The Efficient 3D Object Representation and Recognition using Part-Based Superquadric Models from a Range Image
The Efficient 3D Object Representation and Recognition using Part-Based Superquadric Models from a Range Image

이 논문을 博士學位 論文으로 提出함

2001年 6月

中央大學校 尖端映像大學院
映像工學科 映像情報技術 専攻
李 瑾 鎧
Abstract

Lee, Sun-Ho
Department of Imaging Engineering
Graduate School of Advanced Imaging Science,
Multimedia, and Film, Chung-Ang University

In this paper, we investigate an approach to the problem of efficient 3D object representation and recognition using hierarchical surface and volumetric description. This paper addresses the problem of representing 3D object parts for the purpose of recognition from single viewpoint range data by using a hierarchical surface and volumetric description. We assumed that our target 3D objects are composed of convex volumetric primitive parts and can be modeled by superquadrics, which are composed together in a solid constructive modelling manner using union operation. For the volumetric description, we used a modified superquadric-based recover-and-select segmentation paradigm which takes input as a 3D range image. We segmented and decomposed the input object into several parts because superquadrics can be recovered directly from range data without considering any other geometric models. For each decomposed superquadric part, we calculated surface normals and curvatures to describe surface type and surface relationships for the next recognition level.

We constructed the modelbase which has two layer of hierarchical structure by using surface and volumetric descriptions. In the volume layer, we analyzed volumetric properties of each part-primitive that relates constitutes the model object with superquadric model parameters, and thus the best candidate parts are indexed according to the volume-level feature matching procedure between model objects and input scene object. Surface descriptions such as surface type, surface junction relationships between neighboring surface patches of each part are effectively used in the surface-level matching process for recognition. Because we can use those as matching features to discriminate between two
different objects which have the same volumetric dimensions of superquadric parameters. The proposed system has been tested on both synthetic and real range images, and the related results are presented.

Among various existing part-level models, generalized cylinders and superquadrics are the most widely used models for 3D object recognition in computer vision. Applications of generalized cylinders are, however, often limited because of the difficulty in recovery from range images. On the other hand, superquadrics, are more widely used by the following reasons: (i) They can express wide variety of 3D shapes, like round edges and corners, as well as standard geometric solids, using a small number of model parameters, and (ii) robust methods exist for reconstruction from range images. This paper will presents a novel volumetric approach to model-based 3D object recognition by using superquadric part-primitives. The superquadric part-primitive is defined as volume primitive, which represents a basic geometric solid object (a sphere, a box, a cylinder, etc.). Therefore, it can easily be recovered by using a finite set of superquadric model parameters.

We assumed that our target objects are composed of convex volumetric primitives without any holes, and that the observer can freely choose its viewing position. Objects can be partitioned at concave discontinuities to give parts that can be described by superquadrics part-primitives. For the purpose of recognition through the matching between the scene object and the model object, it is necessary for the input object to be decomposed into several part-primitives. We used the recover-and-select paradigm to segment and decompose the input object because superquadrics can be recovered directly from range data without considering any other geometric models. Note that this fact is contrary to the common belief that the recovery of volumetric models is possible only after the data has been pre-segmented using extensive pre-processing and other geometric models.

For each superquadric part-primitive obtained from volumetric decomposition stage, we should calculate surface normals and curvatures to acquire surface information for the next recognition level. Therefore, we could use both the top-down and the bottom-up approaches together to achieve more robust recognition results. We also analyzed volumetric properties of each part-primitive that relates constitutes the model object with superquadric model parameters. The proposed model-based approach has the relationships of volumetric part-primitives as well as the surface information: surface type, junction type between neighboring surface patches. These surface attributes and relationships of volumetric primitives are effectively used in the recognition process. Because we can use those as matching features to discriminate between two different objects which have the same dimensions of superquadric parameters.
Our integrated hybrid method is robust to recognize the identity, position, and orientation of randomly oriented objects. Furthermore, we can reduce the effects of self-occlusion and non-linear shape changes according to the viewpoint.
Contents

Abstract v

1 Introduction 1
1.1 Background ......................................... 2
1.2 Problem Statement .................................... 3
1.3 Related Work .......................................... 4
1.4 Paper Overview ....................................... 6

2 3D Range Image Acquisitions 8
2.1 Range Image ........................................... 8
2.2 Range-finding Techniques Survey .................... 10
   2.2.1 Triangulation Sensors ............................ 11
   2.2.2 Radar Sensors ................................... 16
   2.2.3 Stereo ........................................... 17

3 Shape Recovery of Parts using Superquadrics 20
3.1 Definition of Superellipsoids and Superquadrics .... 20
3.2 Geometric Properties of Superquadrics ............... 25
   3.2.1 Superquadric Surface Normal Vector ............. 25
   3.2.2 Distance between a Point and a Superquadric .... 26
3.3 Superquadratic Deformation ............................ 27
   3.3.1 Tapering Deformation ............................ 28
   3.3.2 Bending Deformation ............................. 29
3.4 Input Data to Superquadric Model Recovery .......... 32
   3.4.1 Range Images ................................... 32
   3.4.2 Stereo .......................................... 33
   3.4.3 Single Intensity Images ........................ 33
3.5 Minimization of an Objective Function ............... 34
3.6 Non-deformed Part Shape Recovery using the Inside-Outsides Superquadric Function ....................... 37
   3.6.1 Superquadrics in General Position ............... 37
   3.6.2 Objective Function ................................ 40
   3.6.3 Parameter Constraints ........................... 42
6 Experiments

7 Conclusions and Discussion

A Range image acquisition system
   A.1 IVP RANGER2500 Laser Range Finder System .......... 219
   A.2 RIEGL LMS-Z210 Laser Mirror Range Scanner ....... 221

B Differential Geometry of 3D surface .......... 229

References .......... 237
List of Tables

5.1 Estimated superquadric parameters. ............................................. 118
6.1 Results of superquadric parameter matching. ................................. 145
6.2 Matching parameters of junction relationships. ......................... 145
6.3 Table of the threshold values for model recovery and the constants for model selection. ........................................... 146
6.4 Surface geometric unary features for the part 0. ........................... 202
6.5 Surface adjacency binary feature for the part 0. ............................. 202
6.6 Surface geometric unary features for the part 1. .......................... 203
6.7 Surface geometric unary features for the part 2. ........................... 204
6.8 Surface geometric unary features for the part 0. ........................... 205
6.9 Surface adjacency binary feature for the part 0. ............................ 205
6.10 Surface geometric unary features for the part 1. .......................... 206
6.11 Surface geometric unary features for the part 2. ........................... 207
6.12 Surface adjacency binary feature for the part 2. ........................... 207
6.13 Surface geometric unary features for the part 0. ........................... 208
6.14 Surface adjacency binary feature for the part 0. ............................ 208
6.15 Surface geometric unary features for the part 1. .......................... 209
6.16 Surface geometric unary features for the part 2. ........................... 210
6.17 Surface adjacency binary feature for the part 2. ........................... 210
6.18 Surface geometric unary features for the part 3. ........................... 211
6.19 Surface adjacency binary feature for the part 3. ........................... 211
6.20 Surface geometric unary features for the part 0. ............................ 212
6.21 Surface adjacency binary feature for the part 0. ............................ 212
6.22 Surface geometric unary features for the part 1. ........................... 213
6.23 Surface geometric unary features for the part 2. ........................... 214
6.24 Surface adjacency binary feature for the part 2. ............................ 214

B.1 Classifying surface types using Gaussian and mean curvature information. ........................................... 233
B.2 Classifying surface types using principal curvature information. .......... 233
List of Figures

2.1 An example of the intensity and the range image of target object:
(a) Intensity image, (b) Range image. .......................... 10
2.2 The law of sines used in triangulation sensor. ............... 12
2.3 The single spot range imaging. ................................ 13
2.4 The sheet-of-light range imaging. .............................. 14
2.5 Acquisition principle of coded light range sensor. .......... 15
2.6 Basic structure of sheet-of-light range imaging system. .... 15
2.7 Laser (left) and camera (right) occlusion. ................. 16
2.8 Pinhole camera model. ....................................... 17
2.9 Stereo sensor geometry. ..................................... 18
2.10 Epipolar geometry of stereo sensor. ........................ 19

3.1 Superquadric shape recovery procedure. .................... 21
3.2 A 3D vector which defines a closed 3D surface can be obtained
by a spherical product of two 2D curves. ......................... 22
3.3 Superellipses for different values of parameter $\varepsilon$. .... 23
3.4 Superellipsoids with different values of exponents $\varepsilon_1$ and $\varepsilon_2$. Size
parameters $a_1$, $a_2$, and $a_3$ are kept constant. Superquadric-
centered coordinate axis $z$ is upward. ............................ 24
3.5 Geometric interpretation of radial Euclidean distance. ...... 27
3.6 Invalid tapering deformation due to $K_x, K_y > 1.0$. ....... 29
3.7 Original superquadric and its bending deformation. ........ 30
3.8 Bending plane and bending angle. ............................ 30
3.9 Example of superquadric deformations with tapering and bend-
ing parameters. .................................................. 32
3.10 A Superquadric in general position. .......................... 38
3.11 Volumetric primitives. ..................................... 46
3.12 Superquadric shape recovery of volumetric primitives. ... 48
3.13 Superquadric model recovery when some data loss in object part
shape exists, (a) Original cylinder image, (b) Cylinder with top
loss, (c) Cylinder with bottom loss, (d) Shape recovery for the
top loss, (e) Shape recovery for the bottom loss. ............ 49
3.14 Superquadric model recovery for a split cylinder due to occlusion, (a) Original cylinder image, (b) Shape recovery for the split cylinder, (c) Trace of the shape parameters, (d) Trace of the size parameters. ................................................. 52

3.15 Deformed superquadric model recovery for a synthetic cone object, (a) Original cone image, (b) Surface normal image, (c) Superquadric model recovery, (d) Trace of the tapering parameters. 54

3.16 Deformed superquadric model recovery for a real bent cylinder object, (a) Original cone image, (b) Surface normal image, (c) Superquadric model recovery, (d) Trace of the bending angle parameter, (e) Trace of the bending curvature parameter. .......... 57

4.1 Comparison of surface and volumetric description of L-shaped object. .................................................. 64

4.2 Greedy algorithm procedure. ................................................. 72

4.3 Comparison between recover-then-select method and recover-and-select method. ................................................. 74

5.1 3D surface geometry showing principal directions. .................. 97

5.2 Block diagram of overall surface description procedure. .......... 100

5.3 Spherical surface. ......................................................... 108

5.4 Cylindrical surface. ....................................................... 109

5.5 Plane-plane binary surface relationship. ............................ 111

5.6 Plane-cylinder binary surface relationship. ......................... 112

5.7 Plane-sphere binary surface relationship. .......................... 113

5.8 Cylinder-cylinder binary surface relationship. .................... 114

5.9 Cylinder-sphere binary surface relationship. ....................... 115

5.10 Sphero-sphere binary surface relationship. ......................... 115

5.11 Hierarchical model construction. ...................................... 117

5.12 Comparison between the volumetric and the surface description. 119

5.13 Matching features of the hierarchical model. ...................... 122

5.14 The outline of recognition scheme. .................................. 127

5.15 Surface segmentation and volumetric decomposition. ................ 129

5.16 Construction of model object's PBSM. ................................. 130

5.17 Part segmentation and PBSM of scene object. ...................... 131

5.18 Definition of the junction distance (a) and the junction angle (b). 132

5.19 The proposed matching procedure. .................................. 133

6.1 The model objects used in the experiments. ......................... 135

6.2 Experiment for the noise reduction using median filtering. ....... 138

6.3 Experiment for real range image captured from RANGER system. 139
6.4 The PBSM of the model objects.  ........................................ 140
6.5 The scene objects to be matched with the model object 1.  ........ 141
6.6 PBSM of the scene object 1 (SO1) to be matched with the model object 1 (MO1). .................................................. 142
6.7 PBSM of the scene object 2 (SO2) to be matched with the model object 1 (MO1). .................................................. 143
6.8 PBSM of the scene object 3 (SO3) to be matched with the model object 1 (MO1). .................................................. 144
6.9 Complex synthetic range images. ....................................... 147
6.10 Real range images. ......................................................... 147
6.11 The intensity and the real range images acquired from the RANGER2500 laser range finder system. ............................ 148
6.12 Volumetric decomposition for a complex synthetic COMP1 model using basic recover-and-select algorithm without post-processing. 154
6.13 Volumetric description for a complex synthetic COMP1 model using basic recover-and-select algorithm without post-processing. 156
6.14 Volumetric description for a complex synthetic COMP1 model after region improvement post-processing. .................... 158
6.15 Volumetric decomposition for a complex synthetic COMP2 model using basic recover-and-select algorithm without post-processing. 160
6.16 Volumetric description for a complex synthetic COMP2 model using basic recover-and-select algorithm without post-processing. 162
6.17 Volumetric description for a complex synthetic COMP2 model after region improvement post-processing. .................... 164
6.18 Volumetric decomposition for a complex synthetic T-PIPE model using basic recover-and-select algorithm without post-processing. 166
6.19 Volumetric description for a complex synthetic T-PIPE model using basic recover-and-select algorithm without post-processing. 167
6.20 Volumetric description for a complex synthetic T-PIPE model after region improvement post-processing. .................... 168
6.21 Volumetric decomposition for a complex synthetic HAMMER2 model using basic recover-and-select algorithm without post-processing. 170
6.22 Volumetric description for a complex synthetic HAMMER2 model using basic recover-and-select algorithm without post-processing. 172
6.23 Volumetric description for a complex synthetic HAMMER2 model after region improvement post-processing. .................... 173
6.24 Volumetric decomposition for a complex synthetic GEARBOX model using basic recover-and-select algorithm without post-processing. 175
6.25 Volumetric description for a complex synthetic GEARBOX model using basic recover-and-select algorithm without post-processing. 177
6.26 Volumetric description for a complex synthetic GEARBOX model after region improvement post-processing.

6.27 Volumetric decomposition for a complex real L-SHAPE model using basic recover-and-select algorithm without post-processing.

6.28 Volumetric description for a complex real L-SHAPE model using basic recover-and-select algorithm without post-processing.

6.29 Volumetric description for a complex real L-SHAPE model after region improvement post-processing.

6.30 Volumetric decomposition for a complex real T-PIPE model using basic recover-and-select algorithm without post-processing.

6.31 Volumetric description for a complex real T-PIPE model using basic recover-and-select algorithm without post-processing.

6.32 Volumetric description for a complex real model after region improvement post-processing.

6.33 Volumetric decomposition for a complex real GEARBOX model using basic recover-and-select algorithm without post-processing.

6.34 Volumetric description for a complex real GEARBOX model using basic recover-and-select algorithm without post-processing.

6.35 Volumetric description for a complex real GEARBOX model after region improvement post-processing.

6.36 Volumetric decomposition for a complex real DATA1 model using recover-and-select algorithm with post-processing.


6.40 Surface description for the decomposed part 0 of a complex synthetic COMPl model.

6.41 Surface description for the decomposed part 1 of a complex synthetic COMPl model.

6.42 Surface description for the decomposed part 2 of a complex synthetic COMPl model.

6.43 Surface description for the decomposed part 0 of a complex synthetic COMPl model.

6.44 Surface description for the decomposed part 1 of a complex synthetic COMPl model.

6.45 Surface description for the decomposed part 2 of a complex synthetic COMPl model.
6.46 Surface description for the decomposed part 0 of a complex real
DATA1 model. ........................................ 208
6.47 Surface description for the decomposed part 1 of a complex real
DATA1 model. ........................................ 209
6.48 Surface description for the decomposed part 2 of a complex real
DATA1 model. ........................................ 210
6.49 Surface description for the decomposed part 3 of a complex real
DATA1 model. ........................................ 211
6.50 Surface description for the decomposed part 0 of a complex real
DATA2 model. ........................................ 212
6.51 Surface description for the decomposed part 1 of a complex real
DATA1 model. ........................................ 213
6.52 Surface description for the decomposed part 2 of a complex real
DATA2 model. ........................................ 214

A.1 The RANGER2500 laser range camera system. ............... 220
A.2 The target object projected with laser slit beam. .......... 220
A.3 RANGER2500 laser range finder using sheet-of-light. .... 222
A.4 Snapshot of the sensor area. ................................ 223
A.5 The example of 3D shape profile for hand range image. .. 223
A.6 The surface plot of the hand range image. .................. 224
A.7 The RIEGL LMS-Z210 laser mirror range scanner. ....... 225
A.8 Measurement principle of the pulsed range finder. ....... 225
A.9 The principle of operation. ................................ 226
A.10 Example of image acquisition of RIEGL laser mirror range scanner. 227
A.11 Experiment for the 3D reconstruction with texture. ...... 228

B.1 Local coordinate frame of a surface point. .................. 230
Chapter 1

Introduction

Three-dimensional (3D) object recognition using range images is very important in many industrial applications of machine vision, such as robotics and computer vision [9, 27, 87, 107]. Obviously, complete model description of 3D object and an elaborate matching algorithm are critical steps in building an efficient 3D object recognition system. In general, human visual system perceives 3D objects intuitively by decomposing entire object into its subset parts based on the theory of Recognition By Components (RBC) [14, 15]. This is achieved during process of image segmentation that partitions the 3D data set into data subsets corresponding to individual parts. The parts subsets are then usually represented by some sort of a parametric models to achieve sufficient data reduction. In this point of view, one of most appropriate part-level representations is the superquadric model [4, 106]. The advantage of superquadric primitives is that they can describe a large variety of shapes with a finite number of parameters including translation, rotation and global deformation.
1. Introduction

1.1 Background

In order to recognize an object from a range image, a computer vision system must convert the image data into a symbolic description or model of the object shape which is somehow consistent with the actual object shape. Especially, part-level shape description is important for tasks involving spatial reasoning, object manipulation, and structural object recognition. The primary reason for this is that part-based descriptions help to bridge gap between image features and symbolic descriptions of objects. Therefore they can be more robustly and efficiently indexed into a database than some other features, such as edges or surfaces. Many objects consist of parts or component which can be distinguished perceptually, geometrically or functionally from each other. Many researchers have attempted to define parts as perceived by human vision in mathematical terms. Koenderink and van Doorn defined part intersections as "parabolic lines on the surfaces of objects." Hoffman and Richards refined the definition by using instead the "negative minima of principal curvature" which works even in case of figure/ground reversal when the perceived part structure completely changes. Such definitions of parts which are articulated in terms of analytical geometry are difficult to apply on real images because of noise and non-smooth surfaces. Computation of second order partial derivatives, which is necessary in the process of part boundary detection, requires excessive smoothing of edges and other sharp discontinuities in images. Among various existing part-level models, generalized cylinders and superquadrics are the most widely used for 3D object recognition in computer vision. The first parts representation of generalized cylinders was suggested by Binford. Unfortunately, the recovery of this type of representation seems to require elaborate line grouping and reasoning, which is difficult and largely unsolved. Moreover, because
1. Introduction

such descriptions are often not unique, it is unclear how they aid in object recognition. On the other hand, superquadrics, are more widely used by the following reasons: (i) They can express wide variety of 3D shapes, like round edges and corners, as well as standard geometric solids, using a small number of model parameters, (ii) robust methods exist for reconstruction from range images, and (iii) superquadrics can be enhanced by adding global and local deformations. And one of the most important features of superquadric models is their interchangeable implicit and explicit defining function. The explicit form is convenient for rendering, while the implicit equation is especially suited for model recovery from images and for testing of intersection.

1.2 Problem Statement

In this paper, we present a novel method to construct geometric models of objects composed of different parts. We assume that the parts can be modeled by superquadrics, which are composed together in a solid constructive modeling manner using union operation. The superquadric part-primitives are defined as kinds of volume primitives, which represent basic shapes of geometric solid objects (spheres, cubes, and cylinders, etc.). We assumed that our target objects are composed of convex volumetric primitives without any holes, and that the observer can freely choose its viewing position. And it can easily be recovered by using a finite set of superquadric model parameters for each parts using by recover-and-select paradigm. In spite of many advantages above, only the use of superquadrics has difficulties in recognizing 3D object. This paper summarizes these reasons as following. At first, superquadrics cannot provide a unique representation of 3D object, so it is too difficult to build model base constantly. Secondly, volumetric representation has no surface information: surface type
1. Introduction

and relationship, which is widely used for a precise recognition. In this manner, there is the trade-off between surface and volume description for the purpose of the effective 3D object representation using hierarchical surface and volume description from single range image.

Therefore, to overcome these difficulties, we combine this volumetric superquadric description and surface description, which is used as features for the recognition. For the purpose of recognition through the matching between the input object and the model object, it is necessary for the input object to be decomposed into several part-primitives.

1.3 Related Work

This paper relates to research in range image segmentation, modeling, and recognition. Most of conventional researches using superquadrics are concentrated on 3D shape recovery and segmentation of range images through various kinds of parameter estimation. Pentland[16] and also Solina and Bajcsy[1] used superquadrics for the recovery of compact volumetric models from 3D range points. Model recovery was implemented by a least squares minimization of a cost function whose value was varied according to distance of the data points from the model's surface and the global parameters of the model. However, the class of objects that could be recovered accurately are limited to single-part convex shapes. Ferri, Lagarde and White[9] proposed a bottom-up strategy to deal with the problem of multi-part objects by segmenting the range data based on curvature consistency. Each part of the segmented range image is then modeled with superquadrics. But the experimental results are limited to cases where each segmented surface patch corresponds to a single volumetric model. Also this approach is inadequate to represent the segmented parts uniquely. This
1. Introduction
can lead to ambiguity in the recognition process. Similar bottom-up approach, but more refined and adjusted for use of superquadrics as a final representation, is proposed by the work of Gupta and Bajcsy[2]. They used biquadric surface patches within recover-and-select paradigm to pre-segment a range image. To solve a problem of multiple surface patches belonging to a single superquadric model they proposed a strategy of grouping surfaces along convex discontinuities of the surface normal. Although such an approach enlarges the class of images that can be segmented with volumetric models recovered from surface segments, the approach cannot deal with an L-shaped object, since the surface shape of the object would require the algorithm to split up a surface instead of grouping surfaces together. Metaxas and Terzopoulos[18] developed dynamic models based on global and local deformation properties from superquadric ellipsoids and membrane splines. However, the parameters for local deformation are only used in the reconstruction process and only the gross shape information is used for recognition. We used the recover-and-select paradigm to segment and decompose the input object because it is not based on exhaustive search through superquadric parameter space and superquadrics can be recovered directly from range data without considering any other geometric models. Note that this fact is contrary to the common belief that the recovery of volumetric models is possible only after the data has been pre-segmented using extensive pre-processing and other geometric models. At this stage, superquadrics not only play the role of ultimate modeling primitives, but also used to derive the segmentation process. For each superquadric part-primitive obtained from volumetric decomposition stage, we should calculate surface normal and curvatures to extract surface information for the next recognition level. Therefore, we used both the top-down and the bottom-up approaches together to achieve more robust recognition results. We also analyze volumetric proper-
ties of each part-primitive that relates constitutes the model object with superquadric model parameters. The proposed model-based approach has the relationships of volumetric part-primitives as well as the surface information: surface type, junction type between neighboring surface patches. These surface attributes and relationships of volumetric parts are effectively used in the recognition process. Because we can use those as matching features to discriminate between two different objects which have the same dimensions of superquadric parameters. Our integrated hybrid method is robust to recognize the identity, position, and orientation of randomly oriented objects. Furthermore, we can reduce the effects of self-occlusion and non-linear shape changes according to the viewpoint.

1.4 Paper Overview

This paper consists of seven chapters. The second chapter introduces the 3D range image acquisition and pre-processing through unwanted noise removal. The third chapter describes the superquadrics and their geometric properties including surface normals and Euclidean distance, and also their deformations such as bending and tapering. It is also mentioned about the recovery of individual parts using the inside-outside superquadric function and shows examples of non-deformed and deformed objects from pre-segmented range images based on least-squares minimization. In Chapter 4, a summary of superquadric-based segmentation methods is given and the volumetric description using recover-and-select paradigm is also described to show how the recovery of superquadrics can be tightly integrated with volumetric description to achieve part decomposition. The fifth chapter presents the surface segmentation and the description for the planar and the curved surfaces respectively by using surface equation
and surface normal to extract the geometrical parameters and introduces the hierarchical modelbase construction by combining surface and volumetric descriptions and derives the hierarchical part-based matching procedure between model object and input scene object. The sixth chapter shows extensive experimental results of the proposed hierarchical part-based superquadric model and surface description on the various of synthetic and real range images. Finally the seventh chapter summarizes this paper with conclusion and further research.
Chapter 2

3D Range Image Acquisitions

2.1 Range Image

The model-based vision system described in this dissertation uses range images as its input. Most of computer vision research performed during the last 40 years has concentrated on using digitized gray-scale intensity images as sensor data. It has proven to be extraordinarily difficult to program computers to understand and describe these images in a general purpose way. One important problem is that digitized intensity images are rectangular arrays of numbers which indicate the brightness at individual points on a regularly spaced rectangular grid and contain no explicit information that is relevant to depth perception. Yet human beings are able to correctly infer depth relationships quickly and easily among intensity image regions whereas automatic inference of such depth relationships has proven to be remarkably complex. In recent years digitized range data has become available from both active and passive sensors, and the quality of this data has been steadily improving. Range data is usually produced in the form of a rectangular array of numbers, referred to as a depth map or range
2. 3D Range Image Acquisitions

image, where the numbers quantify the distance from the sensor plane to the surfaces within the field of view along the rays emanating from a regularly spaced rectangular grid. Not only are depth relationships between depth map regions approximate the three dimensional shape of the corresponding object surfaces in the field of view. Therefore, the process of recognizing objects by their shape should be less difficult in depth maps than in intensity images due to the explicitness of the information. For example, since correct depth map information depends only on geometry and is independent of illumination and reflectivity, intensity image problems with shadows and surface markings do not occur. Nevertheless, it seems that existing vision techniques have influenced many investigators and this has led to restricted approaches to processing range data.

Range imaging sensors collect 3D coordinate data from visible surface of objects in the scene. The output of a range sensor is termed a range image and also known as depth maps. For example, the image from a general camera depicts the intensity distribution of the viewed scene as in the left image in Figure 2.1(a). A range finder on the other hand, displays the distance from the camera to the objects of the scene as in the right image in Figure 2.1(b), where dark areas are far away from the range camera, and light areas close. Range image acquisition can be defined as the process of determining the distance (or depth) from a given observation point in a known reference coordinate system to all surface points of consideration in a scene. In range images, image pixels explicitly represent scene surface geometry in a sampled form. Since three-dimensional information is directly available in a range image, the problem of recognizing and localizing three-dimensional objects in a range image is to a great extent simplified. The technology of range finders is not new and has been used for many years in a wide variety of computer vision application areas such
2. 3D Range Image Acquisitions

(a)                   (b)

Figure 2.1: An example of the intensity and the range image of target object: (a) Intensity image, (b) Range image.

as automated surface inspection, robot bin-picking, autonomous navigation, 3D object modeling and recognition, and data fusion.

2.2 Range-finding Techniques Survey

In order to use sensor data to understand the range image formation process. At each pixel in a range image, the digitized depth value encodes information about (1) surface geometry and viewing geometry in terms of the distance from the sensor to the object surface, and (2) the rangefinder characteristics which include spatial resolution, range resolution, noise parameters, and other rangefinder parameters which depend on the type of rangefinder used. One important difference between the intensity image formation and the range image formation is that scene illumination and surface reflectance are not directly encoded in range values. Moreover, rangefinders directly produce the depth information.
2. 3D Range Image Acquisitions

In this section, we briefly describe most of the techniques currently being used to acquire range image from a scene. Range imaging sensors also named as rangefinders are classified as either active or passive. Active sensors project typically optical or sonic energy on the objects in the scene and detect the reflected portion of the energy. Ultrasound and radio wave techniques can be used for range determination, but do not currently possess high enough resolution for most range imaging purposes. Laser can be used as pulse-mode or modulated continuous-wave range sensors. A pulse-mode time-of-flight laser rangefinder determines distance by measuring the elapsed time between pulse transmission and signal reception and therefore requires signal processing electronics with 70 picosecond time resolution to obtain depth resolution of 1 centimeter. Passive sensors, on the other hand, use ambient environmental conditions during the process of data acquisition. The most common passive method is stereo vision [3, 43, 42, 46, 45, 118], where at least two images from different positions are used in a triangulation scheme. Sensors can also be classified as contact or close proximity of the object being sensed, which is not the case with non-contact sensors.

2.2.1 Triangulation Sensors

The Principle of triangulation sensing for the measurement of depth is one of the earliest known ranging techniques. The basis of triangulation sensors is the law of sines as shown in Figure 2.2 and expressed by the equation.

$$\frac{\sin(\angle A)}{BC} = \frac{\sin(\angle B)}{AC} = \frac{\sin(\angle C)}{AB}. \quad (2.1)$$

If the length of the baseline of triangle is known, and if the two included angles and are known, ten the remaining angle and the lengths of sides and can be
computed as shown in Figure 2.2. A single monochrome camera and a light projector are sufficient hardware to construct a triangulation sensor based on structured light [63]. The most common active triangulation methods include illumination with a single spot, a sheet-of-light and coded light. The most common non-triangulation method is time-of-flight (radar), where the time for the emitted light pulse to return from the scene is measured. The high speed of electromagnetic waves makes time-of-flight methods difficult to use for high accuracy range imaging since small differences in range have to be resolved by extremely fine discrimination in time. In single spot range imaging, a single light ray is scanned over the scene and one range datum (rangel) is acquired for each sensor integration and position of the light as shown in Figure 2.3. Thus, in order to obtain an $M \times N$ image, two-axis scanning of $M \times N$ measurements and sensor integrations are needed. In sheet-of-light range imaging a sheet (or stripe) of light is scanned over the scene and one row with $M$ rangels is acquired at each light position and sensor integration as shown in Figure 2.4. In this case only $N$ measurements and integrations are needed for an $M \times N$ image. Finally, in a coded light range, camera the scene is illuminated with several coded patterns. The patterns can for instance be phase coded or gray coded. For instance, with the gray coded approach log$R$ patterns are required to obtain
a resolution of $R$ levels. Therefore $\log R$ integrations are needed to form one $M \times N$ image as depicted in Figure 2.5. It would seem that the spatially coded light method has speed advantages over the other active triangulation methods, since fewer sensor integrations are needed. However, one problem seems to be how to make a high intensity projector that can switch between patterns as fast as the sensor can integrate images. The single spot technique requires advanced mechanics to allow the spot to reach the whole scene. At least two synchronous scanning mirrors are required. The advantage is that a relatively simple linear sensor such as the Position Sensitive Device type can be used. But, the use of only one single sensing element often reduces the acquisition speed.

In the case of sheet-of-light systems, the projection of the light can be made
Figure 2.4: The sheet-of-light range imaging.

with one single scanning mirror which is considerably simpler than the projector design for spatially coded light, or the two mirror arrangement for single spot. Actually, in most sheet-of-light systems the sheet-of-light is not swept at all. Instead the apparatus itself or the scene is moving by using a conveyer belt as displayed in Figure 2.6. Thus, the camera system itself has no moving parts. All triangulation sensor systems suffer from the problem of occlusion such as shown in Figure 2.7. Either the laser light does not reach the area seen by the camera (laser occlusion) or the camera does not see the area reached by the laser (camera occlusion). In both cases the maximum reflection peak found in the sensor row data is the result of noise and/or ambient light.
Figure 2.5: Acquisition principle of coded light range sensor.

Figure 2.6: Basic structure of sheet-of-light range imaging system.
2. 3D Range Image Acquisitions

![Diagram showing laser and sensor with occluded area]

Figure 2.7: Laser (left) and camera (right) occlusion.

2.2.2 Radar Sensors

Radar-based range sensors are often labeled active devices because the sensor illuminate the object with a light source, and range is calculated based on the reflected energy. Most range sensors of this type are named as laser range finders because they employ lasers as the illumination source. Perhaps the most intuitive use of radar techniques for range sensing is simple time-of-flight measurement. Since electromagnetic radiation travels at a constant speed, the distance to a point on an object’s surface can be determined by directing a pulse if laser light to that point and measuring the time interval needed for the reflected pulse to arrive back at the sensor. This sensor is quite feasible for long-distance sensing, such as in laser radar range finders for target acquisition, where the sensor is located several hundred meters from the objects to be sensed. Such sensors often produce both range and intensity images, where the intensity is derived from the amplitude of the returning laser light pulse.
2.3D Range Image Acquisitions

2.2.3 Stereo

For the pinhole camera model, also known as the perspective camera model, light rays from a point in the real world travel in a straight line through the optical center of the camera and are projected on the view plane as shown in Figure 2.8. This is a projection from three-dimensional space onto two-dimensional space through a single point. For two cameras displaced by a rotation and translation, the images are related by another projective transform, known as the Fundamental Matrix. In stereo applications, the fundamental matrix can be derived in two ways, either by knowing the calibration and displacement of the cameras, or by finding eight or more matching points. All rays imaged by the first camera originate from the optical center of the first camera, and the same can be said for the second camera as displayed in Figure 2.9. The line which connects these two optical centres is known as the baseline of the

![Figure 2.8: Pinhole camera model.](image-url)
Figure 2.9: Stereo sensor geometry.

cameras, and the two points at which the baseline intersects the view planes are known as the epipoles. For some point $X$ in space, the plane defined by
the epipoles and the point $X$ is the epipolar plane for that point. Its images
$x$ and $x'$ will lie in the intersection between the image planes and the epipolar
plane, along the lines known as the epipolar lines. As the point moves toward
and away from the cameras, its image moves along the epipolar line. In fact, all
points which exist in one epipolar line will also (subject to occlusion) exist in
the other epipolar line. For cameras where the Fundamental Matrix has been
derived, this reduces the search for corresponding points to a one dimensional
search.

The chief advantage of stereo sensor is passive. The main disadvantage
is that it apparently cannot produce dense range measurements over uniform
regions due to lack of features or presence of too many features for correpon-
Figure 2.10: Epipolar geometry of stereo sensor.

dence. The cost of two cameras is low, but hardware needed to produce the correspondences adds significantly to the cost of a system.
Chapter 3

Shape Recovery of Parts using Superquadrics

In this chapter we define superquadrics after we outline a brief history of their development. Besides giving basic superquadric equations we derive also some their useful geometric properties of superquadrics. This chapter also gives a thorough review of the superquadric recovery method to be used in our volumetric description stage for the part model shape from a set of $n$ range data points, which was proposed by Solina and Bajcsy in 1990. The part shape recovery method is based on least squares minimization of a fitting function derived from the inside-outside superquadric function. This overall recovery process is shown in Figure 3.1.

3.1 Definition of Superellipsoids and Superquadrics

A 3D surface can be obtained by a spherical product of two 2D curves [4]. A unit sphere, for example, is produced when a half circle in a plane orthogonal
3. Shape Recovery of Parts using Superquadrics

Figure 3.1: Superquadric shape recovery procedure.

to the \((x, y)\) plane in Figure 3.2.

\[
m(\eta) = \begin{bmatrix} \cos \eta \\ \sin \eta \end{bmatrix}, \quad -\pi/2 \leq \eta \leq \pi/2 \tag{3.1}\]

is crossed with the full circle in \((x, y)\) plane.

\[
h(\omega) = \begin{bmatrix} \cos \omega \\ \sin \omega \end{bmatrix}, \quad -\pi \leq \omega \leq \pi \tag{3.2}\]

\[
r(\eta, \omega) = m(\eta) \otimes h(\omega) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \eta \cos \omega \\ \cos \eta \sin \omega \\ \sin \eta \end{bmatrix}, \quad -\pi/2 \leq \eta \leq \pi/2, -\pi \leq \omega \leq \pi \tag{3.3}\]
3. Shape Recovery of Parts Using Superquadrics

Figure 3.2: A 3D vector which defines a closed 3D surface can be obtained by a spherical product of two 2D curves.

In two dimensions, analogous to a circle, the superellipse centered at the origin is defined in parametric form

$$\left(\frac{x}{a}\right)^{\frac{2}{\epsilon}} + \left(\frac{y}{b}\right)^{\frac{2}{\epsilon}} = 1$$

(3.4)

can be written as

$$s(\phi) = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \cos^{\frac{2}{\epsilon}} \phi \\ b \sin^{\frac{2}{\epsilon}} \phi \end{bmatrix}, \quad -\pi \leq \epsilon \leq \pi.$$  (3.5)

Figure 3.3 shows how the parameters $a$ and $b$ determine superellipse overall size, while parameter $\epsilon$ determines its shape. All these three parameters are positive real numbers with 0 included. As the parameter $\epsilon$ changes from 0 toward 1, the shape changes from a rectangle to an ordinary ellipse, and at value 2 reaches a shape of a deltoid. Superellipse with $\epsilon > 2$ are concave and tend toward cross...
like shapes as \( \varepsilon \) increases towards \( +\infty \). Superellipsoid explicit equation can therefore be obtained by a spherical product of a pair of such ellipses

\[
\mathbf{r}(\eta, \omega) = s_1(\eta) \otimes s_2(\omega) = \begin{bmatrix} \cos \varepsilon_1 \eta \\ \sin \varepsilon_1 \eta \end{bmatrix} \otimes \begin{bmatrix} a_1 \cos \varepsilon_2 \omega \\ a_2 \sin \varepsilon_2 \omega \end{bmatrix} = \begin{bmatrix} a_1 \cos \varepsilon_1 \eta \cos \varepsilon_2 \omega \\ a_2 \cos \varepsilon_1 \eta \sin \varepsilon_2 \omega \\ a_3 \sin \varepsilon_0 \eta \end{bmatrix} .
\]

Parameters \( a_1, a_2 \) and \( a_3 \) are scaling factors along the three coordinate axes. \( \varepsilon_1 \) and \( \varepsilon_2 \) come from the exponents of the two original ellipses. \( \varepsilon_2 \) determines the shape of ellipsoid cross section parallel to the \( xz \) plane, while \( \varepsilon_1 \) determines the shape of ellipsoid cross section in a plane perpendicular to the \( xy \) plane and containing \( z \) axis as shown in Figure 3.4. Superquadrics are a family of shapes that includes not only superellipsoids, but also superhyperboloids as well as supertoroids. In computer vision literature, it is common to refer to superellipsoids by the more generic term of superquadrics. In this paper, we also use the term superquadrics as a synonym for superellipsoids. An alternative,
Figure 3.4: Superellipsoids with different values of exponents $\varepsilon_1$ and $\varepsilon_2$. Size parameters $a_1$, $a_2$, and $a_3$ are kept constant. Superquadric-centered coordinate axis $z$ is upward.

An implicit superquadric equation can be derived from the above explicit equation using the equality $\cos^2 \alpha + \sin^2 \alpha = 1$ as follows,

$$\left( \left( \frac{x}{a_1} \right)^{\varepsilon_2} + \left( \frac{y}{a_2} \right)^{\varepsilon_2} \right)^{\frac{\varepsilon_1}{\varepsilon_2}} + \left( \frac{z}{a_3} \right)^{\varepsilon_1} = 1.$$  \hspace{1cm} (3.7)

All points with coordinates $(x, y, z)$ that correspond to the above equation lie by definition on the surface of the superquadrics. The function

$$F(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\varepsilon_2} + \left( \frac{y}{a_2} \right)^{\varepsilon_2} \right)^{\frac{\varepsilon_1}{\varepsilon_2}} + \left( \frac{z}{a_3} \right)^{\varepsilon_1}$$  \hspace{1cm} (3.8)
3. Shape Recovery of Parts using Superquadrics

is also called the inside-outside function because it provides a simple test whether a given point lies inside or outside the superquadric. If \( F < 1 \), the given point \((x, y, z)\) is inside the superquadric, if \( F = 1 \), the corresponding point lies on the surface of the superquadric, and if \( F > 1 \), the point lies outside the superquadric.

### 3.2 Geometric Properties of Superquadrics

In this section, derivations of superquadric normal vector and radial Euclidean distance between a point and a superquadric are given.

#### 3.2.1 Superquadric Surface Normal Vector

Normal vector at a point \( \mathbf{r}(\eta, \omega) \) on the superquadric surface is defined by a cross product of the tangent vectors along the coordinate curves

\[
\mathbf{n}(\eta, \omega) = \mathbf{r}_\eta(\eta, \omega) \times \mathbf{r}_\omega(\eta, \omega)
\]

\[
= \begin{bmatrix}
-a_1 \varepsilon_1 \sin \eta \cos^{\varepsilon_1-1} \eta \cos^{\varepsilon_2} \omega \\
-a_2 \varepsilon_1 \sin \eta \cos^{\varepsilon_1-1} \eta \sin^{\varepsilon_2} \omega \\
a_3 \varepsilon_1 \sin^{\varepsilon_1-1} \eta \cos \eta
\end{bmatrix} \times \begin{bmatrix}
-a_1 \varepsilon_2 \cos^{\varepsilon_1} \eta \sin \omega \cos^{\varepsilon_2-1} \omega \\
a_2 \varepsilon_2 \cos^{\varepsilon_1} \eta \cos \omega \sin^{\varepsilon_2-1} \omega \\
0
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-a_2 a_3 \varepsilon_1 \varepsilon_2 \sin^{\varepsilon_1-1} \eta \cos^{\varepsilon_1} \eta \cos \omega \sin^{\varepsilon_2-1} \omega \\
-a_1 a_3 \varepsilon_1 \varepsilon_2 \sin^{\varepsilon_1-1} \eta \cos^{\varepsilon_1} \eta \sin \omega \cos^{\varepsilon_2-1} \omega \\
-a_1 a_2 \varepsilon_1 \varepsilon_2 \sin \eta \cos^{\varepsilon_1} \eta \sin^{\varepsilon_2-1} \omega \cos^{\varepsilon_2-1} \omega
\end{bmatrix}
\]  \( (3.9) \)

The above equation can be simplified defining following common term

\[
f(\eta, \omega) = -a_1 a_2 a_3 \varepsilon_1 \varepsilon_2 \sin^{\varepsilon_1} \eta \cos^{\varepsilon_2-1} \eta \sin^{\varepsilon_2-1} \omega \cos^{\varepsilon_2-1} \omega,
\]  \( (3.10) \)
so that the normal vector can be written as
\[
\mathbf{n}(\eta, \omega) = f(\eta, \omega) \begin{bmatrix}
\frac{1}{a_1} \cos^{2-\epsilon_1} \eta \cos^{2-\epsilon_2} \omega \\
\frac{1}{a_2} \cos^{2-\epsilon_1} \eta \cos^{2-\epsilon_2} \omega \\
\frac{1}{a_3} \sin^{2-\epsilon_1} \eta
\end{bmatrix}.
\] (3.11)

The scalar function \(f(\eta, \omega)\) term can be dropped out if we only need the surface normal direction. By doing this we actually get a dual superquadric to the original superquadric \(r(\eta, \omega)\)
\[
\mathbf{n}_d(\eta, \omega) = \begin{bmatrix}
\frac{1}{a_1} \cos^{2-\epsilon_1} \eta \cos^{2-\epsilon_2} \omega \\
\frac{1}{a_2} \cos^{2-\epsilon_1} \eta \cos^{2-\epsilon_2} \omega \\
\frac{1}{a_3} \sin^{2-\epsilon_1} \eta
\end{bmatrix}.
\] (3.12)

### 3.2.2 Distance between a Point and a Superquadric

The Euclidean distance is defined as a distance between a point and a superquadric along a line through the point and the center of a superquadric as shown in Figure 3.5. For a point defined by a vector \(\mathbf{r}_0 = (x_0, y_0, z_0)\) in the canonical coordinate system of a superquadric, we are looking for a scalar \(\beta\) that scales the vector, so that the tip of the scaled vector \(\mathbf{r}_s = \beta \mathbf{r}_0\) lies on the surface of superquadratics. Thus for the scaled vector \(\mathbf{r}_s\) the following equation holds
\[
F(\beta x_0, \beta y_0, \beta z_0) = \left( \frac{\beta x_0}{a_1} \right)^{\frac{2}{\epsilon_1}} + \left( \frac{\beta y_0}{a_2} \right)^{\frac{2}{\epsilon_2}} + \left( \frac{\beta z_0}{a_3} \right)^{\frac{2}{\epsilon_3}} = 1. \tag{3.13}
\]

From this equation it follows directly
\[
F(x_0, y_0, z_0) = \beta^{-\frac{2}{\epsilon_1}}. \tag{3.14}
\]
Thus the radial Euclidean distance is

\[ d = |r_0 - r_s| = |r_0 - \beta r_0| = |r_0| \left| 1 - F^{-\frac{1}{2}}(x_0, y_0, z_0) \right| \]
\[ = |r_0| \left| F^{\frac{1}{2}}(x_0, y_0, z_0) - 1 \right|. \]  

(3.15)

So for any point \( T \) in space, with given coordinates \((x_0, y_0, z_0)\), we can determine its position relative to the superquadric by simply calculating the value of the \( F(x_0, y_0, z_0) \). If \( F(x_0, y_0, z_0) = 1 \), that is \( \beta = 1 \), then the point \( T \) belongs to the surface of the superquadrics. If \( F(x_0, y_0, z_0) > 1 \), that is \( \beta > 1 \), then the point \( T \) is outside of the superquadrics, and if \( F(x_0, y_0, z_0) < 1 \), that is \( \beta < 1 \), then the point \( T \) is inside.

### 3.3 Superquadratic Deformation

By using non-deformed superquadric models, it is difficult to describe a wide variety of 3D object shapes existed in real world. So by using additional deforma-
tion parameters such as tapering, bending, twisting, cavity, etc., superquadric modeling capabilities are enlarged. If a coordinate $x$ are the points $(x, y, z)$ on the surface of non-deformed superquadrics, and $X$ are the corresponding points $(X, Y, Z)$ after deformation in the object centered coordinate system, we can derive the following equation $X = D(x)$ indicating relationship between $x$ and $X$, where $D$ is shape deformation function. Any translation or rotation is performed after the deformation.

3.3.1 Tapering Deformation

Tapering deformation is to elongate the object shape along the $z$ axis. In this deformation, we assume the following: (a) tapering deformation performed along the $z$ axis; (b) the tapering rate is linear with respect to $z$. By using this tapering, an object can have a wedge-typed shape. Based on these assumptions, tapering deformation is given by

\[
\begin{align*}
X &= f_x(z)x = \left( \frac{K_x}{a_3}z + 1 \right)x \\
Y &= f_y(z)y = \left( \frac{K_y}{a_3}z + 1 \right)y \\
Z &= z
\end{align*}
\]

(3.16)

where $X$ and $Y$ are the transformed coordinates of the primitives after tapering is applied to the coordinates $x$ and $y$.

To be able to recover the deformation parameters, the original surface vector components $x, y, z$ must be expressed in terms of the deformation parameters $f_x, f_y$ and coordinates of points on the surface of a deformed superquadric.
The inverse transformation is given by

\[
\begin{align*}
    x &= \frac{X}{f_x(z)} \\
    y &= \frac{Y}{f_y(z)} \\
    z &= Z
\end{align*}
\] (3.17)

Tapering parameters \( K_x \) and \( K_y \) in the \( x \) and \( y \) coordinates have constraints as \( 0.0 \leq K_x, K_y \leq 1.0 \) to avoid invalid tapering. In the case of \( K_x, K_y > 1.0 \), invalid tapering deformation can be occurred like Figure 3.6. When this expression for \( x, y, z \) are inserted into Eq. (3.31) we get the inside-outside function for a tapered superquadric in general position

\[
F = F(X, Y, Z; \lambda_1, \ldots, \lambda_{11}, K_x, K_y).
\] (3.18)

### 3.3.2 Bending Deformation

The bending operation transforms the \( z \) axis of the shaded original superquadric into a circular section as shown in Figure 3.7. The curvature of the circular section is defined by \( \kappa \). The length of the circular superquadric spine remains
the same as the previous straight spine along the z axis. The bending plane is

![Diagram of superquadric and bending deformation]

Figure 3.7: Original superquadric and its bending deformation.

rotated around the coordinate axis z, and its direction is defined by angle \( \alpha \) as shown in Figure 3.8. The bending deformation is performed first by projecting

![Diagram of bending plane and angle]

Figure 3.8: Bending plane and bending angle.

the \( x \) and \( y \) components of all points on the bending plane, performing the bending deformation in that plane, and then projecting the points back to the
original plane. The projection of a point \((x, y)\) on the bending plane is

\[
r = \sqrt{x^2 + y^2} \cos(\alpha - \beta),
\]

(3.19)

where

\[
\beta = \arctan \frac{y}{x}.
\]

(3.20)

Bending transform \(r\) into

\[
R = \frac{1}{\kappa} - \left(\frac{1}{\kappa} - r\right) \cos(\gamma),
\]

(3.21)

where \(\gamma\) is the bending angle, computed from the curvature parameter \(\kappa\),

\[
\gamma = \frac{z}{\kappa}.
\]

(3.22)

By projecting \(R\) back onto the original plane, which is parallel to the bending plane we get the transformed surface vector

\[
\begin{cases}
X = x + (R - r) \cos(\alpha) \\
Y = y + (R - r) \sin(\alpha) \\
Z = \left(\frac{1}{\kappa} - r\right) \sin(\kappa)
\end{cases}
\]

(3.23)

The inverse transformation is given by

\[
\begin{cases}
X = \frac{X}{I_x(x)} \\
Y = \frac{Y}{I_y(x)} \\
Z = \frac{1}{\kappa} \gamma
\end{cases}
\]

(3.24)

The example of superquadric models after applying above two deformation operations sequentially on the block and the cylinder is shown in Figure 3.9.
3. Shape Recovery of Parts using Superquadrics

3.4 Input Data to Superquadric Model Recovery

Various types of images have been considered as input to superquadric recovery methods. The most convenient type of images are range images and other images with dense and explicit 3D information such as the images obtained by modern imaging techniques. From pairs of intensity images, stereo reconstruction is possible. But the drawback of the stereo method is a sparse and non-uniform distribution of 3D points. When just a single intensity image is available it is more difficult to establish the relationship between the data and the model.

3.4.1 Range Images

Range data is the input of choice for the majority of superquadric recovery methods [90, 106, 20, 48, 47, 119, 74, 73, 72, 71] since 3D models can be fitted straightforwardly to 3D range points. Range data can be acquired with a variety of techniques such as time of flight techniques which can use laser or sonar, stereo, and depth from focus. The best spatial accuracy can be achieved with structural light techniques. Because of self-occlusion and sensor geometry even
3. Shape Recovery of Parts using Superquadrics

range images captured from a single viewpoint do not offer complete 3D data but rather only \( 2 \frac{1}{2} \text{D} \) data [82]. A special type of range data is tactile information which can be obtained by a dextrous robotic hand which is equipped with tactile sensors, a superquadric model of grasped object can be obtained [1].

3.4.2 Stereo

Recovery of superquadrics from stereo images open specific problems related to highly nonuniform distribution of depth data. Metaxas [33, 25] and Chen [26] studied the problem of superquadric recovery from stereo and from multiple views in the framework of qualitative shape recovery [33, 34, 31] and physics-based recovery techniques [83].

3.4.3 Single Intensity Images

An interesting issue is the recovery of superquadrics from single 2D intensity images. Intensity images are normally used for derivation of contours, edges, or silhouettes which serve as input for superquadric recovery [92, 115]. Edge potentials derived from intensity images can serve as minimization force for superquadric recovery [109]. Intensity images can also be used in a more direct way employing the shading information which gives the estimates of the local surface normal. Dickinson [31, 32] proposed a method for part model recovery from contours extracted from intensity images which is based on distributed aspect matching. A qualitative shape recovery which consists of recovering a face topology graph and matching it in a hierarchical way with aspect graphs can be used first for recognition. For a recovered aspect exists a set of possible 3D part primitives, each with a corresponding probability of matching the given image region, which can in turn be used, as proposed in [33], for a quantitative shape recovery stage (superquadric models) using physics-based formulation
3. Shape Recovery of Parts using Superquadrics

in [109]. Pentland used 2D silhouettes for the segmentation part of his two stage superquadric recovery method [91]. 2D silhouettes of 3D superquadric parts of different shapes, scales and of different orientations were matched to 2D silhouettes in binary images.

Recovery of superquadric models directly from shading information was also attempted [23]. The shading flow field extracted from images was directly matched with the isoluminance field of the approximating shapes. Another method compared the intensity output image directly with synthesized intensity images of superquadric models whose parameters were adjusted by a genetic algorithm [98, 99].

3.5 Minimization of an Objective Function

The goal of shape recovery method is to find a model that fits the image data and in consequence its parameter values. Most of recovery methods define an objective function which can be evaluated for a given data. The goal is to find the parameters of superquadrics that minimize the objective function. Depending on the type of the objective functions and the type of input data different strategies of finding the solution are possible. The idea of all minimization approaches is to find a suitable cost, that is a fitting or an objective function which can be evaluated for all considered data points. When the sum of evaluations for all data points is minimized, the parameters (shape, size, orientation, and position) of the best fitting superquadric are revealed. All minimization methods used for superquadric recovery must proceed in an iterative fashion since the objective functions are highly non-linear. Surface data points which are acquired by a single sensor (camera or range scanner) are normally available only one side of an object and therefore additional constraints must be em-
employed to find a unique superquadric model. Pentland proposed such a coarse search through the entire superquadric parameter space [90]. A goodness-of-fit function is evaluated at selected points in the parameter space for many largely overlapping range image regions. The goodness-of-fit function is simply the number of pixels whose range is within \( \pm \sigma \) of the model's range, minus \( \lambda \) times the number of pixels that lie off the figure entirely. The model-data correspondence is along \( z \) axis. This segmentation stage is later followed by a gradient-descent optimization for individual superquadric models. The main drawback of this method is its excessive computational cost. From a computational point of view are much more desirable iterative gradient minimization methods where the minimum of an appropriate objective function can be found in a small number of iterations. Least-squares methods assume that the solution space of the selected objective function is convex enough so that the method eventually comes to an acceptable solution. Simulated annealing and genetic methods, on the other hand, do not put such severe restrictions on the objective function but require in general a much higher number of iterations and a higher computational cost.

Solina and Bajcsy were the first who formulated the recovery of superquadric models from pre-segmented range images as a least-squares minimization [106]. The fitting function is based on the superquadric inside-outside function \( F \), which is known also as algebraic distance. To address the non-even distribution of data points due to self-occlusion, they formulated the fitting function so that among all superquadrics that fit the data points, the preference is given to smaller ones. An iterative gradient descent method is used to solve the non-linear minimization problem.

Boult and Gross followed a similar strategy but studied different error-of-fit measures for recovering superquadrics from range data [20, 47]. In particular,
they reason that that the measure of fit based on the radial Euclidean distance, which is better metric than the algebraic measure based on the inside-outside function $F$. However the most intuitive measure-the exact Euclidean distance from a given point to the nearest point on the superquadric surface-can not be computed. Since the computation of error metrics which are based on the inside-outside function are simpler than the computation of the radial Euclidean distance and since the difference in recovered models is visually negligible, and also the cost function based on the inside-outside function are used more often. Boult and Gross used the Gauss-Newton method for least-squares minimization.

Whaitie and Ferrie addressed the problem of incomplete data due to self-occlusion from a more conservative standpoint [116]. They used the radial Euclidean distance as a measure of error for superquadric recovery and at the same time did not use any additional constraints besides the ones inherent to superquadrics, that is symmetry and convexity. Using a Gaussian noise assumption, they studied the propagation of errors and formulated a shell of uncertainty that encloses the surface of the fitted model and in which there is a given probability that the true surface of the volumetric models lies. The resulting superquadric shells of uncertainty are in general larger than the actual objects.

van Dop and Regtien evaluated the different cost function for gradient least-squares minimization [111]. In particular, they compared the performance of algebraic cost function, which is based on the inside-outside function, and the radial Euclidean distance [116]. They compared superquadrics recovered from several pre-segmented noisy range images. Initially, they got better result with algebraic cost function. By adding to the radial Euclidean distance method, first, background constraint and, second, a robust technique for elimination
3. Shape Recovery of Parts using Superquadrics

of outliers, they finally got better results, but only for cylindric objects. The background constraint assumes that the superquadric models rest on a planar support plane [52]. Since this background constraint cannot be expressed analytically as a constraint on superquadric parameters, points which are on the object's surface are projected onto the supporting plane. These projected points are assumed as supporting points, which are on the surface of the occluded side of the superquadric. These points are at the moment inside of the next iteration in the minimization process. The robust technique for elimination of outliers is based on the least trimmed squares.

3.6 Non-deformed Part Shape Recovery using the Inside-Outside Superquadric Function

3.6.1 Superquadrics in General Position

The inside-outside function defines the superquadric surface in an object centered coordinate system \((x_s, y_s, z_s)\). A superquadric in an object centered coordinate system is defined by 5 parameters such as 3 for size in each dimension and 2 for shape defining exponents. On the other hand, 3D input data points are expressed in a world coordinate system. To model or recover superquadrics from the 3D data, we must represent superquadrics in general position. A superquadric in general position requires 6 additional parameters for expressing the rotation and translation of the superquadric relative to the center of world coordinate system \((x_w, y_w, z_w)\). We use a homogeneous coordinate transformation \(T\) to transform the 3D points expressed in the superquadric centered coordinate system into the world coordinate system as shown in Figure 3.10.
Figure 3.10: A Superquadric in general position.

\[
\begin{bmatrix}
  x_w \\
  y_w \\
  z_w \\
  1
\end{bmatrix}
= T
\begin{bmatrix}
  x_s \\
  y_s \\
  z_s \\
  1
\end{bmatrix},
\]  
(3.25)

where

\[
T =
\begin{bmatrix}
  n_x & a_x & a_x & p_x \\
  n_y & a_y & a_y & p_y \\
  n_z & a_z & a_z & p_z \\
  0 & 0 & 0 & 1
\end{bmatrix}.
\]  
(3.26)

For a given point, transformation \( T \) first rotates that point by the parameters \( n, a \) and \( a \) and then translate it for \([p_x, p_y, p_z, 1]^T\). Since we need to express the points in superquadric centered coordinates, we have to compute them from
the world coordinates

\[
\begin{bmatrix}
    x^s \\
    y^s \\
    z^s \\
    1
\end{bmatrix}
= T^{-1}
\begin{bmatrix}
    x^w \\
    y^w \\
    z^w \\
    1
\end{bmatrix},
\]

(3.27)

Inverting homogeneous transformation matrix \(T\) gives

\[
T = \begin{bmatrix}
    n_x & o_x & a_x & -(p_x n_x + p_y n_y + p_z n_z) \\
    n_y & o_y & a_y & -(p_x o_x + p_y o_y + p_z o_z) \\
    n_z & o_z & a_z & -(p_x a_x + p_y a_y + p_z a_z) \\
    0 & 0 & 0 & 1
\end{bmatrix}.
\]

(3.28)

By substituting Eq. (3.25) and Eq. (3.28) into Eq. (3.8) equation, we get the inside-outside function for superquadrics in general position and orientation

\[
F(x^w, y^w, z^w) = \left(\frac{n_x x^w + n_y y^w + n_z z^w - p_x n_x - p_y n_y - p_z n_z}{a_1}\right)^\frac{2}{\tau_2} + \left(\frac{o_x x^w + o_y y^w + o_z z^w - p_x o_x - p_y o_y - p_z o_z}{a_2}\right)^\frac{2}{\tau_2} + \left(\frac{a_x x^w + a_y y^w + a_z z^w - p_x a_x - p_y a_y - p_z a_z}{a_3}\right)^\frac{2}{\tau_1}
\]

(3.29)

We use Euler angles \((\phi, \theta, \psi)\) to express the elements of the rotational part of transformation matrix \(T\). Euler angles define orientation in terms of rotation \(\phi\) about the \(z\) axis, followed by a rotation \(\theta\) about the new \(y\) axis, and finally,
3. Shape Recovery of Parts using Superquadrics

a rotation $\psi$ about the new $x$ axis.

$$
T = \begin{bmatrix}
\cos \phi \cos \theta \cos \psi - \sin \phi \sin \psi & -\cos \phi \cos \theta \sin \psi - \sin \theta \cos \psi & \cos \phi \sin \theta & p_z \\
\sin \phi \cos \theta \cos \psi - \cos \phi \sin \psi & -\sin \phi \cos \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \sin \theta & p_y \\
-\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta & p_x \\
0 & 0 & 0 & 1
\end{bmatrix}.
$$

(3.30)

The inside-and-outside for superquadrics in general position has therefore 11 parameters

$$
F(x_w, y_w; z_w) = F(x_w, y_w; z_w; a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, \phi, \theta, \psi, p_x, p_y, p_z),
$$

(3.31)

where $a_1$, $a_2$, $a_3$ define the superquadric size; $\varepsilon_1$ and $\varepsilon_2$ the shape; $\phi, \theta, \psi$ the orientation, and $p_x, p_y, p_z$ the position in space. We refer to the set of all model parameters as $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{11}\}$.

3.6.2 Objective Function

Suppose we have a set of $n$ 3D surface points expressed in some world coordinate system $(x_i, y_i, z_i), i = 1, \ldots, n$, which we want to model with a superquadric. A superquadric in general position is defined by the following equation

$$
F(x, y, z; \lambda_1, \ldots, \lambda_{11}) = 1
$$

(3.32)

We want to find such values for the 11 parameters $\Lambda(\lambda_j, j = 1, \ldots, 11)$ that most of the $n$ 3D points $(x_i, y_i, z_i)$ will lay on, or close to the superquadric's surface. There will probably not exist a set of parameters $\Lambda$ that perfectly fits the data. Finding the model for which the algebraic distance from points to model is minimal is defined as a least-squares minimization problem. Since for
any point \((x, y, z)\) on the surface of superquadric holds

\[
F(x, y, z; \lambda_1, \ldots, \lambda_{11}) = 1
\]

we have to minimize the following expression

\[
\min_A \sum_{i=1}^{n} (F(x_i, y_i, z_i; \lambda_1, \ldots, \lambda_{11}) - 1)^2
\] (3.33)

Due to self-occlusion, not all sides of an object are visible at the same time. So we have to consider general viewpoint to provide enough information of whole object. Range data in particular, when captured from a single view, is incomplete not only due to self-occlusion but also as a result of sensor set-up. Range data points then recover typically just a part of imaged objects. Even more, from a particular viewing direction, just one side of a 3D object can be seen. Obviously, from such degenerate views the shape of an object cannot be recovered. But even when a general viewpoint is assumed, certain objects such as parallelepipeds or cylinders (objects with surfaces where at least one principal curvature equals zero) do not provide sufficient constraints for a unique shape recovery using merely inside-outside function. Additional constraint was used by multiplying the minimization terms with \(\sqrt{\alpha_1 \beta_2 \gamma_3}\) which is the same as \(\sqrt{\lambda_1 \lambda_2 \lambda_3}\),

\[
\min_A \sum_{i=1}^{n} (\sqrt{\lambda_1 \lambda_2 \lambda_3} F(x_i, y_i, z_i; \lambda_1, \ldots, \lambda_{11}) - 1)^2
\] (3.34)

\[
R = \sqrt{\lambda_1 \lambda_2 \lambda_3} (F - 1)
\] (3.35)
3. Shape Recovery of Parts using Superquadrics

Since \( R \) is a nonlinear function of 11 parameters, minimization must proceed iteratively until the sum of least squares stops decreasing, or the changes are statically meaningless. We use the Levenberg-Marquardt method for nonlinear least squares minimization.

3.6.3 Parameter Constraints

And we use the following constraints to limit the possible shapes of superquadrics and to avoid singularities during the minimization, inequality constraints are introduced on function parameters \( A \). When \( \varepsilon_1, \varepsilon_2 > 2 \), superquadrics have concavities and we decide not to use such models in our shape vocabulary. Constraints are simple bounds on the parameter values in the form of intervals. Parameter constraints are implemented by a simple projection method. We take the search vector or the trial parameters \( A \) generated by the unconstrained minimization technique, and project the search vector so that it lies in the intersection of the set of constraint intervals. We use following constraints: \( \{a_1, a_2, a_3\} > 0 \) and \( 0.1 < \{\varepsilon_1, \varepsilon_2\} < 2 \). When \( \{\varepsilon_1, \varepsilon_2\} > 0.1 \), the inside-outside function \( F \) might become numerically unstable, although the superquadric shape stays perceptually almost same.

3.6.4 Initial Values Estimation of Superquadric Model

An initial estimate of the set of model parameters \( A_0 \) determines to which local minimum the minimization procedure will converge. During iterative model recovery, very rough estimates of object’s true position, orientation, and size suffice to assure convergence to a local minimum that corresponds to the actual shape. This is important since these parameters can be estimated only from the range points on the visible side of the object and hence the estimates cannot be accurate to begin with. Initial values for both shape parameters, \( \varepsilon_1 \) and \( \varepsilon_2 \)
are always 1.0 so that the initial model is an ellipsoid. Then an initial position of ellipsoid is set to the center of gravity among all \( n \) range points. The initial position \( p_x, p_y, p_z \) (or \( \lambda_9, \lambda_{10}, \lambda_{11} \)) of the initial ellipsoid \( A_0 \) is set to the center of gravity of all \( n \) range points \((x, y, z)\)

\[
\begin{align*}
\bar{x} &= p_x = \frac{1}{n} \sum_{i=1}^{n} x_{wi} \\
\bar{y} &= p_y = \frac{1}{n} \sum_{i=1}^{n} y_{wi} \\
\bar{z} &= p_z = \frac{1}{n} \sum_{i=1}^{n} z_{wi}
\end{align*}
\] (3.36)

To compute the orientation \((\phi, \theta, \psi)\) of the object-centered coordinate system, we compute the first the matrix of the central moments with respect to the center of gravity \((\bar{x}, \bar{y}, \bar{z})\).

\[
M_C = \frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix}
(y_i - \bar{y})^2 + (z_i - \bar{z})^2 & -(y_i - \bar{y})(z_i - \bar{z}) & -(z_i - \bar{z})(x_i - \bar{x}) \\
-(z_i - \bar{z})(y_i - \bar{y}) & (z_i - \bar{z})^2 + (x_i - \bar{x})^2 & -(x_i - \bar{x})(y_i - \bar{y}) \\
-(x_i - \bar{x})(z_i - \bar{z}) & -(y_i - \bar{y})(x_i - \bar{x}) & (x_i - \bar{x})^2 + (y_i - \bar{y})^2
\end{bmatrix}
\] (3.37)

Central moments are moments with respect to the center of gravity \((\bar{x}, \bar{y}, \bar{z})\). We want to find a rotational matrix \(T_R\) which makes the matrix of moments \(M_C\) diagonal [97, 60, 61]. The new diagonal matrix of moments \(D\) is then

\[
D = T_R^{-1} M_C T_R,
\] (3.38)

where \(T_R\) is the rotational part of transform \(T\) in Eq. (3.30). On the other hand, matrix \(M_C\) can be diagonalized with a diagonalization matrix \(Q\), whose
columns are eigenvectors of matrix $M_C$.

$$D = Q_R^{-1}M_CQ_R.$$  \hfill (3.39)

Comparing Eqs. (3.38) and (3.39) gives

$$T_R = Q.$$ \hfill (3.40)

Rotation matrix $T_R$ that orients the object centered coordinate system along the axis of minimum inertia can thus be assembled out of eigenvectors of matrix $M_C$. Eigenvector $e_1$ with the smallest eigenvalue $\kappa_1$ corresponds to the minimum-inertia line. The minimum inertia line is also known as the principal axis [97]. We use Jacob method for computing the matrix of eigenvectors which consists of sequence of orthogonal similarity transformations designed to annihilate one of the off-diagonal matrix elements. Under the assumption that bending and tapering deformations normally affect objects along their longest side, we decided to orient the object-centered coordinate system so that the $z$ axis lies along the longest side for elongated object (axis of least inertia) and along the shortest flat objects (axis of largest inertia). This causes the $z$ coordinate axis to coincide with the axis of the rotational symmetry. For round flat objects, on the other hand, we want to the $z$ coordinate axis to coincide with the axis of the rotational symmetry. Given the three eigenvectors $e_1$, $e_2$, $e_3$, we have to assign to them coordinate axes labels. For ordered eigenvalues $\kappa_1 < \kappa_2 < \kappa_3$, of the three corresponding eigenvectors $e_1$, $e_2$, $e_3$, the $z$ axis is assigned according to the following rule

if $|\kappa_1 - \kappa_2| < |\kappa_2 - \kappa_3|$ then $z = e_3$

else $z = e_1$
This condition puts the $z$ axis along the longest side of elongated object and perpendicular for flat, rotationally symmetric objects. From the elements of the rotation matrix $T_R$ which makes up the rotational part of the homogeneous transformation $H$, Eq. (3.26), we compute the equivalent Euler angles $\phi_0, \theta_0, \psi_0$ (or $\lambda_6, \lambda_7, \lambda_8$) For evaluating the inside-outside Eq. (3.29), we could use the elements of the rotational matrix $T_R$ directly, but the partial derivatives required for minimization of the fitting function are all expressed in terms of Euler angles. The size $a_1, a_2, a_3$ (or $\lambda_1, \lambda_2, \lambda_3$) of the initial ellipsoid $A_0$ are computed from the bounding box of the range points whose sides are aligned with the new object-centered coordinate system. The initial estimates computed in the described fashion are sometimes very close to the actual parameter values. But even when the initial estimates for rotation are quite poor-this is the case with objects which are not elongated- the correct model is recovered. This would suggest that the estimation of orientation is not crucial at all. The initial ellipsoid could have a default orientation, for example, the same as the world coordinate system. This would be acceptable for blob-like objects, but especially for elongated objects we want to align the $z$ coordinate of the object centered coordinate system with the longest axis of the superquadric so that it can be subject to bending, if necessary. Recovery of a single superquadric model requires on the average only about 20 iterations. The required time for each iteration depends on the number of range points. Since the computations involving individual range points are independent, the method lends itself naturally to a parallel implementation. Since the perfect segmentation is assumed, the method by itself does not cope well with outliers. In the context of the recover-and-select segmentation method, the outlier problem is greatly reduced since the inclusion of range points into an individual superquadric model is dynamically adjusted by the segmentation process.
3. Shape Recovery of Parts using Superquadrics

(a) Block.  (b) Cylinder.  (c) Sphere.

(d) Surface plot of block.  (e) Surface plot of cylinder.  (f) Surface plot of sphere.

Figure 3.11: Volumetric primitives.

3.7 Experiments for the Superquadric Model Recovery

The synthetic range images are useful because they are taken from objects designed with a CAD modelling system, allowing exact control over the elongation and other shape attributes. We assume that the 3D input object is mainly consisted of blocks, cylinders, and spheres. Figure 3.11 shows volumetric primitives used in this experiment, where block in Figure 3.11(a) has a dimension of $40 \times 40 \times 160$, cylinder in Figure 3.11(b) has the radius value of 15 and the height of 70, sphere in Figure 3.11(c) has the radius of 30. Figure 3.12 shows the
result of shape recovery by using non-deformed superquadric model recovery. In Figures 3.12(a) 3.12(d), 3.12(b) 3.12(e), and 3.12(c) 3.12(f) are the results after 1st, 5th, and 15th iteration respectively. To recover superquadric model from input range image, we used nonlinear iterative optimization method of Levenberg-Marquardt till maximum 15 iterations. The volumetric primitives used in this experiment are simple part objects, which have symmetry attributes of the cross-section without any shape deformations. In addition, the used primitives should be elongated at least along one axis direction. If objects have the same size parameters along z, y, and z axes (a1, a2 and a3) and/or has rotationally asymmetric attributes of cross-section, we cannot make a precise recovery. Bending and linear tapering are two deformations commonly applied to superquadrics, which affect the symmetry attributes of the objects as a whole. However, these operations do not affect the symmetry attributes of the cross-section. In order to demonstrate robustness of superquadric shape recovery, this paper shows the experimental results of the input objects having some data loss. We can construct composite objects by combining several volumetric primitives. At this time, there is likely to be existed data loss because of occlusions. The case with no defects is shown in Figure 3.13 (a) and the number of range data points is 5055. Figure 3.13 (b) and (c) include data loss at the bottom of cylinder (3846 data points) and at the top (4415 points) respectively. As shown in Figure 3.13 (d) and (e), we can recover these cylinders compactly in spite of those defects. For the test of robustness in superquadric shape recovery in case of severe occlusion, a split cylinder due to the occlusion by other object is used as shown in Figure 3.14(a). Assuming superquadric model has a symmetry characteristics, the shape recovery of split cylinder is approximated with whole cylinder as shown in Figure 3.14(b). Figure 3.14(c) and Figure 3.14(d) trace the history of the estimated superquadric shape parameter and the size
Figure 3.12: Superquadric shape recovery of volumetric primitives.
Figure 3.13: Superquadric model recovery when some data loss in object part shape exists, (a) Original cylinder image, (b) Cylinder with top loss, (c) Cylinder with bottom loss, (d) Shape recovery for the top loss, (e) Shape recovery for the bottom loss.
parameter respectively.

For the test of deformed superquadric model recovery, we used both synthetic range image of the cone and real range image of the bent cylinder due to tapering and bending deformation respectively. In general nondeformed superquadric model recovery, the superquadric inside-outside fitting function has 11 parameters to be recovered by using the Levenberg-Marquardt method for nonlinear least squares minimization. Deformed superquadrics can be recovered using the same technique as for the recovery of nondeformed superquadrics. The only difference is that some additional parameters describing deformation must be recovered also. Tapering deformation needs additional 2 parameters, \( K_x \) and \( K_y \) like in Eq. (3.16). During model recovery both tapering parameters are adjusted simultaneously with the other 11 parameters. Bending deformation needs additionally 2 parameters, bending curvature \( k \) and bending angle \( \alpha \). The initial values of all these deformation parameters are 0 in the iterative stage except bending curvature \( k \), since \( k = 0 \) means singular condition in Eqs. (3.19)–(3.22). Figure 3.15(a) shows the synthetic cone range image and Figure 3.15(b) describes its surface normal image. Figure 3.15(c) indicates the result of the 15th iteration of superquadric model recovery and the trace of tapering deformation parameters is shown in Figure 3.15(d). The experiment for bending deformation, which uses real range image of the bent cylinder, is shown in Figure. Figure and Figure depict the traces of the bending parameters of curvature and angle respectively. It shows that within the 15th iterations, we can get the best fitting of optimization function.
3. Shape Recovery of Parts using Superquadrics

(a)  

(b)  

Shape parameters

(c)
Figure 3.14: Superquadric model recovery for a split cylinder due to occlusion, (a) Original cylinder image, (b) Shape recovery for the split cylinder, (c) Trace of the shape parameters, (d) Trace of the size parameters.
3. Shape Recovery of Parts using Superquadrics

(a) 

(b) 

(c)
Figure 3.15: Deformed superquadric model recovery for a synthetic cone object. 
(a) Original cone image, (b) Surface normal image, (c) Superquadric model recovery, (d) Trace of the tapering parameters.
3. Shape Recovery of Parts using Superquadrics
3. Shape Recovery of Parts using Superquadrics

![Graph showing bending angle](image)

(d)
Figure 3.16: Deformed superquadric model recovery for a real bent cylinder object, (a) Original cone image, (b) Surface normal image, (c) Superquadric model recovery, (d) Trace of the bending angle parameter, (e) Trace of the bending curvature parameter.
Chapter 4

Volume Decomposition using Recover-and-Select Paradigm

A common underlying task of most recognition applications is building the scene description in terms of symbolic entities. A challenging problem in scene understanding is segmentation, where each piece of information must be mapped either to a shape primitive or discarded as noise. At the same time, there should be a minimum number of such primitives involved, so as to get as a compact description as possible. There have been several attempts to segment and recover volumetric models from the range data. In order to partition the data into their parts that can supposedly be represented with a single volumetric model, these approaches usually involve various kinds of procedures which are mostly applied in a hierarchical fashion to have ranges from the estimation of local surface properties such as curvature, to more complex symmetry seeking procedure. Such approaches, in fact, isolate segmentation stage from the representation stage and significant efforts are necessary to combine, usually surface type descriptions into volumetric models.
4. Volume Decomposition using Recover-and-Select Paradigm

4.1 Overview of Superquadric-Based Segmentation Methods

There have been many kinds of approaches to scene segmentation using superquadrics. In general, they have been applied together with the recovery of superquadric part-models from image data. This approach to segmentation can be roughly divided into following two categories.

4.1.1 Segment-then-Fit Method

Pentland proposed a method of matched filters to segment binarized image data into superquadric part-structure (Pentland, 1990). The first part of this two-stage method is based on matching 2D silhouettes (2D projections of 3D superquadric parts of different shapes, scales and of different orientations) to image data. After part segmentation the actual superquadric recovery was performed in the second stage using a modal dynamics formulation. Here 3D superquadric models were fitted to the range data of individual part regions. The 3D data corresponding to each of the selected patterns was fitted with deformable superquadric based on modal dynamics [88, 92]. The error metric used for numerical minimization is the squared distance along depth axis between the range data and the projected volume’s visible surface. Although Pentland’s segmentation method is quite robust and efficient, the assumption that of every part being visible in a 2D projection is frequently violated in cluttered 3D scene. Gupta proposed an edge-based region growing method for segmenting range images of compact objects in a pile [52]. The regions were segmented at jump boundaries, and each recovered region was considered a superquadric model. This method was applied to sorting of postal packages, which, for the most part, can be represented by superquadrics. Problems occur
at the segmentation stage, where thin overlapping objects cannot be segmented at jump boundaries. Ferrie proposed a two-stage data-driven strategy of fitting superquadrics after segmenting range data [40]. They used differential geometric properties and projected space curves modeled as snakes. A Darboux frame is computed at each point by fitting a parabolic quadric surface, which is iteratively refined by a curvature consistency algorithm. The part boundaries exist at critical points defined as negative minima of principle curvature lines. The 2D projected snakes were then used to trace the critical points. Assuming that the objects are composed of convex volumetric primitives and do not contain holes of any kind, and that the observer is free to choose its viewing position, objects can be partitioned at concave discontinuities to give parts that can be described by superquadrics. Solina and Bajcsy also proposed superquadric model recovery method and presented results on complex objects [106]. But the method can fail on noisy data, and if the first critical assumption about the object being piecewise convex joined at concave boundaries is not satisfied. Darrell also used superquadrics to represent the surface segmented in range data [29]. Surfaces, described as bi-quadrics, were segmented using MDL (minimal description length) criterion. Each surface was assumed to map to a superquadric model without further segmentation. This assumption is true only when the object is piecewise convex, and all the surfaces of the parts can be partitioned at the concave boundaries formed at part intersections. It is easy to see that this assumption is frequently violated in the real world, where different parts can have smoothly continuing surfaces at their intersections such as the L-shaped object in Figure. For a qualitative part-based 3D object representation, Raja and Jain constructed 12 geon shapes, and matched them with the recovered aspects of the parts of complex objects [85]. They segment range data into surface patches, classify them according to curvature properties and generate
a surface adjacency graph (SAG) encoding the surface aspect. The subgraphs of the SAG are then matched with the stored catalog to segment the SAG. The recovered parts are then identified by either directly testing the surface attributes or by fitting superquadrics and mapping them to geons. However, like any two-stage approach, it is totally dependent on an independent surface segmentation method, which affects the final result of volumetric segmentation. Furthermore, it has the same assumption with all two-stage methods that each surface in the SAG belongs to only one part. This assumption will not be true for a simple object like L shape to be segmented.

4.1.2 Segment-and-Fit Method

Recovery of superquadric model is sensitive to missing data, noise, and incorrect estimation of orientation. More precisely, model recovery of partial data will be uncertain about the shape, size, and orientation of the model. Consequently, a region growing type of method has to begin with a number of hypotheses about orientation, size, and domain of superquadric model. So segmentation into part models must recover parts by hypothesizing parts and testing them. Pentland was the first who modeled multiple range data with superquadrics [90]. His segmentation consists of three steps. The first step recovers superquadric models on many overlapping image regions by a coarse-grain search through the entire parameter space. This is followed by performing a gradient-descent optimization of the best fitting model in each image region. The best overall description of the data is obtained by picking a minimal covering of the data from among the set of regionally-best-fitting models. This method is computationally expensive due to the coarse search through the entire superquadric parameter space for each image region. The computational load can be reduced by limiting the search regions to the vicinity of the skeleton of the data. And
also the skeletonization of 3D data is by no means trivial, and is in fact similar in complexity to the segmentation problem. However, Pentland's method does demonstrate the feasibility of achieving a multi-primitive description by fitting many independent models and selecting the best among them. In Solina's dissertation, he tried to segment range data of multi-part objects by performing the superquadric recovery method on the entire range point data set [105]. If the initial superquadric encloses the entire set of range data points so that most of the points lie inside of the superquadric, the superquadric will begin to shrink once the minimization starts. By means of adaptive thresholds, points which are too far outside of the current superquadric are temporarily discarded from the minimization. In this way, the superquadric model shrinks until it fits to a stable data set and then grows again until it finds all the points which can be represented by a superquadric model. Unfortunately, this simple part segmentation method based on dynamic adjustment of the range point data set during superquadric recovery was not stable enough.

DeCarlo and Metaxas proposed a much more sophisticated segmentation method along the same general strategy of fitting first a global model over the whole range point data set, and then splitting the model [30]. They named their method shape evolution since it is concerned with how a shape model changes during the course of fitting. The changes of the model include geometric changes such as deformations and representational changes such splitting a model in two parts which are then glued together by way of blending. Estimation of shape using this representation is realized using a physics-based framework.

Gupta and Bajcsy proposed a recursive global-to-local technique for part recovery [53]. A global model is recovered for the entire data, which is evaluated by examining local and global residuals so that the further course of action can be determined. A set of qualitative acceptance criteria defines the
stability of the model as the fraction of data that can be overestimated and underestimated by the model. The scale considerations can be incorporated at two levels such as the tolerance on accepting the description of individual data points, and as the acceptance criteria for the global residuals. Occlusions are accounted for allowing partially occluded parts to be described by a single model. If the model is found to deficient in representing data, then additional part models are hypothesized on the undescribed data. The global model is refitted on the remaining data. Thus, the global model shrinks while the part models grow, yielding a hierarchical part-structure. The residual-based model acceptance criteria are enforced to keep all the models within a predefined tolerance. Further, to make the procedure robust and efficient, surface attributes like bi-quadratic orientation, bi-quadratic patches, 3D concave and convex edges are used, for which the surface segmentation was employed. As seen with two-stage techniques, relying completely on surface segmentation can result in incorrect volumetric segmentation. Gupta and Bajcsy only generate hypotheses about parts using surface information and also use bi-quadratic orientation to orient the superquadric major axis since it is crucial to model recovery [50]. We introduced and modified recover-and-select algorithm to accept input 3D range data points and segment it into subset of volumetric parts after removing unwanted background noise. It has also an advantage that does not need priori knowledge about the object to be segmented. The ability even to identify a set of surfaces as belonging to a given volume is not a trivial task without knowing at least connectedness of surfaces. Moreover, only a surface-level description may not be consistent with the volumetric description. Figure 4.1 show as an example of an L-shape object whose volumetric description cannot be obtained by a simple combination of recovered surfaces.

Specifically, planar patches of the surface-level description in Figure 4.1(c)
4. Volume Decomposition using Recover-and-Select Paradigm

Figure 4.1: Comparison of surface and volumetric description of L-shaped object.
cannot be partitioned in a simple way to correspond to volumetric parts in Figure 4.1(d), since the left surface patch number 11 in Figure 4.1(c) belongs to two different volumetric models.

4.2 Recover-and-Select Segmentation Paradigm

We present here a general outline of the recover-and-select segmentation paradigm proposed by Leonardis [76, 75, 77]. The procedure for the recovery of a single model, which can be roughly partitioned into three distinct modules such as parameter estimation, decision making, and data point classification. In parameter estimation module, given a set of data points, a type of the parameter model, and an estimation method, find an optimal set of parameters and evaluate the goodness-of-fit measure between the model and the corresponding data. In decision making module, if sufficient similarity of goodness-of-fit measure is established, we accept the currently estimated parameters, together with current data set, and proceed with a search for more compatible points. Otherwise, a decision is made whether to terminate procedure. In data point classification module, an efficient search for more compatible points which is performed in the vicinity of the present border of the model is achieved through the extrapolation of the current model. Data points close to the current model are tested for consistency. Outliers, which can be classified either as extreme measurement deviations or as data points that belong to neighboring models, are rejected. New consistent image elements are temporarily included in the data set. This completes one cycle of the algorithm, and the procedure continues at the first module.

The final outcome of the model recovery procedure for a model $M_i$, which has been evolved from the $i$-th seed, consists of three terms such as the region
4. Volume Decomposition using Recover-and-Select Paradigm

$R_i$, the vector of model parameters $\Lambda_i$, and the goodness-of-fit measure $\xi_i$. The region $R_i$ represents the domain of the model $M_i$ and includes image elements that belong to the model. And the goodness-of-fit measure $\xi_i$ evaluates the conformity between the data and the model.

4.2.1 Seed Selection

Initial seeds are placed on the range image in the grid-like pattern of windows. An initial seed encompasses a set of range data points in a small rectangular window, and a superquadric model is fitted to the data set. Next, a decision is made whether all data points belong to that single mode. This decision is based on average error-of-fit measure $\bar{\xi}_i$

$$\bar{\xi}_i = \frac{1}{|R_i|} \sum_{x \in R_i} d(x, M_i) = \frac{1}{|R_i|} \xi_i.$$  \hspace{1cm} (4.1)

It just eliminates those seeds that were placed on data sets that cross part boundaries and thus helps reducing the number of seeds at start of processing.

4.2.2 Superquadric Fitting

The superquadric fitting function $F_s$ can be regarded as an energy function on the space of model parameters. Minimization method can, in general, only guarantee convergence to a local minimum. The starting position in the parameter space determines to which minimum will the minimization procedure converge. Initial values for both shape parameters, $c_1$ and $c_2$ are set to 1, which means that the initial model $\Lambda_0$ is always an ellipsoid. Position in world coordinates is estimated by computing the center of gravity of all points, and the orientation is estimates by computing the central moments with respect to the center of gravity. The initial model $\Lambda_0$ is oriented so that the axis $z$ of
the object centered coordinate system lies along the longest side of the object which is also the same with the axis of least inertia. We then use the model with smaller error-of-fit as a final result for the model recovery.

4.2.3 Decision Making

A decision whether a model should grow further or not, depends on the established similarity between the model and the data. If sufficient similarity is established, we accept the currently estimated parameters, together with the current data set, and proceed with the search for more compatible points. The error-of-fit measure equation is used in this decision making. Due to its dependence on the superquadric size and shape parameters \((a_1, a_2, a_3, \varepsilon_1, \varepsilon_2)\), the algebraic distance is not suitable. Instead, we use the radial Euclidean distance metric between a point \(x\) and the surface of a superquadric \(\Lambda\)

\[
d(x, \Lambda) = |x| \left\| 1 - F^{-\frac{1}{2}}(x, \Lambda) \right\| ,
\]

(4.2)

where \(F\) is inside-outside function. The sum over all data points belonging to the model determines the total error-of-fit \(\xi_i\) of the entire model. This measure is also passed to the selection procedure.

4.2.4 Search for New Compatible Points

An efficient search for more compatible points is performed in the vicinity of the present border points of the region corresponding to particular model. The border points are determined on eight-connectedness and then their distance to the corresponding model is evaluated. Only those points that are close enough to the original model are included in the updated set of points. On this set of points, a new superquadric model recovery procedure is started.
4. Volume Decomposition using Recover-and-Select Paradigm

4.2.5 Model Selection

The model recovery procedure is interrupted by the model selection to eliminate superfluous models. The redundant representation obtained by the model recovery procedure is a direct consequence of the decision that the search for parametric models is initiated everywhere in the image. The redundancy is reflected in the fact that several of the models are completely or partially overlapped. In other words, a single data point can be represented by several models. The final description of an image has to be obtained by fusing different partial descriptions into a globally consistent representation.

The selection procedure is performed on the level of models rather than on the level of the model's constituent elements. Given a set of all recovered models together with their corresponding domains \((M_i, R_i) : i = 1, \ldots, N\), we elect an optimal subset of models which compromise the final result. Thus the core of the problem is how to define optimal criteria for choosing the best description. We introduce MDL (minimum length description) principle for making such judgments. This is an information theoretic approach closely related to maximum-likelihood and maximum-a-posteriori estimation, although with the attractive property that it allows for model selection in addition to parametric estimation [96, 70]. We use this MDL principle as a criterion function only to select a subset of models from all potential models that could possibly be included in the final description. Prior to the recovery of models, the description of an image can only be given in a pointwise form. After recovering a set of models we can describe parts of the image, or possibly the whole image, in terms of a selected subset of the set of all models. Let vector \( m^T = \{ m_1, m_2, \ldots, m_N \} \) denote a set of models, where \( m_i \) is a presence-variable having the value 1 for the presence and 0 for the absence of the model \( M_i \), and \( N \) is he number of all
models. The length of encoding of an image $L_{\text{image}}$ can be given as the sum of two terms

$$L_{\text{image}}(m) = L_{\text{pointwise}}(m) + L_{\text{models}}(m).$$

(4.3)

$L_{\text{pointwise}}$ is the length of encoding of individual data points that are not described by any model, and $L_{\text{models}}$ is the length of encoding of data described by the selected models. The idea is to select a subset of models that would yield the shortest length description. In other words, we should tend to maximize the efficiency of the description, defined as

$$E = 1 - \frac{L_{\text{image}}(m)}{L_{\text{pointwise}}(0)},$$

(4.4)

where $L_{\text{pointwise}}(0)$ denotes the length of encoding of the input data in the absence of models. To produce the best description in terms of models, we define the objective function to be maximized as following form

$$F(m) = m^T Q m$$

(4.5)

$$= \begin{bmatrix} c_{11} & \cdots & c_{1i} & \cdots & c_{1N} \\ \vdots & \ddots & \vdots & & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ c_{N1} & \cdots & c_{Ni} & \cdots & c_{NN} \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ \vdots \\ m_N \end{bmatrix}^T \begin{bmatrix} m_1 \\ \vdots \\ \vdots \\ m_N \end{bmatrix},$$

The diagonal terms of the matrix $Q$ express the cost-benefit value for a particular model $M_i$,

$$c_{ii} = K_1 |R_i| - K_2 \xi_i - K_3 |P_i|,$$

(4.6)
where the region \( R_i \) represents the domain of the model \( M_i \), \( P_i \) is the vector of model parameters, and \( K_1, K_2, K_3 \) are weights which can be determined on a purely information-theoretical basis. The off-diagonal terms handle the interaction between the overlapping models

\[
\sigma_{ij} = \frac{-K_1|R_i \cap R_j| + K_3 \xi_{ij}}{2}, \tag{4.7}
\]

\(|R_i \cap R_j|\) is the number of total points that are included by both models, and \( \xi_{ij} \) is defined as

\[
\xi_{ij} = \max \left( \sum_{x \in R_i \cap R_j} d^2(x, M_i), \sum_{x \in R_i \cap R_j} d^2(x, M_j) \right), \tag{4.8}
\]

The error terms \( d^2(x, M_i) \) and \( d^2(x, M_j) \) are calculated in the region of intersection \( R_i \cap R_j \) and correspond to deviations from the \( i \)-th and the \( j \)-th model, respectively. The objective function takes into account the intersection between different models which may be completely or partially overlapped. From the computational point of view, it is important to notice that the matrix \( Q \) is symmetric, and depending on the overlap of the models, it can be sparse or banded. All three properties of the matrix \( Q \) can be used to reduce the computations needed to calculate the value of \( F(m) \).

### 4.3 Optimization Problem

The optimization problem belongs to the class of problems known as combinatorial optimization. Since the function \( F(m) \) is a quadratic function of variables that can only take the values 0 and 1, the problem is also known as the Quadratic Boolean Problem. The number of models \( N \) is the size of
the problem, and the solution space can be represented by the corners of an \(N\)-dimensional hypercube. Since the number of possible solutions increases exponentially with the size of the problem, \(2^N\), it is usually not tractable to explore them exhaustively except for a very small number of models. In our case, we can obtain a reasonable solution by a direct application of the greedy algorithm which sequentially selects the option which is locally optimal.

### 4.3.1 Greedy Algorithm

We use alternative notion \(f(S)\) for the objective function \(F(m)\) to simply the greedy algorithm description and analysis,

\[
F(m) = f(S), \quad S = i; m_i = 1. \tag{4.9}
\]

The \(f(S)\) denotes the quality of overall description consisting of models in set \(S\) as a sum of elements \(c_{ij}\) of the matrix \(Q\)

\[
f(S) = \sum_{i \in S} \sum_{j \in S} c_{ij}. \tag{4.10}
\]

The greedy algorithm shown in Figure starts with an empty set of selected models \(S\) and a set of candidates \(C\) that contains all the models. At each iteration of the selection procedure, a single model \(y\) from the set of candidates \(C\) is selected such that it maximize the \(f(S \cup \{x\})\) over all elements \(x\) of the set \(C\)

\[
f(S \cup y) = \max_{x \in C} f(S \cup x), \tag{4.11}
\]

and that the equality of the overall description increases by including the model \(y\) in the set \(S\)

\[
f(S \cup y) > f(S). \tag{4.12}
\]
The set $S$ is then replaced with $S \cup \{y\}$ and set $C$ with the set $C\{y\}$. The iteration proceeds until there is no model in set $C$ satisfying conditions 4.11 and 4.12 or the set $C$ is empty. The greedy algorithm can be heuristic which gives an overall optimal solution only under specific conditions. Models grow spatially as long as there are more compatible points left and the deviation between the data and the model is below a prefixed threshold. Depending on how the model growing stops, we can categorize it into following one of three cases:
• A model whose error is within the tolerance stops growing because no more compatible points are found. Note that only the model that found a close match to the data will fall into this category.

• A model stops growing because the initial or intermediate stages of the model recovery included outliers which caused the error term to exceed the tolerated deviation. This situation occurs when some seed regions result in erroneous initial estimates of the parameters of the models. The extent of these models is usually small in comparison to the models that started growing from non-contaminated seeds.

• A model stops growing because the error has slowly accumulated to the maximum value. An analysis of the residual would reveal a systematic error accumulation during the growing process.

When an area in the image is covered with models of the first type, possibly overlaid with models of the second type, the greedy algorithm will produce the optimal result. In the case of models of the third type, the heuristic greedy algorithm gives us a sub-optimal solution in terms of $F(m)$ since it does not possess the look-ahead capability to optimally arrange the models in a piecewise description.

4.3.2 Combination of Model Recovery and Model Selection

There are two major components of the volumetric description procedure, namely the module for model recovery and the module for model selection. They can be combined to obtain a fast and an efficient overall method. For example, these modules can be applied in succession or in a loop as shown in Figure 4.3. In order to achieve a computationally efficient procedure, the model-recovery and the model-selection procedures are combined in an iterative way. To eliminate
Figure 4.3: Comparison between recover-then-select method and recover-and-select method.

Superfluous models, the recovery of currently active models is interrupted by the model-selection procedure which selects a set of currently optimal models which are then passed back to the model-recovery procedure. This process is repeated until the remaining models are completely recovered.

We choose to interrupt the recovery procedure after every \( n \) growing operations, where \( n \) is set to half the seed size at the beginning and then incremented to \( 2n \) after every selection. For example, if the seed size is a square of \( 4 \times 4 \) pixels, the first recovery phase consists of 2 growing iterations, which produces regions of size at most \( 8 \times 8 \) pixels, after the 1st selection the secondary recovery phase consists of 4 growing iterations, leading to a regions of size at most \( 16 \times 16 \) pixels, etc. Since the initial seed regions are placed in a grid like manner, such procedure ensures sufficient overlap of the models, so that the number of models is reduced during selection while it is still not too computationally expensive to let them grow to such a size.
Chapter 5

Part-Based Superquadric Model Construction and Matching

5.1 3D Object Modelling and Recognition

5.1.1 Curvature-Based Representation and Matching

Several groups have focused on curvature properties for object representation and recognition. Some of these efforts have focused only on representation and segmentation while others have attempted at the development of the full system capable of taking range data of a scene and finding a match from a library of known objects.

Nevatia and Binford [85] is probably the first paper published concerning object recognition in range data. The emphasis in this paper is on the analysis of scenes containing curved objects, which are represented as sub-part hierarchies
of generalized cones (cylinders). The data-driven recognition processing of this approach can be summarized as follows:

(a) Range image edge and region features are extracted and organized to create image object descriptions which are structured and symbolic.

(b) Important features of these object descriptions are used to index into a library of object models to retrieve a set of models that are similar to the objects in the image.

(c) The image object description is compared to each of the retrieved models and the best match is chosen.

(d) Verification is performed to see if the differences in the best retrieved object model and the image object description are reasonable. (This step was not implemented.)

Experimental results are discussed for a doll, a horse model, a glove, a ring, and a snake-like object. Objects with different structure were easily distinguished and even moderate amounts of occlusion were handled successfully. This work does not seem to have directly evolved into any more recent range data systems.

Kuan and Drazovich [69] have developed a system that attempts to extend the principles of the ACRONYM approach [21, 22] to range imagery. They use generalized cylinder object models with model priorities and subpart attachment relations to yield multi-level coarse-to-fine object descriptions. They use a model-driven prediction modules that predicts the following features at different levels to enable coarse-to-fine multi-level interpretation:

(a) Object Level: These features include spatial relationships among object components, overall dimensions, extreme points, side view characteristics, and occlusion relationships among object components.
of this

ned to

into a
to the

eved
step
,
and
ished
work
tems.
tend
They
rt at-
They
es at
bject
stics,

(b) Cylinder Level: This level is the most important level because cylinders are the basic symbolic entity of the object description. These features include cylinder contour, cylinder position, and orientation, parallel edge relationships, edge types, cylinder length, extent of overlap with other cylinders, and overall cylinder visibility and obscuration.

(c) Surface Level: These features include information as to whether the surface is planar or curved, surface edge boundary information, and spatial surface relationships.

(d) Edge Level: These features include information as to whether the edge is occluding, convex, or concave.

These predictions give guidance to the low-level feature extraction processes and they also provide mechanisms for feature-to-model matching and interpretation. In contrast to the ACRONYM system, actual measured features are used for matching based on maximizing likelihood rather than creating constraints for later constraint propagation processing. The work seems reasonable for a model-driven approach, but is limited by the dependence of generalized cylinder type objects.

Smith and Kanade [103] discuss a program designed to produce object-centered three-dimensional object descriptions from depth maps. Conical and cylindrical surfaces are used as primitives. The object descriptions derived from their data-driven approach can be used for matching and object recognition. Coherent relationships between sub-cylinders of parts are used to aid the extraction of object surfaces. As an example of this coherency is the relationship between the handle of a pen and the main body of a pan.

Gennery's [42] main concern for object recognition was obstacle avoidance for autonomous vehicle navigation. His algorithm can be summarized as fol-
5. Part-Based Superquadric Model Construction and Matching

Professor Y. Bhanu [13] proposes a 3D scene analysis system for recognizing 3D objects in depth maps. The system uses the object representation and surface extraction method discussed by Henderson [55]. It constructs object models from physical prototypes using multiple view depth maps. A complex curvature-surface automobile part is discussed. Total 8314 3D object surface points were obtained by transforming points from 14 individual views into a common object centered coordinate system. These surface points are used to fit a convex-faced polyhedron using two step algorithm: (1) The three-point seed algorithm is used to group all points into face regions using convexity and narrowness tests (four threshold values needed for this), (2) the face regions are then approximated by 3D planar convex polygons. For the auto part, 85 flat faces are computed to describe the curved surface part. Object recognition is accomplished after model determination as follows. A depth map from an arbitrary view (same scale) is acquired using a rangefinder. The object points are segmented from the background and a polygonal face approximation of the object surface is computed using the same technique mentioned above for model determination. This generates approximately 10-25 faces for unknown views of the auto part. These faces are used to perform object matching using a relaxation-based scheme called stochastic face labeling. The face features of area, perimeter, peroun (the measure of...
the roundness of a region. Peround is computed as the square of the perimeter divided by the area.), length of maximum, minimum, and mean radius vectors from the face centroid, number of vertices, and angle between maximum and minimum radius vectors are used to compute the initial stochastic labeling probabilities. In addition, a face neighbor table is computed where neighbors are ranked according to area. A first stage iteration is performed which involves maximizing the first stage compatibility measure which is defined in terms of a one-largest-area-neighbor compatibility function. Using the labels at the end of the first iteration, a second stage iteration is then performed which involves maximizing the second stage compatibility measure which is defined in terms of a two-largest-area-neighbor compatibility function. Both compatibility functions use following quantities: the distance between neighboring face centroids, the ratio of the areas of neighboring faces, the difference in face orientations, and the rotation information concerning the object can be computed. The method appears to handle arbitrary viewpoints. However, it does seem to rely very heavily on the consistency of the output from the face-finding algorithm. Some face adjacency information is also being ignored.

Ballard and Sabbah [2] have investigated viewer independent shape recognition by factoring an image object description into an object-centered view-independent description and view-dependent view transformation. They emphasize a decoupling of the three subgroups of scale, orientation, and translation parameters. It is assumed that a planar surface patch description of an object is available to them both as a known prototype model and as sensor data from either processed range data or other sources. They also assume that the scale is already known and that the orthographic projection approximation is valid. Their 3D algorithm consists of two main sequential processing steps: (1) Use the Generalized Hough transform (GTH) to compute the three 3D rotational
parameters corresponding to a given view and a given object, and (2) determine the two 2D translational parameters via another GTH. If the correct object is not being matched, unknown view of an object is matched against the correct object, a consistent interpretation is output. They assume there is only be one object per view (image). Their approach is interesting because they determine how an object is oriented (rotation parameters) before they determine where it is translation parameters).

Bolles et al. [19, 18] present a system for recognizing and locating three-dimensional parts in range data which extends previous “local-feature-focus” ideas [17]. Some of their object recognition ideas are quite different from most other researchers:

(a) They prefer to use moderately complex parts instead of polyhedra or quadric surface models because they have found that the abundance of features are helpful for object recognition. They point out that most industrial parts are moderately complex; very few ideal spheres, cylinders, and polyhedra are used as industrial parts.

(b) They also express that only very few features should be used for matching, hopefully only 2 or 3 if possible. For example, if a dihedral edge is found in range data, five degrees of freedom (2 position and 3 rotation) of that edge are determined leaving only one unknown 9the position along the edge). A preliminary planning system should do as much processing as is required up front to select the best features since this computation only needs to be done once.

The recognition process is partitioned into five steps:

(a) Primitive Features Detection: Range edges are detected and linked using

(b) 3DPO
data
part
search
Oel
dimens
maps
c
(2) Su
planar
two separate techniques; one based on discontinuities and the other based on significant second derivative zero-crossing.

(b) Feature Cluster Formation: for example, coplanar edges can be grouped together. circular arcs can be isolated among the coplanar edges.

(c) Hypothesis Generation about possible Objects and Location: For example, the system can hypothesize objects which contain the circular arc as an edge which are appropriately positioned in space.

(d) Hypothesis Verification of Best Object Hypothesis: Objects are checked to see if additional features of each object can be found in the image data or the primitive features already extracted.

(e) Parameter Refinement to Obtain More Precise Information: If additional features are predicted and found, this information can be averaged with the existing information to yield more precise locations.

3DPO uses an extended CAD model to represent objects. A volume-surface-edge-vertex model is extended via the addition of redundant pointers and other data structures to support matching. Object recognition is considered a two-part process: low-level data-driven analysis followed by high-level model-directed search. Their goal seems to be a flexible system capable of performing quick customized recognition algorithms for each object.

Oshima and Shirai [86] have also discussed object recognition using three-dimensional information. Their object recognition system is based on depth maps obtained from a light stripe rangefinder. The range data is processed as follows: (1) Points (range pixels) are grouped into planar surface elements, (2) Surface elements are merged into elementary regions that are classified as planar or curved, (3) Curved elementary regions are merged into consistent
global regions which are fitted with quadric surfaces, and (4) A scene description is generated using global region properties and their relationships with each other. The region properties are based on the best-fit planar region and its boundary and include the following quantities: perimeter, area, peround, minimum, maximum, and mean region radii about the region centroid, and the standard deviation of the radii of the boundary. The region relationships are characterized by distance between region centroids, the dihedral angle between best-fit planes, and the type of intersection curve between the regions. There is a learning process which must be executed for each view of each object that is to be recognized. The recognition process compares unknown scene data against learned scene data. Matching is restricted using a algorithmically selected kernel region that has a principal part and subordinate part. The kernel is matched against each learned scene, and each good match is processed further until a consistent scene description is generated. Two experiments were performed, one using simple objects bounded by only planar or quadric surfaces and the other using machined parts. No bad scene interpretations resulted using an empirically determined set of thresholds for the matching algorithm. The technique is worthwhile because it can handle many objects at once. However, matching will be significantly slowed down when many possible views are allowed because of the view-dependent nature of the stored scene models. This approach appears to be inadequate for single arbitrary view object recognition.

Sato and Honda [101] have investigated pseudodistance measures for recognition of objects that can be placed on a turntable in a stable vertical orientation. A fixed set of horizontal cross-section boundaries is determined for each object to be recognized using a laser projection system and image processor as descriptors are computed for each horizontal cross-section. The object representation then consists of \( N \) sets of \( M \) complex Fourier coefficients. Pseudodistan-
tance measures between two objects representations are defined for elongatedness, horizon strain, section shape, torsion, and displacement. Experimental distance results are shown in the paper for four wood animal models and a doll in three different positions. using a weighted sum of pseudodistance measures and, for example, a minimum distance classifier, unknown curved shapes can be classified. One problem with this method as currently implemented is that disjoint parts of the doll's cross-sections had to be linked manually to create a simple closed curve usable by the Fourier Descriptor algorithm. this system is totally inadequate for single arbitrary view object recognition because of its need for 360 degree view.

Faugeras [38] have devised a 3D object recognition algorithm using geometrical matching between primitive surfaces. The primitive surfaces implemented in the INRIA computer vision system are planes, but quadric surface algorithms are presented in these papers. Each geometric primitive has an associated parameter vector that determines its degree of freedom. For a plane, there are three independent direction parameters and one distance-from-the-origin parameter. Range data is processed to obtain lists of planar regions that corresponds to some object. Object models are created and stored as polyhedra. Matches between the extracted primitives and model primitives lists are hypothesized and verified using an approach which minimizes the mean square error criterion over all plane-to-plane transformation matches. Techniques are used incrementally drop and add primitives in the lists. The rotation and translation matching are decoupled into two separate independent least squares problems. Quaternions are used to convert the non-linear 3D rotation problem into a four-dimensional eigenvalue problem which can be solved directly. The translation problem permits a standard linear squares solution. The rotation and translation matching errors are combined to provide a quantitative mea-
sure of the goodness of the match between the data and a hypothetical model; the best match represents the recognized object. Local consistency tests are used to avoid a full computation on strongly inconsistent plane lists. These ideas result in a computationally efficient method of identifying objects and determining their translation and rotation parameters. Experimental results are shown in [38] for the same automobile part used by Bahlou [13] and Henderson [55]. The precision of rotation angle and translation vector results are slated to be 0.04 radians (2.3 degrees) and 3mm respectively where the accuracy of the range data itself is 1mm. These results are probably the best quoted in the literature.

Horn and Ikeuchi [64, 66, 65, 62] have discussed the use of extended gaussian images (EGI) for object recognition and object attitude determination. 3D object models can be used to compute the prototype surface normal vector orientation histograms for various shapes. depth maps or needle maps computed for real world scene data are processed to create an orientation histogram for the visible half of the Gaussian sphere for pre-segmented objects. The scene object histogram and the prototype object histograms are computed to compute the best match. For a sphere tessellated with 240 triangles, a blind search would require 720 comparison computations in general. the best match determines which object is represented by segmented data and how that object is oriented in space. The extended Gaussian image appears to be ideal for convex object recognition without occlusion because it uniquely determines convex polyhedra. However, the basic EGI is very limited for non-convex objects. Non-convex objects can be handled by creating a separate orientation histogram for every view in a discrete set of views and matching against this enlarged data structure. The EGI approach and the INRIA approach are similar in that they both use surface normal matching procedures and both fail to make explicit use face
5. Part-Based Superquadric Model Construction and Matching

adjacency information.

Besl and Jain [7] have proposed an approach to depth map object recognition that combines surface information, critical point information, and depth-discontinuity edges. No pre-decided surface shapes are used. The approach is motivated by a theorem from differential geometry which states that the coefficients of the first and second fundamental forms of a smooth surface uniquely characterize the shape of that surface. Gaussian curvature and mean curvature are isolated as important features because they combine the information of the two fundamental forms and they are invariant to rotations and translations and to changes in surface parameterization. These surface curvature characteristics generalize the notion of curvature and the boundary values of a surface uniquely determine a graph surface. The method consists of the following steps:

(a) Depth maps are first smoothed to remove noise present in the input data. This smoothed data is then interpreted as samples of an underlying surface.

(b) The smoothed image is convolved with window operators that provide least-squares estimation of the first and second partial derivatives of the underlying surface.

(c) These derivative estimates are used to determine Gaussian curvature, mean curvature, critical points, and depth-discontinuity edges. Ridge-edges are detected by high mean curvature regions. A critical points image is generated using the intersection of the zero-crossing images of the two first partial derivatives.

(d) The sign of the curvature values can be used to place every pixel in one of eight classes: pit, peak, saddle ridge, saddle valley, valley, ridge, minimal,
and flat (planar). Each critical point can be classified into one of the above classes.

(e) Critical points with positive Gaussian curvature are used as starting points for a region growing algorithm which produces a view-independent shape descriptor that can be used to match against a library of pre-computed matching representations of various objects. The depth map is segmented via the matching process. The depth-discontinuity edges are used to verify surface region segmentation.

(f) Possible matches of individual objects are projected back into the depth map format for verification. The best-match depth map surface regions are extracted when found and matching continues on the remaining depth map regions until all objects are explained.

(g) The entire scene model description is then processed by a depth buffer algorithm to create a synthetic scene depth map. Occlusion relationships are checked for correct interpretation. The system outputs the final description listing each distinguishable object, the number of occurrences of each object, and the location and orientation of each object instance. Regions of the depth map that could not be interpreted as an object are characterized and stored for further reference.

This proposed approach will capitalize on the structural scene information available in the 8-level image created by the sign bits of the surface curvature values. Experimental surface curvature results are shown in [8, 10] that indicate the robustness of the computed features.

Vemuri and Aggarwal [112, 114, 113] have addressed complete paradigm in dealing with range data has been addressed. Data is obtained from a light
strip laser imaging system which uses triangulation to compute the three space positions of each point observed in a space. A viewpoint independent region classification scheme based on mean and Gaussian curvatures with 5 types of regions: parabolic umbilic, hyperbolic, planar, and elliptic has been devised. Mean curvature is the average of the minimum and the maximum of principal curvatures at each point while the Gaussian curvature is their product. Analytic surfaces, namely tensor products of smoothing splines under tension, are fit to windows of data so that reliable derivatives can be extracted and used to compute the above properties. Jump boundaries are also computed and marked a simple depth thresholding procedure. The models are built from data directly obtained from the laser range device. Inter-view registration is accomplished by establishing correspondence between calibration patterns visible in each view and deriving the transformation required to register the views. The multiple view information was put into arrays indexed by their cylindrical coordinates. Having all surface information allowed the display of an arbitrary view of the model. The strategy for recognition is to extract lines of constant principal (max or min $\kappa$) curvature from the object and match these to lines of constant principal curvature extracted from its known model. Establishing a one point correspondence between the object's lines and model's lines allowed the object's attitude in three space to be determined. This one point was an extrema of the curves being matched. In cases of multiple choices for a given point, the ambiguity is resolved by utilizing the distance information between points of local maxima. Results were given that showed the system recovered the unknown orientation within about 5~10% for the several examples given. Matching of an unknown view of a known object to its model was performed however matching of an unknown object to a library of models was not done.

The efforts by Venuri and Aggarwal have been an attempt to develop a full
5. Part-Based Superquadric Model Construction and Matching

system encompassing data acquisition, description, modelling, and recognition of smoothly curved or planar objects. However, there are some rather severe limitations. Smoothing splines are fit to the data and the parameters of these splines are interactively specified for each set of data. The data input itself was not made into a depth map but rather kept as an array of triplets. Such a representation required extensive spline fitting making the program computationally very expensive. Although point classification was done, no intermediate level data structure beyond that of a region adjacency graph was done. Neither a high level volumetric model nor a graph nor any other symbolic model composed of regions and distinguishing features was implemented. It is not possible to reconstruct the object with their representation since it is based only on the sign of Gaussian curvature and not its magnitude. The representation scheme is also useful in recognition and localization only when the object being represented is composed of surfaces of variable mean curvature. No attempt was made at matching of an unknown object to a library of models. The coarse matching based on similar region patches proposed is promising however the finer matching of lines of curvature will be difficult.

Hoffman and Jain [58, 56, 57] have attempted a complete recognition system. Digital depth maps of with 8 bits of accuracy are obtained from a TOF laser scanner, patches are classified, images segmented and recognition attempted. Again, a differential geometry approach is taken for the classification and a rule based expert system is used for the matching. In this scheme, points are classified as being one of only three types: planar, convex or concave. This reduction was viewed necessary by the authors in order to avoid problems they were encountering in extracting reliable derivatives from noisy data. Jump boundaries and crease edges are also detected. They used nearest neighbor averages to get smoother data and identified the edges as being of three types: smooth,
jump, or roof. A clustering algorithm called CLUSTER is implemented for
the segmentation. Segments are initially built from points of similar coor-
dinate position, depth and surface normal. Then adjacent patches of similar type
(convex, flat, concave) are merged. These are then evaluated for measures of
size, elongation, etc. In [56] they reported an attempt to match their segmented
images to models and reported some success.

The system covered the entire range from acquisition through segmentation
and finally recognition. The complexity of representable objects seems to be
rather limited by using only three types of surface patch descriptors. Much of
the available information seems to be lost.

Park [87] proposed a new approach of SNI (surface normal images)-based
3D object representation and recognition. The surface normal images of an
object are defined as the projected images obtained from view angles facing
normal to each surface of the object. The recognition is based on the match-
ing of the preconstructed model which consists of SNIs, against rotated input
images (RILs). The RILs are obtained by rotating the input image into several
directions so that the surface normal vector of each scene object surface points
to the view direction, i.e. z-axis direction in the model coordinate system.
Therefore, in the proposed approach, the surface normal view direction readily
provides a straightforward reference for the correct transform between model
and scene objects saving much of the time required to obtain the correspon-
dence between model features and scene features. The difficulty involved in
obtaining a correct correspondence is significantly reduced compared to other
conventional model-based approaches. This approach also gives the advantages
of better transparency of matching and greater tolerance to measurement error,
resulting from the use of frontal view image in the recognition procedure.

But this approach is difficult to apply to curved or concave surfaces, because
5. Part-Based Superquadric Model Construction and Matching

objects processing such surfaces yield self-occluding surfaces consequently resulting in surface shape deformation. The partially occluded surfaces need to be treated in a special way by using the attributes which are not affected by occlusion. Thus SNI-based approach will be effective especially for the recognition of the objects which contain mainly planar surfaces.

5.1.2 Recognition by Parts

The first part-level models in computer vision were generalized cylinders [16]. A generalized cylinder, sometimes called also a generalized cone, is represented by a volume obtained by sweeping a two-dimensional set or volume along an arbitrary space curve. The set may vary parametrically along the curve. Different parametrizations of the above definition are possible. In general, a definition of the axis and the sweeping set are required. The axis can be represented as a function of arc length \( s \) in a fixed coordinate system \( x, y, z \)

\[
a(s) = (x(s), y(s), z(s))
\]

The sweeping set is more conveniently defined in a local coordinate system, defined at the origin of each point of the axis \( a(s) \). The sweeping set can be defined by a cross section boundary, parametrized by another parameter \( r \)

\[
\text{sweping set} = (x(r, s), y(r, s))
\]

This general definition is very powerful so that a large variety of shapes can be described with it. To limit the complexity and simplify the recovery of generalized cylinder models from images researchers often resort to restrictions. For regular objects one can use only straight axes or constant sweeping sets.
Generalized cylinders influenced much of the model-based vision research in the past two decades—vision theory as well as building of actual vision systems. Marr [80, 81, 79], for example, based his hierarchical part scheme for the description of biological forms on cylinder-like primitives. Generalized cylinders were also successful in vision applications. This work is represented by the work of Pentland and that of Brooks who advocate of the recognition by parts approach which they claim has some perceptual foundations. Brooks constructed the ACRONYM system for the identification of three dimensional objects in intensity images. Pentland has been a proponent of using a class of shapes called superquadrics to model general objects.

Brooks [21, 22], even though using intensity data as input to his system, has made significant contributions to the field of three dimensional object representation and recognition. One of the reasons is that the system can be readily re-implemented to use range data by removing the system level which derives three dimensional structure from two dimensional clues and instead starting with three dimensional information obtained from a range finder. ACRONYM was a domain independent method for modeling objects in terms of generalized cylinders. A hierarchy is set up so that classes of objects are established and then particular instantiations of them recognized. The system is frame based and modular with each cylinder having its own coordinate system and the relationship between its parts being qualitatively described. his approach in matching was to look for visual clues that indicated the presence of certain parts and then deduce possible object classes and finally possible objects using a rule-based expert system.

Recovery of generalized cylinders from images was studied by many researchers. Especially notable for his research in recovery of generalized cylinders is Nevatia [85, 96, 84, 110, 122]. In [121] recovery and segmentation of straight
5. Part-Based Superquadric Model Construction and Matching

homogeneous generalized cylinders from real intensity images is addressed. In general the methods of recovering generalized cylinders especially from intensity images seem to be overly complex since they must rely on complicated rules for grouping low level image models (i.e. edges, corners, surface normals) into models of larger granularity (i.e. symmetrical contours or cross-sections) to assemble them finally into generalized cylinders. These problems are due in part to the complicated parameterization of generalized cylinders and to the lack of a fitting function that would enable a straightforward numerical test of how well the model fits the modeled image data.

Superquadric models appeared in computer vision as an answer to some of the problems with generalized cylinders [89]. Superquadrics are solid models that can, with a fairly simple parameterization, represent a large variety of standard geometric solids as well as smooth shapes in between. This makes much more convenient for representing rounded, blob-like shaped parts, typical for objects founded by natural processes. It is interesting to note the circumstances leading to introduction of superquadrics to computer vision. The mathematical function, at least its 2D equivalent, was first described by a French mathematician, Gabriel Lamé, in the beginning of the nineteenth century. In the 1950s a Danish writer and inventor, Piet Hein, introduced superquadrics to architecture and design. In 1960s they were used for lofting in the preliminary design of aircraft fuselage. In the beginning of 1980s Alan Barr introduced them to computer graphics [4]. Finally, a few years later, Alex Pentland brought superquadrics to the attention of the computer vision community. Pentland's influence has produced a lot of papers [90, 89, 47, 50, 106, 51, 76, 75, 78, 123] using a recognition by parts approach based on superquadric surface primitives introduced by Barr [4]. Pentland shown in his SUPERSKETCH system how boolean operations on instantiations of these primitives can produce very good
models. Superquadric surface are similar to the familiar quadric surfaces except that rather than having exponents of two the exponents can take on real non-negative values. The exponent parameters control the shape of the surfaces generated and the scale parameters their size. Additional parameters can be introduced to move the superquadric in space and to induce bending, twisting or tapering. Pentland maintains that these primitives are not merely useful in terms of rendering objects but also as a basis for recognition. Basically, non-linear optimization techniques are applied to the available range data (which is not necessarily dense) to acquire estimates of the 11 superquadric parameters (2 shape, 3 size, 3 position, and 3 orientation) needed to best model that set of data. Various of goodness-of-fit functions are also used in the optimization [90].

Low level processing such as segmentation or edge detection is not addressed. The parts are hypothesized directly from the range data. The RBP approach has promise but will seemingly be plagued by the necessity of solving difficult non-linear optimization problems not guaranteed to coverage on globally optimum solutions especially when involving as many as 15-18 variables are employed. Some segmentation or point classification also seems to be needed as an intermediate step so as to avoid including points from different parts of an objects in the same optimization input set. Such a point classification might also provide some good guesses for the unknown parameters. Although proposed as a more cognitive based method, its present methods of solution are most certainly not perceptual-based.

For even more accurate modeling of shapes, different ways of introducing local deformations were proposed [92, 108, 83, 53, 30]. Local deformations which require a much larger set of parameters can model a large variety of natural and biological shapes up to a desired accuracy. One of most important features of superquadric models is their interchangeable implicit and explicit
defining functions. The explicit form is convenient for rendering, while implicit
equation is especially suited for model recovery from images and for testing of
intersection. The suitability of a particular shape model in computer vision
depends on its usefulness for recognition. It must provide a roughly unique
representation which is compact, has local support, is expressive and preserves
information. The final verdict, however, on actual usefulness of the chosen
model in computer vision depends on how easy it is to recover the model from
image data. Superquadrics are doing quite well on this test. Robust methods
exist for recovery of superquadric models from pre-segmented range data [106,
92] as well as from non-segmented range data [91, 49, 77].

Geons are primitive building blocks for representing parts which were pro-
posed by Biederman in the context of human perception [14]. Biederman for-
mulated a theory of recognition by components which can easily be detected
and differentiated on the basis of their perceptual properties in 2D images.
Geons consist of a set of solid blocks which are derived from generalized cylin-
ders and can describe the wealth of different shapes by combining them like
phonemes in a language. The set of geons consists of up to 36 primitives which
were obtained by analyzing non-accidental qualitative changes on a general-
ized cylinder such as axis shape, cross-section shape, cross-section sweeping
function, and cross-section symmetry. Biederman argued that since these dif-
ferences are qualitative and reflect non-accidentalness, people should be able
to classify them quickly and easily. Geons are in essence conceptual or meta-
models of parts which are normally implemented through generalized cylinders.
But geons can normally be expressed in terms of superquadric models. Sev-
eral methods for qualitative Geon type shape recovery based on superquadric
models were proposed [31, 94, 117].
5.2 Surface Segmentation and Description

For most vision systems, it is necessary to represent data using symbolic descriptions. Segmentation is a process of partitioning the input image into a set of regions such that each region has a certain homogeneous properties. In dense-range images, where the data points are sampled surface points, surface properties are used in the description of the data and, in turn, in achieving segmentation.

For surface description, primitives used in various computer vision systems can be basically divided into two types. One type uses simple surface primitives: plane, quadrics, and straight lines. The other type, however, prefers to use primitives of more general terms in describing the surface shape of an object. These descriptive terms are peak, pit, saddle, valley, roof, critical point, and many others. Even if the latter can handle more general and complicated surfaces, usually complex computations are involved. We are interested primarily in surface curvatures, which completely determine the local surface shape, and allow surface points to be classified into various categories, such as umbilic, saddle, valley, ridge, etc.

5.2.1 Curvature-Based Surface Description

Curvature is defined in terms of a curve in space, where a curve may be thought as a warping of a line, just as a general surface may be considered to be a warping of a plane. To extend the definition of curvature from a property of lines to a property of a surface, we need to first observe that a surface in 3D space may be viewed as a set of vectors, and we must then examine the properties of such a set. We assume that the set of surfaces to be encountered are defined by the implicit quadric equation. In the absence of measurement
noise we would expect surface data points \((x, y, z)\) to satisfy

\[
F(x, y, z) = a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5xz + a_6yz + a_7x + a_8y + a_9z + a_{10} = 0. \tag{5.1}
\]

This model can represent several different surface types including spheres, cylinders, cones, ellipsoids, paraboloids, hyperboloids, and planes [44]. We currently classify surface patches as planar, spherical, or cylindrical since we restricted consideration to man made industrial parts. Hakala et al. [54] report that 85 percent of manufactured parts are reasonably approximated by these quadric primitives. In this case, we need only work with a small set of surface shapes such as planes, cylinders, and spheres since such shapes are common in industrial parts and it is only necessary to recognize and locate a sufficient set of surface regions to uniquely distinguish the part being observed from all others that might present. These descriptions are based on linear or nonlinear fitting techniques and curvature estimations. Segmentation procedures based on qualitative interpretations of surface curvatures have been used over the last several years [12, 36, 113, 120, 59]. Principal curvatures are intrinsic quantities defined at each point \(P\) on a surface. There are two directions tangent to the surface along which the surface curves the most or least. On nondegenerate surfaces, these directions are mutually orthogonal; they are known as principal directions, and the corresponding curvatures are called principal curvatures as shown in Figure 5.1.

Differential-geometric surface description have become quite popular in the computer vision community. Many of the reported surface segmentation and description schemes based on surface curvature operate as follows [11, 37, 113, 100] First, curvature values are estimated at each pixel in the range image.
using a local-surface (a biquadric or a bicubic) fit to the neighborhood of the pixel. Then, the mean and the Gaussian curvature values are used to classify the pixel as planar, umbilic, hyperbolic, etc. In some cases, pixels are grouped based on these estimates to form connected regions of one class in the image.

Ittner and Jain [67] calculate six different curvature values at each pixel in a segment, and classify that segment using the two-sample Kolmogrov-Smirnov test. Besl and Jain [12] and Besl [6] begin by using Gaussian and mean curvature sign labelling (see Table B.1) to achieve a coarse segmentation. For each of the regions segmented, a sub-region, referred to as the seed region, is selected. This selection is based on proximity within each region (towards the center). The seed region is fitted to a surface approximated by a bivariate polynomial of up to the fourth degree. Next, in the region-growing process, all pixels in the image are assigned to the seed region currently under consideration if the following two conditions are met: (1) the difference between the original depth value and the approximating surface is less than a threshold; and (2) the difference between the surface normals computed from the raw data and the estimated normals

Figure 5.1: 3D surface geometry showing principal directions.
using approximating surface is less than a threshold. The largest connected region overlapping the current seed region is extracted to form the new seed region, which is, in turn, used to fit to a new interpolating function. This process is repeated until the sizes of the regions are approximately equal from iteration to iteration or until the fitting error exceeds a threshold. Finally, once the iteration has terminated, surface fits with high error values are rejected. The final segmentation consists of regions which are homogeneous with respect to bivariate polynomials. The authors show good results of segmentation on several real-range images.

For the each of parts extracted from the above recover-and-select segmentation procedure, the surface segmentation is accomplished by clustering image pixel coordinates and associated surface normal, followed by cleaning and merging procedures. In the description process, we attempted to characterize and identify the sub-patches of decomposed volumetric parts. We developed hierarchical surface classification procedure using hypothesis generation and verification. And for each of surface types, we extracted geometric parameters which include surface normal vector for plane’s orientation, direction of rotational axis for cylinder’s orientation, center point for sphere’s position, and radius for sphere and cylinder. Planar surfaces are characterized by containing planar umbilic points. A plane is accepted if the mean-squared-error of planar fit is below a predefined threshold. If the threshold is exceeded, however, we use the minimum curvature direction and corresponding maximum curvature value to estimate the orientation, position, and radius parameters for a hypothesized cylindrical patch. A squared-error statistics is also calculated for this hypothesis; if it exceeds a threshold, the cylindrical fit is rejected, and the patch is labeled as unknown. If, after the planar hypothesis fails, the minimum curvature is not close to zero, we attempt a spherical fit using a nonlinear
optimization algorithm, and again calculate a squared-error statistic. If the
statistic lies within a threshold, the spherical fit is accepted; otherwise, the
patch is labeled as unknown. The overall procedure of surface segmentation is
shown in Figure 5.2.

5.2.2 Planar Surface Description

We determine the surface is planar or curved by describing the method used for
fitting planes to the 3D data. Then two planarity test are introduced. The first
is a $\chi^2$ test based on assumption of normally-distributed measurement noise.
The second test is a nonparametric test based on the relative position of points
with respect to a planar fit.

The traditional method for fitting planes to a set of 3D points is linear
least squares. Unfortunately, the fitting method implicitly assumes that two
of the three coordinates are measures without error, and that these two co-
ordinates adequately support the plane being fitted. To overcome these two
problems, we used a classical multivariate method which explicitly allows us
to extract linear dependencies such as planes in a data set. The method of
principal components [68] yields a planar fit which is optimal in a least squares
sense, with the quality of the fit being independent of the embedding of the
surfaces in 3D scene. The principal components of a set of 3D points may be
obtained by extracting the eigenvalues and corresponding eigenvectors of its
sample covariance matrix $\mathbf{R}$ [68]. Let eigenvalues be denoted as $\lambda_1$, $\lambda_2$, and $\lambda_3$,
with $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and their associated eigenvectors be $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. Un-
der the hypothesis of planarity that points have been sampled from the plane
$ax + by + cz + d = 0$ contaminated by noise, the eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ span
a plane parallel to, and passing through the centroid of the data. The third
eigenvector $\mathbf{v}_3 = (v_{31}, v_{32}, v_{33})$ is perpendicular to that plane. Therefore the
Figure 5.2: Block diagram of overall surface description procedure.

coefficients of the planar equation can be estimated from the third eigenvector
and the mean vector $(\bar{x}, \bar{y}, \bar{z})$:

$$
a = v_{31}, \quad b = v_{32}, \quad c = v_{33}
\quad d = a\bar{x} + b\bar{y} + c\bar{z}
$$

This planar fit minimize the sum of squared perpendicular distances between the data points and the plane, and is independent of the coordinate frame. In this respect, this method of fitting planes is better than other methods of fitting such as least squares which require the specification of an appropriate coordinate system. A chi-square planarity test requires a formal specification of the noise which corrupts the position of the points sampled from a plane. Let $(x'_i, y'_i, z'_i)$ be a noise-free point sampled from the plane $ax'_i + by'_i + cz'_i + d = 0$.

The null hypothesis states that the corresponding observed point $(x_i, y_i, z_i)$ can be written as $x_i = x'_i + \varepsilon_x, y_i = y'_i + \varepsilon_y, z_i = z'_i + \varepsilon_z$, where $\varepsilon_x \sim \mathcal{N}(0, \sigma_x^2)$, $\varepsilon_y \sim \mathcal{N}(0, \sigma_y^2)$, $\varepsilon_z \sim \mathcal{N}(0, \sigma_z^2)$ ($\mathcal{N}(0, \sigma^2)$ denotes the univariate normal distribution with mean zero and variance $\sigma^2$). As before, let $v_3$ be the eigenvector of $\mathbf{R}$ with the smallest corresponding eigenvalue, and let $s_i$ be the score of the $i$-th data point on $v_3$, i.e., the length of its projection on $v_3$:

$$
s_i = v_{31}x_i + v_{32}y_i + v_{33}z_i
$$

Under $H_0$,

$$
s_i \sim \mathcal{N}(0, (v_{31}\sigma_x)^2 + (v_{32}\sigma_y)^2 + (v_{33}\sigma_z)^2).
$$
Therefore, we can define the static $T$ as

$$ T = \sum_{i=1}^{n} \frac{s_i^2}{(v_{31} \sigma_x)^2 + (v_{32} \sigma_y)^2 + (v_{33} \sigma_z)^2} $$

where $n$ is the number of points on the surface. Under $H_0$ the statistic $T$ has a $\chi^2$ distribution with $n - 1$ degrees of freedom, and tests for planarity can be constructed for a given significance level by choosing the appropriate critical value from a table of distribution. The number of degrees of freedom needs to be adjusted if the noise variance in $x$, $y$, and $z$ are estimated from the data.

When the assumption of normally distributed noise is not valid, we have to consider the nonparametric tests. It is assumed that noisy points sampled from a plane are equally likely to lie on either side of the fitted plane. We introduced a multivariate generalization of the Wald-Wolfowitz runs test developed by Friedman and Rafsky [41], and investigated by Smith and Jain [104] to test for the uniformity of multidimensional data. The test procedure is as follows.

The $n$ points are projected onto the best-fitting plane obtained by the principal component analysis. A binary label is assigned to each point so that all points on one side of the projection plane have the same label. A minimal spanning tree (MST) of the $n$ projected points is constructed and then the number of edges in the MST which link points with different labels is counted. Let this number be $T$, and let $C$ be the number of edge pairs in the MST sharing a node. Friedman and Rafsky showed that the conditioned permutation distribution of $T$ is asymptotically normal, with mean

$$ E[T] = \frac{n}{2} $$

Note that asymptotic in a planar test of plane. Two of planar edges to that plane is strong can have of the hypothesis he abandon is reject comparison be mad.
\[ V_{\text{ar}}[T | C] = \frac{n}{2(n - 1)} \left\{ \frac{n}{2} - 1 - \frac{C - n + 2}{n - 3} \right\} \]

Note that the distribution of \( T \) is conditioned upon \( C \). The ratio of the sizes of the two subsets of data must be bounded away from 0 and infinity for this asymptotic result to hold. Intuitively, we can predict the behavior of this statistic in planar and nonplanar situations by considering any two points in the planar projection of the data which are connected by an MST edge. In the planar case, each point is equally likely to be on both sides of the fitted plane. Two of the four possible configurations have the edge crossing the dividing plane. Therefore, the expected value of this statistic under the null hypothesis of planarity is indeed \( n/2 \). In a nonplanar case, we would only expect MST edges to intersect the dividing plane near where the curved surface intersects that plane. A planar indication from both \( \chi^2 \) test and the MST-based test is strong evidence that the surface is truly planar. A rejection by the \( \chi^2 \) test can have several interpretations: an incorrect noise model, incorrect estimates of the noise variance, or nonplanarity in the data. A rejection of the planar hypothesis by the MST-based test indicates that the null hypothesis if spatially homogeneous labeling is not correct. For these reasons, we choose not to abandon the possibility of planarity even when the null hypothesis of planarity is rejected. The best planar approximation to the sample is retained, and a comparison between the planar fit and a subsequent curved surface fit should be made.
5. Part-Based Superquadric Model Construction and Matching

5.2.3 Curved Surface Description

We outline the method for the classification of quadrics into spheres, cylinders, along with the estimation of their geometric parameters. Classification is based on curvature measure, which measure the rate of change of surface orientation as the surface is traversed in a particular direction. These curvature measures describe surface’s local behavior completely, and are independent of the embedding of the surface in 3D.

There are number of different ways to estimate curvature values for a set of 3D points. They are based on different assumptions about measurement error and sampling pattern, and could well give different results for similar input data.

Given a point on a 3D surface and its k nearest neighbors, our first step in estimating curvature is to find an adequate support plane for a cubic surface fit. The principal component fit is used again for this case, but only the point and the k neighbors are used in the fit.

Let \((u, v, w)\) be a local coordinate system defined by the principal components of the point set. The coordinate \(u\) and \(v\) span the locally-fit plane, and \(w\) is the distance from that plane. We fit a bicubic surface to the \(w\) value:

\[
w = f(u, v) = c_1 u^3 + c_2 u^2 v + c_3 uv^2 + c_4 v^3 + c_5 u^2 + c_6 uv + c_7 v^2 + c_8 u + c_9 v + c_{10} \tag{5.3}
\]

To reduce the influence of the noise, we use 20 neighbors \((k = 20)\) and use a least-squares solution to the equation above. It would be desirable to adjust the neighborhood size used in the fitting so that a relatively constant-area region in \((u, v)\) coordinates is employed. This would make estimated curvatures less sensitive to the sampling density on the primitive surface. Once the bicubic fit
has been obtained, curvature values are extracted using partial derivatives of the surface equation. First, let \((u_0, v_0, w_0)\) be the coordinates of the surface point at which curvature is being estimated. Principal (minimum and maximum) curvatures and their associated directions are the eigenvalues and corresponding eigenvectors of the Hessian matrix

\[
\begin{pmatrix}
\frac{\partial^2 f}{\partial u^2} & \frac{\partial^2 f}{\partial u \partial v} \\
\frac{\partial^2 f}{\partial u \partial v} & \frac{\partial^2 f}{\partial v^2}
\end{pmatrix}
\]

(5.4)

evaluated at \((u_0, v_0)\). Let \(k_{\text{min}}\) and \(k_{\text{max}}\) denote the minimum and maximum curvature, respectively. Many curvature measures can be defined in terms of \(k_{\text{min}}\) and \(k_{\text{max}}\) [67].

On a cylindrical surface, for example, minimum curvature is zero, and the corresponding direction is parallel to the cylinder's rotational axis. The maximum curvature direction is perpendicular to the axis. The radius can be approximated by the inverse of maximum curvature. The average of this quantity (over all surface points) is used as the estimate of the cylinder's radius. Since the zero-curvature direction vectors are parallel to the axis, we can obtain orientation angles directly. Also, we can extract a point on the axis by projecting the surface points into the cylinder a distance equal to the radius estimate. They should lie roughly on a line, as well their centroid. The centroid is used as a point fixing the position of the cylinder's axis in space.

Curvature directions on a spherical surface are meaningless, and minimum and maximum curvatures are equal and nonzero. A good estimate of the radius of the sphere is the average (over all surface points) of the inverse of mean curvature. Projecting each surface point into the sphere along its surface normal by a distance equal to the estimated radius produces a swarm of points is used as the estimate of the center.
The sign of these curvatures depend on our choice of coordinates. A negative principal curvature implies that the surface is curving downward with respect to the coordinate frame; conversely, a positive curvature value means that the surface is curving upward in the associated direction.

The initial parameter estimates of spherical and cylindrical surface patches are often quite noisy, due to measurement error and poor-fitting models for local surface characteristics. We use an iterative optimization technique to improve the quality of these parameter estimates. The well-known Levenberg-Marquardt nonlinear least squares algorithm is used to find the best-fitting quadric surface of the specified type [102, 93]. The algorithm minimizes an objective function $F^2$, where $F$ is an implicit quadric equation specific to each surface type. It should be noted that not all of these parameters will be needed in all situations: if, for example, the radii of the spheres in the scene were known a priori, the dimension of the parameter space for spheres could be reduced by one, and the fitting could take place with respect to location only.

For spheres, the geometric parameters are the position $(x_0, y_0, z_0)$ of the center and the radius $r$:

$$F(x, y, z; x_0, y_0, z_0) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2$$  \hspace{1cm} (5.5)

A cylinder has five parameters: two angles $\theta$ and $\phi$ specifying the relative orientation of the axis, the radius $r$, and two of the three coordinates of any point
on the principal axis, \((x_0, y_0)\), from a point \((x_0, y_0, z_0)\). The quadric equation is:

\[
F(x, y, z; r, \theta, \phi, x_0, y_0) = \\
\{x \cos \phi + \sin \phi(y \sin \theta + z \cos \theta) - x_0\}^2 + (y \cos \theta - z \sin \theta - y_0)^2 - r^2
\]

(5.6)

5.2.4 Surface Equations, Normals, and Geometric Parameters

In this section, we will use the surface normals to derive equations of planes, cylinders, and spheres. Let \(N\) be a unit normal vector from a point of a surface.

\[
N = \begin{bmatrix}
N_x \\
N_y \\
N_z
\end{bmatrix}
\]

(5.7)

(a) Plane:

A plane is a set of points that satisfies \(N \cdot x = d\), where \(x\) is a point on a surface, and \(d\) is a constant. Two planes with the same normals and different \(d\) can only be connected by a jump boundary if the distance is not close.

(b) Sphere:

A spherical surface can be determined by four parameters: radius, and the center of sphere which has three parameters. As we can see from the Figure 5.3, the relationships between the radius \(r\), the normal \(N\), the center \(c\) can be represented by the following equation.

\[
(x - c)^t(x - c) = r^2
\]

(5.8)
where, \( c = (x_0, y_0, z_0)^t \). If we expand it, we have

\[
(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0
\]

\[
f(x, y, z) = x^2 + y^2 + z^2 - 2(x x_0 + y y_0 + z z_0) + x_0^2 + y_0^2 + z_0^2 - r^2
\]

Since, \( N = \frac{\nabla f}{\|\nabla f\|} \), where \( \nabla \equiv \left[ \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right]^t \), we have,

\[
N = \frac{1}{r} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}
\]

or

\[
rN - x + c = 0 \quad (5.9)
\]

(c) Cylinder:

For the moment, assume that the axis passes the origin of the coordinate
system. Then the equation of the surface can be represented as follows shown in Figure 5.4,

\[
\begin{align*}
\| x - (x \cdot u)u \|^2 &= r^2 \\
(x - (x \cdot u)u)(x - (x \cdot u)u) &= x^t x - x^t u - u^t x \\
&= x^t (I - uu^t) x \\
&= r^2 \\
f(x) &= x^t (I - uu^t) x - r^2
\end{align*}
\]

where, \( I \) is a 3 by 3 identity matrix and \( u \) is a unit directional vector, \((u_1, u_2, u_3)^t\), of the axis of the cylinder. And,

\[
f(x, y, z) = (1 - u_1^2)x^2 + (1 - u_2^2)y^2 + (1 - u_3^2)z^2 - 2u_1u_2xy - 2u_1u_3xz - 2u_2u_3yz - r^2 = 0
\]
The normal at each point of the cylindrical surface is,

\[ N = \frac{\nabla f}{\|\nabla f\|} = \frac{1}{r} \begin{pmatrix} (1 - u_1^2)x - u_1u_2y - u_1u_3z \\ -u_1u_2x + (1 - u_2^2)y - u_2u_3z \\ -u_1u_3x - u_2u_3y + (1 - u_3^2)z \end{pmatrix} \quad (5.10) \]

or, in vector representation,

\[ rN = (I - uu^t)x \]

If the axis of the cylinder does not pass the origin of the coordinate system, there is a translation vector \( x_0 \). In this case, \( x \) is replaced by \( x - x_0 \) in the above equations.

5.2.5 Surface Relationship between Two Regions

Surface relationship value between two regions will be used to resolve the candidate regions if multiple region pairs are found as candidates for matching an image region pair. The relations are invariant under rotation and translation in 3D space.

(a) Plane - Plane relation:

We denote a planar region by \( (n, d) \), where \( n \) is surface normal, and \( d \) is the distance of a plane from the origin as shown in Figure 5.5. For the relationship between two planes \( P_1 = (n_1, d_1) \) and \( P_2 = (n_2, d_2) \), we use the inner product of \( n_1 \) and \( n_2 \). After transformation of planes become \( P'_1 = (n'_1, d'_1) \), and \( P'_2 = (n'_2, d'_2) \), where \( n'_1 = Rn_1, \quad d'_1 = n'_1 \cdot +d_1 = Rn_1 \cdot t + d_1 \), and \( n'_2 = Rn_2, \quad d'_2 = n'_2 \cdot +d_2 = Rn_2 \cdot t + d_2 \) for rotation \( R \)
Figure 5.5: Plane-plane binary surface relationship.

and translation \( t \). The value of linear product after transformation is,

\[
\mathbf{n}'_1 \cdot \mathbf{n}'_2 = (\mathbf{Rn}_1) \cdot (\mathbf{Rn}_2)
\]

\[
= (\mathbf{Rn}_1)^t(\mathbf{Rn}_2)
\]

\[
= \mathbf{n}'_1^t \mathbf{R}^t \mathbf{Rn}_2
\]

\[
= \mathbf{n}'_1 \mathbf{n}_2
\]

\[
= \mathbf{n}_1 \cdot \mathbf{n}_2
\]

(b) Plane - Cylinder relation:

In this case, the relationship value is determined by the inner product of the normal of the plane with the direction of the axis as shown in Figure 5.6. If the direction of the axis is \( \mathbf{a} \), then \(-\mathbf{a}\) is also a valid direction. Hence we use the magnitude of the inner product.
(b) Plane - Sphere relation:

Let the center of the sphere is \( c \), and the plane be \((n,d)\) as shown in Figure 5.7. We use the distance between the plane and the center of sphere. Projection of \( c \) to the plane is found by \( c_p = (d - n \cdot c)n \). And the distance between the plane and the center of the sphere is distance
between \( c \) and \( c_p \), and is given as \( d - n \cdot c \). After transformation,

\[
\begin{align*}
    c' - c'_p &= (d' - n' \cdot c')n' \\
    &= (Rn \cdot \mathbf{t} + d - (Rn) \cdot (Rc + \mathbf{t}))Rn \\
    &= (Rn \cdot \mathbf{t} + d - (Rn) \cdot (Rc) - (Rn) \cdot \mathbf{t})Rn \\
    &= (d - n \cdot c)Rn
\end{align*}
\]

\[
|c' - c'_p| = ||(d - n \cdot c)Rn|| = d - n \cdot c = |c - c_p|
\]

Figure 5.7: Plane-sphere binary surface relationship.

(c) Cylinder - Cylinder relation:

The relationship is defined by the magnitude of the inner product of the direction of the axis as shown in Figure 5.8.
(d) Cylinder - Sphere relation:
The relationship is defined by the distance from the center of the sphere to the axis of the cylinder. If the cylinder is defined by \((a, x)\) where \(a\) is the direction of the axis and \(x\) is a point on the axis as the following Figure, the distance is given by

\[
d = |(c - x) - ((c - x) \cdot a)a|
\]

(d) Sphere - Sphere relation:
The relationship is defined by the distance between the centers of the spheres as shown in Figure 5.10.
5.3 Hierarchical Model Construction

We introduce hierarchical model description method for 3D object recognition and localization by using combined volumetric and surface description of 3D objects. Superquadric parts models extracted from volumetric description procedure by way of recover-and-select segmentation algorithm represent the global shape description of each part respectively. On the other hand, surface level description of each part through the cleaning and merging procedure im-
plies the local geometric features which could not be detected in the previous volumetric description level. Once good descriptions are extracted from the range image, we need to define a recognition engine to manipulate them. Each new object needs to be stored into the database in such a way that it can be retrieved efficiently when a candidate description is proposed for recognition. Considering efficient model-based recognition system, we have decided to construct the database hierarchically, which combines volumetric and surface level descriptions together, which is shown in Figure 5.11.
Figure 5.11: Hierarchical model construction.
5.3.1 Volumes vs. Surface

The two main representations for capturing the structured nature of an object are surfaces and volumes. Volumetric representations provide the correct level of abstraction for viewpoint tolerant descriptions. In fact, volumes lead themselves more naturally than surfaces to abstract an object’s shape. Surface attributes can be easily determined from a volumetric description, whereas the reverse is not direct. The differences between two objects’ shape may not be directly accessible at a symbolic level when using surface-based description. In fact, surface-based representations of otherwise using similar objects can be drastically different. A volumetric representation provides means to analyze shape similarities (or differences) by using geometric parameters. By decomposing its representation into a set of surface attributes, it becomes easier to compare objects and decide in which ways they are similar or different. Con-

<table>
<thead>
<tr>
<th>object</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$p_x$</th>
<th>$p_y$</th>
<th>$p_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cylinder</td>
<td>44.8</td>
<td>44.2</td>
<td>80.0</td>
<td>120.0</td>
<td>120.1</td>
<td>120.7</td>
</tr>
<tr>
<td>pentagon</td>
<td>47.2</td>
<td>47.2</td>
<td>119.6</td>
<td>120.4</td>
<td>123.2</td>
<td>107.6</td>
</tr>
</tbody>
</table>

Table 5.1: Estimated superquadric parameters.

<table>
<thead>
<tr>
<th>object</th>
<th>$\varepsilon_1$</th>
<th>$\varepsilon_2$</th>
<th>$\phi$</th>
<th>$\theta$</th>
<th>$\psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cylinder</td>
<td>0.1</td>
<td>1.0</td>
<td>4.4</td>
<td>-0.7</td>
<td>1.5</td>
</tr>
<tr>
<td>pentagon</td>
<td>0.1</td>
<td>0.4</td>
<td>4.3</td>
<td>-0.4</td>
<td>2.3</td>
</tr>
</tbody>
</table>

sider the two objects of Figure 5.12 despite their surface differences, we are still able to say that they have a similar structures as shown in Figures 5.12(b), 5.12(e) due to the volumetric parts composing of them. Furthermore, the corresponding parts of the objects have quite different surface representations as shown in Figure ??, a cylindrical and a piecewise polyhedral one for the pentagon prism, for example. But we are still able to judge their similarities(sizes
Figure 5.12: Comparison between the volumetric and the surface description.

along the parts axes and cross-sections), and differences (circular vs. polyhedral cross-sections). This type of analysis is very desirable for recognition and even learning. In this case, the representation is symbolic and discriminative.

5.3.2 Volumetric Model Construction

Assuming 3D object is consists of finite number of parts and each part is composed of several smooth surface patches, we also can easily extract global volumetric information by using recover-and-select paradigm. We defines the volumetric information as parts shape, parts size, and volume. If the object has subparts we also defines parts junction distance and parts junction angle. Among these features, parts shape, volume, and parts junction angle have
property of viewpoint independent characteristics. Parts shape, parts size and volume features can be directly obtained from the extracted superquadric parameters during recover-and-select procedure. In two adjacent parts, the parts junction distance is the Euclidean distance between two neighboring parts’ center of gravity. We compute the parts junction angle, which is an inner-product between the junction unit vector and the principal axis of the biggest volume part. Because the biggest part is less affected by self-occlusion according to the viewpoint, we decide the principal axis as the biggest part’s z axis.

5.3.3 Surface Model Construction

In surface description level, we define useful features as surface type and surface junction relationship. Currently we can obviously identify the surface type into the following categories such as planar, cylindrical, spherical and unknown surfaces. And for each of surface types, we extract geometric parameters which include surface normal vectors for plane’s orientation, direction of rotational axis for cylinder’s orientation, center point for sphere’s position, and radius for sphere and cylinder. Surface types and radius for sphere and cylinder are categorized into unary properties that do not change under rigid motion in 3D space. Therefore, the unary properties are used in the first step in the surface matching process. There may be several candidate regions for a region of an image with the same unary properties. To reduce this ambiguity, not only the properties of a region but also the relationships between regions must be checked. We have to check surface junction relationship as a binary property. This value is measured quantitatively such as inner product of surface normals in planar-planar relationships, the shortest distance from a plane to the center of a sphere in planar-sphere relationships, inner product of the normal of the plan with the direction of the cylinder axis in planar-cylinder relationships,
magnitude of inner product between two axes in cylinder-cylinder relationships, the distance from the center of the sphere to the axis of the cylinder in cylinder-sphere relationships, and the distance between two centers of spheres in sphere-sphere relationships. Figure 5.13 shows the overall matching features of the hierarchical model.
Figure 5.13: Matching features of the hierarchical model.
5.4 Algorithm for the Part-Based Matching and the Surface-Based Matching

After volumetric description and surface description, the features and characteristics of surface regions should be compared to those of stored models to establish matching between the image and models. This object recognition system employing superquadric models as a primary representation reduces the reduction of computation for matching. One reason is that the number of candidates is greatly reduced since a set of volumetric parts is a unit of matching. Superquadrics alone have been used to recognize a large class of objects and have even been used to build composite parts (unions of several superquadrics), but they lack the descriptive ability to effectively capture local surface shape changes even if they have the same shape parameters. We divided the matching procedure into two hierarchical stages such as part-based matching in volume layer and surface-based matching in surface layer. For the purpose of recognition, we should find the best matching between model object and input scene object in both two layers. In volume layer, we can get part-based superquadric models for each of decomposed volumetric parts. The number of part-based superquadric models in a scene object is greater than those of model objects in a modelbase. And also, we defined part similarity using shape parameters and volume similarity measures as a kind of filtering to select the best candidates among model objects. It is efficient to retrieve a model object as a part-based matching. This part similarity of each part means the Euclidean
distance between shape parameters and volumes,

\[ S_{\text{shape}} = w_s \sqrt{ \left( S(P_i^1) - M(P_m^1) \right)^2 + (S(P_i^2) - M(P_m^2))^2 } + w_u \sqrt{ \left( S(P_i^1) - M(P_m^1) \right)^2 + (S(P_i^2) - M(P_m^2))^2 } + w_o \sqrt{ \left( S(P_i^1) - M(P_m^1) \right)^2 + (S(P_i^2) - M(P_m^2))^2 } \]

(5.11)

where \( S(P_i^X) \) and \( M(P_m^X) \) are the i-th part parameters for \( X \) extracted from the scene object, and the m-th part parameters for \( X \) from the model object. The superscript \( X \) can be replaced with shape parameters (\( \varepsilon_1 \) and \( \varepsilon_2 \)), size (\( a_1 \), \( a_2 \), and \( a_3 \)), and orientation (\( \phi \), \( \theta \), and \( \psi \)). Also \( w_s \), \( w_u \), and \( w_o \) indicate the weighing factors for shape parameters, size, and orientation respectively, which can be determined experimentally. If the scene object is composed of multiple parts, then we should check parts junction similarity separately between two neighboring parts,

\[ S_{\text{junction}} = w_d \sqrt{ (S(J_{ij}^d) - M(J_{mn}^d))^2 + w_o \sqrt{ (S(J_{ij}^o) - M(J_{mn}^o))^2 } } \]

(5.12)

where \( S(J_{ij}^d) \) and \( S(J_{ij}^o) \) are respectively parts junction distance and junction angle between the i-th part and the j-th of the scene object and \( M(J_{mn}^d) \) and \( M(J_{mn}^o) \) are respectively parts junction distance and junction angle between the m-th part and the n-th of the model object. Also \( w_d \) and \( w_o \) define the weighing factors for junction distance and junction angle respectively. So the superquadric part models which have the similarity value less than a given threshold are selected as matching candidates in volume layer.

In the surface-based matching procedure, we analyze surface characteristics of different kinds of candidate objects with similar shape and junction relationships above in the part-based matching. This results in the final matching between the scene object and the model. Each decomposed volume part can
have only one or more surface regions depends on segmentation parameters used in the surface description procedure. We use unary and binary properties as matching features in the surface-based matching procedure. A unary property is a feature of a surface region that does not change under rigid motion in 3D space. The obvious unary property is surface types such as planar, cylindrical, and spherical surface. Another unary property is a parameter such as radius of a cylindrical surface or a spherical surface. These unary properties are used in the first step in the surface-based matching process. But there may be several candidate surface regions for a region of an image with the same unary properties. To solve this ambiguity problem, not only the unary property but also the binary property such as region relationship between two regions should be matched. This value is measured quantitatively such as inner product of surface normals in planar-planar relationships, the shortest distance from a plane to the center of a sphere in planar-sphere relationships, inner product of the normal of the plan with the direction of the cylinder axis in planar-cylinder relationships, magnitude of inner product between two axes in cylinder-cylinder relationships, the distance from the center of the sphere to the axis of the cylinder in cylinder-sphere relationships, and the distance between two centers of spheres in sphere-sphere relationships.

5.5 Assembly Parts Recognition based on part-based superquadric model (PBSM)

In general, three major steps are required to establish a 3D model-based object recognition system using superquadrics as object models: database building, scene object model recovery and model matching. First, we need to construct a superquadric database for efficient access of a large number of object models.
Then, the model of the target object is accurately recovered from the range sensor data. Most of conventional researches using superquadrics are mostly concentrated on 3D shape recovery and segmentation of range images through superquadric parameter estimation [50, 106, 89]. After these two steps have been completed, the last step of the system is to correctly and efficiently match a recovered superquadric with a set of superquadrics in the model database. In spite of many advantages, we confront difficulties in recognizing 3D object by using superquadrics alone. This paper summarizes these reasons as following. At first, superquadrics cannot provide a uniform representation of 3D object, then it is too difficult to build model base constantly. Secondly, volumetric representation is a global model and is insensitive to local variations which has no surface information: surface type and relationship. The use of additional surface information is advantageous for a precise recognition.

This section presents an integrated recognition scheme by using both surface and volumetric information. In our algorithm, segmentation provides surface information of the scene object, and we can achieve part decomposition. In the next step, we extract superquadric parameters and the matching features including the number of volumetric primitives and the junction relationships between them. In addition, our method calculates junction vectors between neighboring primitives, which indicate a distance and direction between the centroids of primitives. In the final step, we match the feature values of the scene object with those of the model objects. Figure 5.14 shows our recognition scheme. This section shows that part-based superquadric model (PBSM) can be used as an effective approach to 3D recognition. Since superquadrics is one of volumetric model representations, our method is very similar to human perception, so robust to non-linear shape changes according to viewpoint. Furthermore, the use of both junction vectors between neighbors and surface
adjacency graph (SAG) can solve self-occlusion problem. Experimental results on synthetic and real objects demonstrate the good performance of our proposed method.

![Diagram](image-url)

Figure 5.14: The outline of recognition scheme.

5.5.1 Surface Segmentation and Volumetric Decomposition

In general, the simple 3D assembly part objects can be regarded as the combinations of several volumetric primitives such as sphere, cylinder, block, etc.
The object represented by its volumetrics is sufficiently distinguished from other classes of objects, and can provide an efficient indexing mechanism for recognition systems. We decompose 3D input range data into volumetric primitives, then construct part-based superquadric model (PBSM) through superquadric model recovery. In the first step, the 3D input image is segmented into smooth surface patches by using curvature-based surface segmentation; the Gaussian and the mean curvature. The segmentation is achieved through a hybrid approach of the edge detection and the region growing procedure as presented in references [7, 36, 57]. Discontinuity of depth and orientation is detected for the edge detection. Then, for the segmentation of internal regions surrounded by edges, mean curvatures $H$ and Gaussian curvature $K$ are used. Merging of small regions into neighboring ones is needed, because curvatures $H$ and $K$ are very much sensitive to noise. The final result of the segmented image is shown in Figure 5.15(b). Then, we construct surface adjacency graph (SAG) that represents the relationships of neighboring surfaces, by using the obtained surface information such as the types of the surfaces and the junctions. In SAG, a node indicates the each surface type and a link describes the junction type between two surfaces as shown in Figure 5.15(c). The surface information obtained in segmentation process is useful for an accurate volumetric decomposition and object recognition. Surface segmentation provides the type of the surface and the junction between adjacent surfaces, which are useful for volumetric decomposition. Since self-occlusion can cause a deformation of SAG, the only surface-based approach has a limitation for 3D recognition. When there exist any concave junction and jump edge in SAG, we can split SAG into sub-SAG for an accurate volumetric decomposition. By using the segmented surface patches and SAG of the 3D input object, we can construct superquadric part model. We use block, cylinder, and sphere as volumetric primitives of superquadric part model.
5. Part-Based Superquadric Model Construction and Matching

(a) Range image.  
(b) Surface segmented image.

(c) SAG and volume decomposition.

Figure 5.15: Surface segmentation and volumetric decomposition.

model. A block primitive is consisted of six sides of planar surfaces. However, the number of visible surfaces is three or two if occluded, according to viewpoints. A cylinder has two planes and a cylindrical surface, but the number of visible surfaces of the cylinder is always two. A sphere has only one visible surface regardless of the viewpoint. When there is any concave junction in SAG, the concave junction should be broken to rebuild volumetric primitives, due to our assumption. Therefore a concave junction in Figure 5.15(c) is a reference to be broken and it divides object into two volumetric primitives of a cylinder and a block. In this way, 3D object can be decomposed into finite sets of volumetric primitives for superquadric part models.
5. Part-Based Superquadric Model Construction and Matching

5.5.2 Construction of PBSM and Superquadric Part Model Recovery

For the model object, we estimate superquadric parameters of each volumetric primitive. Eq. (5.13) shows a set of \( N \) representative shapes primitives (RSP). Figure illustrates the results of superquadric parameter extraction for each part of the model object.

\[
MO_1 = RSP_{SQMO_1}, SQMO_2, \ldots, SQMO_{N-1}, SQMO_N
\]  

(5.13)

In the case of composite object, surface information is needed to decompose

![Diagram](image)

Figure 5.16: Construction of model object’s PBSM.

the object into several volumetric parts. Eq. (5.14) indicates the set of \( M \) RSPs.

\[
SO_1 = RSP_{SQSO_1}, SQSO_2, \ldots, SQSO_{M-1}, SQSO_M
\]  

(5.14)

Figure implies the surface and volumetric segmentation and superquadric
model recovery for the scene object SO. For each recovered cylinder, we obtain superquadric parameters and eigenvectors according to the order of volume size.

5.5.3 Algorithm for Object Recognition using PBSM

Both the model object and the scene are represented by using superquadrics. At first, we compare the number of parts of the model with that of the scene. Then if the object is considered to be single, we need the matching procedure of shape and size parameters between the model object and the scene sequentially. In the case of the scene object consisting of multiple parts, we examine the shape parameters of each part. And then, according to the existence of deformation with SAG, we can detect self-occlusions and shape variations. When self-occlusions and shape variations are occurred, we match the orientation parameters of each primitive along z axis, the longest axis. If the object does not contain self-occlusion, we examine the size parameters and junction relation, which represents junction distance and angle. In two adjacent parts, the junction distance is the Euclidean distance between two kinds of center of gravity.
such as Figure 5.18(a). Furthermore, we compute the junction angle, which is an inner-product between the junction unit vector \( \mathbf{u}_1 \) and the principal axis \( \mathbf{u}_2 \) of the biggest part in Figure 5.18(b). Because the biggest part is less affected by self-occlusion according to the viewpoint, we decide the principal axis as the biggest part's \( z \) axis. Figure 5.19 shows the proposed matching procedure above mentioned.

![Diagram](image)

(a)

![Diagram](image)

(b)

Figure 5.18: Definition of the junction distance (a) and the junction angle (b).
Figure 5.19: The proposed matching procedure.
Chapter 6

Experiments

We tested the proposed hybrid 3D object representation method on many of synthetic and real range images. Most of objects were composed of parts that can be modelled well by superquadrics. We generated several synthetic test range images whose size is $240 \times 240$ as shown in Figure 6.1. For the comparison, we used synthetic and real range images in MSU 3D model database [24]. The real range images and intensity images used in this experiments were captured through the RANGER system [28], which is one of 1D laser line scanners using laser triangulation such as sheet-of-light ranging. These captured real range images have some noise and unwanted background due to the table where the objects are placed during scanning. It is helpful to remove these background pixels for further processing. To reduce the noise level and eliminate speckle noise without destroying edge information. We apply the nearest neighborhood smoothing technique. For each pixel, the 5 neighbors in its $3 \times 3$ neighborhood having grey values closest to that of the center pixel is assigned as the average grey value over these 5 neighbors. Also, We used both Gaussian smoothing and median filtering to remove random noise. Figure 6.2 illustrates noise reduction
Figure 6.1: The model objects used in the experiments.
experiment for the MO2 model object which has been added by Gaussian noise of $\sigma=1.5$ using median filtering.

And we assumed that the background has homogeneous characteristics. Once the image has been smoothed, background pixels can be eliminated. To do this, we compare, pixel by pixel, the smoothed range image with a smoothed reference range image (a range image of the table alone) and call the pixel a background pixel if the difference of the two depth values is less than 5 grey values as shown in Figure 6.3.

Figure 6.1 and Figure 6.4 show 12 synthetic range images of the model objects that are mostly constructed with spheres, cylinders and blocks, and their recovered PBSMs, respectively. From these estimated superquadric position parameters, we obtain the junction vectors between parts. In addition, we already knew surface information of model objects, so each model object is described with SAG. For the purpose of recognition, we segment the 3D input object into smooth surface patches by using surface curvature information. In the next, it is followed by volumetric segmentation to decompose the scene object into PBSM. Our model base has surface information (SAG) as well as PBSM. The superquadric parameters and junction relationships of the volumetric parts are used in matching procedure.

The experimental results for the part segmentation and recovery of PBSM are shown in Figure 6.5. Both SO1 and SO3, which are rotated with respect to the viewing direction, should be matched to MO1 in the model base. In particular, non-linear shape changes occurred in SO3 according to the viewpoint. Table 6.1 shows the estimated superquadric parameters of PBSM for both the model objects and the scenes, respectively. The size parameters ($a_1$, $a_2$ and $a_3$) of SO1 and SO2 have little differences with those of SO1 and SO3. Since part 2 is not occluded and larger than part 1, the shape parameters and the size of
6. Experiments

(a) Range image with Gaussian noise.

(b) Surface normal image.

(c) 3D surface plot of (a).
Figure 6.2: Experiment for the noise reduction using median filtering.
Figure 6.3: Experiment for real range image captured from RANGER system.
Figure 6.4: The PBSM of the model objects.
part 1 have much more errors than those of part 2, in every scene objects. In the next step, we compare their junction relationships. Table 2 shows matching parameters of junction relationship: junction distance and angle. Because of self-occlusion occurred by non-linear shape change, we have noticed that SO2 has much different junction properties from those of other scene objects (SO1 and SO3) in Table 2. Our matching module finally determines that among these three inputs, the only SO2 is recognized as a different object. These results show that our method is effective and robust in spite of a little self-occlusion.

Figure 6.5: The scene objects to be matched with the model object 1.
Figure 6.6: PBSM of the scene object 1 (SO1) to be matched with the model object 1 (MO1).
Figure 6.7: PBMS of the scene object 2 (SO2) to be matched with the model object 1 (MO1).
Figure 6.8: PBSM of the scene object 3 (SO3) to be matched with the model object 1 (MO1).
6. Experiments

Table 6.1: Results of superquadric parameter matching.

<table>
<thead>
<tr>
<th>object</th>
<th>MO1</th>
<th>SO1</th>
<th>SO2</th>
<th>SO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>part</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$\varepsilon_1$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>$\varepsilon_2$</td>
<td>1.0</td>
<td>1.0</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>$a_1$</td>
<td>14.5</td>
<td>39.3</td>
<td>15.4</td>
<td>39.3</td>
</tr>
<tr>
<td>$a_2$</td>
<td>13.3</td>
<td>37.6</td>
<td>15.3</td>
<td>37.4</td>
</tr>
<tr>
<td>$a_3$</td>
<td>36.9</td>
<td>37.6</td>
<td>37.2</td>
<td>37.6</td>
</tr>
<tr>
<td>$p_x$</td>
<td>99.9</td>
<td>99.9</td>
<td>116.0</td>
<td>88.8</td>
</tr>
<tr>
<td>$p_y$</td>
<td>59.8</td>
<td>132.3</td>
<td>60.4</td>
<td>130.3</td>
</tr>
<tr>
<td>$p_z$</td>
<td>115.4</td>
<td>90.1</td>
<td>107.5</td>
<td>91.0</td>
</tr>
<tr>
<td>$\phi$</td>
<td>4.7</td>
<td>4.7</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>$\theta$</td>
<td>-1.9</td>
<td>-1.9</td>
<td>-1.9</td>
<td>-1.9</td>
</tr>
<tr>
<td>$\psi$</td>
<td>1.5</td>
<td>1.5</td>
<td>1.2</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 6.2: Matching parameters of junction relationships.

<table>
<thead>
<tr>
<th>Junction unit vector</th>
<th>Junction distance (pixel)</th>
<th>Junction angle (degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>MO1</td>
<td>0.00</td>
<td>-0.94</td>
</tr>
<tr>
<td>SO1</td>
<td>0.36</td>
<td>-0.90</td>
</tr>
<tr>
<td>SO2</td>
<td>-0.28</td>
<td>-0.94</td>
</tr>
<tr>
<td>SO3</td>
<td>0.00</td>
<td>-0.98</td>
</tr>
</tbody>
</table>

The superquadric part decomposition method was tested on a variety of synthetic and real range images such as shown in Figure 6.9 and Figure 6.10, whose resolution is 240×240. Figure 6.11 shows the real intensity and the range images acquired from the RANGER laser range finder system (see Appendix A), which are usually consists of the volumetric primitives of blocks, cylinders or spheres made of painted woods to prevent the specular components. Its original image resolution was 512×512, but it was resized to 256×256 for reducing computational burden. To reduce the computational burden which is occupied
6. Experiments

by the numerical minimization during iterative recovery of superquadrics and
which in turn depends on the number of range data points, the input range
images of size 240×240 or 256×256 pixels were sampled by a factor of four.
Initial size of seeds was 16×16 pixels and seed step was 16. The threshold
values needed in recover-and-select algorithm were experimentally determined.
The actual values of these thresholds which were kept constant throughout the
experiments shown in Table 6.3.

Table 6.3: Table of the threshold values for model recovery and the constants
for model selection.

<table>
<thead>
<tr>
<th>max_point_distance</th>
<th>max_average_model_error</th>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$K_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>5.0</td>
<td>1.0</td>
<td>0.3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Superquadrics seeded on planar regions would become exceedingly thin
during model recovery. To prevent numerical degeneracy during the minimization
of the fitting function of such superquadrics, we used a projection method to
limit the size of superquadric parameters $a_1$, $a_2$, $a_3$ above 1.0. In most cases 15
iterations of the Levenberg-Marquardt method were sufficient for convergence
of the minimization.

On the average it took less than 10 minutes to process each range image on
an SG1 O2 300MHz MIPS R5000 (IP32) processor platform with MIPS R5000
FPU and 256 MB main memory. Note that the total processing time for the su-
perquadric part decomposition mainly depends on the number of superquadric
growth iterations to reach the final size of the models. Therefore the segmen-
tation of a large object which requires a larger number of initial seeds as well as
more growth iterations take more computing time than the segmentation of a
smaller object. However, processing time is not critical since individual models
could be recovered in parallel. Besides, the computation of the superquadric
6. Experiments

(a) L-SHAPE  (b) T-PIPE  (c) GEARBOX

(d) HAMMER2  (e) COMP1  (f) COMP2

Figure 6.9: Complex synthetic range images.

(a) L-SHAPE  (b) T-PIPE  (c) GEARBOX

Figure 6.10: Real range images.
Figure 6.11: The intensity and the real range images acquired from the RANGER2500 laser range finder system.
fitting function and its derivatives is independent for each range point and can also be parallelized at a fine grain level in a straightforward way. Figures 6.12, 6.15, 6.18, 6.21, and 6.24 demonstrate correct decomposition of individual superquadric parts for synthetic range images in terms of their number and qualitative shape. Each figure presents a sequence of images which illustrate the stages in the recover-and-select segmentation. Each image in the sequence shows the original range image overlaid with superquadrics which were recovered up to the particular stage of the segmentation algorithm. Figures 6.27, 6.30, and 6.33 show correct decomposition of individual superquadric parts for real range images in terms of their number and qualitative shape respectively. These Figures also shows how occlusion influences the growth of superquadric models. Even if an object is almost completely occluded through the middle by another object, one superquadric model eventually grows into a complete model of the occluded object since the corresponding region is connected. If the occlusion produces unconnected regions, however, not a single superquadric model can bridge the gap. The result are two superquadric models recovered for the right and for the left part of the occluded object such as shown in Figure 3.14(a) We have demonstrated that the recover-and-select paradigm robustly decompose an object into generic parts which can be modeled well by individual superquadrics in terms of the number of models, their size and shape. However, due to the inherent monotonic growing strategy of the segmentation process, some points are sometimes included into a superquadric model which are sufficiently close to the growing description but in fact belong to another part which touches or penetrate the affected model. Such overgrowing of superquadric models affects their exact size and shape.

For a particular growing description, the final effect of including points from some other part depends on how soon or late in the iterative growing phase the
points get included into the description. The data points from the other parts can be treated as outliers with respect to the recovery process. Assuming that the imaged object is composed of parts that can be perfectly modeled by a chosen type of models, the effect of including points from other parts depends on the ratio of the number of outliers to the number of points belonging to the part we try to recover. Generally the greater the percentage of outliers the lower the chances to recover the correct model.

In real situation, the outliers may enter a description continuously. It is easy to find a case, where outliers gradually influence the description to such an extent, that even more outliers are included. This effect can be reduced by increasing the range image resolution and consequently lowering the threshold values for the maximal distance of points that get included into the description during the growing process. But still can the effect never be completely eliminated just by increasing the resolution.

To analyze possible alternative solutions let us make the following two assumptions:

(a) the scene is composed of parts that can be perfectly modeled by the chosen models (superquadrics),

(b) the regions where recovered models overlap are small in comparison with the regions contributing to the overlap.

If the overlap regions are sufficiently small, we can reliably recover a model from the data simply by discarding the data points in the overlap regions. Overlapping of regions by itself strongly suggests, that the rage points in these regions are not correctly assigned to models. Note that such a conclusion cannot be made simply from the distance of these points from the model, since these distances were limited in advance already affected the model that we use for computing.

For the cave tour step of the regions were analyzed then superquadrics 6.25, 6.29, 6.3:

B. descripti sponding ages capt shown in results after previous by tilting surfaces are not t case of 11 position shape pa same sha procedure surface infor superqua target ob using ans.
computing the distance.

For the objects which are usually included in real range images or have concave touching boundary or penetrating parts, we need to perform an addition step of computing the 8-connectedness closure of points from the overlapping regions with respect to the regions, which they were assigned to. We have analyzed the average errors ($\xi_i$) and the errors distribution of the final improved superquadric models for examples shown in Figures 6.13, 6.16, 6.19, 6.22, 6.25, 6.28, 6.31, and 6.34. Therefore, we finally obtained correct volumetric description of each object with part-based superquadric models for the corresponding decomposed parts as shown in Figures 6.14, 6.17, 6.20, 6.23, 6.26, 6.29, 6.32, and 6.35. The volumetric description tests for the real range images captured from RANGER laser range finder system (see Appendix A) were shown in Figures 6.36, 6.37 and 6.38, 6.39. According to the results, the final results after the post-processing of region improvement have more error than the previous synthetic or calibrated real range images. These images are acquired by tilting the laser slit light source with arbitrarily angle to take more visible surfaces as an input for fitting function. Therefore the calculated depth values are not to be calibrated correctly due to the incorrect configuration. Also in case of the real range image, we can get false result for the volumetric decomposition due to less discrimination and duality characteristics of superquadric shape parameters. Sometimes, even if the 3D shape is totally different, the same shape parameters can be extracted through the volumetric description procedure. Therefore, we need another supporting local features such as surface information to elaborately discriminate this global features of part-based superquadric models. For the surface description of each decomposed part of target object, we calculated principle curvatures for the set of surface points by using analytic regression-based method to segment and iteratively grows into
one of several primitive surface patches such as planar, cylindrical or spherical surfaces for each decomposed part. Its surface description includes binary features of surface geometric parameters such as plane equation coefficients and area for the planar surface, axis, a point, radius, length, and area for the cylindrical surface, and radius, center location and area for the spherical surface. It also describes the binary relationship between neighboring surfaces. The experimental results were shown for the synthetic and real range images respectively to prove the robustness of the proposed approach. The result for the synthetic COMP1 object were appeared from Figure 6.40 to Figure 6.42 and from Table 6.4 to Table 6.7. The result for the synthetic COMP2 object were displayed from Figure 6.43 to Figure 6.45 and from Table 6.8 to Table 6.12. The result for the real DATA1 object were shown from Figure 6.46 to Figure 6.49 and from Table 6.13 to Table 6.19. The result for the real DATA2 object were depicted from Figure 6.50 to Figure 6.52 and from Table 6.20 to Table 6.24. In these real range images, we can get some distorted results for the surface and volumetric description due to the generic characteristics of calibration during image acquisition and noise.
spherical binary features and the cylinder. It is expected that synthetic
from the result are different from the result of 6.49 and were demonstrated in this study.
In each case and during

(a) Range image.  
(b) Initial seeds.

(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.
Figure 6.12: Volumetric decomposition for a complex synthetic COMP1 model using basic recover-and-select algorithm without post-processing.
(a) Part 0 region examination.  
(b) Part 1 region examination.  
(c) Part 2 region examination.  
(d) Error distribution.
Figure 6.13: Volumetric description for a complex synthetic COMP1 model using basic recover-and-select algorithm without post-processing.
(a) Selection result after region improvement.

(b) Output of region improvement.

(c) Part 0 region examination.

(d) Part 1 region examination.

(e) Part 2 region examination.

(f) Error distribution.
Figure 6.14: Volumetric description for a complex synthetic COMP1 model after region improvement post-processing.
(a) Range image.  
(b) Initial seeds.

(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.
(g) Models after the 9th iteration growth.

(h) Remaining models after the 9th selection.

(i) Models after the 12th iteration growth.

(j) Remaining models after the 12th selection.

(k) Final superquadric model selection.

(l) Final decomposed superquadric model.

Figure 6.15: Volumetric decomposition for a complex synthetic COMP2 model using basic recover-and-select algorithm without post-processing.
(a) Part 0 region examination.
(b) Part 1 region examination.
(c) Part 2 region examination.
(d) Error distribution.
Figure 6.16: Volumetric description for a complex synthetic COMP2 model using basic recover-and-select algorithm without post-processing.
6. Experiments

(a) Selection result after region improvement.

(b) Output of region improvement.

(c) Part 0 region examination.

(d) Part 1 region examination.

(e) Part 2 region examination.

(f) Error distribution.
Figure 6.17: Volumetric description for a complex synthetic COM2 model after region improvement post-processing.
(a) Range image.  
(b) Initial seeds.  

(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.  

(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.
6. Experiments

Figure 6.18: Volumetric decomposition for a complex synthetic T-PIPE model using basic recover-and-select algorithm without post-processing.
Figure 6.19: Volumetric description for a complex synthetic T-PIPE model using basic recover-and-select algorithm without post-processing.
Figure 6.20: Volumetric description for a complex synthetic T-PIPE model after region improvement post-processing.
(a) Range image.  
(b) Initial seeds.  
(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.  
(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.
6. Experiments

(g) Models after the 9th iteration growth.

(h) Remaining models after the 9th selection.

(i) Models after the 12th iteration growth.

(j) Remaining models after the 12th selection.

(k) Final superquadric model selection.

(l) Final decomposed superquadric model.

Figure 6.21: Volumetric decomposition for a complex synthetic HAMMER2 model using basic recover-and-select algorithm without post-processing.
(a) Part 0 region examination.

(b) Part 1 region examination.

(c) Part 2 region examination.

(d) Error distribution.
Figure 6.22: Volumetric description for a complex synthetic HAMMER2 model using basic recover-and-select algorithm without post-processing.
Figure 6.23: Volumetric description for a complex synthetic HAMMER2 model after region improvement post-processing.
6. Experiments

(a) Range image.  
(b) Initial seeds.

(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.

Figure model
Figure 6.24: Volumetric decomposition for a complex synthetic GEARBOX model using basic recover-and-select algorithm without post-processing.
(a) Part 0 region examination.
(b) Part 1 region examination.
(c) Part 2 region examination.
(d) Part 3 region examination.
Figure 6.25: Volumetric description for a complex synthetic GEARBOX model using basic recover-and-select algorithm without post-processing.
Figure 6.10
(a) Selection result after region improvement.  
(b) Output of region improvement.  
(c) Part 0 region examination.  
(d) Part 1 region examination.  
(e) Part 2 region examination.  
(f) Part 3 region examination.  
(g) Error distribution.  
(h) Average error for part 0.  
(i) Average error for part 1.
Figure 6.26: Volumetric description for a complex synthetic GEARBOX model after region improvement post-processing.
(a) Range image.  (b) Initial seeds.

(c) Models after the 3rd iteration growth.  (d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth.  (f) Remaining models after the 6th selection.
Figure 6.27: Volumetric decomposition for a complex real L-SHAPE model using basic recover-and-select algorithm without post-processing.
Figure 6.28: Volumetric description for a complex real L-SHAPE model using basic recover-and-select algorithm without post-processing.
Figure 6.29: Volumetric description for a complex real L-SHAPE model after region improvement post-processing.
(a) Range image. (b) Initial seeds.

(c) Models after the 3rd iteration growth. (d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth. (f) Remaining models after the 6th selection.
Figure 6.30: Volumetric decomposition for a complex real T-PIPE model using basic recover-and-select algorithm without post-processing.
Figure 6.31: Volumetric description for a complex real T-PIPE model using basic recover-and-select algorithm without post-processing.
Figure 6.32: Volumetric description for a complex real model after region improvement post-processing.
6. Experiments

(a) Range image.  
(b) Initial seeds.

(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.

Figure using l
Figure 6.33: Volumetric decomposition for a complex real GEARBOX model using basic recover-and-select algorithm without post-processing.
6. Experiments

(a) Part 0 region examination.  (b) Part 1 region examination.

(c) Part 2 region examination.  (d) Part 3 region examination.

Figure 6
Figure 6.34: Volumetric description for a complex real GEARBOX model using basic recover-and-select algorithm without post-processing.
Figure 6. Experiments

(a) Selection result after region improvement.
(b) Output of region improvement.
(c) Part 0 region examination.
(d) Part 1 region examination.
(e) Part 2 region examination.
(f) Part 3 region examination.
(g) Error distribution.
(h) Average error for part 0.
(i) Average error for part 1.
Figure 6.35: Volumetric description for a complex real GEARBOX model after region improvement post-processing.
Figure 6

(a) Range image.
(b) Initial seeds.

(c) Models after the 3rd iteration growth.
(d) Remaining models after the 3rd selection.

(e) Models after the 6th iteration growth.
(f) Remaining models after the 6th selection.
Figure 6.36: Volumetric decomposition for a complex real DATA1 model using recover-and-select algorithm with post-processing.
(a) Part 0 region examination.  
(b) Part 1 region examination.  
(c) Part 2 region examination.  
(d) Part 3 region examination.
6. Experiments

(c) Error distribution.  (f) Average error for part 0.  (g) Average error for part 1.

(h) Average error for part 2.  (i) Average error for part 3.  (j) Decomposed part 0.


Figure 6.37: Volumetric description for a complex real DATA1 model using recover-and-select algorithm after post-processing.
(a) Range image.  
(b) Initial seeds.  
(c) Models after the 3rd iteration growth.  
(d) Remaining models after the 3rd selection.  
(e) Models after the 6th iteration growth.  
(f) Remaining models after the 6th selection.
Figure 6.38: Volumetric decomposition for a complex synthetic DATA2 model using recover-and-select algorithm with post-processing.
6. Experiments

(a) Part 0 region examination.
(b) Part 1 region examination.
(c) Part 2 region examination.
(d) Error distribution.
Figure 6.39: Volumetric description for a complex synthetic DATA2 model using recover-and-select algorithm after post-processing.
Figure 6.40: Surface description for the decomposed part 0 of a complex synthetic COMPL model.

Table 6.4: Surface geometric unary features for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>type coefficients area</td>
<td>plane</td>
</tr>
<tr>
<td></td>
<td>area</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sceneID</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>type coefficients area</td>
<td>plane</td>
</tr>
<tr>
<td></td>
<td>area</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sceneID</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>type coefficients area</td>
<td>plane</td>
</tr>
<tr>
<td></td>
<td>area</td>
</tr>
</tbody>
</table>

Table 6.5: Surface adjacency binary feature for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
Figure 6.41: Surface description for the decomposed part 1 of a complex synthetic COMPI model.

Table 6.6: Surface geometric unary features for the part 1.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>plane</td>
<td>cylinder</td>
</tr>
<tr>
<td>coefficients</td>
<td>-0.000017 0.728689 0.684844 1.401850</td>
<td>0.159436 0.934203 0.319133</td>
</tr>
<tr>
<td>area</td>
<td>0.187182</td>
<td>0.158907 1.085281 0.145655</td>
</tr>
<tr>
<td>sceneID</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td></td>
<td>cylinder</td>
</tr>
<tr>
<td>axis</td>
<td>0.159436 0.934203 0.319133</td>
<td>0.222392</td>
</tr>
<tr>
<td>point</td>
<td>0.158907 1.085281 0.145655</td>
<td>0.795675</td>
</tr>
<tr>
<td>radius</td>
<td>0.222392</td>
<td>0.390844</td>
</tr>
<tr>
<td>length</td>
<td>0.795675</td>
<td></td>
</tr>
<tr>
<td>area</td>
<td>0.390844</td>
<td></td>
</tr>
</tbody>
</table>
(a) Decomposed part 2.  (b) Surface normal.  (c) Surface segmentation.

Figure 6.42: Surface description for the decomposed part 2 of a complex synthetic COMP1 model.

Table 6.7: Surface geometric unary features for the part 2.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>plane</td>
</tr>
<tr>
<td>coefficients</td>
<td>0.000017</td>
</tr>
<tr>
<td>area</td>
<td>0.211695</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sceneID</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>cylinder</td>
</tr>
<tr>
<td>axis</td>
<td>-0.330221</td>
</tr>
<tr>
<td>point</td>
<td>0.368712</td>
</tr>
<tr>
<td>radius</td>
<td>0.212879</td>
</tr>
<tr>
<td>length</td>
<td>1.077835</td>
</tr>
<tr>
<td>area</td>
<td>0.974723</td>
</tr>
</tbody>
</table>
Figure 6.43: Surface description for the decomposed part 0 of a complex synthetic COMP2 model.

Table 6.8: Surface geometric unary features for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>type</th>
<th>coefficients</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>plane</td>
<td>-0.003091 0.728645</td>
<td>0.684884 1.340620</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.149093</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>plane</td>
<td>-0.759467 -0.224480</td>
<td>0.610589 -0.230378</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.326470</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>plane</td>
<td>0.359190 -0.312369</td>
<td>0.879436 -0.149744</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.284182</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.9: Surface adjacency binary feature for the part 0.

<table>
<thead>
<tr>
<th>sceneID1</th>
<th>sceneID2</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>75.110001</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>68.062477</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>70.469894</td>
</tr>
</tbody>
</table>
Figure 6.44: Surface description for the decomposed part 1 of a complex synthetic COMP2 model.

Table 6.10: Surface geometric unary features for the part 1.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>plane</td>
</tr>
<tr>
<td>coefficients</td>
<td>0.361752 0.777253 0.514794 1.569977</td>
</tr>
<tr>
<td>area</td>
<td>0.247560</td>
</tr>
<tr>
<td>sceneID</td>
<td>12</td>
</tr>
<tr>
<td>type</td>
<td>cylinder</td>
</tr>
<tr>
<td>axis</td>
<td>0.435342 0.874186 0.215118</td>
</tr>
<tr>
<td>point</td>
<td>-0.291381 1.745249 -0.419284</td>
</tr>
<tr>
<td>radius</td>
<td>0.329198</td>
</tr>
<tr>
<td>length</td>
<td>1.443240</td>
</tr>
<tr>
<td>area</td>
<td>1.036759</td>
</tr>
</tbody>
</table>
Figure 6.45: Surface description for the decomposed part 2 of a complex synthetic COMP2 model.

Table 6.11: Surface geometric unary features for the part 2.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>type</th>
<th>coefficients</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>plane</td>
<td>-0.001523</td>
<td>0.728963 0.684551 0.721838</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.229200</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>plane</td>
<td>-0.757336</td>
<td>-0.218219 0.615486 -0.227330</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.222248</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>plane</td>
<td>0.360530</td>
<td>-0.312139 0.878969 0.386743</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.169701</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.12: Surface adjacency binary feature for the part 2.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11 74.727371</td>
</tr>
<tr>
<td>10</td>
<td>12 68.061424</td>
</tr>
<tr>
<td>11</td>
<td>12 70.362640</td>
</tr>
</tbody>
</table>
Figure 6.46: Surface description for the decomposed part 0 of a complex real DATA1 model.

Table 6.13: Surface geometric unary features for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>planc</td>
</tr>
<tr>
<td>coefficients</td>
<td>0.160874</td>
</tr>
<tr>
<td>area</td>
<td>0.08029</td>
</tr>
<tr>
<td>sceneID</td>
<td>12</td>
</tr>
<tr>
<td>type</td>
<td>planc</td>
</tr>
<tr>
<td>coefficients</td>
<td>-0.504189</td>
</tr>
<tr>
<td>area</td>
<td>0.256744</td>
</tr>
<tr>
<td>sceneID</td>
<td>13</td>
</tr>
<tr>
<td>type</td>
<td>planc</td>
</tr>
<tr>
<td>coefficients</td>
<td>0.741784</td>
</tr>
<tr>
<td>area</td>
<td>0.179031</td>
</tr>
</tbody>
</table>

Table 6.14: Surface adjacency binary feature for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
Figure 6.47: Surface description for the decomposed part 1 of a complex real DATA1 model.

Table 6.15: Surface geometric unary features for the part 1.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>planc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>coefficents</td>
<td>0.155622</td>
<td>-0.018168</td>
<td>0.987650</td>
<td>-0.405168</td>
</tr>
<tr>
<td>area</td>
<td>0.181271</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sceneID</th>
<th>12</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>cylinder</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>axis</td>
<td>0.002419</td>
<td>0.590268</td>
<td>0.807203</td>
<td></td>
</tr>
<tr>
<td>point</td>
<td>-0.663332</td>
<td>0.812442</td>
<td>-0.591063</td>
<td></td>
</tr>
<tr>
<td>radius</td>
<td>0.243305</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>0.660274</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>area</td>
<td>0.273169</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. Experiments

(a) Decomposed part 2.  (b) Surface normal.  (c) Surface segmentation.

Figure 6.48: Surface description for the decomposed part 2 of a complex real DATA1 model.

Table 6.16: Surface geometric unary features for the part 2.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>plane</td>
</tr>
<tr>
<td>coefficients</td>
<td>0.145921 0.030432 0.988828 0.024316</td>
</tr>
<tr>
<td>area</td>
<td>0.128544</td>
</tr>
<tr>
<td>sceneID</td>
<td>12</td>
</tr>
<tr>
<td>type</td>
<td>plane</td>
</tr>
<tr>
<td>coefficients</td>
<td>-0.667424 -0.533995 0.519032 -0.980538</td>
</tr>
<tr>
<td>area</td>
<td>0.189827</td>
</tr>
<tr>
<td>sceneID</td>
<td>13</td>
</tr>
<tr>
<td>type</td>
<td>plane</td>
</tr>
<tr>
<td>coefficients</td>
<td>0.594114 -0.678689 0.431752 -0.683855</td>
</tr>
<tr>
<td>area</td>
<td>0.220900</td>
</tr>
</tbody>
</table>

Table 6.17: Surface adjacency binary feature for the part 2.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11 66.447334</td>
</tr>
<tr>
<td>10</td>
<td>12 60.464146</td>
</tr>
<tr>
<td>11</td>
<td>12 79.048187</td>
</tr>
</tbody>
</table>
Figure 6.49: Surface description for the decomposed part 3 of a complex real DATA1 model.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>area</th>
<th>type coefficients</th>
<th>planec</th>
<th>0.133748</th>
<th>0.023848</th>
<th>0.990728</th>
<th>-0.517789</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.608791</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sceneID</td>
<td>area</td>
<td>type coefficients</td>
<td>planec</td>
<td>0.584566</td>
<td>-0.653635</td>
<td>0.480670</td>
<td>-0.440782</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.378901</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sceneID</td>
<td>area</td>
<td>type coefficients</td>
<td>planec</td>
<td>-0.667729</td>
<td>-0.514371</td>
<td>0.538108</td>
<td>-0.536195</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.271305</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.19: Surface adjacency binary feature for the part 3.
Figure 6.50: Surface description for the decomposed part 0 of a complex real DATA2 model.

Table 6.20: Surface geometric unary features for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>type</th>
<th>coefficients</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>plane</td>
<td>-0.148162</td>
<td>0.001676 0.988962 0.137349</td>
</tr>
<tr>
<td></td>
<td>area</td>
<td>0.265944</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sceneID</th>
<th>type</th>
<th>coefficients</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>plane</td>
<td>-0.474477</td>
<td>-0.773702 0.419830 -0.273962</td>
</tr>
<tr>
<td></td>
<td>area</td>
<td>0.357021</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sceneID</th>
<th>type</th>
<th>coefficients</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>plane</td>
<td>0.764259</td>
<td>-0.487222 0.422520 -0.275496</td>
</tr>
<tr>
<td></td>
<td>area</td>
<td>0.068028</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.21: Surface adjacency binary feature for the part 0.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
Figure 6.51: Surface description for the decomposed part 1 of a complex real DATA1 model.

Table 6.22: Surface geometric unary features for the part 1.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>cylinder</td>
</tr>
<tr>
<td>axis</td>
<td>-0.253448 0.609865 0.750885</td>
</tr>
<tr>
<td>point</td>
<td>-0.308243 0.416866 -0.647301</td>
</tr>
<tr>
<td>radius</td>
<td>0.202792</td>
</tr>
<tr>
<td>length</td>
<td>0.773089</td>
</tr>
<tr>
<td>area</td>
<td>0.193463</td>
</tr>
</tbody>
</table>
6. Experiments

Figure 6.52: Surface description for the decomposed part 2 of a complex real DATA2 model.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>type coefficients</th>
<th>area</th>
<th>sceneID</th>
<th>type coefficients</th>
<th>area</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>plane</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.174269</td>
<td>0.007933</td>
<td>0.984666</td>
<td>0.353732</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.164716</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>plane</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.849405</td>
<td>-0.463038</td>
<td>0.253193</td>
<td>-0.508213</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.427888</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>plane</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.391425</td>
<td>-0.733345</td>
<td>0.555871</td>
<td>-0.346262</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.547130</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.24: Surface adjacency binary feature for the part 2.

<table>
<thead>
<tr>
<th>sceneID</th>
<th>angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusions and Discussion

We have proposed a new efficient hybrid 3D object representation and recognition method by combining volumetric and surface description together hierarchically. Only conventional shape based description approach has difficulties to describe parts of 3D objects because it is not well defined to derive volumetric information from surface descriptions. Volumetric superquadric description by using recover-and-selective segmentation paradigm can be directly used to decompose a 3D object into its consisting parts uniquely, which is global shape representation of 3D object. The knowledge about objects in terms of stored models with superquadrics as part-primitives has not so far been exploited despite the compact representation which may facilitate fast database matching. One of reasons that superquadrics have not been used much for object recognition is the difficulty of matching superquadrics recovered from image data with superquadrics in the model database. Data-driven bottom-up strategy of fitting superquadrics to image data does not assure the recovery of canonical superquadric models. The two main problems of matching recovered superquadrics with a set of superquadrics in a model database are the non-unique
7. Conclusions and Discussion

parameterization of the same shapes which is a result of different possible orientations of the local superquadric-centered coordinate system and the nonlinear relationship between superquadric parameters.

We have constructed the hierarchical part-based superquadric models to be used in matching procedure for the object recognition, which are consisted of the volume and the surface layers respectively. In volume layer, superquadrics are used in the first stage for a stable recovery of the overall shape of objects and to provide a convenient abstraction mechanism for robust indexing. In the second layer, finer level of surface layer consists of surface patches of uniform principal curvatures which locally describes the each of decomposed part of the objects. We assumed that our target 3D objects are composed of the convex volumetric primitive parts and can be modeled by superquadrics, which are composed together in a solid constructive modelling manner using union operation. We tested proposed method on a variety of the real and the synthetic range images of 3D objects and demonstrated that a direct segmentation of range images into part-level volumetric models, which is stable with regard to the number of parts and their generic shapes, is possible. To prevent false volumetric decomposition using basic recover-and-select segmentation paradigm, we adopted the post-processing of region improvement using 8 neighborhood connectedness and improved the results that we want. But in case of real range images, it is much easier to get false result for the volumetric decomposition due to less discrimination and duality characteristics of superquadric shape parameters. It is not desirable to use superquadric shape parameters directly in the object recognition procedure. Therefore, we need another supporting local features such as surface information to compensate this global features. For the surface description of each decomposed part of target object, we calculated principle curvatures for the set of surface points by using analytic regression-based method to seg-
ment and iteratively grows into one of several primitive surface patches such as planar, cylindrical or spherical surfaces for each decomposed part. Its surface description includes binary features of surface geometric parameters such as plane equation coefficients and area for the planar surface, axis, a point, radius, length, and area for the cylindrical surface, and the radius, the center location and the area for the spherical surface. It also includes the description of the binary relationships between neighboring surfaces: plane-plane, plane-cylinder, plane-sphere, cylinder-cylinder, cylinder-sphere, and sphere-sphere. Therefore, our integrated hybrid 3D object representation method has potential robustness to recognize the identity, position, and orientation of randomly oriented objects. Furthermore, we could reduce the effects of self-occlusion according to the viewpoint.

The primary role of superquadrics has been mainly focused on the volumetric representation in conjunction with shape recovery and segmentation. Despite initial reluctance of using superquadrics due to their nonlinear form, they have proven to be the primitives of choice for many applications that require volumetric models. It is important to understand the limitations of the basic superquadric models, namely their limited shape vocabulary, and the fact that they are really coarse volumetric model descriptions suitable, in particular, for object-centered generic shape descriptions. When selecting a volumetric representation for a vision application, one has to keep in mind the facts that superquadrics have nonlinear relation among parameters, a smooth surface, no real existence of edges, and inherent symmetry assumptions which makes them insensitive to small occlusion and change of viewpoint. Superquadrics are used in computer vision applications as volumetric shape primitives for 3D representation, segmentation, object classification and recognition, and tracking. Superquadrics are good at capturing the global coarse shape of 3D objects or
of their constituent parts.

Currently, we have constraints in our target objects to have convex volumetric primitive parts that can be recovered well with non-deformable superquadric models. But for the recognition of more natural complex objects, we need advanced deformable models rather than superquadrics. And for the real range images, we have to consider more reliable and stable methods for calculating the local surface characteristics. We are now using only single range image as an input. But if we want to recognize the objects which have the same dimension and the different colors, we should extract the color information for the target object to be discriminated.
Appendix A

Range image acquisition system

A.1 IVP RANGER2500 Laser Range Finder System

The RANGER2500 laser range finder system acquires profiles of 3D target objects using laser triangulation, and the technique is known under a variety of names such as sheet-of-light range imaging, laser profiling and light-stripe ranging. We call it sheet-of-light range imaging. A configuration of our range sensing system appears in Figure A.1. The RANGER system contains two different sub-systems, the Smart Camera(s) and the PC interface. Each Smart Camera contains the Smart Vision Sensor (SVS), a control processor (an Intel 386) and an IVP HSSI (High Speed Serial Interface). The Smart Camera is connected to the system PC via a COM-port and with an HSSI interface on a PCI board called SC adapter. Figure A.2 shows the example of the target object projected with laser slit beam. In RANGER2500 system, range data
A. Range image acquisition system

Figure A.1: The RANGER2500 laser range camera system.

Figure A.2: The target object projected with laser slit beam.
is acquired with triangulation. The offset position of the reflected light on the sensor plane depends on the distance (range) from the light source to the object in Figure A.3(a). Using trigonometry we can solve the equations for the range if we know the distance of the baseline between the laser and the optical center of the sensor, the direction of the transmitted ray. The third triangulation parameter, the direction of the incoming light ray, is given by the sensor offset position. For each position of the sheet-of-light the depth variation of the scene creates a contour which is projected onto the sensor plane in Figure A.4. The example of 3D profiles for the hand range image is shown in Figure A.5 in which red straight line shows the range values from 0 to 255 and blue dotted line describes the intensity values from 0 to 255 respectively. In a 3D plot example, Figure A.6, the recorded range profiles are converted to polygon surface patches and visualized using a 3D renderer. If intensity data has been acquired this is texture mapped onto the polygon model. If we extract the position of the incoming light for each sensor row, we obtain an offset data vector, or profile, that serves as input for the triangulations. To make a sheet-of-light the sharp laser spot-light passes through a lens, see Figure A.3(b). The lens spreads the light into a sheet in one dimension while it is unaffected in the other. The lens can either be a glass cylinder or a diffractive element. A sheet-of-light can also be made using a fast scanning mirror mechanism, an array of LED's, or with a slit projector.

A.2 RIEGL LMS-Z210 Laser Mirror Range Scanner

The 3D laser mirror scanner LMS-Z210 shown in Figure A.7 is one of radar sensors, and a surface imaging system based upon accurate distance measurement by means of electro-optical range measurement and a two axis beam
Figure A.3: RANGER2500 laser range finder using sheet-of-light.
scanning mechanism. The 3D images are gained by performing a number of independent laser range measurements in different, but well-defined angular directions. These range data together with the associated angles from the basis of the 3D images. This rangefinder system is based upon the principle of time-of-flight measurement of short laser pulse in the infrared wavelength region. Figure A.8 shows the following measurement principle of the pulsed range finder. An electrical pulse generator periodically drives a semiconductor laser diode sending out infrared light pulses, which are collimated by the transmitter lens. Via the receiver lens, part of the echo signal reflected by the target hits
A. Range image acquisition system

Figure A.6: The surface plot of the hand range image.

A photodiode which generates an electrical receiver signal. The time interval between the transmitted and the received pulses are counted by means of clock frequency. The calculated range value is fed into the internal microcomputer which processes the measured data and prepares it for data output. The range finder electronics (1) of the 3D scanner LMS-Z210 shown in Figure A.9 is based upon the RIEGL LD90-3 laser distance meter, optimized in order to meet the requirements of high speed scanning (fast laser repetition rate, fast signal processing and high speed data interface). The fast angular deflection ("line scan") of the laser beam (2) is realized by a rotating polygon (3) with a number of reflective surfaces. It rotates continuously at adjustable speed to provide an unidirectional scan within an angle of \( \theta = 80^\circ \). The slow scan ("frame scan") is provided by rotating the complete optical head (4) up to 340\(^\circ\). The gained information of RANGE, SIGNAL AMPLITUDE, and ANGLE is provided via an 8 bit parallel data output which can be connected directly to the ECP compatible LPT printer port (5) of a PC (6). Laptop, or equivalent.
A. Range image acquisition system

Figure A.7: The RIEGL LMS-Z210 laser mirror range scanner.

Figure A.8: Measurement principle of the pulsed range finder.

This PC can be equipped with the RIEGL SCAN-software (7). It is a program running under Win95 / WinNT for data acquisition and real time display. The displayed 3D range images can be zoomed and copied to the clipboard. For fur-
ther processing the scanner data are logged to disk. Figure A.10 presents the

Figure A.9: The principle of operation.

example of acquired pseudo-color range image, intensity image, and true color image respectively. And also for the 3D reconstruction test, the wall image is reconstructed with the texture of true color image in Figure A.11.
Figure A.10: Example of image acquisition of RIEGL laser mirror range scanner.
(a) True color image of the building wall.

(b) 3D reconstructed image with texture.

Figure A.11: Experiment for the 3D reconstruction with texture.
Appendix B

Differential Geometry of 3D surface

In Figure B.1, a regular surface in the 3D Euclidean space is defined, parametrically, as

\[ \vec{r}(u, v) = (x(u, v), y(u, v), z(u, v)) \]  \hspace{1cm} (B.1)

where \( \vec{r} \) is the vector from the origin of an arbitrary 3D coordinate system to the sampled point, \( x, y, z \) are its 3D coordinates, and \( (u, v) \) are the parameters in the 2D space in which the data is embedded. Assuming that the surfaces are twice differentiable, the basic equations such as the first fundamental form and the second fundamental form have been used in the past to define many surface properties. The first fundamental form is

\[ I(u, v, du, dv) = d\vec{r} \cdot d\vec{r} \]  \hspace{1cm} (B.2)

\[ = Edu^2 + 2Fdu dv + Gdv^2 \]  \hspace{1cm} (B.3)
\[ E = \vec{F}_u \cdot \vec{F}_u, \quad F = \vec{F}_u \cdot \vec{F}_v, \quad G = \vec{F}_v \cdot \vec{F}_v \]  \hspace{1cm} (B.4)

\[ \vec{F}_u(u, v) = \partial \vec{F} / \partial u, \quad \vec{F}_v(u, v) = \partial \vec{F} / \partial v \]  \hspace{1cm} (B.5)

\( \vec{F}_u \) and \( \vec{F}_v \) are linearly independent tangents to the parameter curves and define a plane known as the tangent plane as shown in Figure B.1. This implies that all tangents to the parameter curves on the surface at a point \((u, v)\) lie on this plane. Although the two vectors may not be orthogonal, they form a basis for the tangent plane at the point \((u, v)\). The first fundamental form measures the change in \( \vec{F}_u dv + \vec{F}_v dv \) with respect to the corresponding change of \((du, dv)\) in the parameter plane. \( I(u, v, du, dv) \) is invariant to the rotation, the translation, and the parameterization of the surface, implying that the first fundamental form is an intrinsic surface property. Intrinsic properties are significant since the same intrinsic surface primitive viewed from two different viewpoints remains
the same for each viewpoint. On the contrary, extrinsic properties dependent on how the surface is embedded in the 3D space and will be different for the two viewpoints; they are not useful in matching the models and the objects in the scene. The functions $E$, $F$, and $G$ are invariant with respect to coordinate transformations but not with respect to parameter transformations. The second fundamental form is defined as

$$ II(u, v, du, dv) = -d\vec{\mathbf{r}} \cdot d\vec{n} $$

$$ = L du^2 + 2M du dv + N dv^2 $$

where

$$ \vec{n} = \frac{\vec{r}_u \times \vec{r}_v}{||\vec{r}_u \times \vec{r}_v||} $$

$$ L = \vec{r}_{uu} \cdot \vec{n}, \quad M = \vec{r}_{uv} \cdot \vec{n}, \quad N = \vec{r}_{vv} \cdot \vec{n}. $$

$$ \vec{r}_{uu}(u, v) = \frac{\partial^2 \vec{r}}{\partial u^2}, \quad \vec{r}_{uv}(u, v) = \frac{\partial^2 \vec{r}}{\partial u \partial v}, \quad \vec{r}_{vv}(u, v) = \frac{\partial^2 \vec{r}}{\partial v^2}. $$

The vector $\vec{n}$ is known as the surface normal vector and is perpendicular to the tangent plane. Together with the two unnormalized vectors $\vec{r}_u$ and $\vec{r}_v$, $\vec{n}$ forms a local coordinate system at the point $\vec{r}(u, v)$. The second fundamental form measures the correlation between the change of the normal vector and $\vec{r}$ as a function of $(du, dv)$, hence, demonstrating that $II(u, v, du, dv)$ is an extrinsic surface property. However, the second fundamental form is invariant under parameter transformations with a positive Jacobian.

An additional and important surface property is the curvature, $\kappa$. Of particular interest are the principal curvatures, $\kappa_1$ and $\kappa_2$, referred to as the max-
B. Differential Geometry of 3D surface

 собой curvature and the minimum curvature, respectively. $\kappa_1$ and $\kappa_2$ may be calculated by solving

\[
\det \begin{bmatrix}
\kappa E - L & \kappa F - M \\
\kappa F - M & \kappa G - N
\end{bmatrix} = 0.
\]  
(B.11)

The directions, in the parameter plane, of the maximum and minimum $\kappa$ are referred to as the principal directions. The principal directions are the roofs of

\[
\det \begin{bmatrix}
\tau^2 & -1 & 1 \\
E & F & G \\
L & M & N
\end{bmatrix} = 0.
\]  
(B.12)

where $\tau = du/dv = \tan \theta$, and $\theta$ is the angle in the parameter $(u,v)$-plane. On any given regular surface, by connecting the points that have the same principal directions along the minimum and maximum directions, two families of curves are obtained known as the lines of curvature of the surface.

Using $\kappa_1$ and $\kappa_2$, Gaussian $K$ and mean $H$ curvature values may also be defined:

\[
K = \kappa_1 \kappa_2 = \frac{LN - M^2}{EG - F^2}  
\]  

\[
H = \frac{1}{2}(\kappa_1 + \kappa_2) = \frac{EN + GL - 2FM}{2(EG - F^2)}  
\]  

Unlike $\kappa_1$ and $\kappa_2$, where the directions are needed as a part of the description of the surface shape, the Gaussian curvature allows for a representation of the curvature by a single number and is invariant to all rigid geometrical transformations.
The principal curvatures and $K$ and $H$ are related by

$$\kappa^2 - 2H\kappa + K = 0,$$

and have many differences and similarities [6, 5]. Considering the signs of the quantities, $\{\kappa_1, \kappa_2\}$ may be used to classify the surface points into six types, whereas $\{H, K\}$ determine eight types of surfaces (see Table B.1).

Table B.1: Classifying surface types using Gaussian and mean curvature information.

<table>
<thead>
<tr>
<th>$K &gt; 0$</th>
<th>$K = 0$</th>
<th>$K &lt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H &lt; 0$</td>
<td>peak</td>
<td>ridge</td>
</tr>
<tr>
<td>$H = 0$</td>
<td>-</td>
<td>flat</td>
</tr>
<tr>
<td>$H &gt; 0$</td>
<td>pit</td>
<td>valley</td>
</tr>
</tbody>
</table>

Table B.2: Classifying surface types using principal curvature information.

<table>
<thead>
<tr>
<th>$\kappa_1 &gt; 0$</th>
<th>$\kappa_1 = 0$</th>
<th>$\kappa_1 &lt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\kappa_2 &lt; 0$</td>
<td>saddle</td>
<td>ridge</td>
</tr>
<tr>
<td>$\kappa_2 = 0$</td>
<td>valley</td>
<td>flat</td>
</tr>
<tr>
<td>$\kappa_2 &gt; 0$</td>
<td>pit</td>
<td>valley</td>
</tr>
</tbody>
</table>

Gaussian curvature is an intrinsic property of the surface, while mean curvature and principal curvatures are extrinsic properties (however, these properties are invariant to parameter transformations with a positive Jacobian). If the surface under consideration is convex, Gaussian curvature can uniquely determine the surface shape; $\{\kappa_1, \kappa_2\}$ are unable to provide such uniqueness. On the other hand, since mean curvature is found by averaging $\kappa_1$ and $\kappa_2$, it is less sensitive to noise than the Gaussian and principal curvatures. For example, if for two
B. Differential Geometry of 3D surface

smooth surfaces, $S_1$ and $S_2$, $E_1 = E_2$, $F_1 = F_2$, $G_1 = G_2$, $L_1 = L_2$, $M_1 = M_2$, and $N_1 = N_2$, then $S_1$ and $S_2$ have identical shapes and made be coincide using appropriate rotation and translation matrices. Also, using the elements of the second fundamental form, namely $L$, $M$, and $N$, the shape of the surface near the point of tangency may be classified into one of four groups:

- If $LN - M^2 > 0$, then the neighborhood is elliptic, This implies that the surface lies entirely one side of the tangent plane. If $II(u, v, du, dv)$ is positive, the surface, at the neighbor of the tangent plane, lies on the same side as the surface normal point; otherwise, the neighborhood resides on the other side. Simple examples of elliptic surfaces include spheres and ellipsoids.

- If $LN - M^2 < 0$, then the neighborhood is hyperbolic, also known as the saddle point. Assuming that the tangent plane is divided up into four regions, then the surface alternatively lies above and below the tangent plane.

- If $LN - M^2 = 0$, but not all $L$, $M$, and $N$ are equal to zero, then the surface is parabolic at the neighborhood point lying entirely on one side of the tangent plane. This is an intermediate stage between elliptic surfaces and hyperbolic surfaces.

- If $L = M = N = 0$, no information regarding the shape of the surface can be provided using such a classification scheme.

In some literature, a third fundamental form, $III$, is defined by combining Gaussian and mean curvatures and the first and second fundamental forms:

$$III(u, v, du, dv) = 2II(u, v, du, dv) - KI(u, v, du, dv).$$  \hspace{1cm} (B.16)
All sign changes of $II$ caused by parametric transformations are cancelled by corresponding changes in the sign of $H$; therefore, the sign of the third fundamental form is invariant to such transformations. However, the third fundamental form has not been widely used in the computer vision literature in the past.

If at a point $(u_0, v_0)$ the Jacobian $\partial(x, y)/\partial(u, v) \neq 0$, then in the neighborhood of $(x(u_0, v_0), y(u_0, v_0))$ $u$ and $v$ can be expressed as single-valued continuous functions of $x$ and $y$; we can let $x(u, v) = u$ and $y(u, v) = v$, representing the surface by

$$\mathbf{r}(u, v) = (u, v, f(u, v)), \quad (B.17)$$

known as the graph surface representation. In most instances, the sampled surfaces are assumed to be graph surfaces. Using this representation, the partial derivatives defined above are simplified as follows:

$$\mathbf{r}_u = (1 \ 0 \ f_u), \quad \mathbf{r}_v = (0 \ 1 \ f_v) \quad (B.18)$$

$$\mathbf{r}_{uu} = (0 \ 0 \ f_{uu}), \quad \mathbf{r}_{uv} = (0 \ 0 \ f_{uv}), \quad \mathbf{r}_{vv} = (0 \ 0 \ f_{vv}), \quad (B.19)$$

and the surface normal becomes

$$\mathbf{n} = \frac{1}{\sqrt{f_u^2 + f_v^2 + 1}}(-f_u \ - \ f_v \ 1). \quad (B.20)$$
The components of the first and second fundamental forms are then

\[ L = \frac{f_{uu}}{\sqrt{f_u^2 + f_v^2 + 1}}, \]
\[ M = \frac{f_{uv}}{\sqrt{f_u^2 + f_v^2 + 1}}, \]
\[ N = \frac{f_{vv}}{\sqrt{f_u^2 + f_v^2 + 1}}, \]
\[ E = 1 + f_u^2, \quad F = f_u f_v, \quad G = 1 + f_v^2, \]
(B.21)

The Gaussian and mean curvatures are defined as

\[ K = \frac{f_{uu} f_{vv} - f_{uv}^2}{(f_u^2 + f_v^2 + 1)^2}, \]
\[ H = \frac{(1 + f_u^2) f_{uu} + (1 + f_v^2) f_{vv} - 2 f_u f_v f_{uv}}{2(f_u^2 + f_v^2 + 1)^{3/2}}. \]
(B.24)

Such representation allows for the easy calculation of surface properties using simple convolutions [11].

In addition to regular surface representations, surfaces may be represented intrinsically [39]:

\[ F(x, y, z) = a_1 x^2 + a_2 y^2 + a_3 z^2 + a_4 xy + a_5 xz + a_6 yz \]
\[ + a_7 x + a_8 y + a_9 z + a_{10} = 0. \]
(B.25)

Using the signs of the coefficients, \( a_i \), the points on the surface may be classified [35]. Several other measures have been used in the past to aid in the matching of objects and models. For example, the surface area, \( S_a \), of the surface \( S \) is given by integrating the area elements, \( \sqrt{EG - F^2} \), over the surface:

\[ S_a = \int_S \sqrt{EG - F^2} \, du \, dv. \]
(B.26)
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


감사의 글

먼저 미안하지만 이 논문을 끝까지 무사히 완성할 수 있게 부족한 저에게 건강과 지혜를 허락해 주시고, 인도해주시기 하나님께 모든 영광을 돌립니다.

이 논문을 완성하기까지 주위의 모든 분들로부터 수많은 도움을 받았습니다. 특히 중앙대학교에 입학한 이후로 줄곧 가까이에서 아버지처럼 늘 저를 응

답없이 격려해주시고, 풍부한 지도와 지도가 지도를 일관하여 지도로 가는 감사를 드립니다. 그러

고 영상처리 연구실은 백준기 교수님과 컴퓨터 비전 및 그래픽스 연구실의 제

선배님과 저가 존경하는 홍현기 교수님께 진심으로 감사드립니다. 사실 제가 이

나아 이렇게 논문을 완성할 수 있게 된 것은 모두 분들께 배운 결과가 아닌

가합니다. 또한 저의 보살핌 없는 논문을 심사해주시던 남서울대학교 김태온

교수님과 열려 미국 Tennessee주 Tennessee 대학의 M. A. Abidi 교수님들께도

두손으로 감사드립니다. 전자공학과 학부와 전자공학과 대학원 그리고 첨단영

상 대학원 두루 거쳐서 여러 교수님들의 정성어린 지도를 통해 배운 지식들

이 오늘 제가 이렇게 결실을 맺을 수 있게 되는 빌바오가 되었습니다. 그분들

께 진심으로 다시 한번 고개숙여 감사드립니다.

중-고등학교 시절부터 생활을 같이 하며 동생탁한해진 친구들에게 감사의

말을 전합니다. 김유승, 강용화, 김영현, 박성호, 이우승, 이들은 저의 가장 가

까운 사람들로며, 함께 한 시간들은 저의 성장의 활성소로서 항상 큰 힘이 되

았습니다. 그리고 같은 연구실에서 함께 밤을 갖는 체 발을 주고받으며 생활

한 선배들인 지식의 보고 장성갑 교수님, 언제나 신사인 용관형, 밀을먹는 정

수형, 과학기술 선배이자 친형 같은 경수형, 뷔심이 좋은 용인형, 그리고 동료

들인 노총각 대환이, 유직한 뎅쟁이, 재치있는 대현이, 장완, 난우, 경진, 정훈,

감히 이를 부르기로 갈색의 작은종수, 상훈, 국보 등 모두에게 감사의 말을 전

합니다.

끝으로 오늘의 제가 있을 수 있도록 사랑으로 키워 주신 아버님에게 계시

는 나의 영원한 명을 아버지와, 지금까지도 고생많으신 어머님과 하나밖에 없

는 여동생, 또한 친아비처럼 저 뒤에서 독수리가 자를 독특히 응원해주시는

저의 장인여론, 제 아내보다 저에게 신경을 더 써주시는 강보남, 그 외 여러 가

족친지를에게 감사드립니다. 저의 이 작은 결실이 그분들께 조금이나마 보답

이 되기를 바랍니다. 무엇보다 제 뒤에서 몹시가 지나도록 눈물과 인내로 그림

자처럼 손발이 되어 자를 뒷바라지 해준 제 사랑하는 아내 현선, 이 세상에서

저를 담은 제 분신인 풍직한 제 두아들, 주익, 주원이에게 이 논문을 바칩니다.