooth surface. In fact, using the techniques of *differential topology* [GP74], it can be proved that, when two surfaces surfaces intersect they do so transversally. The importance of *transversality* regularity in context of part segmentation was first observed by Hoffman *etal* [HR85, BH87], and recommended as a partitioning rule for surfaces. The theoretical treatment [BH87, HR85, KvD82, GP74, PB84, BPYA85, Lan84] of surface boundaries has received considerable attention in the past, along with the singularities on the surfaces, like umbilical points [SZ88, BH77, Por83] and parabolic points. Unfortunately, detecting these boundaries in real images has proved to be extremely difficult. The methods used for reflectance images are of no use in detecting C_1 discontinuities, much less the smooth contours. Clearly, we need a different approach for range images.

Piecewise continuous patches are delineated by surface boundaries of *some* physical or differential geometric significance. So, given surface boundaries, patch description is trivial to obtain. On the other hand, surface boundaries enclose patches, and hence, given patches, boundaries are trivial to obtain. Where does one start? This chicken-and-egg problem was noted by Leclerc and Zucker [LZ87] in dealing with discontinuities in one dimension. They concluded that the two tasks are inseparable. It is clear that both the descriptions have to go together, if we want to segment a complex surface into meaningful parts. It is however not very clear how one goes about obtaining the two descriptions simultaneously in two dimensions. Besl and Jain [BJ86a] have used significant local surface features to extrapolate preliminary patches into variable order (upto fourth order) surface patches, generating a piecewise continuous surface description. However, they do not emphasize the significance of discontinuities at surface intersections. T. J. Fan [Fan88] has computed the jump boundaries and creases from sign of principal curvatures. His method does not give closed boundaries of the regions and explicit gap filling of 5 pixels is performed to obtain patches, which are then defined as second order surfaces. The major difference between the two approaches is that Besl and Jain aggregate patches with same differential geometric properties and fit variable order patches in a systematic procedure. While Fan's procedure computes boundaries, which are considered final segmentation of the scene. Patches are used to simply describe the closed regions. We propose to combine the two basic procedures of region growing and contour detection, as gives better localization for the 3-D edges and classifies them. The surface representation used by the former is of type :

$$S = (x, y, z) \quad where \quad z = f(x, y) \text{ is a polynomial}$$

which does not admit important second order surfaces like cylinders and spheres and is not a suitable global representation for patches. The general equation for a quadric patch is given by :

$$F(x, y, z) = \sum_{i,j,k=0}^{i+j+k\leq 2} a_{ijk} x^i y^j z^k \; = \; 0$$

It should be mentioned that we have made a distinction between local and global representation of surfaces. For local estimation of the surface properties we use the bicubic $z = f(x, y)$ representation, while for global representation, we use the general quadric $F(x, y, z) = 0$ representation. As with every choice of representation, we have to justify our choice of second-order patches. Why

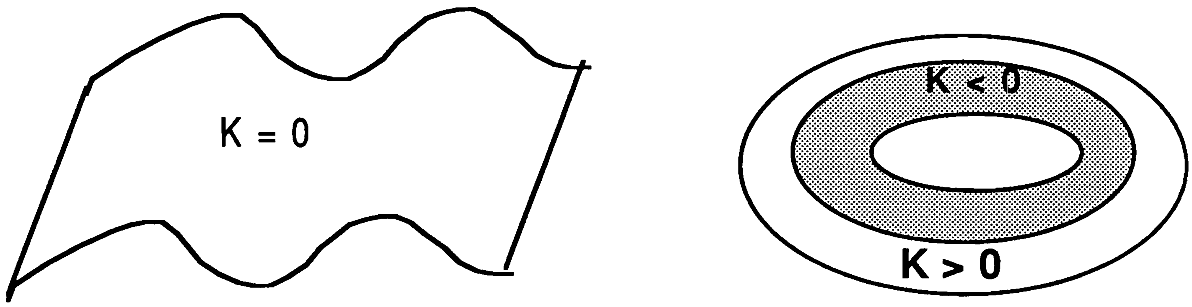


Figure 4.3: **Patches of constant Gaussian Curvature sign that cannot be described by second-order surfaces.**

not third-order or fourth-order patches or combinations thereof? Let us first mention the following property of second order patches [HC52]. "On any second-order surface the Gaussian curvature is either positive everywhere, as on the ellipsoid, or negative everywhere, as on the hyperboloid of one sheet, or everywhere zero, as on the cylinder and the cone." Is the converse true? Unfortunately not, as shown in the figure4.3, smooth cylindrical surfaces can only be approximated as piecewise second-order with boundaries at the zero-crossings of the curvature. Also, parts of torus cannot be modeled as a second-order surface. Interestingly, the sign of mean curvature divides the smooth undulated surfaces into concave and convex ridges with boundary at the zero-crossing contour. Figure 4.7 shows the division of the surface by the sign of mean curvature. Why do we need to decompose a smooth surface into parts at all? Firstly, such a surface cannot be described as a fixed order patch. Secondly, from the perceptual organization point of view, segmentation into piecewise smooth patches is carried out by human observers. Koenderink and van Doorn[KvD82] suggested *parabolic contour* segmentation rule, which rules out segmenting such surfaces. Certainly this is not desirable. Bennett and Hoffman [BH87] suggested partitioning at the *minima contours.* But decomposition based on minima contours is not describable by second or even third order patches, as the patch is no longer singly curved. We are avoiding higher order patches because they introduce oscillations and computational problems. If such oscillations are present, they can be readily described by piecewise continuous patches. Another consideration is the volumetric (superquadric) representation, which is essentially a modified quadric surface. Detecting the minima contours and the zero-crossing contours reliably is very difficult. Typically, lines of curvature are needed to compute them, whose detection is computationally expensive and unreliable. As shown in figure 4.7, they are marginally visible in the sign map of mean curvature. Thus, the sign maps of Gaussian and mean curvature are good starting points for both, quadric surface fitting as well as boundary detection. We have to further investigate how to extend the local description to obtain patches and patch boundaries.

4.3 Computing Local Surface Properties in Range Images

Computation of curvature involves computing first and second order derivatives at every pixel in the image. Let us first review different methods used by researchers to approximate derivatives and compute surface properties. Haralick *et al* [HWL83] have described a facet model for describing the topographic primal sketch of the underlying gray tone intensity surface of a digital image. They use first and second directional derivatives to classify each picture element as one of peak,pit,ridge,ravine,saddle,flat, and hillside. Derivatives were computed by least square fitting a bicubic patch locally at every point. Brady *etal* [BPYA85, PB84] described a computational method of tracing lines of curvature and obtaining a curvature primal sketch of the surface. Tracing lines of curvature in real range images is very unreliable due to the low x-y resolution of the scanner and quantization and other sensing errors.. Besides it is noise sensitive and computationally expensive. Besl and Jain [BJ86a, BJ86b] have done a comprehensive study of invariant surface characteristics and presented an algorithm for variable order surface fitting for image segmentation. They have summarized the field of 3-D object recognition in their survey [BJ85].

A scale-space based algorithm for extraction and representation of physical properties of a surface, using curvature properties of the surface is discussed in Fan [Fan88]. Nackman [Nac84] has described the two dimensional critical point configuration graphs for describing the behavior of smooth functions of two variables by extracting peaks (local maxima), pits(local minima) and passes (saddle points) of a surface. Yang and Kak [YK86] computed derivatives by fitting B-splines and used local curvature information to label the object as flat and curved. There are scanner-specific methods available to process images acquired using a light-stripe rangefinder. Smith and Kanade [SK85] have done contour classification of light-stripes to produce object centered 3-dimensional descriptions. Another method by Martin Herman [MA83] extracts detailed, complete descriptions of polyhedral objects from light-stripe rangefinder data.

To compute local properties of the surface points one has to calculate the Gaussian and mean curvature. To compute surface curvature we need to know the estimates of the first and second partial derivatives of the depth map. This requires estimating the surface type in the neighborhood of the point by fitting an anaylitic surface. Since the estimation is done only in the neighborhood of a point, it is possible [BJ86b, BPYA85, YK86, Gup88] to reliably estimate the first and second order derivatives by fitting a biquadric or bicubic patch of the form (of a graph surface [dC76]) :

$$\mathbf{x}(u,v) = (u, v, f(u,v)) \quad \text{where } f \text{ is a biquadric or bicubic function of } (u,v)$$

Where $u = x, v = y$. The simplicity in parametrization gives following formulas for the surface partial derivatives and the surface normal :

$$\mathbf{x}_u = \begin{pmatrix} 1 & 0 & f_u \end{pmatrix} \quad \mathbf{x}_v = \begin{pmatrix} 0 & 1 & f_v \end{pmatrix} \quad \mathbf{x}_{uu} = \begin{pmatrix} 0 & 0 & f_{uu} \end{pmatrix}$$

$$\mathbf{x}_{vv} = \begin{pmatrix} 0 & 0 & f_{vv} \end{pmatrix} \quad \mathbf{x}_{uv} = \begin{pmatrix} 0 & 0 & f_{uv} \end{pmatrix}$$

$$\mathbf{n}(u,v) = \frac{\begin{pmatrix} -f_u & -f_v & 1 \end{pmatrix}}{\sqrt{1 + f_u^2 + f_v^2}}.$$

and the six fundamental form coefficients :

$$g_{11} = 1 + f_u^2 \quad g_{22} = 1 + f_v^2 \quad g_{12} = f_u f_v$$

$$b_{11} = \frac{f_{uu}}{\sqrt{1 + f_u^2 + f_v^2}} \quad b_{12} = \frac{f_{uv}}{\sqrt{1 + f_u^2 + f_v^2}} \quad b_{22} = \frac{f_{vv}}{\sqrt{1 + f_u^2 + f_v^2}}$$

The expression for *Gaussian* curvature is given by :

$$K = \frac{f_{uu} f_{vv} - f_{uv}^2}{(1 + f_u^2 + f_v^2)^2}$$

And the expression for *mean* curvature is given by:

$$H = \frac{f_{uu} + f_{vv} + f_{uu} f_v^2 + f_{vv} f_u^2 - 2 f_u f_v f_{uv}}{2(1 + f_u^2 + f_v^2)^{3/2}}$$

Thus if we are given a depth map function $f(u,v)$ that possesses first and second partial derivatives, Gaussian and mean curvature can be computed directly.

4.3.1 Estimation of partial derivatives

Partial derivatives of the range image can be obtained by fitting a continuous differentiable function that best fits the data. There are various techniques available in mathematics that have been used by computer vision researchers to determine partial derivatives of depth maps. Let us briefly outline approaches used by researchers to compute derivatives. Besl and Jain [BJ86b] used discrete quadratic orthogonal polynomial fitting at each pixel to estimate derivatives. A quadratic surface is fit at each pixel in the image, using a window convolution operator of size desired by the user. Brady *etal* [BPYA85] used 3×3 difference operators derived by least squares fitting a quadratic to a 3×3 facet of the surface. Yang and Kak [YK86] have derived 3×3 operators using B-splines

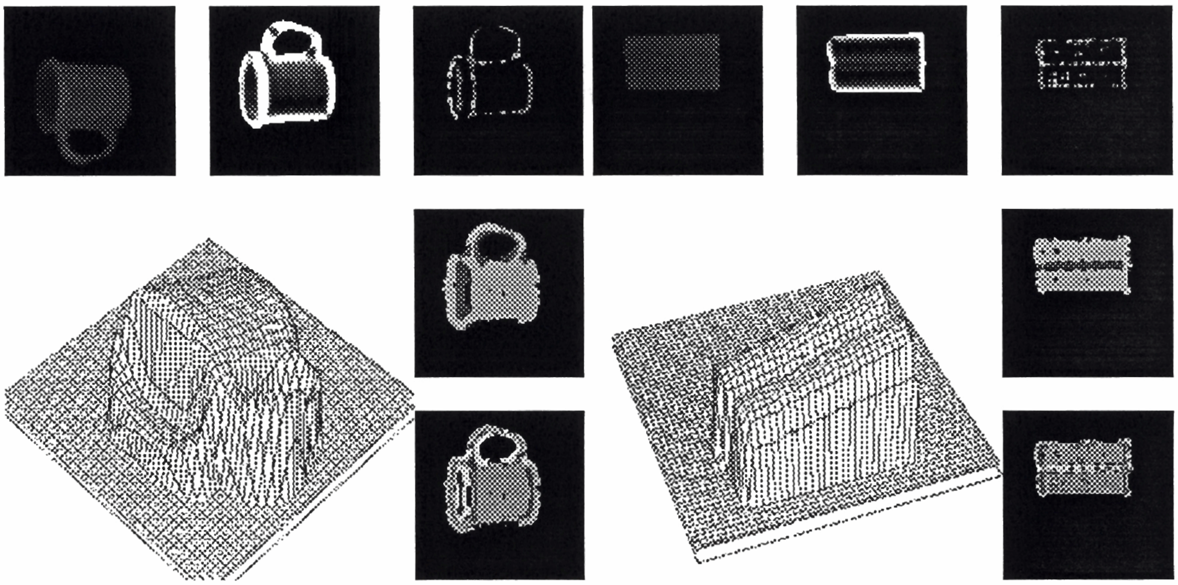


Figure 4.4: **Analysis of Cylinderical surfaces** : Coffee cup (left) and joined cylinders (right). Clockwise from top : Original image, error in local bicubic fit, sign map of Gaussian and Mean curvature, labeled image, perspective plot of image

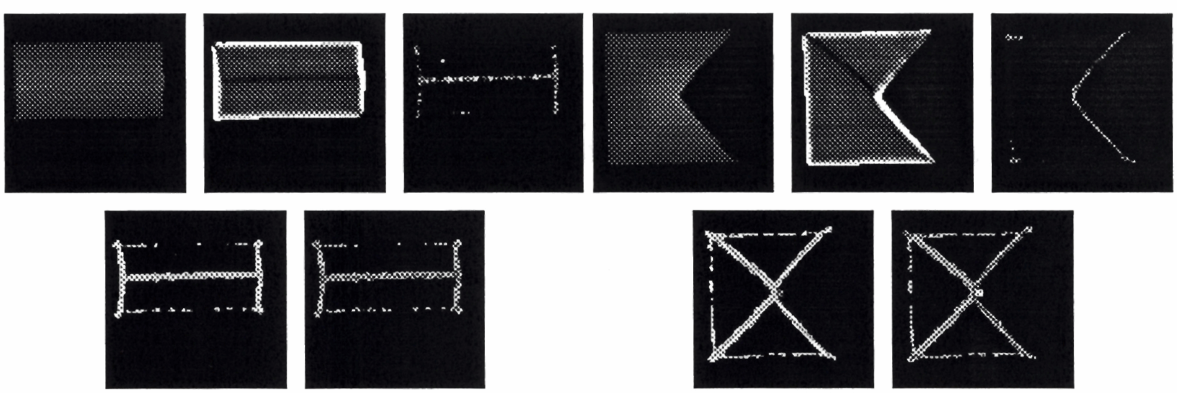


Figure 4.5: **Analysis of flat surfaces** : Prism (left) and pyramid (right). Clockwise from top : Original image, error in local bicubic fit, sign map of Gaussian and Mean curvature, labeled image

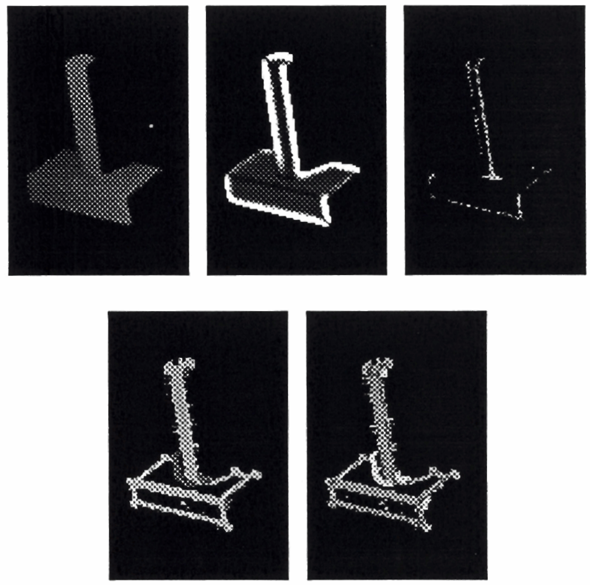


Figure 4.6: **Analysis of a Composite object** : Cylinder joined to box. Clockwise from top : Original image, error in local bicubic fit, sign map of Gaussian and Mean curvature, labeled image

for computing partial derivatives of a range map. These can be combined with Gaussian operator to increase the window size and reduce sensitivity to noise. Sander and Zucker[SZ88] have taken a parabolic quadric surface as the local model.

We have used a fast least squares fitting method to derive partial derivatives in the *symmetric* Neighborhood of a pixel. This method allows the Neighborhood size to be controlled A surface fit of order n can be written as :

$$f(x,y) = \sum_{i,j=0}^{i+j\leq n} a_{ij}x^i y^j$$

We have used third-order ($n = 3$) fitting in the Neighborhood of every pixel to compute first and second order derivatives. Clearly, since the pixel at which derivatives are computed is at the origin, we get :

$$x = 0 \; and \; y = 0$$

$$a_{10} = \frac{\partial f(x,y)}{\partial x} \quad a_{01} = \frac{\partial f(x,y)}{\partial y} \quad 2a_{20} = \frac{\partial^2 f(x,y)}{\partial x^2}$$

$$2a_{02} = \frac{\partial^2 f(x,y)}{\partial y^2} \quad a_{11} = \frac{\partial^2 f(x,y)}{\partial x \partial y} = \frac{\partial^2 f(x,y)}{\partial y \partial x}$$

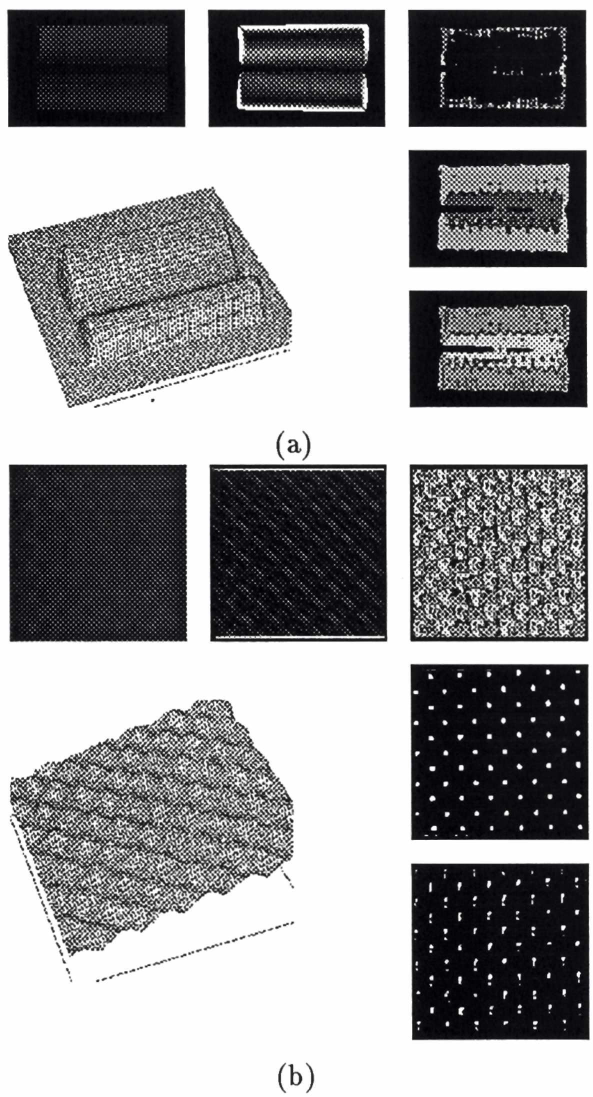


Figure 4.7: **Analysis of smooth surfaces :** (a) Smooth cylindrical surface (outputs as before). (b) Surface with peaks and pits : Clockwise from top : Original image, error in local bicubic fit, labeled image, peak surfaces, and pit surfaces.

Thus derivatives are read off directly from the coefficients. For the purpose of computing derivatives we always have symmetric Neighborhood around the pixel. This fact simplifies the least squares equations.

Using this procedure, we analyzed surfaces in real range images (Figures 4.4 to 4.7) obtained from the GRASP lab range finder. The resolution of the scanner is 1.5mm/pixel. All the images were smoothed by a 5×5 ($\sigma = 1.0$) Gaussian window. The results are shown for objects (figures 4.4,4.5 and 4.6) with cylindrical and flat surfaces, and also for regular objects having undulated surfaces (figure 4.7). The outputs show the original range image, the error in locally estimating the bicubic surface, the sign map of Gaussian and mean curvature, and the image labeled by eight surface types. The black label in the sign image reflects zero value of the curvature, white and gray reflect negative and positive values respectively. The cylindrical surfaces are easily identified by zero Gaussian curvature. Sign of mean curvature determines if they are convex or concave. For example, in the cup image, the visible part of cavity is concave while the external body is convex. Both these can be modeled as quadric patches separately or a cylindrical superquadric collectively. Along the rim, Gaussian curvature indicates an elliptical boundary between the two, while a hyperbolic boundary is seen between the cup and the background. Error image indicates that the error near jump boundaries makes curvature computation unreliable. But, the sign of curvature is generally correct as observed before. Error is high near boundaries and the effect is propagated depending on the window size. The results on the cup image show that it is difficult to locate the discontinuity where the handle and body of the cup join. What is more, in the real world these joins are normally smooth. Thus, information from occluding contour is needed along with patch growing to effectively segment the cup into body and handle.

In the previous section we noted that smooth contours like zero-crossing of the curvature can be located as a boundary formed by two patches of zero Gaussian curvature but with opposite mean curvature sign. In figure 4.7, it is evident that region growing is needed to approximate the contour. It is interesting to see that C_1 discontinuities (roof and ramp edges) appear as locally cylindrical in smoothed images (figures 4.5 and 4.4), while error image indicates a nice fit on such boundaries. So, mean curvature information is useful in detecting creases. In case of composite object formed by cylinder glued to the box, the transversal join is labeled by negative (i.e. concave) mean curvature. While mean curvature sign is important in locating these edges, Gaussian curvature is zero there because of the locally cylindrical shape obtained after uniform smoothing. The final result on the undulating surface in two dimensions (figure 4.7) shows peak surfaces and pit surfaces, which are locally spherical.

Chapter 5

Superquadrics : Deformable Part Models

Volumetric primitives give object-centered descriptions of the object parts. Generalized cylinders [Kli78] proposed for use in vision by Binford [Bin71] have been used as volumetric primitives for their rich vocabulary of shapes. However, this vocabulary of shapes is very difficult to recover from vision data, limiting the actual vocabulary to simple linear-straight-homogeneous-cylinders. Recently, Terzopolous *etal* [TWK88] suggested a deformable model based on the concept of generalized cylinders. The model needs segmented data and user intervention for the initial approximation and is computationally expensive. Superquadric primitives can model only a subset of generalized cylinders shapes, but provide a good compromise for the representation and computational effectiveness. They are capable of modeling tapering and bending deformations, and are recovered effectively by a stable numerical procedure. In this chapter we will first give the definition of deformable superquadrics as given by Solina [Sol87, BS87], and then outline the model evaluation criteria developed by us.

5.1 Introduction

Superquadrics are a family of parametric shapes that have been used as primitives for shape representation in computer vision [Pen86, Sol87, BG87] and computer graphics [Bar81, Bar84]. Superquadrics are like lumps of clay that can be deformed and glued together into realistic looking models.

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

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

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

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

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

If both ε_1 and ε_2 are 1, the surface defines an ellipsoid. Cylindrical shapes are obtained for $\varepsilon_1 < 1$ and $\varepsilon_2 = 1$. Parallelopipeds are obtained for both ε_1 and ε_2 are < 1. We have restricted the model recovery procedure to fit the models with $0 \leq \varepsilon_1, \varepsilon_2 \leq 1$.

5.1.1 Applying Deformations to Superquadrics

The representational power of superquadrics increase further by applying various *deformations* on the basic model. Deformations that we have included in our vocabulary are **tapering** and **bending**.

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

$$X = f_x(z)\,x \quad \text{where} \quad f_x(z) = \frac{K_x}{a_3}z + 1$$

$$Y = f_y(z)\,y \quad \text{where} \quad f_y(z) = \frac{K_y}{a_3}z + 1$$

$$Z = z$$

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

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

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

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

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

Bending transforms r into

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

Where γ is the bending angle

$$\gamma = zk^{-1}$$

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

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

Thus bending and tapering introduce two parameters each in the final superquadric equation, bringing total parameter count to 15. The minimization procedure is capable of recovering all 15 parameters simultaneously. The above equation describes the volumetric model used to describe parts in our system. Henceforth, the term superquadrics will refer to **X** defined above.

5.2 Criteria for Model Evaluation

A superquadric model obtained by least-square fitting the inside-outside function is an overconstrained estimation of data, with more constraints than parameters. Like any parametric approach the goal is to describe a large chunk of data by a few parameters. Such a compact representation comes at a certain price. The recovery procedure assigns equal importance to each point, no matter where the point lies in 3-D space, with the central goal of including the point in the global estimation. The model recovered by such a procedure needs to be analyzed for its suitability in describing data by studying both quantitative measures and qualitative measures. We have identified the following measures for model evaluation in the context of the shape recognition problem :

1. The **goodness-of-fit** measure based on the inside-outside function.

2. The **least squares error** measure based on the true Euclidean distance of individual points from the model surface.

3. The **difference map** produced by comparing the apparent contour formed by the model in the viewpoint direction with the occluding contour of the object.

4. The **error map** produced by comparing the superquadric surface with the points in the range image in the direction of viewpoint.

The first two are global and *quantitative* measures, while the last two are local and *qualitative* in nature.

Now we outline the methods to compute the qualitative measures from a given superquadric model. Computation of the difference map and error map is an issue to be addressed in the chapter on integration. However, generation of the apparent contour and the superquadric surface in image coordinate system (for eventual comparison) are pertinent here.

5.2.1 Goodness-of-fit measure

The inside-outside function for an object centered superquadric model is given by :

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

It determines where a point lies relative to the superquadric surface. If $F(x,y,z) = 1$, point (x,y,z) lies on the surface of the superquadric. If $F(x,y,z) < 1$, the point lies inside and if $F(x,y,z) > 1$, the point lies outside the superquadric. The minimization procedure optimizes the inside-outside function of deformed superquadrics in general position given by :

$$F(x,y,z) = F\left(x,y,z;a_1,a_2,a_3,\varepsilon_1,\varepsilon_2,\phi,\theta,\psi,p_x,p_y,p_z,K_x,K_y,k,\alpha\right)$$

Where ϕ,θ,ψ define the orientation and p_x,p_y,p_z define position of superquadric in space.

Goodness-of-fit is simply the sum of the inside-outside function values at all the points, divided by the total number of points. To use this normalized value of F for model evaluation, we have to assign a meaning to it. In other words, what does it mean for a point to have a goodness-of-fit value? It is certainly not related to the Euclidean distance. We now describe the significance of the goodness-of-fit measure.

Interpretation of Goodness-of-fit

The outermost exponent ε_1 in the inside-outside function F was added by Solina [Sol87] to cancel out the effect of ε_1 in the equation. This modification resulted in better recovery of cylindrical objects. Solina noted only the qualitative effect of the modification, and no mathematical justification was given for it. We provide an explanation which gives an intuitive meaning to the values of inside-outside function, and makes it possible to use this measure for model evaluation.

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

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

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

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

Solving for β yields :

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

It follows from the definition of F that :

$$F = \beta^2.$$

This result shows that the value of inside-outside function F for a point (x, y, z) is nothing but square of the factor by which the axes of superquadric S_1 have to be scaled to make it pass through (x, y, z). This factor can be seen as the amount a superquadric has to be *expanded* or *contracted* (figure 5.1) to make it pass through an arbitrary point in 3 space. This result provides an intuitive explanation for the values of F, with values > 1 indicating expansion and < 1 indicating dilation of the superquadric.

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

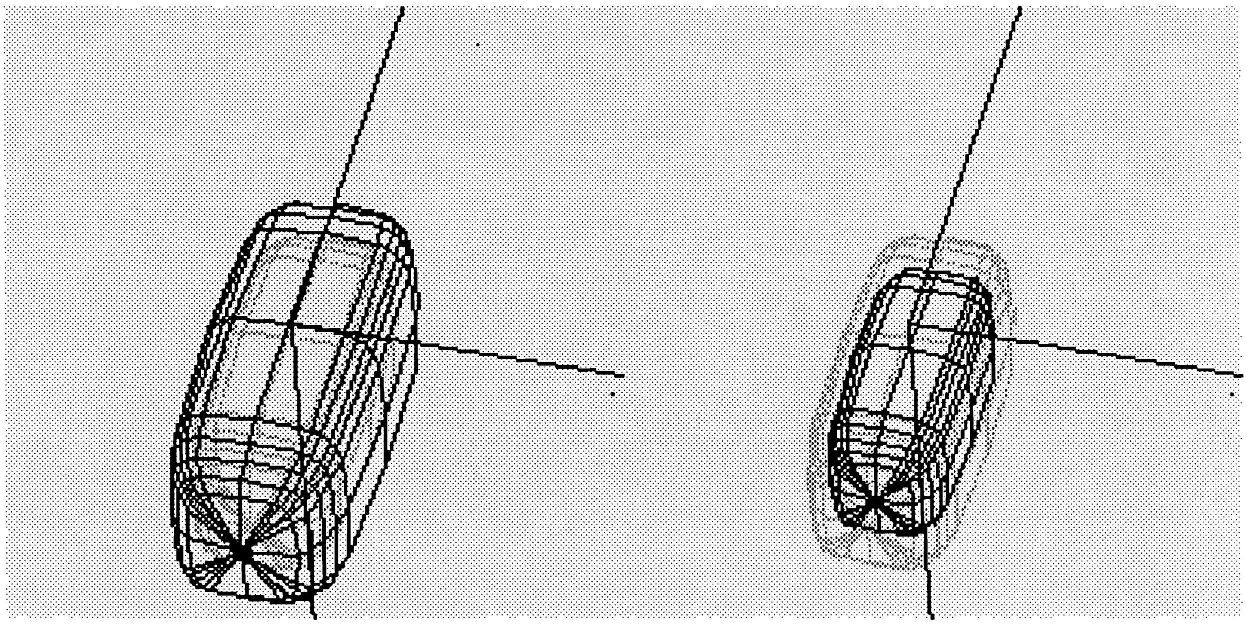


Figure 5.1: β **expansion and contraction of a superquadric model.** left : $\beta = 1.2$, right = $\beta = 0.8$.

5.2.2 Euclidean distance measure

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

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

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

The distance of an arbitrary point in 3 space from a given superquadric model is difficult to compute because of multiple solutions of the analytical formulation of the problem as the non-linear root finding problem. Further, it is not possible to obtain a closed form solution for the problem. We have posed it as a minimization problem, that iteratively minimizes d for a given point and a given deformed superquadric (figure 5.2). In any minimization problem it is imperative to have a close initial approximation. Superquadric surfaces are parametrized by η and ω, and most importantly do not have local minima. Thus the problem is formulated as :

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

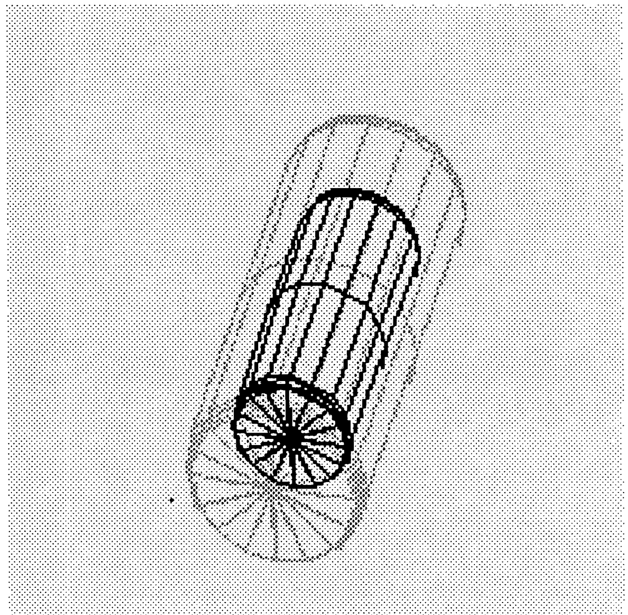


Figure 5.2: **Euclidean distance** and initial approximation for the iterative procedure.

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

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

To ensure convergence to the right solution, a close initial approximation is obtained by extending the expansion/contraction approach introduced in the previous section (figure 5.2. Corresponding to the point $P(x_1, y_1, z_1)$ in 3 space, there is a point $Q(x_2, y_2, z_2)$ on the original superquadric S_1 :

$$x_2 = x_1/\beta,$$

$$y_2 = y_1/\beta,$$

$$z_2 = z_1/\beta,$$

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

The objective is to find R. The function d of two variables is minimized given the initial approximation η and ω, using a quasi-Newton method[1] and a finite-difference gradient. The method requires only function values, a finite-difference method is used to estimate the gradient internally. Though d is differentiable at all points (even with deformations), we have found that supplying external gradient values does not speed up the iterative process in general. The method was found to be accurate upto sixth decimal place for experimental data. We can settle for lower accuracy for faster convergence. The method has been successfully tested on deformed superquadrics.

5.2.3 Apparent Contours of Superquadrics

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

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

$$\mathbf{V}.\mathbf{N} = 0$$

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

$$V_x n_x + V_y n_y + V_z n_z = 0$$

Substituting for $\mathbf{N}$ gives :

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

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

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

Figure 5.3 (a and b) shows the apparent contours of superquadrics generated by the above equation. Unfortunately, there is no closed form solution for a general deformed superquadric, as the surface normal vector N has to undergo deformation by the following rule (derived by Barr [Bar84]) :

$$N' = det\mathbf{J}\mathbf{J}^{-1^T}\mathbf{N}$$

[1] Minimization routine *duminf* from the IMSL version 10.0 library was used with double precision mathematics.

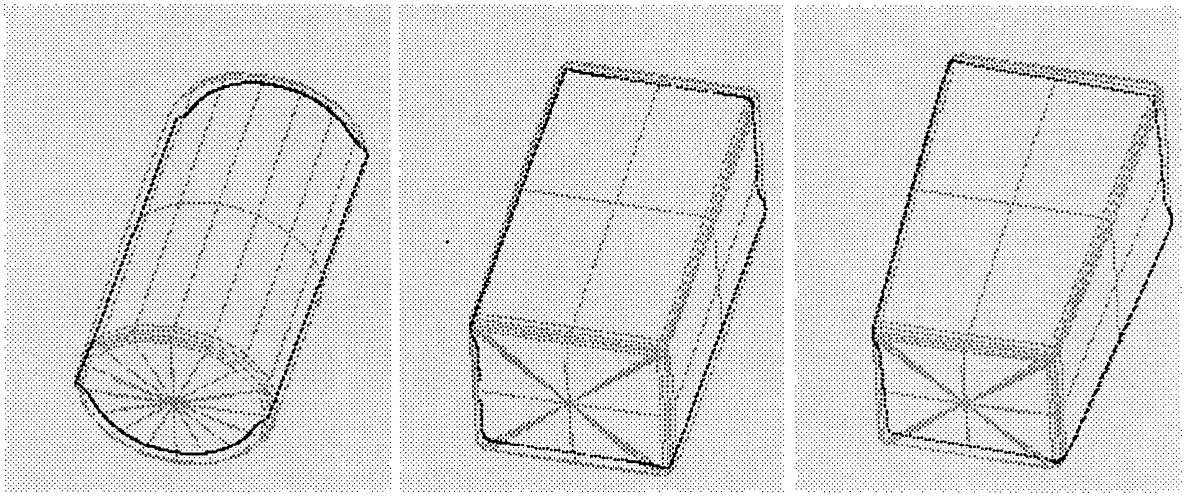


Figure 5.3: **Apparent contours of Superquadrics : for non-deformed box and cylinder, and for a tapered box.**

where **J** is the Jacobian of the deformed superquadric. To trace the apparent contour of a deformed superquadric, we have to vary the angles η and ω systematically. Points on the contour are accumulated in such a way that a closed contour is formed (see figure 5.3(c)). This contour is then orthographically projected on the image coordinate system to make comparisons with the image contour.

5.2.4 Difference map of Superquadric model

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

1. Compute the distance in world coordinate system. We have implemented an iterative procedure based on $\beta-$ expansion and dilation method described earlier.

2. Reconstruct the superquadric surface in the image coordinate system and then perform point by point comparison in z direction to compute the difference map.

The first method needs the occluding contour of the superquadric to determine if a point has distance from the superquadric surface along the given direction. The second method simply transforms the superquadric into image coordinate system, where both the difference map as well as occluding contour can be traced by the same method as image contour tracing. We have implemented both the methods, but the results shown in the proposal are computed using second method.

Chapter 6

Research Proposal : An Integrated Approach

Having discussed the shape primitives individually and identified the role of each primitive in shape segmentation and description, we now focus our attention on the goal of this research, which is to develop an effective control structure that works in conjunction with these modules to extract the part-structure of a complex object. The primitives give a hierarchy of shape descriptions, ranging from the planar contour level to the three-dimensional volumetric level. The problem that we wish to solve can be stated in the following way. Given that we have all three different modules for extracting volume, surface and boundary properties, how should they be invoked, evaluated and integrated? There are two possibilities. The first one is to apply all three modules simultaneously. The second is to apply them strictly in a predetermined sequence. In the parallel approach conflicting hypotheses can arise that would have to be resolved. The sequential method may lead the segmentation process in a wrong direction so that backtracking would sometimes be necessary. A combined approach where all three methods could interact would not be so vulnerable. This opens up the problem of evaluating and comparing information embedded in models built by different aggregation methods. How to evaluate the models individually and collectively by comparing against one another? What do you do when different types of models do not reinforce each other? Some method of resolving the conflicts has to be devised that assigns confidence levels to each primitive. How do we know when to trust a model and when not to? To provide motivation for our approach, we will first provide examples of simple situations that highlight these issues. We will then describe our proposed approach and progress made so far. Finally, we will summarize our proposal.

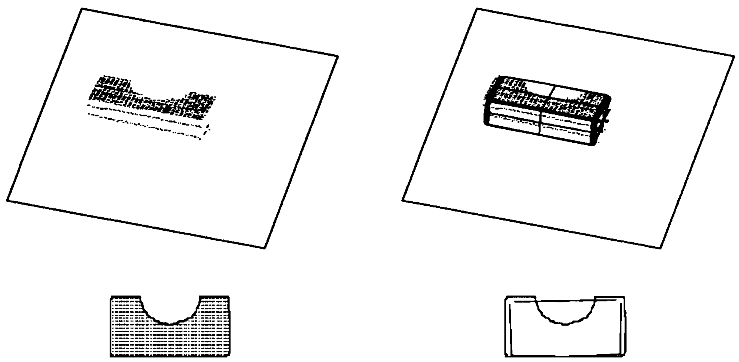


Figure 6.1: **Box with a circular cutout (an arch) :** Though the volumetric model gives acceptable fit in terms of error function, it does not account for the cutout.

6.1 Motivation

Before we propose our control strategy, it is instructive to study the behavior of the shape primitives on the actual data consisting of objects of varying complexity. The volumetric shape recovery procedure [Sol87] was applied to a set of range images of single objects (Figures 6.1 to 6.6). The contour obtained by tracking the occluding boundary and the contour of the recovered volumetric model are compared in all the cases. For the objects in figures 6.4 to 6.6, surfaces reconstructed from the superquadric model are compared with the original range data.

While the volumetric model gives a holistic explanation of the whole object it can miss details that are beyond the scope of the model. An overall measure of goodness of fit, like the residual from least-squares fit, or the distance measure does not always give an accurate evaluation of the appropriateness of the volumetric model. Although models can have acceptable overall goodness-of-fit, like the volumetric model for the box with cut-out (figure 6.1), they need not be the acceptable representations of the object. On the other hand, for value of the goodness-of-fit in same range, volumetric models for the vase (figure 6.5) and the box-with-jagged-edge are more or less acceptable volumetric representations of the actual object. This argues for a measure other than the quantitative measure of goodness-of-fit or Euclidean distance. The qualitative measure obtained by comparing the local boundary of the object in the range image with the boundary of the recovered volumetric model can point out the limitations of the volumetric model and suggest improvements in segmentation or refinement in shape representation. When boundaries do not

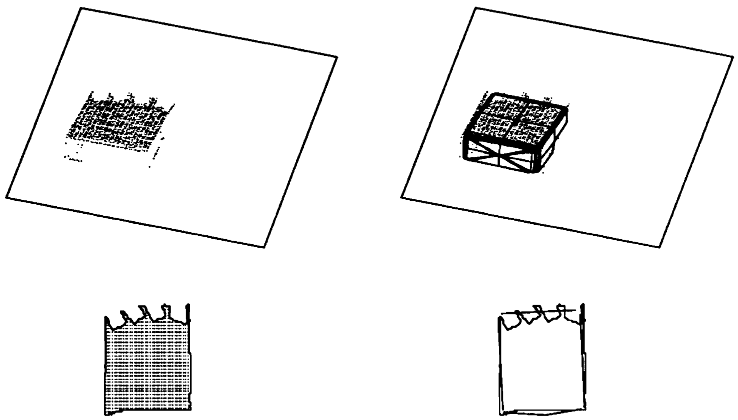


Figure 6.2: **Box with jagged edge :** The difference between the two outlines is small in comparison with the overall size of the object. The jagged edge could be brushed away as a detail.

coincide, preference should be given to actual boundary in the range image, but the possibility of missing data (due to self occlusion) must also be considered.

The Part *versus* detail issue can be addressed at individual primitive levels as well as collectively. For example, the vase in figure 6.5 is formed of three second-order surface patches, collectively organized in a cylindrical shape. At the volumetric level, a cylindrical model is sufficient to describe the overall shape. Details have to be obtained in terms of second order patches at the surface level. Contour analysis signals the presence of details on the object, and accepts the superquadric model. However, the superquadric model is accepted only after the surface comparison yields acceptable error. Thus, both the qualitative measures are essential for model evaluation. The presence of details in the form of a jagged edge is similarly detected in figure 6.2. It should be noted that the details are not neglected in the final description. They are ignored by only the volumetric model. Contour and surface description are generated in detail with the final decision of assigning labels postponed to the domain-dependent processing. For example, a pitcher's small dent on the rim is necessary for recognition, so it cannot be ignored by a bottom-up shape description process. However, the decision to segment the object into volumetric primitives has to be taken at the geometric level.

Closely tied to the issue of part-detail is the issue of part-whole relationships. What cannot be brushed away as a detail has to be considered a part at the volumetric level. It is easy to detect presence of distinct parts in the object (figures 6.3,6.4 and 6.6), by contour and surface

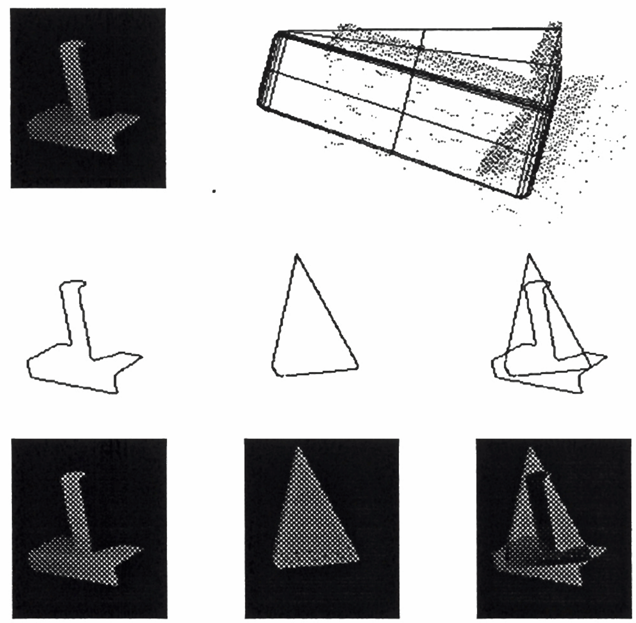


Figure 6.3: **A composite object (cylinder glued to box):** The poor approximation of the object reflects need for segmentation.

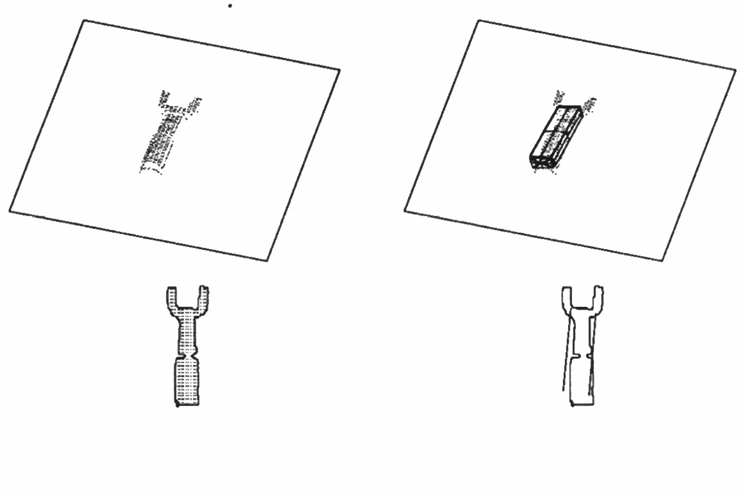


Figure 6.4: **Object with parts (a wrench) :** The two boundaries coincide in only part of the image alerting to the fact that the object has parts.

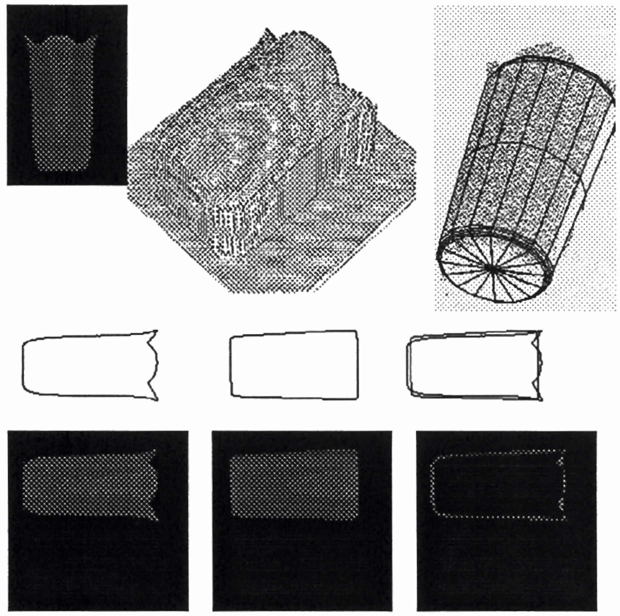


Figure 6.5: **Object with surface dėtail (A vase) :** The difference between the two outlines is negligible compared to the overall size of the object. However, to recover more detail, and to define the internal boundaries, surface description is necessary.

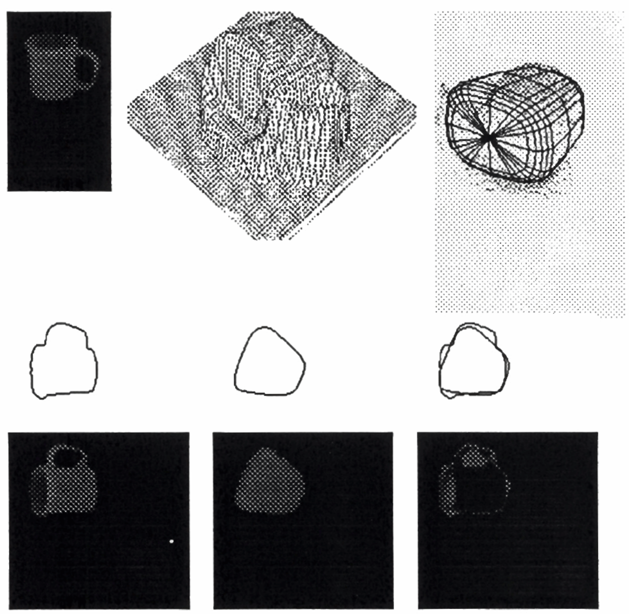


Figure 6.6: **Object with hole and cavity :** Surface and contour information is required to effectively segment it into parts and to define concavities on the surface.

comparisons. It is another matter to recover them in terms of primitives. It needs partitioning the object into parts at surface boundaries and contour concavities. How do surfaces and contours interact to generate hypotheses about parts and then use superquadrics to verify the hypotheses? What if there is no volumetric description possible for the part? What is the best approximation for such a part? What do we mean by acceptable shape description? To attempt answers to these questions we propose our approach next.

6.2 The Proposed Approach

The detailed flow diagram of our proposed approach is shown in the figure 6.7. The past research of 3-D part segmentation has been mostly theoretical. To satisfy the practical constraints of computability and robustness we propose a parallel closed-loop segmentation process with active feedback between different description modules. From the examples in the previous section it is clear that interaction among different primitives is imperative.

To incorporate the best of the coarse to fine and fine to coarse segmentation strategy we propose to perform volume, surface, and boundary fitting in parallel on the input data. The volumetric shape recovery is a global method, going from very coarse to fine fitting on the part level while surface and boundary detection going from fine to coarse. These two processes are complementary in the approach of explaining the data, accounting for global position, orientation, size and shape such that the descriptions obtained at the global and local levels support each other. Thus, it is the *local* processing by the *Occluding contour* and the *Surface* modules that is done in parallel and has to be done only once. The global description at the contour and surface level is obtained by refining these initial measures in a closed-loop feedback. The *Curve Segmentation* module and the *Surface Segmentation* module perform the refinements in a typical fine to coarse manner through an internal feedback as well as an external feedback from the *control module* (figure 6.7). For example, fitting global second order patches on the surface needs *intra-primitive* feedback from the surface level itself, while detecting surface boundaries also needs *inter-primitive* feedback from the occluding contour. The segmented descriptions are evaluated and integrated at the inter-primitive level by the control module along with the evaluation of superquadric model to combine the descriptions. Since the superquadric model estimation treats data globally, the initial estimation might not be acceptable due to presence of parts. Once the control module (the global segmentor) generates hypotheses about parts, the superquadric procedure gives the best fitting models for verification of the hypotheses. Thus the model recovery procedure works as the hypotheses verifier at the volumetric level. It then follows that *part-segmentation* is the core of the problem.

To achieve an effective segmentation of a single viewpoint scene, the control structure has to

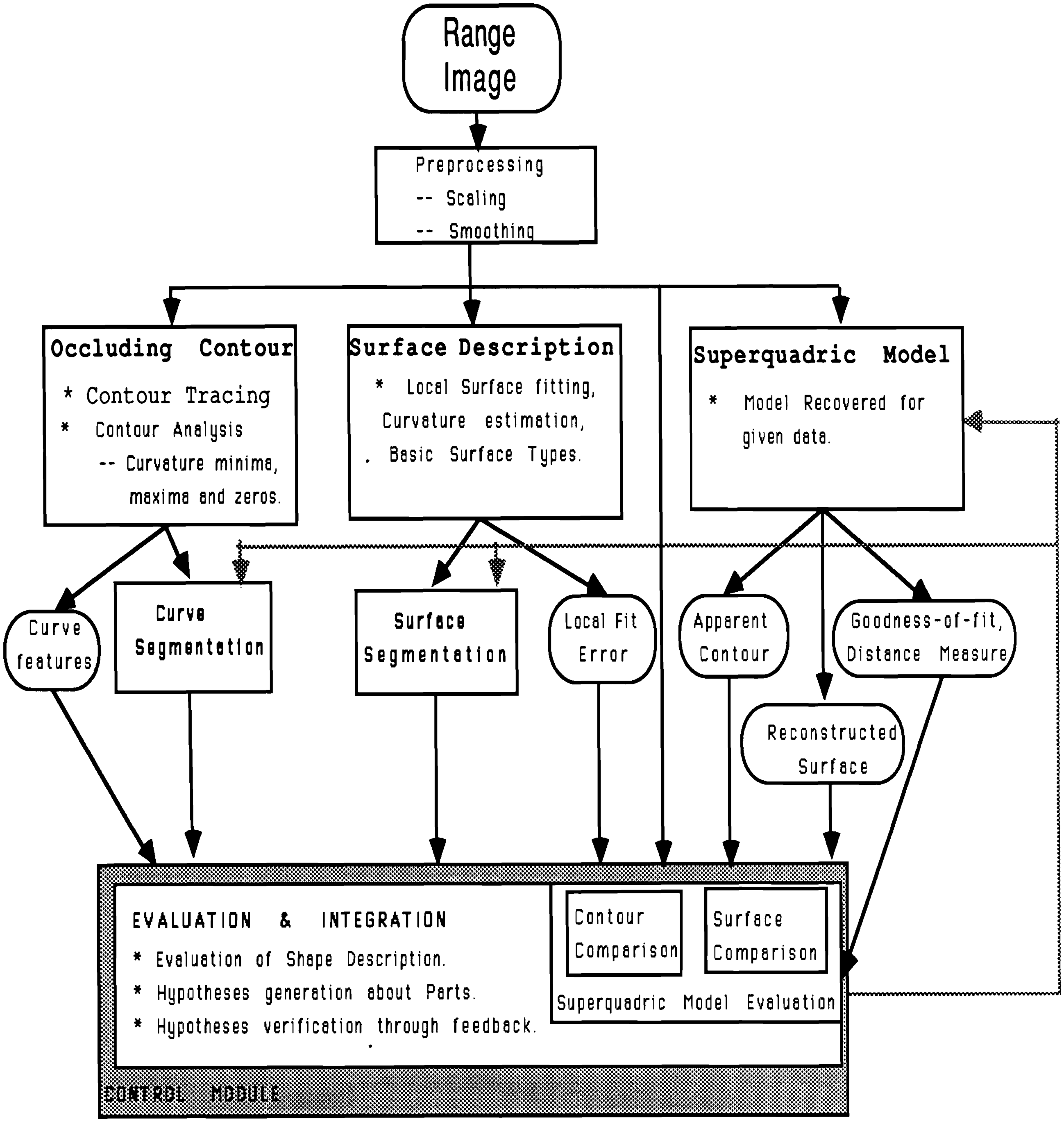


Figure 6.7: **Detailed block diagram of our proposed approach.**

determine the reliability of information obtained from each primitive. Superquadrics being part-models, need to be compared with the bounding contour and available surface points to evaluate suitability of the recovered model. Surfaces, for most part, complement the information provided by bounding contours. Bounding contours are viewpoint dependent and may not account for all relevant contours needed for complete segmentation or description. This is obviously the case when viewpoint is not general. Thus, in some cases, when volumetric information is not available, surface information along with bounding contour can determine if the object is in a general position or not and ask for information from different viewpoint (or rotate the object). For some objects, it may not be possible to obtain data from a viewpoint such that the object can be segmented by analyzing only the contour. In such a case, if surface information strongly suggests segmentation along a surface discontinuity, bounding contour should not lower our confidence in surface information. On the other hand, if contour suggests a possible segmentation and there is no support from surfaces, a decision will have to be made about the possibility of segmentation assuming a possible smooth join between part and object body. Superquadrics essentially provide global description of individual parts and give the feedback as to the possibility of a further segmentation of that part. They lack the local information needed to suggest possible segmentation sites. Contour and Surfaces, on the other hand, actively hypothesize and carry out segmentation. The process continues until a satisfactory description of parts is achieved.

How do we evaluate the intermediate descriptions? As seen in the examples, the global feedback loop between the individual descriptors and the control module gives a set of "difference measures" at the contour and surface level. Many techniques are available for planar contour matching and surface matching in pattern recognition literature. We want to use this feedback for evaluation of the intermediate descriptions as well as for further segmentation. The differences can be interpreted as "overestimation" or "underestimation" of actual data by recovered models. Since superquadrics tend to undersegment (figure 6.3), and bring in symmetry considerations, the difference patterns generated by them consist of overestimated and underestimated regions (e.g. cup in figure 6.6).

What do you do if different types of models do not mutually reinforce each other? In such cases, one would normally prefer models of smaller granularity that are less prescriptive models that closely follow the data in the image. Contour description which is local by the nature of the data can guide segmentation. But this has to be distinguished from the case when the information that could give rise to low level models is not present. A good example are the well known phenomena of illusory contours in human perception. We can perceive solid shapes although a large part of boundary lines physically do not exist. Though *perceptual* shape resulting from subjective contours or illusions is not our concern in this research since we are dealing with *physical* shape only, the observation is relevant. In conflicting situations information has to be reorganized and the control system adapted. Also, in simple situations like that in figure 6.3 contours may not give exact site for segmentation. True, the pair of concavities in the contour segment the *contour* into two parts belonging to two distinct parts in 3-D, they do not provide a *mechanism* to segment the 3-D object as such. Indeed, partitioning into relevant parts requires surface boundaries (figure 6.3, shown in the mean curvature sign map). This example presents the case for *not* relying entirely on contour information for 3-D segmentation, although contour level segmentation from the same information is correct. Also, discontinuities in surfaces may not project as discontinuities in the planar contour. Thus, the control module has to account for disagreement among primitives, by choosing the one that is most plausible under single viewpoint.

A pertinent issue to address at this time is are we doing too much by simultaneously describing shape at three levels? Is there some way of recognizing the dimensionality of the scene and applying only the primitives needed to the scene? It is true that in a restricted domain, dimensionality is known and an elaborate approach is not needed. We are proposing a general approach that is not tied to a domain of particular dimension. It is certainly possible to recognize some aspects of shape by low-level models, and adapt the control structure accordingly. If all the objects are in the scene are flat, then description can be achieved in terms of only contour primitives, though flat models exist in superquadric vocabulary. Surface models are not at all needed. But the superquadric models will still provide a global region-based shape measure that is not possible to obtain with our contour primitive. A typical way of achieving this in our design is to apply all three primitives as usual. The fact that the scene is two-dimensional will be apparent from the results of all the three modules. The control module can then decide not to go for surface segmentation at global level. Let us consider another scenario. If the object has a hole (visible as an occluding contour, figure 6.6), there is a good probability of not obtaining a superquadric model for it. However, this is not always true, take for example, a box with a cylindrical hole through it. A model for the box exists and is recoverable.

During the segmentation process the control module has also to decide on part/whole (or part/detail) relationships. This requires determining the scale of a potential part given the overall size of the object and deciding to consider it a part or just a detail of the object that can be ignored (implying that current description is adequate). This requires that the global control program must have the resolution of the parameters and thresholds predetermined, or if possible, adjusted during the process. Some of those parameters are the following:

1. The size (or range of sizes) of the local neighborhood for local processing.

2. Acceptable tolerance for error in model evaluation, keeping in view the limitations of shape models.

3. The size and shape of models. When does a circular cylinder become elliptical, or at what angle two planes must meet for a roof edge to exist?

4. The number (or range) of expected segmented units,

5. The thresholds for partitioning and aggregation.

6. The level of details that we wish to explain.

We now briefly describe the progress in implementing our approach. As evident from the results shown in the proposal, we have completed the implementation of the bulk of individual description modules. The contour description module needs reliable computation of contour features, for which we are investigating the possibility of incorporating scale-space approach to the Rosenfeld's algorithm. Preliminary results are encouraging, as seen for the cup image in chapter 3. Surface boundary detection is an open problem, and we plan to deal with it in conjunction with the occluding contour and quadric patch growing. We are confident that our parallel approach of surface boundary and surface patch description will provide better localization and reliability for the boundaries. Beyond the "black box" of superquadric model recovery procedure, we have implemented algorithms for apparent contour generation, model reconstruction in image coordinate system, Euclidean distance computation, and goodness-of-fit interpretation. The next step is to design and implement the control module as discussed above.

6.3 Proposal Summary

The goal of this research is to obtain structured shape descriptions of complex three-dimensional objects in range images in terms of parts defined by a hierarchy of shape primitives. We posed the shape recognition problem as a combination of shape description and shape segmentation problems and presented arguments for using shape primitives at multiple levels. We then described the criteria for selection of shape primitives and selected hierarchical shape description model consisting of contour, surface and volumetric primitives. The chapters on shape primitives outlined the shape description and decomposition methods based on them. Rules for partitioning of objects as proposed by vision researchers were discussed for all the three primitives. We observed that most of the work on part segmentation is theoretical in nature, and the crucial aspect of computability is seldom addressed. Segmentation techniques based on single primitives have severe restrictions on the shape vocabulary and the scope of description. It was observed that certain vital issues like surface boundary detection are still unsolved in computer vision. With computability and robustness as our primary concern we proposed a parallel closed-loop segmentation process with active feedback between different description modules. The descriptions thus obtained are independent of position, orientation, scale, domain and domain properties, and are extremely useful for top-down high-level domain-dependent symbolic reasoning processes.

Bibliography

[AB73] G. J. Agin and T.O. Binford. Computer description of curved objects. In *Proceedings of International Joint Conference of Artificial Intelligence*, pages 629–640, 1973.

[AB86] H. Asada and Michael Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):2–14, January 1986.

[Arn74] R. Arnheim. *Art and Visual Perception.* University of California Press, 1974.

[Att54] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61:183–193, 1954.

[AWB87] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. In *First International Conference on Computer Vision*, pages 35–54, London, 1987.

[BA84] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, 3(3):36–61, 1984.

[Baj89] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 1989.

[Bar81] Alan H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):, January 1981.

[Bar84] Alan H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18(3):21–30, July 1984.

[BB82] Dana H. Ballard and Christopher M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[BG87] T.E. Boult and A. D. Gross. Recovery of superquadrics from depth information. In *Proceedings of Spatial Reasoing and Multi-Sensor fusion workshop*, pages 128–137, St. Charles,IL, 1987.

[BG88] Terrance E. Boult and Ari D. Gross. On the recovery of superellipsoids. In *AAAI Workshop*, 1988. Submitted.

[BH77] M. V. Berry and J.H. Hannay. Umbilic points on gaussian random surfaces. *Journal Phys. A : Math. Gen.*, 10(11):1809–1821, 1977.

[BH87] B. M. Bennett and D. D. Hoffman. *Image Understanding 1985-86*, chapter Shape decompositions for visual recognition: the role of tranversality. Ablex Publishing Corporation, Norwood, NJ, 1987.

[Bie85] I. Biederman. Human image understanding: recent research and theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.

[Bie87] I. Biederman. Matching image edges to object memory. In *Proc. of International Conference on Computer Vision*, pages 384–392, 1987.

[Bin71] T. O. Binford. Visual perception by computer. In *IEEE Conference on Systems and Control*, Miami, 1971.

[Bin82] Thomas O. Binford. Survey of model-based image analysis systems. *International Journal of Robotics Research*, 1(1):18–64, Spring 1982.

[BJ85] Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1), March 1985.

[BJ86a] P. J. Besl and R. C. Jain. Segmentation through symbolic surface descriptions. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 77–85, 1986.

[BJ86b] P. J. Besl and Ramesh C. Jain. Invariant surface characteristics for three dimensional object recognition in range images. *Computer Vision, Graphics, and Image Processing*, 33(1):33–88, 1986.

[BN78] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[BPYA85] M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing surfaces. *Computer Vision, Graphics, and Image Processing*, 32(1):1–28, 1985.

[Bra83] M. Brady. *Human and Machine Vision*, chapter Criteria for representation of shape. Academic Press, Orlando,FL, 1983.

[Bro83] Rodney Brooks. Model-based 3-d interpretations of 2-d images. *IEEE Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.

[BS87] R. Bajcsy and F. Solina. Three dimensional shape representation revisited. In *First International Conference on Computer Vision*, pages 231–240, London, 1987.

[BSG88] Ruzena Bajcsy, Franc Solina, and Alok Gupta. *Segmentation versus object representation - are they separable?* Technical Report MS-CIS-88-58, University of Pennsylvania, 1988. Also to appear in the Range Image Understanding book published by Springer.

[BY84] M. Brady and A. Yuille. An extremum principle for shape from contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(3):288–301, 1984.

[CA87] C. C. Chien and J. K. Aggarwal. Shape recognition from single silhouettes. In *Proceedins of the International Conference on Computer Vision*, pages 481–490, 1987.

[CA89] C. H. Chien and J.K. Aggarwal. Model construction and shape recognition from occluding contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(4):372–389, April 1989.

[COCD87] David Cyganski, John A. Orr, T. A. Cott, and R. J. Dodson. Development, implementation, testing, and application of an affine invariant curvature funtion. In *Proceedings of the International Conference on Computer Vision*, pages 496–500, London, 1987.

[Dav77] Larry S. Davis. Understanding shape : angles and sides. *IEEE Transactions on Computers*, C-26(3):236–242, March 1977.

[dC76] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, NJ, 1976.

[Fan88] T. J. Fan. *Describing and Recognizing 3-D Objects Using Surface Properties.* PhD thesis, University of Southern California, August 1988. Technical report IRIS 237.

[FB86] M. A. Fischler and R.C. Bolles. Perceptual organization and curve partitioning. *IEEE Pattern Analysis and Machine Intelligence*, 8:100–105, 1986.

[FD77] H. Freeman and Larry S. Davis. A corner-finding algorithm for chain-coded curves. *IEEE Transaction on Computers*, 297–303, 1977.

[Fre67] H. Freeman. *Models for the Perception of Speech and Form*, chapter On the classification of line drawing data, pages 408–412. MIT Press, 1967.

[Fre74] H. Freeman. Computer processing of line-drawing imgaes. *Computing Surveys*, 6:57–97, 1974.

[Gom72] E. H. Gombrich. *Art and Illusion.* Phaidon Press, London, 1972.

[GP74] V. Guilleman and A. Pollack. *Differential Topology.* Prentice-Hall, 1974.

[Gup88] A. Gupta. *Range Image Segmentation for 3-D Object Recognition.* Master's thesis, University of Pennsylvania, Philadelphia, 1988. Technical Report : MS-CIS-88-32.

[HC52] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination.* Chelsea Publishing Co., 1952.

[Hor83] B.K.P. Horn. The curve of least energy. *ACM Trans. on Mathematical Software*, 9(4):441–460, December 1983.

[HR82] D. D. Hoffman and W. A. Richards. Representing smooth plane curves for recognition : implications for figure-ground reversal. In *Proc. of National Conference on Artificial Intelligence*, Pittsburgh, 1982.

[HR85] D.D. Hoffman and W.A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.

[HWL83] R. M Haralick, L. T. Watson, and T. J. Laffey. The topographic primal sketch. *The International Journal of Robotics Research*, 2(1):50–72, Spring 1983.

[KD98] D. Kuan and R. Drazovich. Model-based interpretation of range imgery. In *Proceedings of AAAI Conference*, pages 210–215, 198?

[KK87] M. W. Koch and R. L. Kashyap. Using polygons to recognize and locate partially occluded objects. *IEEE Pattern Analysis and Machine Intelligence*, PAMI-9(4):483–494, July 1987.

[Kli78] W. Klingenberg. *A course in Differential Geometry. Graduate Texts in Mathematics*, Springer-Verlag, 1978.

[KvD82] J. Koenderink and A. van Doorn. The shape of objects and the way contours end. *Perception*, 11:129–137, 1982.

[KWT87] Michael Kass, Andrew Witkin, and Demetri Terzopolous. Snakes: active contour models. In *Proceedings of the International Conference on Computer Vision*, London, 1987.

[Lan84] D.J. Langridge. Detection of discontinuities in the first derivatives of surfaces. *Computer Vision, Graphics and Image Processing*, 27:291–30, 1984.

[Low88] D. G. Lowe. Organization of smooth image curves at multiple scales. In *Proceedings of the International Conference on Computer Vision*, pages 558–567, 1988.

[LZ87] Y. G. Leclerc and S. W. Zucker. The local structure of image discontinuities in one dimension. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-9(3):341–355, May 1987.

[MA77] J. W. McKee and J.K. Aggarwal. Computer recognition of partial views of curved objects. *IEEE Transactions on Computers*, C-26(8):790–800, 1977.

[MA83] W. Martin and J.K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Trans. on Pattern Analysis and Machine Intelligencve*, PAMI-5(2), 1983.

[Mal88] S. G. Mallat. *Multiresolution representation and Wavelets.* PhD thesis, University of Pennsylvania, Philadelphia, 1988.

[Mar77] D. Marr. Analysis of occluding contour. *Proc. of Royal Society of London*, B(197):441–475, 1977.

[Mar82] D. Marr. *Vision.* Freeman, San Francisco, 1982.

[Mil88] E. E. Milios. Recovering shape deformation by an extended circular image representation. In *Proceedins of the International Conference on Computer Vision*, pages 20–29, 1988.

[MM86] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Pattern Analysis and Machine Intelligence*, PAMI-8(1):34–43, January 1986.

[Mok88] F. Mokhtarian. Evolution properties of space curves. In *Proceedings of International Conference on Computer Vision*, pages 100–105, 1988.

[Nac84] Lee R. Nackman. Two-dimensional critical point configuration graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):442–449, July 1984.

[NB77] R. Nevatia and T.O. Binford. Description and recognition of complex-curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[NP85] L. R. Nackman and S. M. Pizer. Three-dimensional shape description using the symmetric axis transform i:theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):187–202, March 1985.

[Pav77] Theo Pavlidis. *Structural Pattern Recognition.* Springer-Verlag, Heidelberg, 1977.

[Pav80] T. Pavlidis. Algorithms for shape analysis of contours and waveforms. *IEEE Trans. on Pattern Analysis and Machine Intelligencve*, PAMI-2(4):301–312, July 1980.

[PB84] J. Ponce and Michael Brady. Toward a surface primal sketch. In *Proc. of Conf. on Robotics and Automation*, pages 420–425, March 1984.

[Pen86] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 1986.

[Pen87] Alex P. Pentland. Recognition by parts. In *International Conference on Computer Vision*, pages 612–620, London, 1987.

[Pen88] Alex Pentland. *Automatic Extraction of Deformable Part Models.* Technical Report 104, MIT Media Lab, July 1988.

[PH74] T. Pavlidis and S. L. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, c-23(8):860–870, 1974.

[Por83] I. R. Porteous. The normal singularities of surfaces in r3. In *Proceedings of Symposia in Pure Mathematics*, 1983. Volume 40, Part 2.

[Rao88] Kashipati Rao. *Shape Description from Sparse and Imperfect Data.* PhD thesis, University of Southern California, December 1988. Technical report IRIS 250.

[RJ73] A. Rosenfeld and Emily Johnston. Angle detection on digital curves. *IEEE Transactions on Computers*, 875–878, September 1973.

[RK82] A. Rosenfeld and A.C. Kak. *Digital Picture Processing.* Academic Press, Orlando, FL, 1982.

[Ros78] E. Rosch. *Cognition and Categorization*, chapter Principles of Categorization. Erlbaum, Hillsdale, NJ, 1978.

[RW75] A. Rosenfeld and J. S. Weszka. An improved method of angle detection on digital curves. *IEEE Transactions on Computers*, September 1975.

[SB78] B. Soroka and R. Bajcsy. A program for describing complex three-dimensional objects using generalized cylinders as primitives. In *Proc. of the IEEE Conference on Pattern Recognition and Image Processing*, pages 331–339, 1978.

[Sha80] L. G. Shapiro. A structural model of shape. *IEEE Trans. Pattern Analysis and MAchine Intelligence*, PAMI-2(2):111–126, March 1980.

[SK85] D. R. Smith and Takeo Kanade. Autonomous scene description with range imagery. *Computer Vision, Graphics and Image Processing*, 31(3):322–334, 1985.

[Sol87] Franc Solina. *Shape Recovery and Segmentation with Deformable Part Models.* PhD thesis, University of Pennsylvania, 1987. Technical Report MS-CIS-87-111.

[Stewn] K. A. Stevens. The line of curvature constraint and the interpretation of 3-d shape from parallel surface contours. In *Proceedings of the International Joint Conference in Computer Vision*, pages 1057–1061, unknown.

[SZ88] P. T. Sander and S.W. Zucker. *Inferring Differential Structure from 3-D Images: Smooth Cross Sections of Fibre Bundles.* Technical Report TR-CIM-88-6, McGill University, 1988.

[TH84] B. Tversky and K. Hemenway. Objects, parts, and categories. *Journal of Experimental Psychology:General*, 113(2):169–193, 1984.

[TWK88] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: recovering 3d shape and nonrigid motion. *Artificial Intelligence*, 91–123, August 1988.

[Wit83] A. P. Witkin. Scale space filtering. In *Proc. 8th Int. Joint Conf. on Artificial Intelligence*, pages 1019–1022, West Germany, 1983.

[WT83] Andrew P Witkin and Jay M. Tenenbaum. *Human and Machine Vision*, chapter On the Role of Structure in Vision. Academic Press, Orlando,FL, 1983.

[YK86] H. S. Yang and A. C. Kak. Determination of the identity, position and orientation of the topmost object in a pile. *Computer Vision, Graphics, and Image Processing*, 36:229–255, 1986.

[Yui89] A. L. Yuille. Zero crossings on lines of curvature. *Computer Vision, Graphics and Image Processing*, January 1989.