To the Graduate Council:

I am submitting herewith a thesis written by Sreenivas Rangan Sukumar entitled "Curvature Variation as Measure of Shape Information." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Mongi A. Abidi

Major Professor

We have read this thesis and
recommend its acceptance:

Michael J. Roberts

David L. Page

Andrei V. Gribok

Accepted for the Council:

Anne Mayhew

Vice Chancellor and
Dean of Graduate Studies

(Original signatures are on file with official student records.)

Curvature Variation as Measure of Shape Information

A Thesis
Presented for the
Master of Science Degree
The University of Tennessee, Knoxville

Sreenivas Rangan Sukumar

December 2004

Acknowledgements

"Experience is the toughest of teachers, she gives me tests first and lessons later.
What I learn is simply information, Experience of information is knowledge.
I've learnt that science is organized knowledge, but wisdom is organized life.
And most importantly, I have learnt that I definitely have a lot more to learn…!"

It was not long ago when I used to think that this section of the document was just another formality until I realized its significance as a medium to express my gratitude to all the people without whose contribution this work that I am presenting would have remained a dream.

Words are not enough to express what they have done to me. They have given me the life, the vision and their happiness for my well being and but for their monotonically increasing affection I am sure I would not be what I am today. It is my pleasure to dedicate this work to my parents Chellappa Sukumar and Malathi Sukumar.

There is an adage that says "You can only take the horse to the pond but cannot make it drink". It has been an excellent learning experience under the academic guidance of Dr. Mongi Abidi. He showed me the pond in the Imaging, Robotics and Intelligent Systems Lab and provided me with the right kind of academic, financial and philosophical support all through my pursuit.

I shall never forget the quotation on his desk "It is not important what you learn but it is important how you teach it to others". It takes a lot to be an unselfish teacher that you have been to me. "Thanks a million Dr. Page". I should thank Dr. Andrei Gribok for the lively discussions of high technical impact on this work. I would like to take this opportunity to appreciate the efforts of Dr. Andreas Koschan and Dr. Besma Abidi whose rigorous review and feedback has added to my learning experience in the lab. Dr. Roberts has helped me with the documentation and review of my work. It is not fair if I do not mention the efforts of Tak Motoyama in the data acquisition process.

A significant amount of the learning process at the graduate level has to be attributed to my peers at the lab. Faysal, Brad and Yohan have been my inspirations towards a PhD degree and the weekly brainstorming sessions with them have been a good platform to launch new ideas. I would like to extend a sincere thanks to Umayal for bequeathing her experience with the IVP Ranger to me and I shall never forget our evenings at the "Motor pool" scanning under vehicle range data. It's my pleasure to acknowledge Ashwin, Madhan, Sampath, and Rishi with whom I share "the" cherishable moments at Knoxville.

Sincere thanks to you all…

Abstract

In this thesis, we present the Curvature Variation Measure (CVM) as our informational approach to shape description. We base our algorithm on shape curvature, and extract shape information as the entropic measure of the curvature. We present definitions to estimate curvature for both discrete 2D curves and 3D surfaces and then formulate our theory of shape information from these definitions.

With focus on reverse engineering and under vehicle inspection, we document our research efforts in constructing a scanning mechanism to model real world objects. We use a laser-based range sensor for the data collection and discuss view-fusion and integration to model real world objects as triangle meshes. With the triangle mesh as the digitized representation of the object, we segment the mesh into smooth surface patches based on the curvedness of the surface. We perform region-growing to obtain the patch adjacency and apply the definition of our CVM as a descriptor of surface complexity on each of these patches. We output the real world object as a graph network of patches with our CVM at the nodes describing the patch complexity. We demonstrate this algorithm with results on automotive components.

Contents

1 INTRODUCTION 1
1.1 Motivation 2
1.2 Proposed Approach 4
1.3 Document Organization 6

2 LITERATURE REVIEW 7
2.1 Cognition and Computer Vision 7
2.2 Shape Analysis on 2D Images 8
2.2.1 Classification of Methods 8
2.2.2 Contour-Based Description 10
2.2.3 Region-Based Description 13
2.3 Shape Analysis on 3D Models 16
2.3.1 Classification of Methods 16
2.3.2 Feature Extraction 17
2.3.3 Descriptive Representation 19
2.3.4 Shape Histograms 20
2.3.5 Topology Description 21
2.4 Summary 23

3 DATA COLLECTION AND MODELING 26
3.1 Range Data Acquisition 26
3.1.1 Range Acquisition Systems 26
3.1.2 Range Sensing Using the IVP Range Scanner 27
3.2 Solid Modeling from Range Images 33
3.2.1 Modeling Automotive Components for Reverse Engineering 33
3.2.2 Modeling Automotive Scenes for Under Vehicle Inspection 36

4 ALGORITHM OVERVIEW 39
4.1 Algorithm Description 39
4.1.1 Informational Approach to Shape Description – Curvature Variation Measure 40
4.1.2 Curvature-Based Automotive Component Description 41
4.2 Building Blocks of the CVM algorithm 43
4.2.1 Differential Geometry of Curves and Surfaces 43
4.2.2 Curvature Estimation 45
4.2.3 Density Estimation 48
4.2.4 Information Measure 56

5 ANALYSIS AND RESULTS 59
5.1 Implementation Decisions on the Building Blocks 59
5.1.1 Analysis of Curvature Estimation Methods 59
5.1.2 Density Estimation for Information Measure 63
5.2 State-of-the-Art Shape Descriptors 66
5.3 Results of our Informational Approach 70
5.3.1 Intensity and Range Images 70
5.3.2 Surface Ruggedness 70
5.3.3 3D Mesh Models 72

6 CONCLUSIONS 81
6.1 Contributions 81

6.2 Directions for the Future 82
6.3 Closing Remarks 83

BIBLIOGRAPHY.....84

VITA 100

List of Tables

Table 2.1:Qualitative comparison of 3D shape analysis methods with focus on algorithm efficiency. 24

Table 2.2: Qualitative comparison of 3D shape analysis methods with focus on effective description. 25

Table 4.1: Kernel functions. 52

Table 4.2: List of entropy type measures of the form $H(P)=h\left(\frac{\sum_{i=1}^{M} v_i \cdot \varphi_1(p_i)}{\sum_{i=1}^{M} v_i \cdot \varphi_2(p_i)}\right)$. 57

List of Figures

Figure 1.1: Engineering and reverse engineering 2
Figure 1.2: Under vehicle inspection and surveillance 3
Figure 1.3: Proposed approach 5
Figure 2.1: Classification of shape description and representation adapted from [Zhang, 2004] 9
Figure 2.2: [Reproduced from Belongie, 2003] Shape Contexts 11
Figure 2.3: Classification of methods on 3D data 17
Figure 3.1: IVP Ranger SC-386 range acquisition system 28
Figure 3.2: Triangulation and range image acquisition 30
Figure 3.3: The process of calibration 32
Figure 3.4: Graphical User Interface 33
Figure 3.5: Block diagram of a laser-based reverse engineering system 34
Figure 3.6: Model creation 35
Figure 3.7: Data acquisition for under vehicle inspection 38
Figure 4.1: A circle and an arbitrary object 40
Figure 4.2: Block diagram of our CVM as the informational approach to shape description 41
Figure 4.3: Block diagram of curvature-based vehicle component description algorithm including path decomposition and CVM computation 42
Figure 4.4: Illustration to understand curvature of a surface 44
Figure 4.5: Illustration that shows the effect of bin width on density estimation using a histogram 49
Figure 4.6: Different methods used to estimate the density of the same dataset. Reprinted from [Silverman, 1986] 51
Figure 4.7: Effect of bandwidth parameter on kernel density 53
Figure 4.8: Resolution issue with Shannon type measures 58
Figure 5.1: Neighborhood of a vertex in a triangle mesh 60

Figure 5.2: Curvature analysis – Multi-resolution error analysis experiment with four different approaches to curvature estimation on triangle meshes 62

Figure 5.3: Curvature analysis – Error in curvature of a sphere at multiple resolutions 64

Figure 5.4: Curvature analysis – Variation in curvature for surface description 65

Figure 5.5: Curvature-based descriptors..... 67

Figure 5.6: Implementation of Shape Distributions 68

Figure 5.7: Shape Distributions and its uniqueness in description 69

Figure 5.8: Shape complexity measure– using Shannon's definition of information . 71

Figure 5.9: Shape information and surface ruggedness 72

Figure 5.10: Shape information divergence from the sphere – Experimental results on super quadrics 73

Figure 5.11: Surface description results - surface, curvature and density of curvature of (a) Spherical cap (b) Saddle (c) Monkey saddle 75

Figure 5.12: Multi resolution experiment on the monkey saddle – The surface, its curvature density and the measure of shape information 76

Figure 5.13: CVM graph results on simple mesh models: curvedness-based edge detection, smooth patch decomposition and graph representation..... 78

Figure 5.14: CVM graph results on automotive parts: curvedness-based edge detection, smooth patch decomposition and graph representation 79

Figure 5.15: CVM graph results on an under vehicle scene 80

1 INTRODUCTION

Have we ever realized how easy it has been for us to locate a friend at the shopping center? How quickly we recollect something by looking at a photograph, and how accurately we approximate distance? It is indeed amazing to realize the design of 126 million receptors compactly packed into nerve endings and muscles that coordinate so impeccably well to process visual information that would require a bandwidth of 600 terahertz and processing capability of 2 terabytes per second. We are just measuring the sensing capability of the eye; not to forget the extremely fast and meticulous brain that does the processing at that bandwidth and with incredible accuracy and precision.

As computer vision researchers, we acknowledge the uncanny ability of our human visual system in object detection and recognition, to address the complexities involved in imparting this intelligence to a computer. The first and foremost computational hurdle is that of variability. A vision system needs to generalize across huge variations of an object to viewpoint, illumination, occlusions and many such factors and still be very specific. For more than two decades researchers have been fighting such factors and the lack of important depth information with intensity images. With increase in computational speed and capabilities of the electronic world, we now deal with 3D data. The 3D sensors, in addition to having the capabilities of traditional cameras, require processing resources to extract depth information. By 3D data, we mean digitized representations of the real world objects that we can visualize and understand using a computer. Computers can be programmed to understand a specific domain of objects by extracting features from their digital representation. An important feature used for image understanding is shape. Shape is interpreted as the geometric description of an object, and shape analysis refers to the process of feature extraction followed by feature matching. In this thesis we present the pipeline for 3D data collection and discuss a new shape analysis algorithm that we have developed. We base our algorithm on a feature that we define as the Curvature Variation Measure (CVM). We have implemented the algorithm in an application to reverse engineering and vehicle inspection that we elaborate in Section 1.1.

1.1 Motivation

Computer aided design (CAD) combined with computer aided manufacturing (CAM) has revolutionized many engineering disciplines since the 1980's. In particular, CAD and CAM technologies have catered to the needs of the automobile manufacturers. A designer can now rapidly fabricate a real-world tangible object from a conceptual CAD description. The process of designing and manufacturing components using a computer is often referred to as computer aided engineering. In this context, we would like to introduce the idea of reverse engineering that begins with the product and works through the design process in the opposite direction to arrive at a product definition statement. In doing so, it uncovers as much information as possible about the design ideas that were used to produce that particular product. By design ideas, we mean the shape and topology of the surfaces used at the time of modeling. At this point, we would like to emphasize that our focus is only on the geometric aspect of reverse engineering and not on the functional aspect of these mechanical components.

Reverse engineering aids the electronic dissemination and archival of information in addition to the prospect of re-creating an out-of-production component. More recently, reverse engineering techniques play a significant role in real-time rapid inspection and validation in the production line. The traditional approach to reverse engineering has been the use of coordinate measuring machines (CMM) that require a probe in contact with the object at the time of digitization. Though CMMs are accurate some applications demand non-contact digitization.

In Figure 1.1 we illustrate the process of reverse engineering as the reversal of CAM. We show that the reverse engineering of the disc brake involves acquiring 3D position data in the point cloud. We then represent geometry of the object in terms of surface points and tessellated piecewise smooth surfaces. We now need to represent the point cloud in a form that the CAM system can interpret and manufacture.

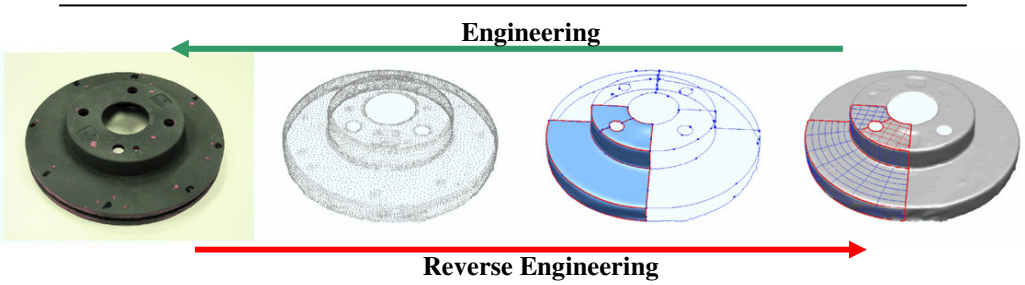


Figure 1.1: Engineering and reverse engineering.

Another application that our research efforts target is that of under vehicle inspection. Vehicle inspection has been traditionally accomplished through security personnel walking around a vehicle with a mirror at the end of a stick. The inspection personnel are able to view underneath a vehicle to identify weapons, bombs and other security threats. The mirror-on-the-stick system allows only partial coverage under a vehicle and is restricted by ambient lighting. The inspecting personnel are also at risk. As part of the Security Automation and Future Electromotive Robotics (SAFER) program we aim at developing a robotic platform that deploys "sixth sense" sensors for threat assessment. We propose the idea of incorporating a 3D range sensor on the robotic platform. The idea is to be able to extract the 3D geometry of the undercarriage of automobiles. With prior manufacturer's information on the components that make the undercarriage of the vehicle, we believe that it will be possible to identify foreign objects in the scene. For example in Figure 1.2 we show the robotic platform and the 3D geometry of the scene containing the muffler, shaft and the catalytic converter. It will not be possible to extract complete geometry of the undercarriage without dismantling the automobile. We hence need a representation scheme that maps the shape sensed from the scene to the CAD description and that is robust with occluded data.

Though vehicle inspection and reverse engineering appear as different applications, they share the same processing pipeline as a computer vision task of designing a system that can capture the geometric structure of an object and store the subsequent shape and topology information. We discuss the use of laser-based range scanners for the extraction of 3D geometry and a curvature-based shape analysis algorithm based on our CVM to interpret surface topology.



Robotic Platform **Under Vehicle Scene** **3D Geometry**

Figure 1.2: Under vehicle inspection and surveillance.

1.2 Proposed Approach

Shape analysis is an age-old research topic and has been pursued since the dawn of image processing and computer vision. Literature on shape extraction from intensity images is vast and gives good insight into why vision research with intensity images has not been very successful. Most of the methods we have studied show promise with more and almost complete information with 3D data. Though 3D data acquisition and processing is relatively new, there are a few important contributions in our context of shape similarity and shape description all motivated by the challenge of object recognition. We survey the literature on shape analysis applied to intensity images and also summarize recent and ongoing work in 3D computer vision.

Computer vision systems seek to develop computer models of the real world through processing of image data from sensors. In Figure 1.3(a) we present the flow diagram of our proposed approach. We begin with the data acquisition (Figure 1.3 (b)) using laser-based range scanners and the process of creating CAD models using these scanners. We acquire range images using the laser-illuminated active range sensor from the Integrated Vision Products Inc. (IVP). A range image is a 2D matrix with values proportional to the distance between the sensor and the object. We acquire range information from multiple views of the object to make sure that we have sufficient data to represent the object completely. We then transform the range data from the camera coordinate frame to the real world and integrate the multi-view point clouds into a single global reference frame. We reconstruct triangle meshes from the point clouds and use it as our input for the shape analysis.

We base our shape analysis algorithm on the part-based perception model [Stankiewicz, 2002]. With automotive components, our task is simplified because the components are man-made and manufacturing limitations restrict us to smooth (mostly planar and cylindrical) patches. We hence propose that surface shape description of each of the parts and the connectivity of parts can uniquely describe the object. In describing surfaces and surface complexity we chose curvature to extract "shape information". We chose curvature because it is an information preserving feature, invariant to rotation and possesses an intuitively pleasing correspondence to the perceptive property of "simplicity". We decompose the object of interest into a set of patches and assign a Curvature Variation Measure (CVM) to each of these patches and represent the object as a patch adjacency graph. Our graph representation when extended to scenes with occlusions can still yield satisfactory results.

Consider the example in Figure 1.3 again. We first decompose the triangle mesh model into smooth patches. We show the disc brake model and decompose it into four parts. We have shaded each of these parts with a different color. We base our surface patch decomposition on the definition of curvedness in [Dorai, 1996]. Curvedness identifies sharp edges and creases.

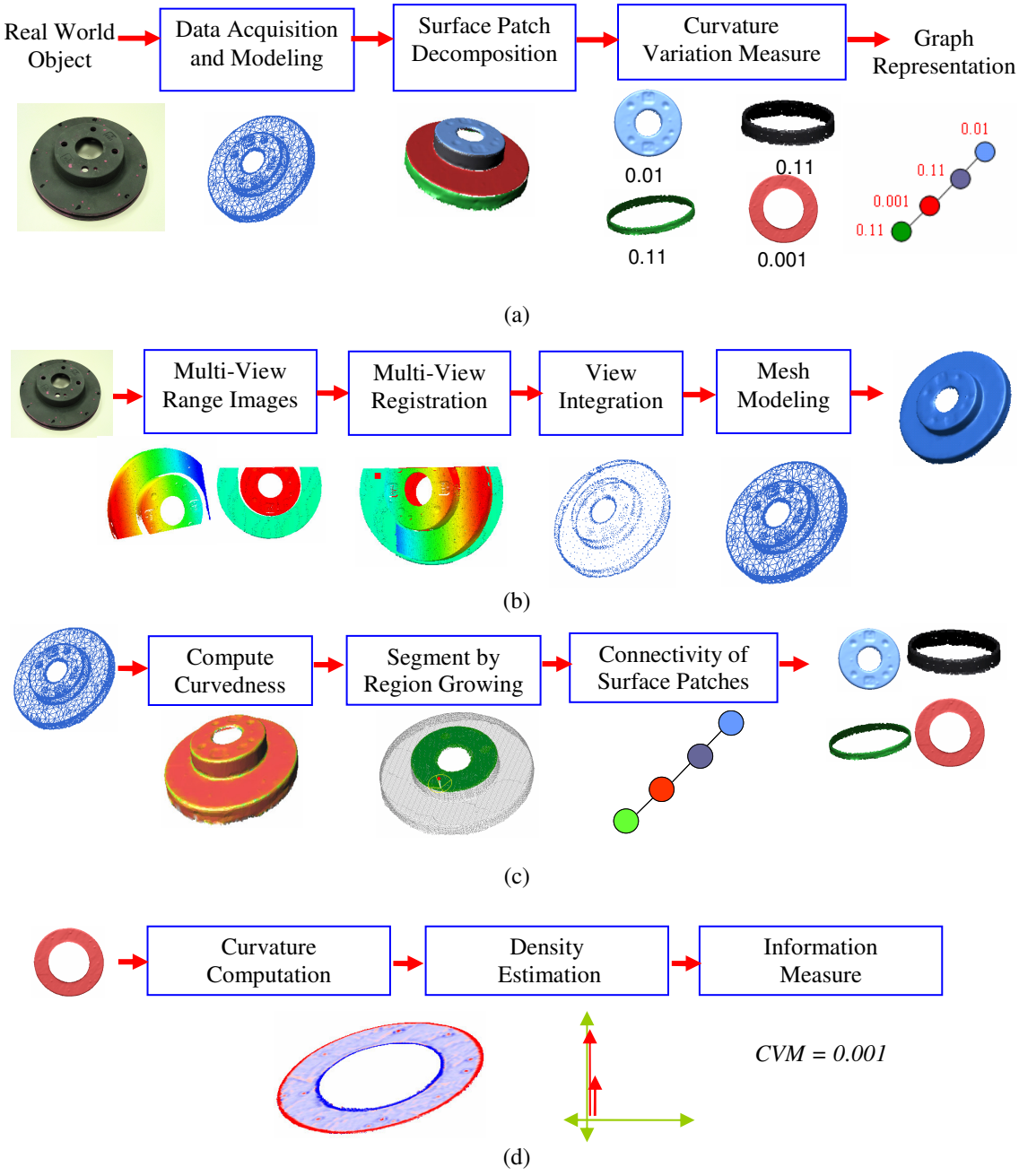


Figure 1.3: Proposed approach. (a) Shape analysis based on our curvature variation measure - flow diagram. (b) Data acquisition and modeling. (c) Surface patch decomposition. (d) Curvature variation measure.

We then perform region-growing segmentation and save the patch adjacency information as illustrated in Figure 1.3(c).

Now that we have segmented surface patches that make the object, we compute the curvature variation measure on each of these patches (Figure 1.3(d)). We have borrowed concepts from Shannon's idea [Shannon, 1948] of measuring information on a probabilistic framework. We hence define the curvature variation measure as the entropy of curvature along that surface. We present a brief analysis on various curvature estimation methods on triangle meshes and reiterate the importance of bandwidth optimized density estimation to stabilize the information measure. Our modification of Shannon's definition of entropy is normalized and invariant to scale. The normalized resolution invariant measure attempts to quantify the complexity of the surface by a single number. Similar shapes at different scales will have equal measures.

1.3 Document Organization

The remainder of this thesis documents the theory and results of our data collection and CVM algorithm. Chapter 2 presents a survey of the literature on the shape analysis and description of 2D images and 3D models. Here we explain why methods in 2D cannot be extended to 3D and discuss the scope for extending the state of the art. Then, we present our experience with the data acquisition using a laser-based scanner for creating 3D models of automotive components and scenes under the vehicle in Chapter 3. Chapter 4 documents the theory that supports our shape analysis algorithm. We test our algorithm on the acquired data and present our results in Chapter 5. These experimental results demonstrate capabilities of our algorithm and its scope as an object recognition system. Finally, we conclude with possible extensions in Chapter 6.

2 LITERATURE REVIEW

In this chapter we present a review of the research literature. In Section 2.1 we introduce the reader to shape and its implication to computer vision and briefly review some key methods on 2D images in Section 2.2. We discuss contemporary research in 3D computer vision for shape analysis in Section 2.3 and summarize our survey in Section 2.4.

2.1 Cognition and Computer Vision

The human cognitive system is designed to interpret sensory data with such remarkable speed and accuracy that we fail to appreciate millions of computations involved in a common event of identifying an object. An impressive component in human perception is our ability to recognize 3D objects from their 2D retinal projections. Stankiewicz outlines human visual perception into three possible hypotheses, namely the feature model approach, alignment model approach and the part-based approach [Stankiewicz, 2002]. Feature models propose that the visual system does not match a precise numerical array of an object with another but remembers a collection of features in memory. According to this approach the location of the features in a particular image is less significant than its presence in the image. The feature model approach fails with increasing occlusions and is less reliable when the spatial relationship between the features and the image are vital in recognizing the object. Alignment models make use of the spatial information to compensate for viewpoint changes but do not consider occlusions. They can handle Euclidean transformations such as the rotation, translation and scaling and are accepted to be robust in comparison with the feature models. Part-based models operate by decomposing an object into its constituent parts. The approach uses image features to describe the shape of parts in addition to documenting relationships between parts. The part-based model has not met with great success in computer vision because of the insufficiency in intensity images to segment objects as parts, but with the increasing computational capabilities and improvements in sensor technology towards 3D imaging, part-based models are a good prospect.

Shape is the geometric information invariant to a particular class of transformations such as affine, translation, rotation and scaling and is considered to be the "words" of the visual language. Shape analysis is an important aspect in image understanding. Since so many objects in our world are strongly determined by geometric properties, the applications of shape analysis extend over a broad spectrum of science and technology. Indeed, when properly and carefully applied shape analysis provides rich potential for applications in diverse areas, spanning computer vision, graphics, material sciences, biology and even neuroscience.

2.2 Shape Analysis on 2D Images

Shape description looks for effective and perceptually important shape features based on either shape boundary information or interior content. By perceptually similar shapes we are referring to shapes that are rotated, translated, scaled and are affine transformed. Many shape representation techniques have been developed in the past and shape analysis still remains as an interesting field of research. A few such representation techniques are the shape signatures, shape histograms, moments, curvature, shape context and shape matrix. We would like to direct the reader to [Zhang and Lu, 2004] for a recent and comprehensive survey on 2D shape representation for various applications.

2.2.1 Classification of Methods

Shape representation techniques are generally classified into two classes based on whether shape features are extracted from the contour only or from the whole region. Zhang and Lu [Zhang and Lu, 2004] subdivide each of these classes further into structural and global approaches based on the primitives used to describe the shape. They discuss methods that operate on the space domain and transform domain to extract shape information and classify shape description methods as shown in Figure 2.1.

Contour-based approaches are more popular in computer vision literature. Such methods assume that human beings discriminate shapes mainly by their feature contours. The contour-based approach is limited by noise and by data that do not have sufficient information (occlusions) in the boundary contour. Region-based methods are considered to be more robust and are dependable for accurate retrieval as they attempt to extract shape information from the entire region and not just its boundary.

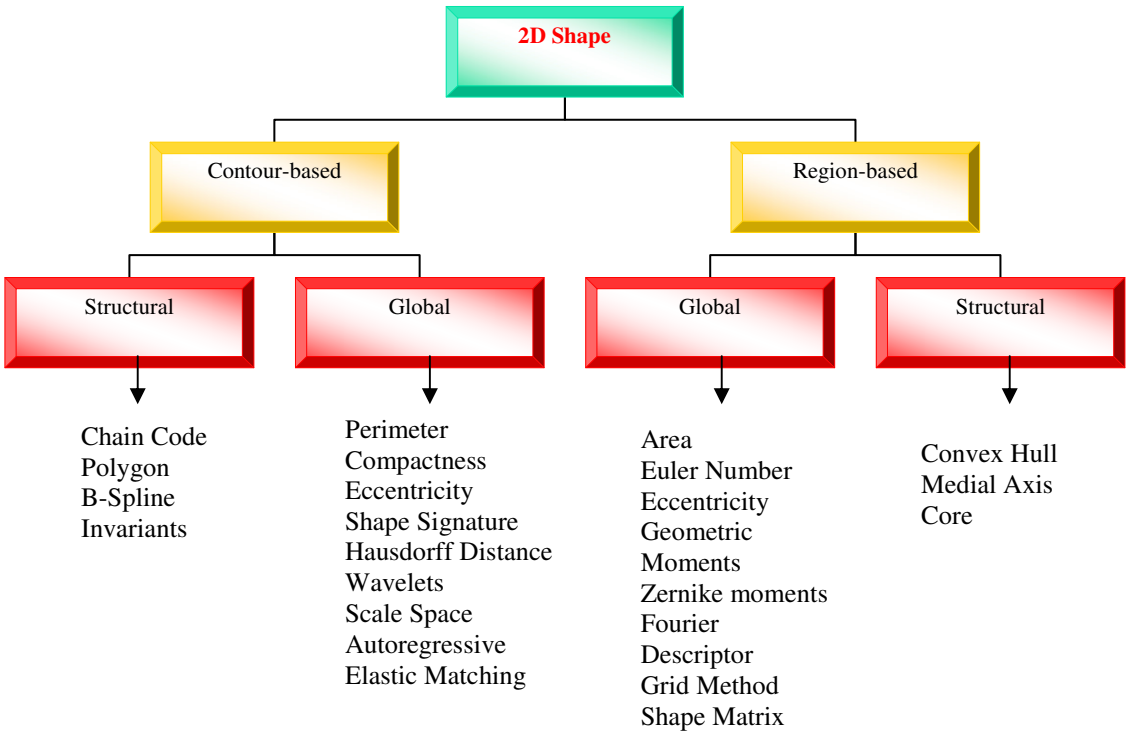


Figure 2.1: Classification of shape description and representation adapted from [Zhang, 2004].

2.2.2 Contour-Based Description

Contour-based shape representation techniques extract shape information from the boundary. There are generally two approaches for contour shape modeling: the continuous global approach and discrete structural approach. The global approach makes use of feature vectors derived from the boundary to describe shape. The measure of shape similarity is the metric distance between feature vectors. The discrete approach to shape analysis represents the shape into a graph or tree of segments (primitives). The shape similarity is deduced by string or graph matching.

We begin our analysis with the contour-based global shape description methods. The most commonly used global shape descriptors are surface area, circularity, eccentricity, convexity, bending energy, ratio of principle axis, circular variance and elliptic variance and orientation. These simple descriptors are not suitable standalone descriptors but are usually used to discriminate shapes with large differences or to filter false hits. Some of these are also used in combination with the other descriptors for shape description. The efficiency of such descriptors is discussed in [Peura and Ivarinen, 1997].

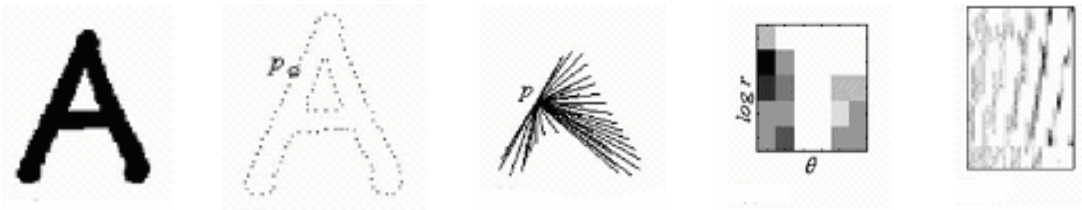
A few space-domain techniques compute correspondence-based shape measures using the point-point match where each point on the boundary is considered to be a contributor to shape. Hausdorff distance is a classical correspondence-based shape matching method, often used to locate objects in an image and measure similarity between shapes as discussed in [Huttenlocher, 1992].

Given two shapes $S_1 = \{a_1, a_2,,\ldots\ldots ,a_p\}$ and $S_2 = \{b_1, b_2,,\ldots\ldots ,b_p\}$ represented as two sets of points, the Hausdorff distance is defined as

$$H_d(S_1,S_2) = max(h(S_1,S_2),h(S_2,S_1)\},\ \ h(S_1,S_2) = \max_{a\in S_1} \min_{b\in S_2} \| a-b \| \tag{2.1}$$

where ||.|| refers to the Euclidean distance.

The Hausdorff distance measure is too sensitive to noise and is useful for partial matching invariant to rotation, scale and translation. Rucklidge improves it with a new measure between two datasets using a prohibitively expensive matching procedure that tackles different orientations, positions and scales [Rucklidge, 1997]. A more recent but similar kind of approach to shape matching was introduced by the name of "shape contexts" in [Belongie et al., 2002].Shape contexts claim to extract global feature at every point reducing the point-point matching into a matrix matching of contexts. To extract the shape context at a point *p* on the boundary, the vectors that connect *p* and each of the other points on the boundary are computed. The length and orientation of these vectors are quantized into a log-space histogram map for that point *p* to account for additional sensitivity to neighboring points. These histograms are flattened and concatenated to form the context of the shape as shown in Figure 2.2.



(a) (b) (c) (d) (e)

Figure 2.2: [Reproduced from Belongie, 2003] Shape Contexts. (a) A character shape. (b) Edge image of (a). (c) The histogram is the context of the point p. (d) The log-space histogram. (e) Each row of the context map is the flattened histogram of each point context, the number of rows is the number of sampled points.

Davies [Davies, 1997] describes shape signatures as a one-dimensional function derived from the shape boundary points. Some shape signatures that can be found in the literature are the centroidal profile, complex coordinates, tangent angle, cumulative angle, chord length and curvature. Shape signatures are usually normalized in scale. Translational and rotational invariance is achieved by a shift search procedure of the one dimensional function extracted from the shape boundary. Shape signatures require further processing in addition to the high matching cost to overcome their sensitivity and improve robustness. Autoregressive models [Chellappa and Bagdazian, 1984] are stochastically defined predictor-based methods dependent on modeling the shape into a 1D function.

Boundary moments are extensions of shape signatures to reduce the dimensionality of the boundary representation. If *z(i)* is an extracted shape signature of a boundary, the rth moment and the central moment μ_r can be estimated as shown in Equation 2.2 and 2.3.

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r \tag{2.2}$$

$$\mu_r = \frac{1}{N}\sum_{i=1}^{N}[z(i) - m_1]^r \tag{2.3}$$

where *N* is the number of points representing the boundary. The normalized moments are invariant to shape translation, rotation and scaling. As discussed in [Gonzalez, 2002] the amplitude of the shape signature can be treated as a random variable and its moments computed using its histogram. These moments are easily computable but have no physical significance.

Bimbo [Bimbo, 1997] implements elastic matching for shape-based image retrieval. A deformed template is generated as the sum of the original template and a warping deformation. The similarity between the original shape of the template and the object is obtained by minimizing the compound function, which is the sum of the strain energy, bending energy and the deviation measure of the deformed template with the object. He defines shape complexity as the number of curvature zero crossings and a correlation between curvature functions of the template and the object. The classification is performed by a back propagation algorithm neural network.

Most of the space-domain techniques discussed in literature are sensitive to noise and boundary deviations. Spectral-domain techniques resolve the noise issues. The simplest of the spectral domain descriptors are the Fourier descriptors [Zhang and Lu, 2002] and the wavelet descriptors [Yang et al., 1998]. They are derived from the one-dimensional shape signatures of the shape function. They are easy to compute, normalize and bypass the complex matching stages of the shape signature based methods. Zhang and Lu [Zhang and Lu, 2002] argue that the centroidal profile is the most efficient shape descriptor to be used in combination with Fourier descriptors.

Structural shape representation is yet another approach to analysis of shape description as shown in Figure 2.1. With the structural approach, shapes are broken down into segments called shape primitives. Structural methods differ in the selection of primitives and organization of primitives for shape representation. Some of the common methods of boundary decomposition are based on polygonal approximation, curvature decomposition and curve fitting. The result of the decomposition is encoded in a general string form that can be used with a high-level syntactic analyzer for shape comparison tasks.

Chain code described by [Freeman and Saghri, 1978] is a sequence of unit-size line segments with a given orientation. The unit vector method describes any arbitrary curve as a sequence of small vectors of unit length in a set of directions. The chain codes need to be independent of the starting boundary pixel. The independence is achieved by a good scheme that defines the characteristics of a starting pixel or by representing the chain code as differences in successive directions. Chain codes used for object recognition and matching are not scale invariant though they are rotation invariant. Polygonal decomposition methods discussed in [Groskey et al., 1992] break a given boundary into line segments by using polygon vertices as primitives. Feature strings are created with four elements such as the internal angle, distance from the next vertex and the coordinates of the vertex. The similarity of shapes is the editing distance between two feature strings representing the shape. Mehrotra and Gary in [Mehrotra and Gary, 1995] represent shape as a series of interest points from the polygonal boundary approximation. These points are mapped onto a new scale and rotation invariant basis to represent shape in a new coordinate system. Berretti et al. [Berretti et al., 2000] extend [Groskey and Mehrotra, 1990] for shape retrieval by

defining tokens as the zero crossings of Gaussian curvature and shape similarity is the Euclidean distance between primitives. Dudek and Tsotsos [Dudek and Tsotsos, 1997] use curvature scale spaces for shape matching. In this approach, shape primitives are first obtained from a curvature-tuned smoothing technique. A segment descriptor consists of the segment's length, ordinal position, and curvature turning value extracted from each of these primitives. A string of segment descriptors is then created to describe the shape. For two shapes A and B represented by their string descriptors, a model-by-model match using dynamic programming is exploited to obtain the similarity score of the two shapes. To increase robustness and to save matching computation, the shape features are put into a curvature scale space so that shapes can be matched even in different scales. However, due to the inclusion of length in the segment descriptors, the descriptors are not scale invariant.

Another interesting approach to the analysis of shape is syntactic analysis in [Fu, 1974] that attempts to simulate the structural and hierarchical nature of the human vision system. In syntactic methods, shape is represented by a set of predefined primitives. The set of predefined primitives is called the codebook and the primitives are called code words. The matching between shapes can use string matching by finding the minimal number of edit operations in trying to convert one string to another However, it is not practical in general applications due to the fact that it is not possible to infer a pattern of grammar which can generate only the valid patterns. In addition, this method needs a priori knowledge for the database in order to define codeword or alphabets. Shape invariants make use of simple shape descriptors such as the cross ratio, length and area to derive a multi-valued signature. Kliot and Rivlin [Kliot and Rivlin, 1998] propose a multi valued matrix that can be used for matching two curves. This method can be improved with a histogram matching stage before the matrix matching. Squire and Caelli in [Squire and Caelli, 2000] use the density function of piecewise linear curves for their shape invariant. The histogram of the shape invariant signature is fed into a neural network for classification.

2.2.3 Region-Based Description

Region-based techniques take into account all the pixels within a shape region to obtain the shape representation, rather than only use boundary information as in contour-based methods. Common region-based methods use moment descriptors to describe shapes. Structural methods include grid method, shape matrix, convex hull and medial axis. Global methods treat shape as a whole; the resultant representation is a numeric feature vector which can be used for shape description while structural methods break down the shape into segments. Similarity between global shape descriptors is simply the metric distance between their feature vectors. Some of the global descriptors are the geometric moment invariants and the algebraic moment invariants. One of the oldest global methods implemented for region-based description

is from Hu [Hu, 1962]. He used the work of nineteenth century mathematicians on images for pattern recognition.

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y), \quad p, q = 0,1,2.... \tag{2.4}$$

Lower-order geometric moments from Equation 2.4 are easy to compute and are sufficient for representing simple shapes. Algebraic moments [Taubin and Cooper, 1991] and [Taubin and Cooper, 1992] on the other hand are based on the central moments of predetermined matrices that can be constructed for any order and are invariant to affine transformations. Teague [Teague, 1980] defines orthogonal moments by replacing the x^p y^q term by the Zernike polynomials. Moment shape descriptors are concise, robust, and easy to compute and match. The disadvantage of moment methods is that it is difficult to correlate higher-order moments with the shape's physical features.

Among the many moment shape descriptors, Zernike moments [Jeannin, 2000] are the most desirable for shape description. Due to the incorporation of a sinusoid function into the kernel, they have similar properties of spectral features, which are well understood. Although Zernike moment descriptors have a robust performance, they have several shortcomings. First, the kernel of Zernike moments is complex to compute, and the shape has to be normalized into a unit disk before deriving the moment features. Second, the radial features and circular features captured by Zernike moments are not consistent, one is in the spatial domain and the other is in spectral domain. This approach does not allow multi-resolution analysis of a shape in the radial direction. Third, the circular spectral features are not captured evenly at each other and can result in loss of significant features which are useful for shape description.

To overcome these shortcomings, a generic Fourier descriptor (GFD) has been proposed by Zhang and Lu [Zhang and Lu, 2002]. The GFD is acquired by applying a 2D Fourier transform on a polar-raster sampled image using the Equation 2.5.

$$PF_2(\rho, \phi) = \sum_r \sum_i f(r, \theta i) exp\left[j2\pi \left(\frac{r}{R}\rho + \frac{2\pi i}{T}\varphi \right) \right] \tag{2.5}$$

Zhang and Lu show that GFD outperforms contour shape descriptors such as curvature scale spaces, Fourier descriptors and moment-based descriptors.

The grid shape descriptor proposed by [Lu and Sajjanhar, 1999] has been used in [Chakrabarti et al., 2000] and [Safar et al., 2000]. Basically, a grid of cells is overlaid on a shape; the grid is then scanned from left to right and top to bottom. The result is saved as a bitmap. The cells covered by the shape are assigned one and those not covered by the shape are assigned zero. The shape can then be represented as a binary

feature vector. The binary Hamming distance is used to measure the similarity between two shapes. To account for the invariance to Euclidean transformations the shape needs to be normalized. Chakrabarti et al. [Chakrabarti et al., 2000] improve the grid descriptor by using an adaptive resolution (AR) representation acquired by applying quad-tree decomposition on the bitmap representation of the shape.

Typically, shape methods use rectangular-grid sampling to acquire shape information. The shape representation so derived is usually not translation, rotation and scaling invariant. Extra normalization is therefore required. Goshtasby [Goshtasby, 1985] proposes the use of a shape matrix which is derived from a circular raster sampling technique. The idea is similar to normal raster sampling. However, rather than overlay the normal square grid on a shape image, a polar raster of concentric circles and radial lines is overlaid in the center of the mass. The binary value of the shape is sampled at the intersections of the circles and radial lines. The shape matrix is formed such that the circles correspond to the matrix columns and the radial lines correspond to the matrix rows. Prior to the sampling, the shape is scale normalized using the maximum radius of the shape. The resultant matrix representation is invariant to translation, rotation, and scaling. Since the sampling density is not constant with the polar sampling raster, Taza et al. represent shape using a weighed shape matrix, which gives more weight to peripheral samples in [Taza et al., 1989]. However, since a shape matrix is a sparse sampling of shape, it is easily affected by noise. Besides, shape matching using a shape matrix is expensive. Parui et al. propose a shape description based on the relative areas of the shape contained in concentric rings located in the shape center of the mass in [Parui et al., 1986].

Structural methods for region-based shape description usually involve the convex hulls and medial axis described in [Davies,1997], [Blum,1967] and [Morse,1994]. A region R is convex if and only if for any two points x_1; $x_2 \in$ R, the whole line segment x_1x_2 is inside the region. The convex hull of a region is the smallest convex region H which satisfies the condition R $\subset$ H. The difference H − R is called the convex deficiency D of the region R. The extraction of the convex hull can be achieved either using the boundary-tracing method from [Sonka et al., 1993] or by using morphological methods from [Gonzalez and Woods, 1992]. Since shape boundaries tend to be irregular because of digitization noise and variations in segmentation result in a convex deficiency that has small, meaningless components scattered throughout the boundary. Common practice is to first smooth a boundary prior to partitioning. The polygon approximation is particularly attractive because it can reduce the computation time taken for extracting the convex hull from $O(n^2)$ to $O(n)$ (n being the number of points in the shape). The extraction of convex hull can be a single process which finds significant convex deficiencies along the boundary. A fuller representation of the shape is obtained by a recursive process which results in a concavity tree. Here the convex hull of an object is first obtained with its convex deficiencies, then the convex hulls and deficiencies of the convex deficiencies are

found, and the recursion follows until all the derived convex deficiencies are convex. The shape can then be represented by a string of concavities (concavity tree). Each concavity can be described by its area, bridge (the line that connects the cut of the concavity) length, maximum curvature, distance from maximum curvature point to the bridge. The matching between shapes becomes a string or a graph matching.

Like the convex hull, a region skeleton is also employed for shape representation. A skeleton may be defined as a connected set of medial lines along the limbs. The basic idea of the skeleton is to eliminate redundant information while retaining only the topological information concerning the structure of the object that can help with recognition. The skeleton methods are represented by Blum's medial axis transform (MAT) [Blum, 1967]. The medial axis is the locus of centers of maximal disks that fit within the shape. The bold line in the figure is the skeleton of the shaded rectangular shape. The skeleton can then be decomposed into segments and represented as a graph according to a certain criteria. The matching between shapes becomes a graph matching. The computation of the medial axis is a rather challenging problem. In addition, medial axis tends to be very sensitive to boundary noise and variations. Preprocessing the contour of the shape and finding its polygonal approximation has been suggested as a way of overcoming these problems. But, as has been pointed out by Pavlidis [Pavlidis, 1982] obtaining such polygonal approximations can be quite sufficient in itself for shape description. Morse [Morse, 1994] computes the core of a shape from medial axis in scale space.

We conclude this section with a note that shape description from intensity images have to deal with view occlusions and lack of sufficient information. We now study some important methods used for shape analysis on 3D mesh models in Section 2.3.

2.3 Shape Analysis on 3D Models

In Section 2.2, we have reviewed techniques implemented for shape extraction in 2D intensity images. In the following section we present a classification of methods in the literature on digitized 3D representations. We follow the classification with a brief description of some interesting methods.

2.3.1 Classification of Methods

There is a multitude of techniques to assess the similarity among 2D shapes as discussed in the Section 2.2. Most of the techniques do not extend to 3D models because of the difficulty of extending parameterization of the boundary curve extracted from 2D to 3D. In simple words, given a 2D shape, its parameterization is a straightforward 1D curve. With a 3D real world object it is difficult because when it is projected onto a 2D image plane, one dimension of object information is lost. The 3D domain requires dealing with

objects of different genus which makes it impossible for most of the 2D similarity assessment methods extendable to 3D. The challenge in 3D computer vision is more than just the lack of information as in the 2D case and needs to address the computational effort and descriptive representation. The 3D data usually are represented as meshes or assemblies of simple primitives. The representation scheme is suitable for visualization but not for recognition and computer vision tasks. The process of shape assessment hence becomes a two step process: (1) the shape signature extraction and (2) the comparison of shape signatures with distance functions. Based on how the shape is extracted from the 3D model representation techniques can be classified as shown in Figure 2.3.

2.3.2 Feature Extraction

Feature extraction techniques usually attempt to represent the shape of the 3D object by a combination of one-dimensional feature vectors. A common approach for similarity models is based on the paradigm of feature vectors. A feature transform maps a complex object onto a feature vector in a multidimensional space. The similarity of two objects is then defined as the vicinity of their feature vectors in the feature space. Geometric parameters and ratios such as the surface area, volume ratio, compactness, Euler numbers and crinkliness have been used with limited discrimination capabilities.

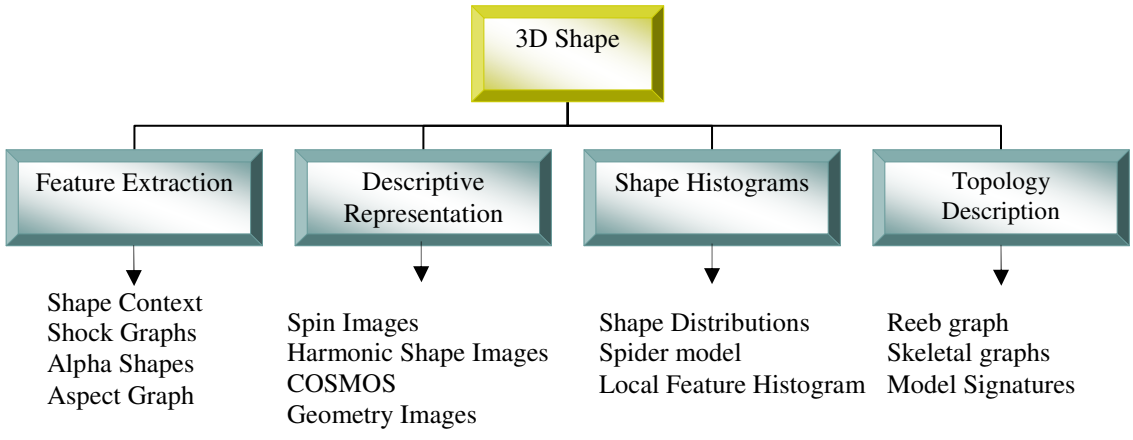


Figure 2.3: Classification of methods on 3D data.

Kortgen et al. [Kortgen et al., 2003] achieves shape matching by extending the 2D shape contexts [Belongie, 2003] to 3D. They use the shape context at a point on the surface as the summary of the global shape characteristics invariant to rotation, translation and scaling. Vranic and Saupe [Vranic and Saupe, 2001] propose a new method for shape similarity search on polygonal meshes. They characterize spatial properties of 3D objects such that similar objects are mapped as close points in the feature space. They then perform a coarse voxelization of the object in the canonical coordinate frame and compute the absolute value of the 3D Fourier coefficients as the feature vector. Vranic improves it further in [Vranic, 2003]. Ohbuchi et al. [Obhuchi et al., 2003] describe a multi-resolution analysis technique for the task of shape similarity comparison. They use 3D alpha shapes to generate a multi-resolution hierarchy of shapes of a given query object. They then follow that by applying a simple shape descriptor such as the D_2 shape function introduced by [Osada et al., 2002] on each of the multi-resolution representations and call it the multi-resolution shape descriptor.

Automated feature recognition has also been attempted by extracting instances of manufactured features from engineering designs. Henderson [Henderson et al., 1993] is an extensive survey of such methods that make use of a library of machining features for description. With the assumption of primitives, procedural methods proposed by Elinson et al. [Elinson et al., 1997] and Mukai et al. [Mukai et al., 2002] have applied constructive solid geometry (CSG's) to classify CAD models of mechanical parts. Their methods however cannot be extended to a more general class of shapes represented as point sets and meshes. Biermann et al. in [Biermann et al., 2001] propose Boolean operations of primitives for shape description. However, direct assessment of similarity between 3D models using Boolean operations is computationally slow due to the difficulty in aligning the models before performing the operation. With a large database, it is not a pragmatic solution. Zhang and Chen [Zhang and Chen, 2001] discuss efficient global feature extraction methods from the mesh representation.

Duda and Hart [Duda and Hart, 1973] have been extended by Khotanzad et al. [Khotanzad et al., 1980] to a subset of 3D moments that are invariant to rotation, translation and scaling that can be used as feature vectors for shapes as shown in Equation 2.6.

$$m_{pqr} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q z^r \, \rho(x, y, z) \, dxdydz$$

where $\rho\,(x, y, z)$ represents the point cloud of the model. (2.6)

Cybenko et al. [Cybenko et al., 1997] use second-order moments, spherical kernel moment invariants, bounding-box dimensions, object centroid and surface area along with a correlation metric for shape-similarity measurement. Elad et al. [Elad et al.,

2001] implement support vector machines for adaptively selecting weights for distance measurements between moments for shape similarity. Corney et al. [Corney et al., 2002] compute the Euclidean distance between simple geometric ratios for a shape similarity measure. Cyr and Kimia [Cyr and Kimia, 2001] use a shock graph-based shape similarity metric to assess the similarity between 3D models. Adjacent views are clustered in, thus generating the aspect using a seeded-region growing technique that satisfies the local monotonicity and specific distinctiveness of the aspect view criteria. The comparison of two 3D models is achieved by matching the 2D aspect views.

2.3.3 Descriptive Representation

In this category of methods, shape matching is achieved through an intermediate representation that aides a matching stage. These methods are usually robust but are computationally expensive. Usually the 3D information is broken down into a stack of 2D descriptors on which robust 2D shape matching techniques can be applied.

Dorai presents COSMOS (Curvedness-Orientation-Shape Map on Sphere) [Dorai, 1996] as a representation scheme for 3D free form objects from range data without occlusions. According to this scheme, the object is represented concisely in terms of maximal surface patches of constant shape index. The shape index is a quantitative measure of shape complexity of the surface and is based on the principle curvatures at a point on the surface. The patches are mapped onto a sphere based on the orientations and aggregated using shape spectral functions. Surface area, curvedness and connectivity are utilized to capture global shape information. She derives a shape spectrum and experiments on its efficiency of recognition.

Johnson and Hebert [Johnson and Hebert, 1999] introduce spin images for a 3D shape-based object recognition system towards simultaneous recognition of multiple objects in scenes containing clutter and occlusion. The spin image is a data level descriptor that is used to match surfaces represented as surface meshes. They describe surface shape as a dense collection of points and surface normals and associate a descriptive image with each surface point that encodes global properties of the surface using an object centered coordinate system. The spin image is created by constructing a local basis at an oriented point on the surface of the object and accumulates geometric parameters in a 2D histogram. In simple words, the spin image can be visualized as a sheet spinning about the normal at that point. The image is descriptive because it accounts for all the points on the surface and is invariant to rigid transformations.

Kazhdan et al. [Kazhdan et al., 2003] outline an algorithm for 3D shape matching using a harmonic representation of 3D polygonal meshes. They rasterize the 3D mesh into 64 x 64 x 64 voxel grids and center the object as the center of the grids so that the

bounding sphere is of radius 32 voxels. They then treat the object as the function in 3D space and decompose it into 32 spherical functions by considering spheres of radii 1 through 32. They further decompose each of the functions into 16 harmonic components and the 32 x 16 harmonics constitute the harmonic representation of the 3D model. They compare two harmonic representations with the Euclidean distance. Zhang and Hebert propose the harmonic shape images as a 2D representation of surface patches [Zhang and Hebert, 1999]. The theory of harmonic maps studies the mapping between different metric manifolds from the energy minimization point of view. With the application of harmonic maps, a surface representation called harmonic shape images is generated to represent and match 3D freeform surfaces. The basic idea of harmonic shape images is to map a 3D surface patch with disc topology to a 2D domain and encode the shape information of the surface patch into the 2D image. This simplifies the surface-matching problem to a 2D image-matching problem.

Shum et al. address the problem of 3D shape similarity between closed surfaces in [Shum et al., 1996]. He defines a shape similarity metric as the L_2 distance between the local curvature distributions over the spherical mesh representations of the two objects. He achieves the similarity measure in $O\ (n^2)$ complexity where n is the number of tessellations in the object mesh. Their experiments on simple shapes show good shape similarity measurements.

2.3.4 Shape Histograms

The histogram-based methods reduce the cost of complex matching schemes but sacrifice efficiency and robustness to the methods discussed in Section 2.3.2. These methods compare shapes on the basis of their statistical properties.

Ankerst et al. [Ankerst et al., 1999] introduce 3D shape histograms as an intuitive powerful approach to structural classification of proteins. They decompose a 3D object into three models (shell model, sector model and spider web) around an object's centroid and process model similarity queries based on a filter refinement architecture. A similar search technique for mechanical parts using histograms was proposed in [Kriegel et al., 2003]. The models are normalized into a canonical form and voxelized into axis parallel equal partitions. Each of these partitions is assigned to one or several bins in a histogram depending on the specific similarity model.

Besl et al. [Besl et al., 1995] consider histograms of crease angles for all edges in a triangle mesh to describe shape. Their method does not match non-manifold surfaces and is not invariant to changes in mesh tessellation. Osada et al. in [Osada et al., 2002] presents Shape Distributions for a shape similarity search engine by extending Besl's approach. According to his technique, random points from the surface of a model are

extracted. Shape functions D_1, D_2, D_3, D_4, and A_3 are computed at each of these random points.

- D_1: Distance between a fixed point (centroid) and a random point.
- D_2: Distance between two random points.
- D_3: Square root of the area of triangle formed by three random points.
- D_4: Cube root volume of the tetrahedron of four random points.
- A_3: Angle between three random points.

They suggest the use of D_2 shape function for computing Shape Distributions due to its robustness and efficiency along with invariance to rotation and translation. The D_2 distances between random points are normalized using the mean distance. The shape distribution is the histogram that measures the frequency of occurrence of distances within a specified range of distance values. Once the Shape Distributions are generated the distance between the two solid shapes is computed using L_N norm. Usually L_2 norm is used for comparison, though other distances such as Earth Mover's distance or match distances can also be used.

This technique is robust and efficient for simple objects and gross shape similarity. As the resolution of the 3D model increases the comparison becomes more robust, but the computational time increases. Furthermore as objects become more and more complex, the Shape Distributions tend to assume similar shape resulting in inaccurate comparison of solid models. Shape Distributions have been experimented with limited success on mechanical parts and real laser scanned data. Ohbuchi et al. in [Ohbuchi et al., 2003] improve the performance of Shape Distributions with a 2D histogram of angle-distance and absolute angle distance that can be computed from the D_2 shape distribution. Page et al. [Page et al., 2003b] define shape information as the entropy of the curvature density. They use it to describe the complexity of the 3D shape. Hetzel et al. [Hetzel et al., 2001] present an occlusion robust algorithm for 3D object recognition that makes use of local features such as the shape index, pixel depth and the surface normal characteristics in a multidimensional histogram. Histograms of two objects are matched and verified using the chi–squared hypothesis test to achieve shape recognition.

2.3.5 Topology Description

The topology of a 3D model is an important property for measuring similarity between different models. Topology of models is typically represented in the form of a relational data structure such as trees or directed acyclic graphs. Subsequently, the similarity estimation problem is reduced to a graph or tree comparison problem.

Gotsman et al. describe the fundamentals of spherical parameterization for 3D meshes [Gotsman et al., 2003]. They argue that closed manifold genus-zero meshes are topologically equivalent to a sphere and assign a 3D position on the unit sphere to each of the mesh vertices. They use barycentric coordinates for the planar parameterization. Leibowitz et al. [Leibowitz et al., 1999] share their memory intensive experience in implementing geometric hashing for the comparison of protein molecules represented as 3D atomic structures.

In [McWherter et al., 2001] model signature graphs have been proposed for topological comparison of solid models. They extend attribute adjacency graphs, mentioned in [Joshi and Chang, 1998], to consider curved surfaces. Model signature graphs are constructed from boundary representation of the solid. This graph forms the shape signature of the solid model. Once a model signature graph is constructed, the solid models are compared using spectral graph theory [Chung, 1997]. The eigen values of the Laplacian matrix are used in the comparison. The eigen values of the Laplacian are strongly related to other graph properties such as the graph diameter. The graph diameter is the largest number of vertices, which must be traversed, to travel from one vertex to another in the graph. Another technique proposed for comparing the graphs is the use of graph invariance vectors [McWherter et al., 2001]. The vectors are then compared using L_2 norm to determine similarity between the graphs and hence the solid models. The graph invariants that form the graph invariance vectors include node and edge count, minimum and maximum degree of the nodes, median and mode degree of the nodes, and diameter of the graph. The use of graph invariance vectors improves the efficiency of the method. However it results in decrease in the accuracy of comparison. This technique has been applied to mechanical parts and is applicable to product design and manufacturing domain. The paper [Cardone et al., 2003] is a comprehensive survey on shape-similarity based assessment for product design applications.

Multi-Resolution Reeb Graphs presented in [Hilaga et al., 2001] have been used for modeling 3D shapes. The Reeb graph is derived from the triangle mesh models by defining a suitable function such as the geodesic curvature. The choice of the function depends on the topological properties selected. The range of the function over the object is split into smaller bins. The number of bins is the resolution of the Reeb graph. Each connected region in the bin will map into a node of the Reeb graph, and the adjacent nodes will be connected by edges. The Reeb graph construction has a time complexity of $O(N \log N)$, N being the number of vertices in the mesh. The Reeb graphs of two objects can be used for maximizing a similarity function at corresponding nodes. This technique is not invariant to Euclidean transformations.

We have very briefly described some of the key methods for shape analysis on 2D intensity images and 3D mesh models. In the next section we present two tables that

contain a qualitative comparison based on algorithm efficiency and descriptive capability of the key methods presented in Section 2.3.

2.4 Summary

We would like to summarize the literature review in this section. We have presented 3D shape searching as applied in diverse fields such as computer vision, mechanical engineering, bio-informatics and bio-medical imaging. In Tables 2.1 and 2.2 we compare the description and search efficiency of a few key methods before concluding our summary.

Shape signatures are abstractions of 3D shapes and have limited discrimination capabilities. They are application specific and hence the complexity involved in matching and computation cannot be compared on the same domain for effectiveness. Therefore a good strategy to shape analysis would be the choice of a signature that is computationally efficient producing lesser false positives followed by another one that needs computational effort to remove those false positives. With the popularity of 3D scanning and CAD models we emphasize the necessity of a quick and information preserving shape representation than a time consuming exact isomorphic representation.

Shape analysis has been pursued by researchers for the task of multi-modal data fusion (registration), object recognition, object visualization and compression. Most of the methods developed are bounded by an application specific heuristic constraint that bridges the user's notion and the computer's notion of shape similarity. We would like to conclude the literature survey as our knowledge base for further research and development.

We discuss range data acquisition and solid modeling of mechanical parts and automotive scenes in the next chapter with illustrative examples. We emphasize that it is important to understand and interpret the data before analysis and so introduce range acquisition systems and the process of 3D model creation using a range sensor.

Table 2.1: Qualitative comparison of 3D shape search methods with focus on algorithm efficiency.

Searching Technique	**Computational Cost**	**Comparison Cost**	**Test Data**	**Key Methods**
Feature (Global)	$O(N)$ where N is the number of voxels under consideration.	$O(F)$ where F is the number of features extracted.	Methods span synthetic mesh datasets to complex real datasets.	[Elad et al., 2001] [Zhang, and Hebert,1999]
Intermediate Description	$O(V \log V)$ in the worst case where V is the number of vertices.	$O(R^2)$ where R is the resolution of the intermediate representation.	Range and Triangle Mesh real world data sets of scenes and objects.	[Johnson and Hebert,1999] [Dorai,1996]
Manufacturing and Product based Description	$O(P)$ where P is the number of primitives.	$O(F^2)$ where F is the number of features extracted.	CAD models of mechanical components.	[Mukhai et al.,2002]
Histogram-based	$O(SB)$ where S is the number of sample points and B is the number of bins.	$O(B)$ where B is the number of bins.	Simple and low resolution synthetic models.	[Ankerst et al., 1999] [Osada et al.,2002]
Topological Graph Methods	$O(N)$ where N is the number of voxels considered.	Worst case $O(N^3)$ where N is the number of nodes in the graph.	Low resolution synthetic data sets.	[Hilaga et al.,2001] [Leibowitz et al., 1999] [McWherter et al., 2001]

Table 2.2: Qualitative comparison of 3D shape search methods with focus on effective description.

		Comparison Criterion			
Shape Category	**Method**	**Scale Invariance**	**Local Saliency**	**Advantages**	**Disadvantages**
Feature (Global)	Moments	No	No	Computationally fast	Different shapes can have same moments.
	Spherical Harmonics	No	No	Used in general shape classification.	Low stability
Intermediate Description	COSMOS	Yes	Yes	Curvature-based	Assumes ideal data.
	Spin Images	No	Yes	Robust to occlusions	Storage of spin images and 2D image matching.
	Gaussian Images	No	No	Useful for pruning	Low computational efficiency.
Manufacturi-ng and Product based Description	Feature Graphs	No	No	Useful for mechanical parts.	Shape recovery is difficult with more primitives.
	String Description	No	No	Useful for mechanical parts.	Cannot be automated.
Histogram based	Shape histograms	Yes	No	Simple and easy description.	Not very robust
	Shape Distributions	Yes	No	Good for clustering	Uniqueness of the distribution is not justified
Topological Graph Methods	Skeletal Graph	Yes	Yes	Topologically correct with local saliency support.	Important local feature extraction stage
	Reeb Graph	Yes	No	Multi-resolution Analysis	Choice of Reeb function
	Geometric Hashing	No	No	Exact matching	High storage requirements

3 DATA COLLECTION AND MODELING

The computer vision approach to reverse engineering and under vehicle inspection requires digitized data. We hence require a system that can automatically (or with minimal manual intervention) capture geometric structure of an object and store the subsequent shape and topology information as a digitized model. We make use of 3D range scanners for this task. We introduce in this chapter the process of range data acquisition and solid modeling geared towards generating mesh models using a sheet-of-light laser scanning mechanism and share our experience with the IVP range sensor to create 3D models of automotive parts and automotive scenes.

3.1 Range Data Acquisition

Range images are a special class of digital images. Each pixel of a range image expresses the distance between a known reference frame and a visible point in the scene. Therefore, a range image produces the 3D structure (though not completely) of a scene and can be best understood as a sampled surface in 3D. Range images (often referred as depth maps, depth images, xyz maps, surface profiles and 2.5D images) are obtained using range sensors. Range sensors are devices that make use of optical phenomena to measure range. In general range image acquisition systems are classified into one of the following types based on their principle of operation: triangulation (passive or active), time of flight, focusing, holography and diffraction. We discuss each of these methods very briefly in Section 3.1.1 and document the principle of operation and calibration details of our range sensor in Section 3.1.2.

3.1.1 Range Acquisition Systems

We begin our discussion with triangulation-based techniques. Passive triangulation (stereo) is the way humans perceive depth. It involves two cameras taking a picture of the same scene from two different locations at the same instant of time. Depth cues are extracted by matching correspondences in the two images and using epipolar geometry. Passive triangulation is however challenged by the ill-posed problem of correspondence in stereo matching. The correspondence problem is eliminated by

replacing one of the cameras by a moving light source (preferably a laser light source). This technique is called active triangulation where a pattern of light (energy) is projected on the scene and is detected to obtain range measurements. Time of flight range finders determine range by measuring the time required for a signal to travel, reflect and return. Holographic interferometry uses split beam interference to produce an image which when processed further, yields the range image. A moiré interference pattern is created when two gratings with regularly spaced patterns are superimposed on each other. "Moiré" sensors project such gratings onto surfaces, and measure the phase differences of the observed interference pattern. Distance hence becomes a function of such phase differences. Focusing and defocusing have also been used to derive range information. These methods infer range from two or more images of the same scene, acquired under varying focus settings. For example, shape from focus sensors vary the focus of a motorized lens continuously, and measure the amount of blur for each focus value. Once the best focused image is determined, a model linking focus values and distance is used to approximate distance. The decision model makes use of the law of thin lenses and computes range based on the focal length of the camera and the image plane distance from the lens' center. While triangulation methods and time of flight methods have been extensively used for computer vision tasks, methods based on holography, focusing and diffraction are sidelined because of their fundamental performance limitations and their inability to meet real-time imaging requirements of speed and accuracy. We direct the reader to [Besl, 1988] and [Trucco and Verri, 1998] for further reading on range image acquisition and processing.

We concentrate on triangulation-based range sensors. The main reason behind this choice is that such sensors are based on intensity cameras, giving us a chance to exploit the concepts that we know on intensity imaging. They also give accurate 3D coordinate maps and are easy to understand and build for real-time imaging.

3.1.2 Range Sensing Using the IVP Range Scanner

The IVP RANGER system as shown in Figure 3.1 consists of two different subsystems; the Smart Camera and the PC Interface. Each Smart Camera contains a Smart Vision sensor, a control processor (Intel 386) and an IVP HSSI (High speed serial interface). The Smart Camera is connected to the system PC via a COM port and an HSSI Interface on a PCI board called the SC adapter. The IVP Ranger is implemented on the MAPP2200 (MAPP stands for Matrix Array Picture Processor), MAPP 2500 and LAPP1530 (Linear Array Picture Processor) Smart Vision Sensors from the IVP. The total integration of the sensor, A/D converter and the processor on the same parallel architecture allows image processing at a very high speed. The Smart Camera acquires the range profiles autonomously and outputs the profiles to the host via the HSSI interface. The host PC can then manipulate these profiles.

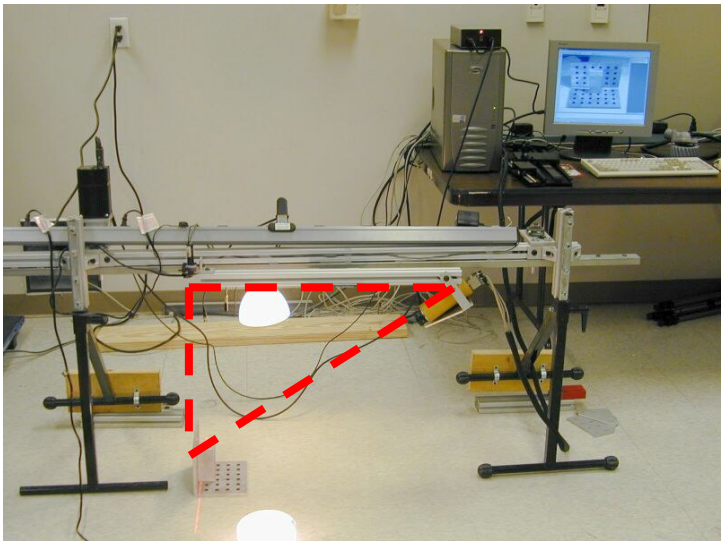


Figure 3.1: IVP Ranger SC-386 range acquisition system.

The IVP Ranger uses an active triangulation scheme where the scene is illuminated from one direction and viewed from another. The illumination angle, the viewing angle, and the baseline between the illuminator and the viewer (sensor in this case) are the triangulation parameters.

The Ranger consists of a special 512 x 512 pixel camera and a low-power stripe laser. The design of the Ranger is specifically tailored for the camera and the supporting electronics to integrate image processing functions onto a single parallel-architecture chip. This chip contained within the camera housing has a dedicated range processing function that allows for high-speed acquisition of nearly one million points per second. The most common arrangement of the system is to mount the camera and the laser source relative to the proposed target area to form a triangle where the camera, laser, and target are each corners of the triangle. The angle where the laser forms a corner is typically a right angle such that the laser stripe projects along one side of the triangle. The angle, α, at the camera corner is typically 30-60 degrees. The baseline distance between the camera and the laser, denoted by B, specifies the right triangle completely (see dotted line in Figure 3.1). We would like to summarize our experience of the IVP Ranger as a sensor that outputs range values as a function of illumination, relative motion, temperature and surface reflectance as shown in Equation 3.1.

$$r = F(i, j, \alpha, \beta, B, T, \eta, \chi, \mu) \tag{3.1}$$

where i and j respectively are the horizontal and vertical pixel positions, β (= 90 degrees) and α is the illumination angle and the camera view angle respectively, and B is the base line distance between camera and the laser source. These are the important design parameters that decide the field of view of the scanning mechanism. External parameters such as temperature (T), environmental light (η), surface reflectance and color of the objects (χ) and the trajectory of the relative motion (μ) also influence the quality of range scans. We have characterized the scanner to minimize the effect of such external factors. We have deduced that the warm up time of 40-50 minutes yielded stable and reliable data. We ignore effects of environmental temperature. We also realize that the Ranger is sensitive to light and tends to introduce significant error when the ambient illumination is strong. Most of the scanning that we do inside the lab is performed with minimal lighting. We have learned that the effect of illumination can be compensated by the use of a powerful laser (> 100mW and wavelength 685nm) that we propose to use for scanning under the vehicle. We have performed a simple experiment to characterize the sensor's behavior to the color and reflectance of the objects. We have tried to image wooden and metal rectangular blocks of the same size and compared the range measurements. We have concluded that the IVP range sensor is not influenced by surface reflectance but black objects because of their laser beam reflectance characteristics need modification. We have simply painted the object with a lighter color to work around this sensitivity. We have simulated the triangulation geometry of the Ranger system in MATLAB to understand the effect of different sensor parameters that influence the scanning mechanism and the process of calibration.

In Figure 3.2 we demonstrate the principle behind range acquisition using the IVP Range scanner. We show the sheet-of-the-light laser falling on a target object. The laser line that provides cues about the surface shape of the object is called a surface profile. By traversing the entire object either by moving the sensor setup or the scene, a sequence of surface profiles is accumulated as a range image.

Equation 3.2 is the reduced form of range r as a function of the geometry, focal length (b_0 is the distance between the lens and the sensor approximated as the focal length of the lens) and sensor offset position in the 512 x 512 CCD chip assuming that we have compensated for the external sensor sensitivity parameters.

$$r(s) = B\frac{(b_0 \tan\alpha - s)\cos\alpha}{\frac{b_0}{\cos\alpha} - (b_0 \tan\alpha - s)\sin\alpha} = B\frac{f\tan\alpha - s}{f + s\tan\alpha} \tag{3.2}$$

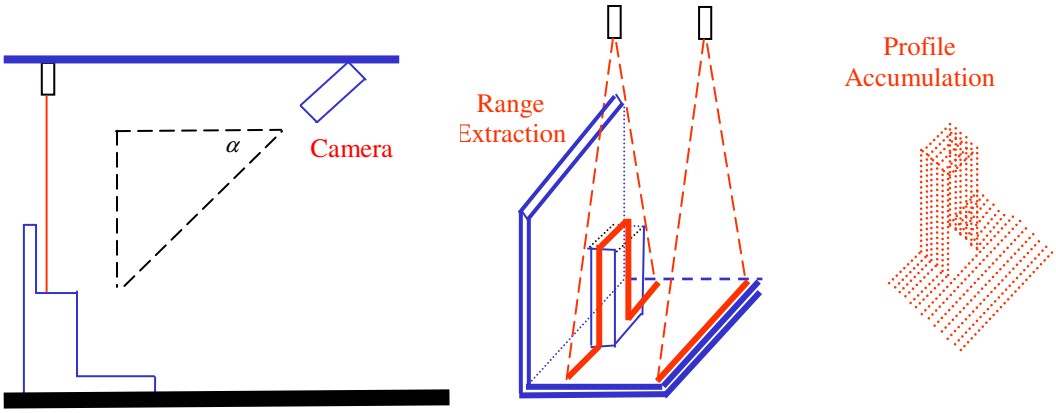


Figure 3.2: Triangulation and range image acquisition.

The differential of the range equation in Equation 3.2 is the resolution of the sensor as shown in Equation 3.3

$$\Delta r = \Delta s \frac{-Bb_0}{(b_0 \cos\alpha + s \sin\alpha)^2} \tag{3.3}$$

The maximum range that a particular sensor arrangement with a baseline B and angle α (Equation 3.4) can measure is obtained by maximizing the function for range (Equation 3.2) in terms of the sensor size N and resolving capability Δx.

$$R_T = \frac{4Bb_0 N\Delta x(1+\tan\alpha)^2}{4b_o^{\,2} - (N\Delta x \tan\alpha)} = \frac{4BfN\Delta x(1+\tan\alpha)^2}{4f^2 - (N\Delta x \tan\alpha)} \tag{3.4}$$

These equations are important when the decision between field of view and resolution has to be made. We make use of these equations for the design of our scanning mechanism but not for range measurements. We follow a much more robust calibration procedure that models the world to sensor coordinate transformation as a combination of translation (of the world coordinate system to the optical coordinate system), a rotation (to align optical axis with real world axis) and a projection from world to sensor coordinate system. Equation 3.5 is the transformation from the world to the sensor coordinate system where w_p is the sensor coordinate system scale factor proportional to the baseline distance B, (u, v) refers to the position in the sensor coordinates; (X, Y, Z) are the real world position coordinates, f is the focal length of the optics. (u_0 , v_o) is the position where the optical axis meets the sensor, and k_v, k_u

and θ are the skew and tilt compensation factors. The s_{ii} matrix takes care of the rotation in the three rectangular axes while (x_0, y_0, z_0) compensates for the translation.

$$\begin{bmatrix} w_p u \\ w_p v \\ w_p \end{bmatrix} = \begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sin\theta}{k_u} & \frac{\cos\theta}{k_u} & 0 \\ 0 & \frac{1}{k_v} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f & o & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -x_0 \\ 0 & 0 & 0 & -y_0 \\ 0 & 0 & 0 & -z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.5}$$

Equation 3.5 can be simplified into Equation 3.6 with 12 unknown parameters that can be determined with at least 6 points positioned in the world coordinate system projected into the sensor coordinates.

$$\begin{bmatrix} w_p u \\ w_p v \\ w_p \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.6}$$

After calibration we know the equation for all rays hitting the sensor plane. However, we still do not know from which point along the ray that it started. To find out where our sheet-of-light rays start we introduce a simple calibration step (Figure 3.3 (b)) using the sheet-of-light to calibrate a single profile. By finding the sensor positions where the light sheet hits the calibration target we can compute the world coordinates for the laser plane. Thus, calibration gives us the rays for each sensor coordinate, and the laser plane equation, using which we can find the world coordinates for each point.

This process of calibration can be better understood with the help of Figure 3.3. Figure 3.3(a) is the status of the CCD when it is viewing the laser line (white line on CCD shown in Figure 3.3(b)). The sensor position is detected with sub pixel accuracy (based on the intensity on the CCD because of the laser line) for the range measurement. We solve for the 12 unknown parameters as a system of linear equations. Theoretically, for the system described in Equation 3.6 we need six equations to compute the parameters. We increase the reliability and reduce possible error by using 40 points on the calibration target. With the 40 real-world coordinates as in Figure 3.3(c) known we compute a transformation matrix that maps the sensor coordinates to the real world in 3D. We use this transformation matrix for our future scans without disturbing the geometry of the scanning mechanism.

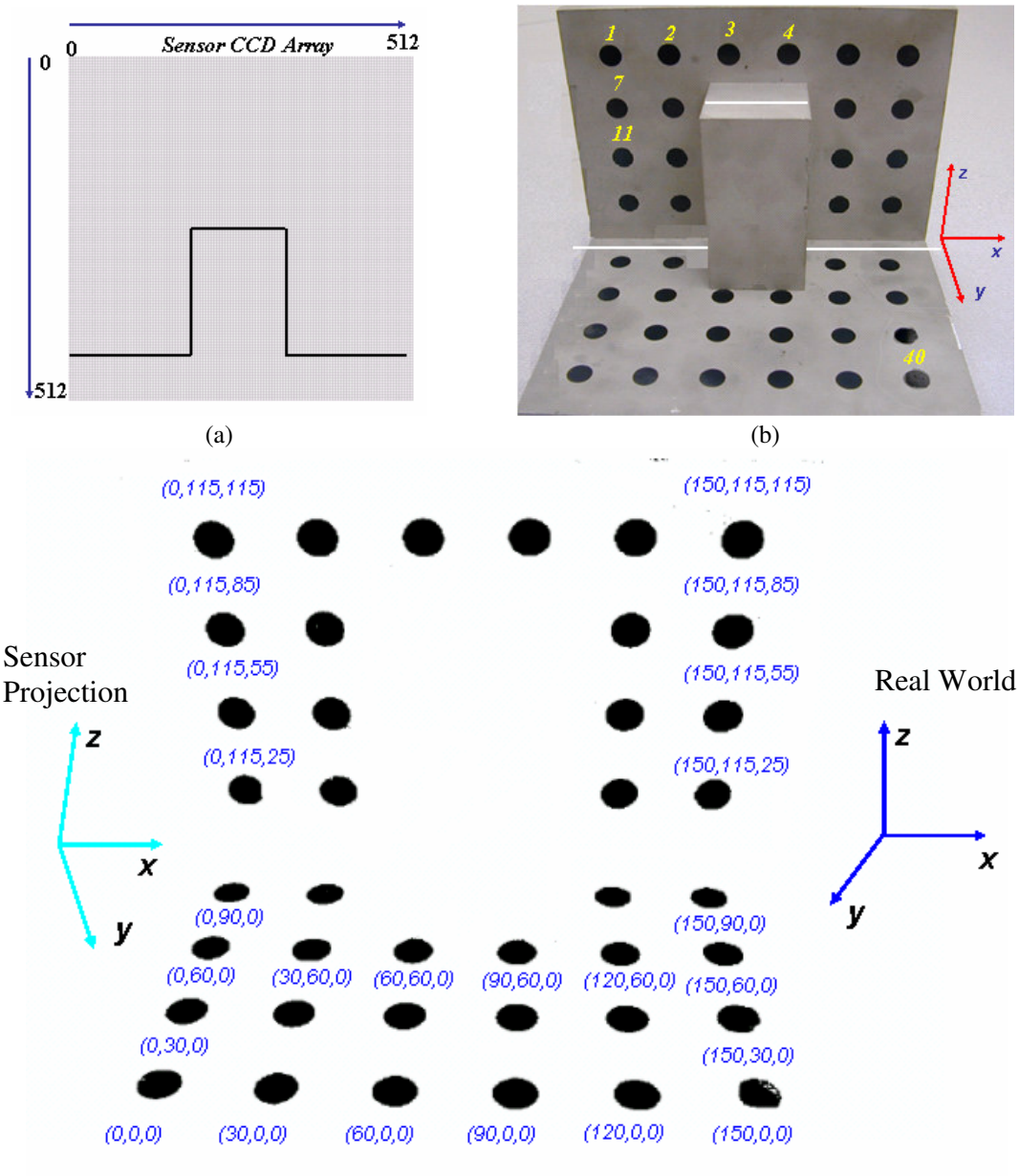


Figure 3.3: The process of calibration. (a) Single profile calibration. (b) The physical calibration target designed to compute the 12 unknown parameters using 40 points. (c) The transformation from the sensor projection coordinates to the real world.

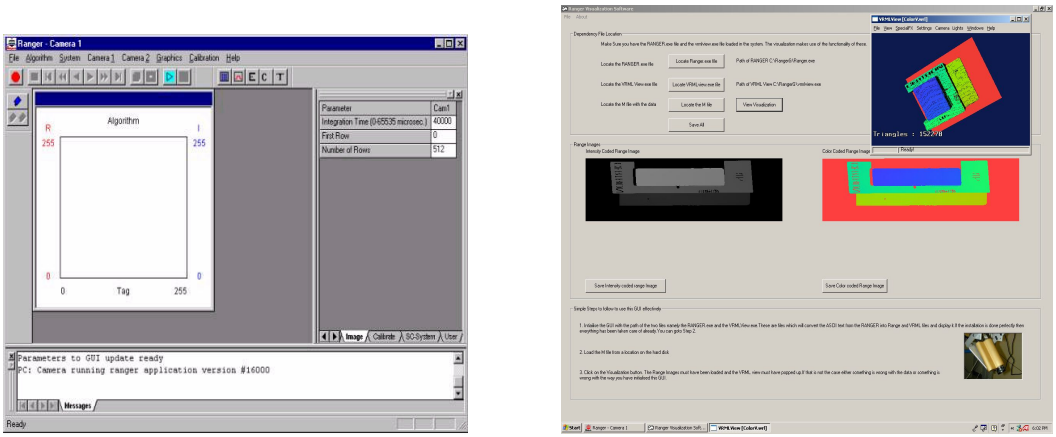
The IVP range scanner is capable of acquiring 2000 profiles in one second. We have controlled the relative motion of the sensor arrangement with a precise smart motor. The collection of profiles spanning that particular view of the object is represented as a range image. We have built a graphical user interface (GUI) for visualizing range images and their corresponding 3D triangle meshes acquired using the scanner. Figure 3.4(a) is the acquisition and control interface provided by the IVP and Figure 3.4(b) is the snapshot of our visualization GUI in action.

3.2 Solid Modeling from Range Images

Range data acquisition is a digitization process and is only the first step towards model generation. We now need to process the range information for better visualization and representation. In Section 3.2.1 we explain the processing pipeline for creating mesh models of objects for the task of reverse engineering and extend our implementation to a more challenging task of modeling automotive scenes in Section 3.2.2.

3.2.1 Modeling Automotive Components for Reverse Engineering

Reverse engineering is the ability to create computer aided design models of existing objects. It is often considered as a feedback path for inspection and validation in a rapid manufacturing system. Bernardini et al. in [Bernardini et al., 1999] stress on the promise and impact of computer aided reverse engineering in the process of system design while



(a) (b)

Figure 3.4: Graphical User Interface. (a) Snapshot of the GUI for acquisition from the IVP. (b) Snapshot of the GUI for visualization.

Thompson et al. [Thompson et al., 1999] apply reverse engineering as a process that will enable the recreation of objects that are out of production.

With the emergence of high speed accurate laser scanners reverse engineering is moving away from the traditional tedious but accurate coordinate measuring machines (CMM). As discussed in the previous section, we use an active range sensor to acquire an ensemble of range images to reconstruct a CAD-like model of the object. We begin reverse engineering with acquisition of range images using the IVP sensor, which provides speed and accuracy. We would like to summarize the process of model creation as a block diagram in Figure 3.5.

After data acquisition we have a set of range images representing multiple view points around an object. The task is now to reconstruct the CAD model from these range images. The fundamental challenge in modeling the range images is that of reconstruction as discussed in [Hoppe et al., 1992]. The challenge lies in aligning multiple views into a global coordinate frame (also called as the process of registration) and integrating and merging aligned views into a CAD representation. As discussed earlier, multiple views of an object are necessary to overcome occlusions. As the camera moves to the new view, the resulting data is relative to the new view position.

Registration is the process where we align these multiple views and their associated coordinated frames into a single global coordinate frame. The registration problem is essentially recovering the rigid transformation from the new range data. We define rigid transformation as

$$y = Rx + t \tag{3.7}$$

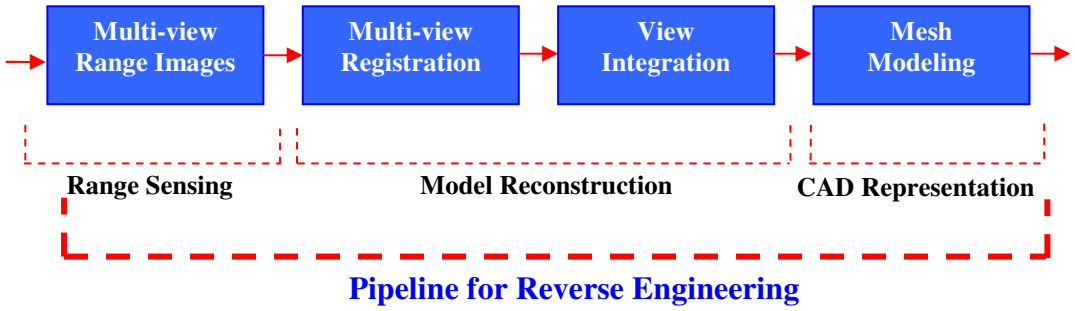
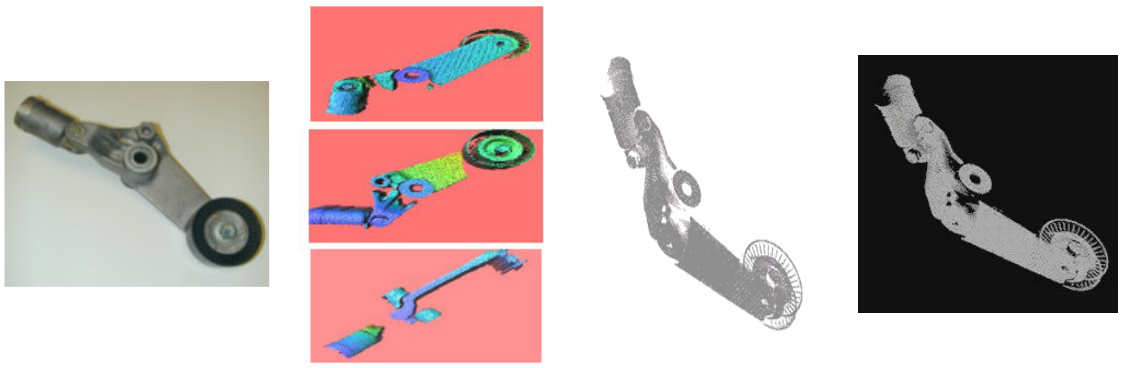


Figure 3.5: Block diagram of a laser-based reverse engineering system.

where R represents the rotation matrix and t is the translation vector. The point y is the same as x but in the global coordinate frame. Registration is the process of finding R and t. The registration process tries to interpret common geometric information from two calibrated range images at two different poses (views).

According to [Horn et al.,1988], given three or more pairs of non-coplanar corresponding 3D points between views, the unknown rigid transformation of rotation and translation has a closed form solution. The registration problem can hence be approached as a point matching problem. The most popular of registration algorithms is the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992]. We have used the implementation of ICP in Rapidform (a reverse modeler software package) for the task of surface registration. It allows us to initialize the ICP algorithm by manual point picking. The three pairs of corresponding points so picked are iteratively refined up to a particular threshold before merging the two point clouds.

Having overcome the problem of occlusions by registering multiple views, we now need to integrate these views into a single surface representation. We consider the registered range data as a cloud of points and reconstruct the topology of that object from its range samples. A simple shape may require just a few views while a complicated object may require significantly more. Page et al. [Page et al., 2003a] document this systematic procedure in the literature as a method of reconstructing mechanical components. Figure 3.6(a) shows a part that we would like to reverse engineer. We present results of multiple view range image acquisition process in Figure 3.6(b). The point cloud in Figure 3.6(c) is the result of reconstruction that we triangulate to represent as a CAD model in Figure 3.6(d). We use polygonal meshes to represent the CAD model.



(a) (b) (c) (d)

Figure 3.6: Model creation. (a) Photograph of the object. (b) Multiple-view range maps. (c) View integrated point cloud. (d) Rendered triangle mesh model.

A polygonal mesh is a piece-wise linear surface that comprises vertices, edges and faces. A vertex is a 3D point on the surface, edges are the connections between two vertices, and a face is a closed sequence of edges. In a polygonal surface mesh, an edge can be shared by, at most, two adjacent polygons and a vertex is shared by at least two edges. We use triangle meshes to represent the discrete approximations to 3D surfaces. A triangle mesh is a pair $T=<\nu, \tau>$ where $\nu=\{\nu_1, \nu_2, \nu_3, \ldots, \nu_n\}$ is a set of vertices, and $\tau=\{\tau_1, \tau_2, \ldots, \tau_m\}$is the set of triangles that approximate the surface.

3.2.2 Modeling Automotive Scenes for Under Vehicle Inspection

Under ideal laboratory conditions, data collection with a range scanner is straightforward. Underneath a vehicle however, we address several challenges. The most significant of those is the design of the scanning mechanism. The field of view is limited by the ground clearance and the huge variation in size of the components that make up the scene under the vehicle. The distance (range) is too small for the use of time-of-flight scanners and laser triangulation scanners but too large for photogrammetric measurements.

Real-time 3D data acquisition is a research challenge in computer vision. Before we start thinking of a robot mountable design for vehicle inspection, we would like to briefly survey 3D data acquisition systems that optimize the process of digitization of real world scenes and objects for speed and accuracy. One such expensive effort is the digitization of statues in the "Digital Michelangelo" project that involves the close scanning of statues for cultural heritage recording. Levoy et al. in [Levoy et al., 2000] suggest a configuration for high speed laser triangulation that involves a light projector recording video, that is processed later to fill holes and registered using the ICP algorithm for the complete 3D model. The paper [Takatsuka et al., 1999] proposes a low cost interactive active monocular range finder. Davis and Chen [Davis and Chen, 2001] present the design of a laser range scanner designed for minimum calibration complexity. They specifically state that despite the simple geometry and components, laser scanners must be engineered and calibrated with extremely high precision. Champleboux et al. [Champleboux et al., 1992] examine the process of registration of multiple 3D data sets obtained with a laser range finder. They propose a new sensor calibration technique based on the conjunction of a mathematical camera model and further discuss an algorithm based on octree splines for recovering rigid transformation, for rotational and translational rectification between two 3D data sets obtained from the range sensor. Having considered so many options and as a tradeoff between resolution and field of view we decided to jack the vehicle up by a meter and use the inverted triangle mechanism for scanning. We calibrated the sensor arrangement as discussed in Section 3.1.2 and without disturbing it we inverted it and moved the sensor arrangement on a conveyer belt to reconstruct the 3D scene.

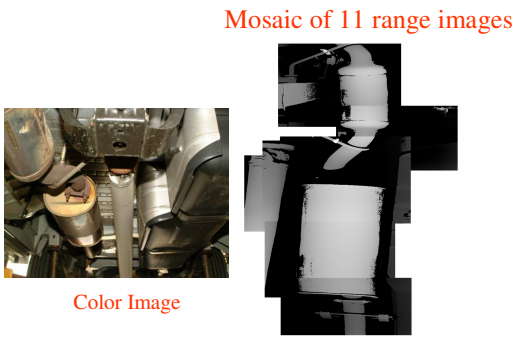
Although a powerful laser was used to counter ambient lighting, we could not compensate for spectral reflections since the metallic surfaces under a vehicle exhibit strong spectral reflection properties. A laser further complicates this problem as internal reflections lead to significant errors in range estimation. A promising approach for this problem involves the use of an optical filter tuned to the frequency of the powerful laser. The filter allows the data collection to isolate the initial reflection of the laser and thus improve the range sensing capabilities. The other noise issue that we would like to discuss involves the jerks in trajectory of the scanning mechanism. We have assumed a linear and smooth trajectory under the vehicle in the data that we have presented.

Another significant problem in range scanning underneath a vehicle is that of view occlusions. The obvious occlusion is that the camera can only view one side of a component (the bottom side facing straight down towards the ground). The muffler for example in the Figure 3.7(d) is a one-sided view. Without dismantling the car, range scanner cannot extract geometry of the other side of the muffler. Such an occlusion should illustrate the potential of other occlusions such as one object partially covering another object from the range sensor. The objects underneath a vehicle have various shapes and scales located at different depths. For example in Figure 3.7(b), the bent pipe that connects the muffler and the catalytic converter is occluded by the muffler at the time of scanning. The solution to this problem is to use multiple scans to fill as much as possible the areas without information. This solution is a laborious one because multiple fields of view imply multiple calibration procedure iterations. The different views and scanning angles are extremely restricted by the low ground clearance under a vehicle. Thus an integration and fusion of multiple scans only partially fills the occlusion holes, but significantly enhances the data. As a result, we scan underneath a vehicle with multiple passes and at different angles. The final challenge that we consider with the data collection is the data redundancy inherent to laser range scanning. A single range image with 512 x 512 pixels yields over 250,000 data points. With additional scans to overcome occlusions and to achieve full coverage under a vehicle, this number quickly grows to several million data points. This large data set allows high fidelity geometry that other 3D sensors do not offer, but the price is that of data redundancy and a potential data overload. The data that we present in Figure 3.7(d) is a 40 megabyte VRML model with 10 million vertices and 15 million triangles.

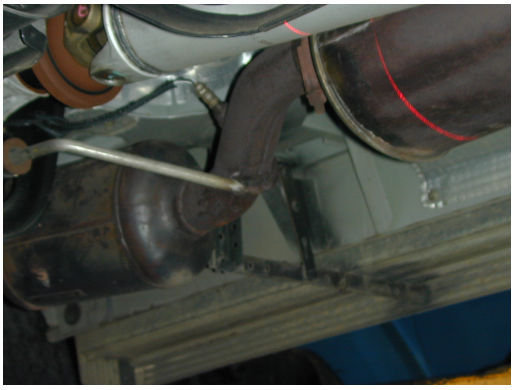
We have presented the procedure and capability of data collection using a 3D range scanner in this chapter. We now have real world objects in a format that computers can attempt to understand. In Chapter 4, we would like to outline our approach to shape description and discuss the building blocks of our algorithm in detail.



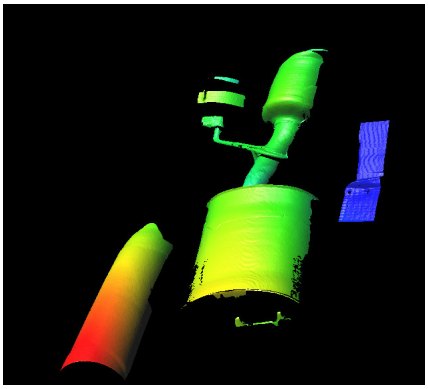
(a)



(b)



(c)



(d)

Figure 3.7: Data acquisition for under vehicle inspection. (a) A pre-calibrated scanning mechanism in action. (b) The mosaic of range images as the output from the scanner. (c) Close-up color image of the scene. (d) Snapshot of the registered 3D model.

4 ALGORITHM OVERVIEW

In this chapter, we describe our CVM algorithm as the informational approach to shape description. We first discuss CVM for 2D in the context of intensity and range images and extend it with modifications to 3D models. We also explain in detail each of the building blocks of the algorithm.

4.1 Algorithm Description

Before we discuss the details of the algorithm we would like to introduce some of the key papers that have influenced our work. Arman and Aggarwal present a survey on model based object recognition strategies on dense range images in [Arman and Aggarwal, 1993]. More recently, Campbell and Flynn survey free form object representation and recognition in [Campbell and Flynn, 2001]. We focus our algorithm development with these surveys as our knowledge base on object representation and recognition.

We are inspired by the COSMOS framework for free form object representation [Dorai, 1996] for the development of our CVM algorithm. Dorai defines shape index and curvedness as indicators of shape and constructs a shape spectrum for object analysis. She models range images as a combination of maximally sized surface patches of constant shape index to get around segmentation issues and uses a graph representation on her range data. She assumes that there are no occlusions in her image. Her method of computing curvature that assumes the uniform grid structure however is not suited for mesh models. With CVM, we hence analyze various curvature estimation methods for triangle meshes and propose a graph representation based on curvedness segmentation and a normalized surface variation measure based on curvature. Our approach is analogous to the shape index that Dorai uses for segmentation on the range image and the curvedness map on the sphere for shape analysis. We chose surface representation because it directly corresponds to the features that will aid recognition even with view occlusions in the sensed data. To illustrate the CVM better, we introduce in Section 4.1.1 the idea of using information theory for shape complexity description on 2D contours. We discuss the algorithm with a block diagram and describe how we extend it to the description of 3D mesh models in Section 4.1.2.

4.1.1 Informational Approach to Shape Description – Curvature Variation Measure

We would like to formulate our algorithm on the basis that shape information is directly proportional to the variation in curvature (curvature of the boundary for 2D curves and curvature of surfaces on 3D surfaces) and inversely proportional to symmetry. We propose to extract shape information from the images and analyze a procedure to discriminate objects based on a single number that is a measure of its visual complexity.

To understand the basis of our algorithm better, let us start with a small and simple example. In Figure 4.1, we show a circle and an arbitrary contour. Visually, the more appealing of the two is the circle while the complex of the two contours is the arbitrary contour. We propose that smoothly varying curvature conveys very little information while sharp variation in curvature increases the complexity for shape description. In this context, we would like to refresh the fact that more likely an event is, lesser the information it conveys. The circle has uniformly varying curvature; that is there is no uncertainty involved in the variation of its curvature, which means it has the least shape information. Figure 4.2 is the block diagram of our algorithm for 2D silhouettes and segmented boundary contours. The block diagram represents the CVM algorithm for 2D contours. We get back to the example of the circle again. The curvature of a circle is uniformly distributed. The density of curvature hence is a Kronecker delta function of strength one. The entropy of a Kronecker delta function is zero. This result implies that circular and linear contours convey no significant shape information. We note that circles of different radii will also have zero shape information. We argue that change in scale (radius) adds no extra shape information. On the other hand, the broader the curvature density, the higher the entropy and more complex the shape is. The most complex shape hence would be the one that has randomly varying curvature at every point on the boundary.



Figure 4.1: A circle and an arbitrary object.

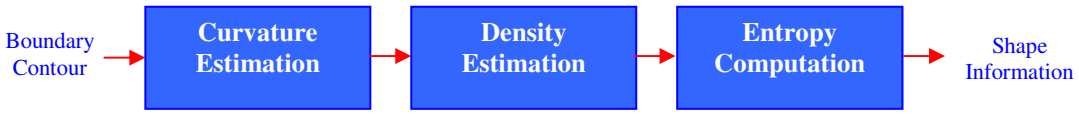


Figure 4.2: Block diagram of our CVM as the informational approach to shape description.

We use curvature because it is an information preserving feature, invariant to rotation and possesses an intuitively pleasing correspondence to the perceptive property of simplicity. Curvature completely parameterizes the boundary contour for efficient shape description of 2D curves and boundary contours. We counter the inverse relation to symmetry by using information theory. Symmetry does not contribute to more shape information (entropy) but rather reduces it.

4.1.2 Curvature-Based Automotive Component Description

Our CVM measure of shape on geometric curves is the entropy of curvature along the boundary contour. Curvature along the boundary provides us with sufficient detail in the 2D case but with 3D models and surfaces, it is not enough. We extend our idea of shape information on 3D meshes to describe surface variation of the smooth surface patches that make up the object and store the list of connected patches. We assume that we can reasonably describe most objects as a unique network of smooth patches. Then, the uniqueness of our description is to measure the variation in curvature across each of these patches.

We describe the algorithm in a pictorial fashion with a block diagram in Figure 4.3. Our description which could be used for purposes such as reverse engineering and inspection takes triangle meshes as the input. We take the example of the disc brake again. We break down the triangle mesh into surface patches based on the Dorai's [Dorai, 1997] definition of curvedness. Curvedness identifies sharp edges and abrupt surface changes. We perform a simple region growing segmentation by identifying a point and collecting the vertices whose face normal deviation is less than a particular angle. This angle is a free parameter. We have used 85 degrees as the maximum threshold angle before we meet an edge in the growing procedure. We save the connectivity information of each of these surface patches. Our segmentation is a crude

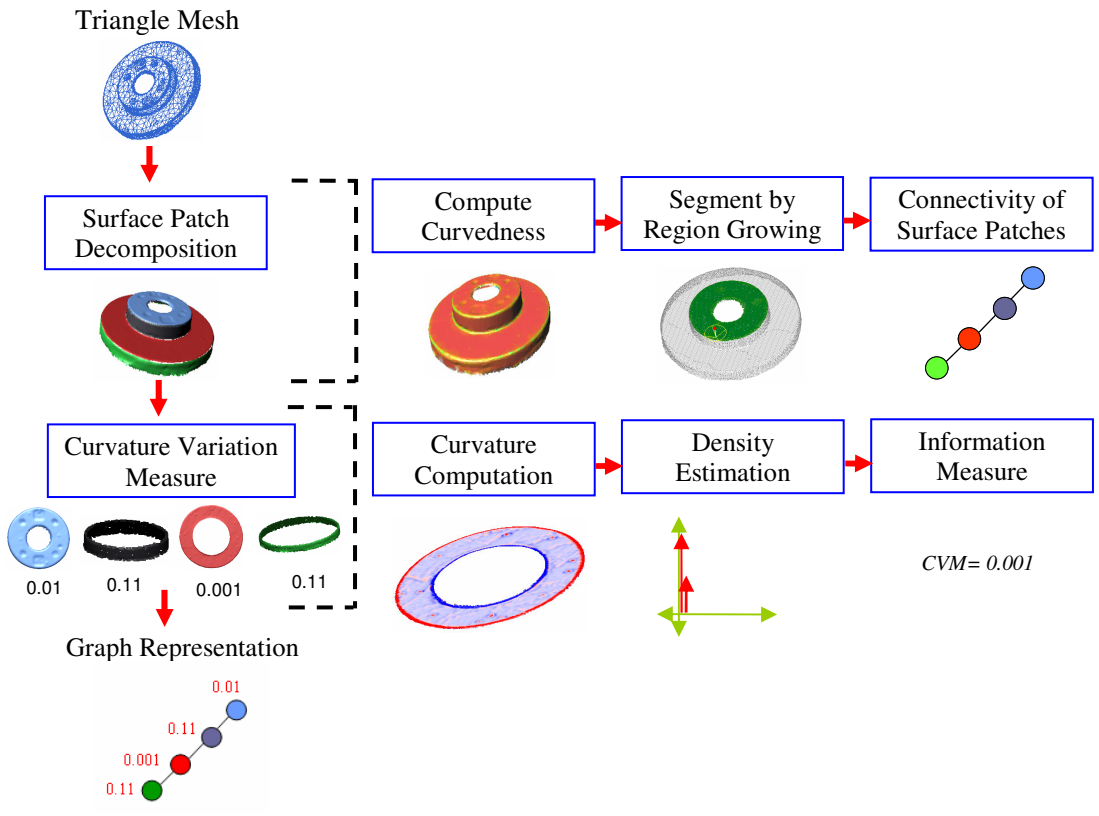


Figure 4.3: Block diagram of curvature-based vehicle component description algorithm including path decomposition and CVM computation.

implementation of Guillaume's algorithm [Guillaume, 2004]. He presents a more efficient algorithm for the decomposition of 3D arbitrary triangle meshes into surface patches. The algorithm is based on the curvature tensor field analysis and presents two distinct complementary steps: a region-based segmentation, which decomposes the object into known and near constant curvature patches and a boundary rectification based on curvature tensor directions, which correct boundaries by suppressing their artifacts and discontinuities.

We then analyze each surface patch individually to compute the CVM, which is the entropy of curvature. We compute the Gaussian curvature on each of those surface patches. The kernel density of the Gaussian curvature is estimated. We optimize the bandwidth of the kernel density using the plug-in method to ensure stability in the resolution normalized entropy. This log scale measure from the curvature density is the curvature variation measure (CVM). We then combine the surface connectivity information and the curvature variation measure into a single graph representation.

We call our CVM algorithm a curvature-based approach because the segmentation and the description require computation of curvature. (Curvedness is a function of principal curvatures.) However, the surface variation measure that we describe is not invariant to scale. We would like to emphasize that our algorithm can be used for describing occluded scenes as well but at the cost of partial graph matching if we have to attempt object recognition.

4.2 Building Blocks of the CVM algorithm

As background, we first present a brief overview of surface curvature in the important context of differential geometry. We in particular deal with curvature of a surface in Section 4.2.1 because we have assumed that curvature intrinsically describes the local shape of that surface. The differential geometry section helps us understand curvature estimation on triangle meshes. We present a brief survey on curvature estimation techniques in Section 4.2.2 and then discuss the theory behind the other building blocks of the algorithm in Section 4.2.3 and Section 4.2.4.

4.2.1 Differential Geometry of Curves and Surfaces

First, let us consider the continuous case for 2D curves. Using [Carmo, 1976], we arbitrarily define a planar curve $\alpha: I \rightarrow R^2$ parameterized by arc length s such that we have $\alpha(s)$. We carefully choose, without loss of generality, this parameterization such that the vector field $T = \alpha'$ has unit length. With this construction, the derivative $T' = \alpha''$ measures the way the curve is turning in R^2 and we term T' the curvature vector field. Since T' is always orthogonal to T, that is normal to, we can write that $T' = \kappa N$

where N is the normal vector field. The real valued function κ where $\kappa(s) = || \alpha''(s) ||$, s the curvature function of α and completely describes the shape of α in R^2, up to a translation and rotation. This curvature function is what we would like to exploit to define shape information of a curve. We would like to formulate the task of curvature estimation on discrete samples of such curves. For a planar curve α, we have samples $\alpha_j = \alpha(s_j)$. We assume that uniform sampling across the arc length of the curve such that $\Delta s = s_j - s_{j-1}$ is a constant. This approach leads to N samples over the curve α. Since we have uniform sampling along the curve κ_j is directly proportional to the turning angle θ_j formed by the line segments from end point α_{j-1} to α_j and from α_j to α_{j+1}.With 2D curves the definition and hence the computation of curvature is straightforward while its extension to 3D surfaces require some concepts in differential geometry.

On a smooth surface S, we can define normal curvature as a starting point. Consider Figure 4.4, the point p lies on a smooth surface S, and we specify the orientation of S at p with the unit-length normal N. We define S as a manifold embedded in R^3. We can construct a plane Π_p that contains p and N such that the intersection of Π_p with S forms a contour α. As before, we can arbitrarily parameterize $\alpha(s)$ by arc length s where $\alpha(0) = p$ and $\alpha'(0) = T$. The normal curvature $\kappa_p(T)$ in the direction of T is thus $\alpha'' (0) = \kappa_p(T)N$ .This single $\kappa_p(T)$ does not specify the surface curvature of S at p since Π_p is not a unique plane. If we rotate Π_p around N, we form a new contour on S with its own normal curvature. We can see that we actually have an infinite set of these normal curvatures around p in every direction. Fortunately, herein enters the elegance of surface curvature. For this infinite set, we can construct an orthonormal basis $\{T_1, T_2\}$ that completely describes the set. The natural choice for this basis is the tangent vectors associated with the maximum and minimum normal curvatures at p since the directions of these curvatures are always orthogonal.

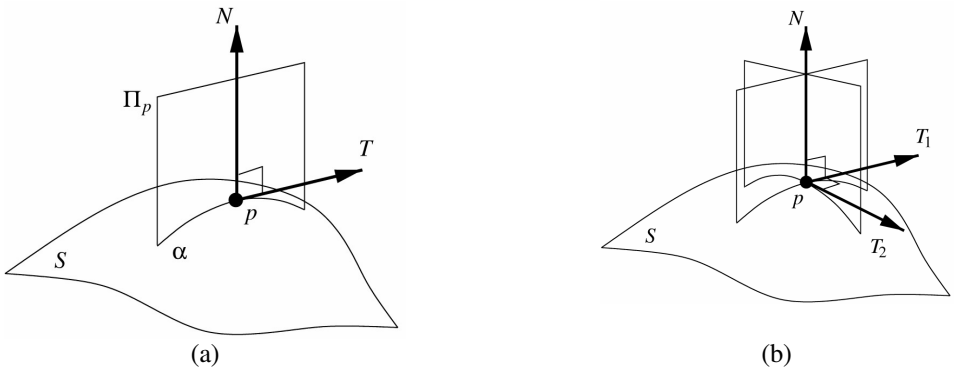


Figure 4.4: Illustration to understand curvature of a surface.

These maximum and minimum directions $\{T_1, T_2\}$ are the principal directions. The added benefit of choosing the principal directions as the basis set is that the curvatures $\kappa_1 = \kappa_p(T_1)$ and $\kappa_2 = \kappa_p(T_2)$ associated with these directions lead to the following relationship for any normal curvature at p:

$$\kappa_p(T_\theta) = \kappa_1 \cos^2(\theta) + \kappa_2 \sin^2(\theta), \tag{4.1}$$

where $T_\theta = \cos(\theta)T_1 + \sin(\theta)T_2$ and $-\pi \le \theta \le \pi$ is the angle to vector T_1 in the tangent plane. The maximum and minimum curvatures are known as the principal curvatures. The principal directions along with the principal curvatures completely specify the surface curvature of S at p and thus describe the shape of S. Combinations of the principal curvatures lead to other common definitions of surface curvature. The most commonly used is the Gaussian curvature, and is the product of the principal curvatures as shown in Equation 4.2.

$$K_p = \kappa_1 \kappa_2 \tag{4.2}$$

This definition highlights that negative surface curvature that occurs at hyperbolic occur where only one principal curvature is negative. The second definition of curvature is mean curvature. We specify mean curvature as the average of both principal curvatures (Equation 4.3). Mean curvature gives insight to the degree of flatness of the surface.

$$H_p = (\kappa_1 + \kappa_2)/2 \tag{4.3}$$

4.2.2 Curvature Estimation

Curvature estimation is a challenging problem on digitized representations of curves and surfaces. Consider a 2D function $y=f(x)$.The curvature of the continuous function y is mathematically defined as shown in Equation 4.4.

$$\kappa = \frac{\frac{d^2y}{dx^2}}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{3/2}} \tag{4.4}$$

Equation 4.4 assumes the rectangular coordinate system. If we parameterize $y = f(x)$ in the polar coordinate system the curvature equation can be rewritten as in Equation 4.5.

$$\kappa = \frac{r^2 + 2r_\theta^2 - rr_{\theta\theta}}{\left(r^2 + r_\theta^2\right)^{3/2}}; \quad r_\theta \Rightarrow \partial r/\partial\theta\,. \tag{4.5}$$

These equations for continuous functions of x can be extended to contours of images without much error by using the difference operator. We identified two key methods of computing curvature for 2D contours from [Oddo, 1992] and [Abidi, 1995]. Oddo follows the strict definition of curvature in the continuous case and extends it to the digitized curves. He argues that the turning angle at every pixel on the boundary with two other points at a fixed pixel distance is proportional to the curvature at that pixel. Abidi [Abidi, 1995] uses a method based on polar coordinates to estimate curvature. We have used a second order differential operator on the boundary contour to approximate curvature which is also proportional to the second derivative of a function for our implementation.

Curvature estimation on surfaces is a more challenging research area. After a detailed survey of the literature we would like to emphasize the fact that most of the research on curvature estimation is in the context of range images with very little of it suited for the general problem on surface meshes. Flynn et al. [Flynn and Jain, 1989] and Suk et al. [Suk and Bhandarkar, 1992] offer us with surveys on curvature from range methods. These methods give an insight of the fundamental problems that we might encounter with surface meshes. One of the major assumptions that we make with range images that prevents us from extending them to triangle meshes is that of a regular grid structure and consistent topology that might not be always the case with polygonal meshes. In the next few paragraphs, we present a brief survey on different curvature measures on triangle meshes. Triangle meshes are the most common output of 3D scanners and is assumed as a piecewise approximation to a surface.

Surface fitting methods try to apply concepts of differential geometry on surface approximations. An analytic surface is fit to the region of interest and curvature is computed from that functional approximation. Surface fitting methods do not differ much from the curvature-from-range methods because of the planar topology of fitted surfaces and range images. Surface fitting aside, researchers have tried to estimate curvature using curve fitting methods as well. A family of curves is fit around a point on the surface and the ensemble is used to compute principal curvatures. Besl and Jain [Besl and Jain, 1986] construct a local parameterization of the surface and estimate curvature by fitting orthogonal polynomials followed by a series of convolution operations. Stokely and Wu [Stokely and Wu, 1992] present five practical solutions, the characterized Sander-Zucker approach, two novel methods based on direct surface mapping, a piecewise linear manifold technique, and a turtle geometry method. One of the new methods, called the cross patch (CP) method, is shown to be very fast, robust in the presence of noise, and is based on a proper surface parameterization, provided the perturbations of the surface over the patch neighborhood are isotropically distributed. Kresk et al. [Kresk et al., 1998] summarize their experience with circle fitting, paraboloid fitting and the Dupin cyclide method. These three methods do not assume that the sample points be on a regular grid. They accept the speed and accuracy of the circle fitting method but doubt the robustness on dense polygonal

meshes. With paraboloid fitting which is slower than the circle fitting method, they point out the systematic error that is introduced by the procedure in estimating curvature of smooth and uniformly varying surfaces such as spheres and cylinders. The Dupin Cyclide method turns out slower and inaccurate compared to the paraboloid fitting method.

Another approach to curvature estimation is to use the geometry and topology of the surface approximation to estimate curvature. These methods compute total curvature as a global feature at each of the vertices of the triangle mesh though theoretically each sample point on the mesh is a singularity. Lin and Perry [Lin and Perry, 1982] use the angle excess around a vertex and extend the Gauss-Bonnet Theorem in differential geometry to define a total curvature measure. They relate it to the Gaussian curvature of the surface. Desbrun et al. in [Desbrun et al., 1999] derive an estimate of mean curvature on a triangle mesh based on the loss of angle approach. Delingette [Delingette, 1999] lays out a framework called simplex meshes as a dual of triangle meshes for surface representation and formulates curvature measures on the surface very similar to the angle excess method on the triangle mesh. Gourley in [Gourley,1998] attempts to approximate a curvature metric based on the dispersion of face normals around a vertex while Mangan and Whitaker [Mangan and Whitaker,1999] refine a curvature measure further as the norm of a covariance matrix for the face normals. Chen and Schmitt [Chen and Schmitt, 1992] formulate a quadratic representation of curvature at each vertex to derive principal curvatures by minimizing least squares. Taubin [Taubin, 1995] enhances Chen's approach into an elegant algorithm that defines a symmetric matrix that has the same Eigen vectors as the principal directions and Eigen values that are related by a homogenous linear transformation to principal curvatures. Talking of Eigen analysis [Page, 2001] proposes the idea of normal vector voting that selects a geodesic neighborhood around each vertex. The triangles in this neighborhood vote to estimate the curvature at the specified vertex. He collects these votes in a covariance matrix and uses Eigen analysis of the matrix to estimate curvature. The relative size of the neighborhood controls the trade-off between algorithm robustness and accuracy.

We would like to summarize by saying that the surface fitting methods require the most computational effort since they typically employ optimization in the fitting process. They are robust to noise but cannot deal with discontinuities. Curve fitting methods on triangle meshes are extremely sensitive to noise yet very simple. Of the methods discussed in the previous paragraphs, we have decided to perform an analysis as to which of these would help us in characterizing surfaces and their complexity. We chose to compare Gaussian curvature estimates using the paraboloid fitting method, Taubin's method, angle deficit as curvature and the Gauss-Bonnet's extension to curvature estimation. Our comparison differs from [Surazhsky,2003] and [Meyer,2000] because we not only focus on the absolute error in the estimation of curvature, but also the effect of resolution on the same surface and how well each one

of these methods can be exploited as surface shape complexity descriptors. We present the implementation issues and the results of analysis in the next chapter and justify the use of Gauss-Bonnets method of computing curvature.

4.2.3 Density Estimation

Probability density functions are ubiquitous when it comes to intelligent decision making and modeling. In this section of the document, we survey some of the key density estimation techniques. Research in this field of density estimation dates back to the early 1950's and was proposed by Fix and Hodges in 1951 [Fix and Hodges, 1951] as a breakthrough of freeing discriminant analysis from rigid distributional assumptions. Since then it has undergone application oriented metamorphosis. Rosenblatt introduces the concept of non-parametric density estimation as an advanced statistical method [Rosenblatt, 1956]. Parzen follows that up with remarks on a model that aims at non-parametric estimation of a density function [Parzen, 1962].

Density estimation is generally approached in two different ways. One of them is the parametric approach that assumes that the data has been drawn from one of the established parametric family of distributions such as the Gaussian and Rayleigh with a particular mean, variance and other well defined statistical parameters. The density *f* underlying the data could then be estimated by finding the estimates of the mean and the variance from the data and then substituting these values into the formula of the assumed density. The parametric approach to density estimation is bounded by the rigid assumption of the shape of the density function independent of the observed data. The non-parametric approach however is less rigid in its assumptions. The data speak for themselves in determining the estimate of *f*. Silverman [Silverman, 1986] traces the evolution of density estimation techniques for a uni-variate dataset represented as a sample of *n* observations of a data set $X = \{x_1, x_2, x_3, x_4, \ldots, x_n\}$. We briefly survey such techniques in the next few paragraphs.

The oldest and probably the most widely used non-parametric density estimate is the histogram. A histogram is constructed by dividing the real line into equally sized intervals, often called bins. The histogram is then a step function with heights being the proportion of the sample contained in that bin divided by the width of the bin. If *h* denotes the width of the bins (bin width) and *n* represents the number of samples in the dataset then the histogram estimate at a point *x* is given by

$$\hat{f}(x) = \frac{(number\ \ of\ \ X_i\ in\ the\ same\ \ bin\ as\ \ x)}{n\ h} \tag{4.6}$$

The construction of the histogram depends on the origin and bin width, the choice of the bin width primarily controlling the inherent smoothing of the density estimate.

Histograms are good representation tools but not efficient density estimates. We discuss the effect of bin width on the histogram with a simple example in Figure 4.5. It is important to note the significant change in shape and the density of the estimate.

Another method that is an improvement on the histogram used to estimate the density is the naïve estimator. It is based on the fact that if the random variable y has density f then

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h < X < x + h). \tag{4.7}$$

Thus a natural estimator f of the density can be obtained by choosing a small number h as shown in Equation 4.6.

$$\hat{f}(x) = \frac{[number\ of\ x_1, x_2 \ldots . x_n\ falling\ in\ (x - h, x + h)\ ]}{2 \times h \times n} \tag{4.8}$$

The naïve estimator can also be mathematically expressed as follows

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} w\left(\frac{x - X_i}{h}\right) \tag{4.9}$$

where $w(x)$ represents a rectangle function of height 0.5 and width of 2.

It is easy to generalize the naive estimator to overcome its rugged nature of the density by replacing the weight function w by a kernel function K which satisfies the condition described in Equation 4.10.

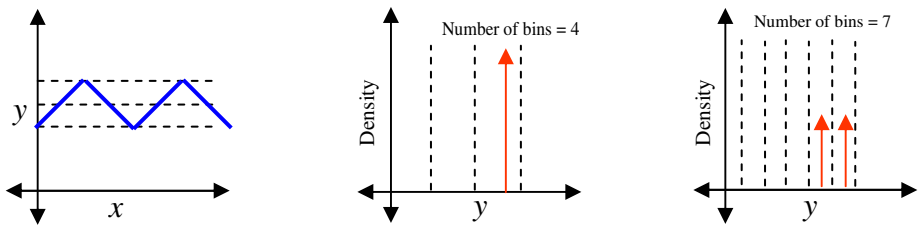


Figure 4.5: Illustration that shows the effect of bin width on density estimation using a histogram.

$$\int_{-\infty}^{\infty} K(x)\,dx = 1. \tag{4.10}$$

Analogous to the definition of the naive estimator, the kernel estimator with kernel K is defined by

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{4.11}$$

While the naive estimator can be considered as a sum of boxes centered at the observations, the kernel estimator is a sum of bumps placed at the observations. The kernel function K determines the shape of the bumps while the window width h determines their width. It suffers inaccuracy with long tailed distributions because of the fixed bandwidth throughout the process of density estimation.

The nearest neighbor class of estimators represents an attempt to adapt the amount of smoothing to the `local' density of data. The degree of smoothing is controlled by an integer k, chosen to be considerably smaller than the sample size; typically $k \approx n^{1/2}$. Define the distance $d(x, y)$ between two points on the line to be $|x - y|$ in the usual way, and for each t define $d_1(t) \le d_2(t) \le \ldots \le d_n(t)$ to be the distances, arranged in ascending order, from t to the points of the sample.

The kth nearest neighbor density estimate is then defined by

$$\hat{f}(t) = \frac{k}{2nd_k(t)}. \tag{4.12}$$

While the naive estimator is based on the number of observations falling in a box of fixed width centered at the point of interest, the nearest neighbor estimate is inversely proportional to the size of the box needed to contain a given number of observations. In the tails of the distribution, the distance $d_k(t)$ will be larger than in the main part of the distribution, and so the problem of under smoothing in the tails is reduced. Like the naive estimator, to which it is related, the nearest choice neighbor estimate as defined is not a smooth curve. The function $d_k(t)$ can easily be seen to be continuous, but its derivative will have a discontinuity. We would like to achieve stability with the information measure and since most of the surfaces that we are interested in have smooth analytical parameterization, we are inclined to chose the continuous and smooth looking kernel density estimate. We show how each of these methods estimate the density of the same dataset in Figure 4.6. We have reproduced Figure 4.6 from [Silverman, 1986].

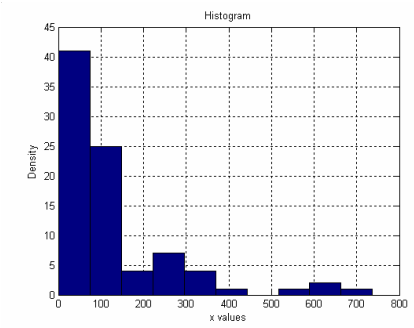
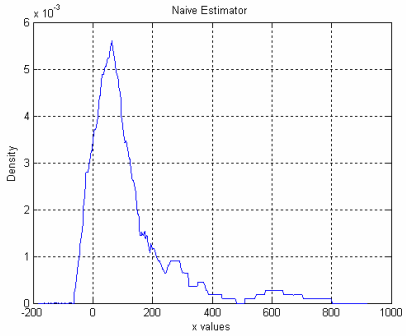
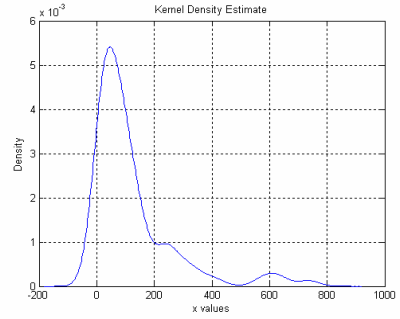
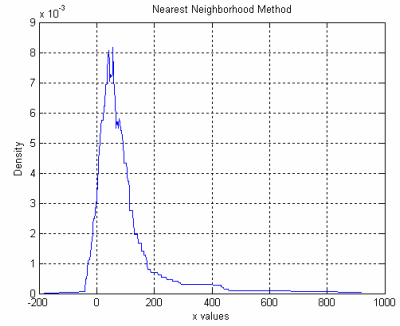
Method used to compute density	**Plot of the density function**
Histograms	
The Naïve Estimator	
The Kernel Density Estimator	
The Nearest Neighborhood Method	

Figure 4.6: Different methods used to estimate the density of the same dataset. Adapted from [Silverman, 1986].

There is a plethora of research in automating the process of bandwidth estimation that will give us the best estimate of the density as possible. Recollecting Equation 4.9, the parameters that influence the density estimate are the kernel function, density span and the kernel bandwidth. We ignore the effect of kernel function and density span because of the assumption that we have represented the digitized surface with enough points to represent a continuous surface. We hence assume that our dataset is too large to react to the effect of the different kernel functions listed in Table 4.1. We have decided to use the Gaussian kernel for our implementation for its continuity though the Epanechnikov kernel is considered to be the most efficient of kernel functions.

We have performed a simple experiment on a normally distributed pseudo random data set with zero mean and unit variance at 512 points to study the effect of the bandwidth parameter on density estimation. In Figure 4.7, the red curves represent the ground truth Gaussian density function and the blue ones represent the estimated density. Figures 4.7(a-f) portray the amount of smoothing that the bandwidth parameter imposes on the estimated density. Figure 4.7 illustrates the importance of the bin width parameter in density estimation with a simple example. Though the figures represent the density of the same dataset, we are able to anticipate the instability of its information measure.

The paper [Turlach, 1996] is an excellent survey on bandwidth selection in kernel density estimation. The books [Silverman, 1986] and [Wand, 1995] are classics in the field of kernel estimation and kernel smoothing and have detailed descriptions of kernel

Table 4.1: Kernel functions.

Kernel	K(u)
Uniform	$\frac{1}{2} I(\lvert u \rvert \leq 1)$
Triangle	$(1-\lvert u \rvert) I(\lvert u \rvert \leq 1)$
Epanechnikov	$\frac{3}{4}(1-u^2) I(\lvert u \rvert \leq 1)$
Triweight	$\frac{15}{16}(1-u^2)^2 I(\lvert u \rvert \leq 1)$
Quartic	$\frac{35}{32}(1-u^2)^3 I(\lvert u \rvert \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}} e(\frac{-u^2}{2})$
Cosinus	$\frac{\pi}{4} cos(\frac{\pi u}{2}) I(\lvert u \rvert \leq 1)$

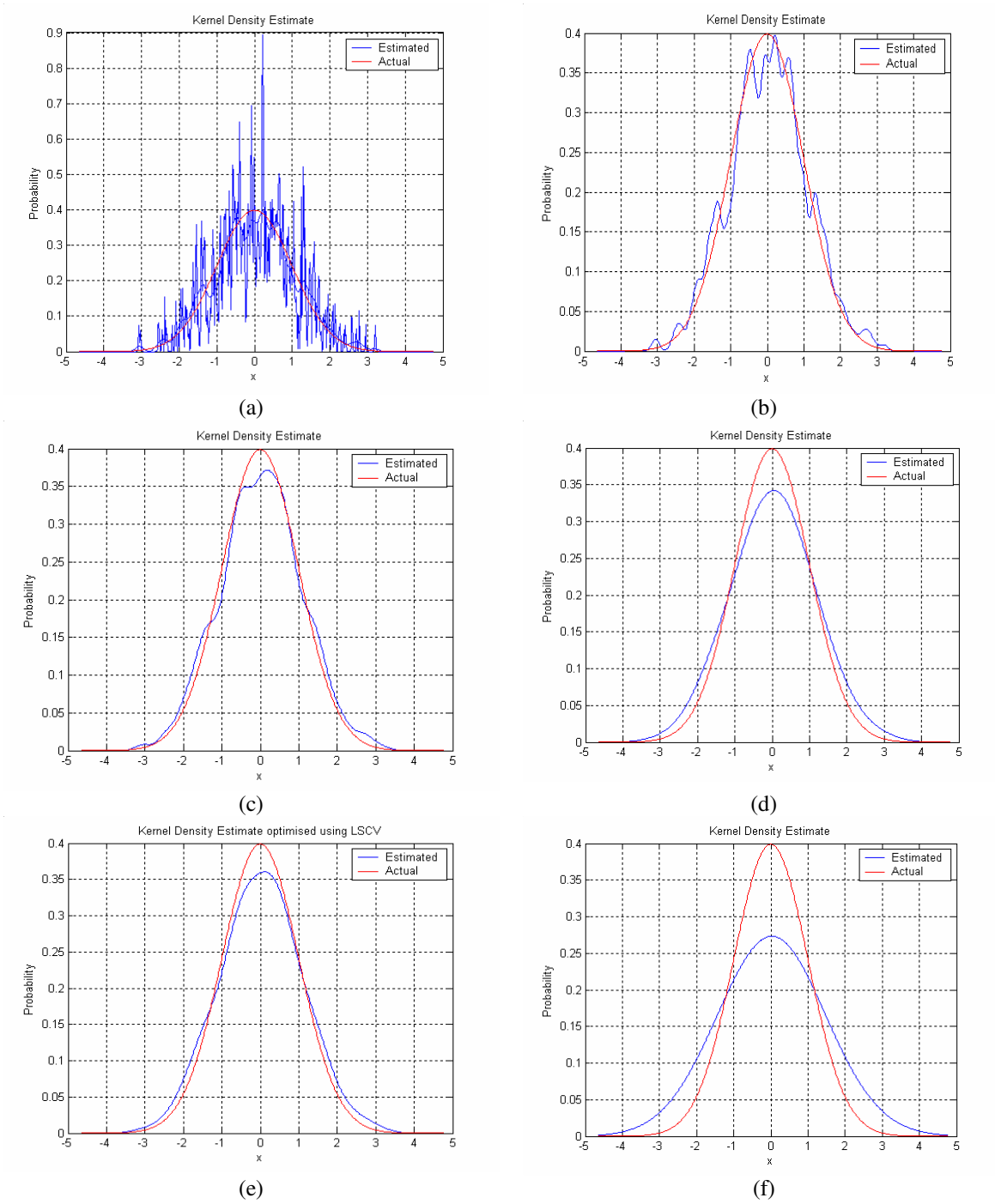


Figure 4.7: Effect of bandwidth parameter on kernel density. (a) KDE for h = 0.01. (b) KDE for h = 0.1. (c) KDE for h = 0.3. (d) KDE for h = 0.5. (e) KDE for h = 0.328(optimal). (f) KDE for h = 1.

smoothing as applied to uni-variate and multi-variate datasets. Papers that discuss the information bound bandwidth selection methods are [Wu and Lin, 1996] and [Jones et al., 1996].

The bandwidth selection for the process of density estimation is important to assert the accuracy of the density estimate. The choice of the bandwidth at least theoretically can be derived to minimize the mean integrated square error between the actual density and the computed density. Some methods that are used for this purpose are

- ❖ Distribution Scale Methods,
- ❖ Cross Validation Methods,
- ❖ Plug-In Methods, and
- ❖ Bootstrap methods.

In the next few paragraphs, we will very briefly discuss the rationale behind these objective methods for bandwidth selection. Assume that f is the actual density of the data and $\hat{f}$ is the estimated density. The process of bandwidth selection is aimed at minimizing the integrated mean square error between the actual and the estimated density. The integrated mean square error is defined as the expected value of the integrated square error and is given by Equation 4.13.

$$MSE\{ f(x;h)\} = n^{-1}\{(K_h^2 * f)(x) - (K_h * f)^2(x)\} + \{(K_h * f)(x) - f(x)\}^2. \qquad (4.13)$$

The Mean Integrated Square Error (*MISE*) is the integral of the mean squared error that can be simplified as shown below.

$$MISE\{\hat{f}(x,h)\} = \int MSE\{\hat{f}(x;h)\}dx = E\int\{\hat{f}(x;h) - f(x)\}^2 dx$$
$$MISE\{\hat{f}(.;h)\} = n^{-1}\int\{(K_h^2 * f)(x) - (K_h * f)^2(x)\}dx + \int\{(K_h * f)(x) - f(x)\}^2 dx$$
$$MISE\{\hat{f}(.;h)\} = (nh)^{-1}\int K^2(x)dx + (1-n^{-1})\int (K_h * f)^2(x)dx - 2\int (K_h * f)(x)f(x)dx + \int f(x)^2 dx \qquad (4.14)$$

MISE is the sum of the integrated square bias and the integrated variance and hence minimization of that error is effectively the tradeoff between the bias and variance. The closed form solution that is derived for the optimal bandwidth by minimizing the MISE is the h_{opt} in Equation 4.15.

$$h_{opt} = \left[\frac{\int K(z)^2 dz}{n(\int z^2 K(z)dz)^2(\int f''(x)^2 dx)}\right]^{1/5} \qquad (4.15)$$

The problem with using this closed form solution is the dependence of the optimal bandwidth on the second derivative of the density function *f* that we are trying to compute. By using the Gaussian kernel for our implementation we have ensured the differentiability of the estimated density and also justified the reason for not choosing the naïve estimator or its rugged counterparts.

Two popular but quick and simple bandwidth selectors are based on the normal scale rule and maximum smoothing principle. For example, an easy approach would be to make use of a standard family of distributions to assign a value to the double derivative term. In Equation 4.12 we assume normal density and compute the second derivative. This method can lead to gross errors in cases when the data is not distributed the way it was assumed.

$$\int f''(x)^2 dx = \sigma^{-5} \int \phi''(x)^2 dx \approx 0.212\sigma^{-5} \tag{4.16}$$

The rationale behind the principle of cross validation is to use the same dataset to extract data points partially as a construction set and a training set. A model is fit assuming the correctness of the training dataset and is tested for accuracy with the construction dataset. The error in the estimate is minimized by defining a cost function of the error. Based on the construction of the cost function, methods are named as least squares cross validation, biased cross validation and likelihood cross validation. More advanced bandwidth selectors are the plug-in and the bootstrap methods that "plug-in" estimates of the unknown quantities that appear in the formulae for asymptotically optimal bandwidth. Bootstrap methods make use of a pilot bandwidth to initialize the density estimation process and improve the pilot bandwidth based on the data. In Equation 4.15, we show the plug-in method of bandwidth selection. Plug-in methods involve the estimation of the integrated squared density derivatives called functionals.

$$\hat{h} = \left[\frac{243R(K)}{35\mu_2(K)^2 n} \right]^{\frac{1}{5}} \hat{\sigma} \quad where\ R(K) = \int K(t)^2 dt, \mu_2(K) = \int t^2 K(t) dt \tag{4.17}$$

$$\text{and } \hat{\sigma} = med_j \left| X_j - med_i(X_i) \right| \text{ is the absolute deviation.} \tag{4.18}$$

We discuss implementation issues in the next chapter. Our next building block is the information measure on the accurate density of curvature estimated using the bandwidth optimized kernel density estimators.

4.2.4 Information Measure

Information theory is a relatively new branch of mathematics that began only in the 1940's. The term "information theory" still does not possess a unique definition but broadly deals with the study of problems concerning systems that involve information processing, information storage, information retrieval and decision making.

The first studies in this direction were undertaken by Nyquist in 1924 and by Hartley in 1928 (Equation 4.17) that recognized the logarithmic nature of the measure of information. In 1948, Shannon published his seminal paper on the properties of information sources and of the communication channels used to transmit the outputs of these sources and the important definition of entropy as the measure of information (Equation 4.18).

$$H_{Hartley}(p_1, p_2, \ldots p_n) = log\,|\{i; p_i > 0, 1 \le i \le n\}| \tag{4.19}$$

$$H_{Shannon} = -\sum_{i=1}^{n} p_i \, log \, p_i \tag{4.20}$$

In the past fifty years, literature on information theory has grown quite voluminous and apart from communication theory it has found deep applications in many social, physical and biological sciences, economics, statistics, accounting, language, psychology, ecology, pattern recognition, computer sciences and fuzzy sets.

A key feature of Shannon's information theory is the term "information" that can often be given a mathematical meaning as a numerically measurable quantity, on the basis of a probabilistic model. This important measure has a very concrete operational interpretation for the communication engineers. We would like to summarize the various definitions of entropy in the literature as Table 4.2.

The list that we have presented in Table 4.2 is not an exhaustive one though we have spanned a few important definitions involving parameters and weights. Pap in [Pap, 2002] briefs the history of information theory and discusses various measures of information while Reza [Reza, 1994] approaches information theory from the coding aspect of communication theory. We would like to emphasize that the difference in using a discrete random variable and a continuous random variable. The analogous definition of Shannon's entropy in the continuous case is called the differential entropy (Equation 4.21).

$$H_{differential} = \int_{-\infty}^{\infty} p(x) log(p(x)) dx \tag{4.21}$$

Table 4.2: List of entropy type measures of the form $H(P)=h\left(\frac{\sum_{i=1}^{M} v_i \cdot \varphi_1(p_i)}{\sum_{i=1}^{M} v_i \cdot \varphi_2(p_i)}\right)$.

Measure	$h(x)$	$\phi_1(x)$	$\phi_2(x)$	v_i
1	x	$-x\log(x)$	x	v
2	$(1-r)^{-1}\log x$	x^r	x	v
3	x	$-x^r\log(x)$	x^r	v
4	$(s-r)^{-1}\log(x)$	x^r	x^s	v
5	$(1/s)\arctan(x)$	$x^r\sin(s\log x)$	$x^r\cos(s\log x)$	v
6	$(m-r)^{-1}\log(x)$	x^{r-m+1}	x	v
7	$(m(m-r))^{-1}\log(x)$	$x^{r/m}$	x	v
8	$(1-t)^{-1}\log x$	x^{t+s-1}	x^s	v
9	$(1-s)^{-1}(x-1)$	x^s	x	v
10	$(t-1)^{-1}(x^t-1)$	$x^{1/t}$	x	v
11	$(1-s)^{-1}(e^x-1)$	$(s-1)x\log(x)$	x	v
12	$(1-s)^{-1}(x^{\frac{s-1}{r-1}}-1)$	x^r	x	v
13	x	$-x^r\log(x)$	x	v
14	$(s-r)^{-1}x$	x^r-x^s	x	v
15	$(\sin s)^{-1}x$	$-x^r\sin(s\log x)$	x	v
16	$(1+\frac{1}{\lambda})\log(1+\lambda)-\frac{x}{\lambda}$	$(1+\lambda x)\log(1+\lambda x)$	x	v
17	x	$-x\log(\frac{\sin(sx)}{2\sin(s/2)})$	x	v
18	x	$\frac{\sin(xs)}{2\sin(s/2)}\log(\frac{\sin(sx)}{2\sin(s/2)})$	x	v
19	x	$-x\log(x)$	x	w_i
20	x	$-\log(x)$	1	v_i
21	$(1-r)^{-1}\log x$	x^{r-1}	1	v_i
22	$(1-s)^{-1}(e^x-1)$	$(s-1)\log(x)$	1	v_i
23	$(1-s)^{-1}(x^{\frac{s-1}{r-1}}-1)$	x^{r-1}	1	v_i

With the help of Figure 4.8 we would like to explain an issue with the Shannon type entropy measures. As the resolution of the data increases, the number of points in the density is also going to increase and Δ tends towards zero. Using Reiman's definition of integrals we can rewrite Equation 4.18 as

$$-\sum \Delta f(x_i)\log(\Delta f(x_i)) = -\sum \Delta f(x_i)\log(f(x_i)) - \sum \Delta f(x_i)\log(\Delta)) \tag{4.22}$$

$$-\int_{-\infty}^{\infty} f(x)\log f(x)dx = \lim_{\Delta\to 0}(H_{Shannon} + \log(\Delta)) \tag{4.23}$$

We see that as the number of points approaches the continuous random variable, there is a quantum jump in the amount of information measured. We needed a measure that is normalized and improves with resolution. The measures that we have presented in Table 4.2 have an upper limit that is directly proportional to the number of characters in a symbol. Since we need to have the shape information quantized and independent of resolution, we have studied different divergence measures such as KL divergence (Equation 4.24), Jenson-Shannon divergence (Equation 4.25) and Chi-Squared divergence measures before extending Shannon's definition for our CVM.

$$H_{KL} = -\sum p(x)\log\frac{p(x)}{q(x)} \tag{4.24}$$

$$H_{JS} = H_{Shannon}\left(\frac{p+q}{2}\right) - \frac{H_{Shannon}(p) + H_{Shannon}(q)}{2} \tag{4.25}$$

where p is the density of the object of interest and q is the density of the reference.

We have discussed the supporting theory for the proposed CVM algorithm. In the next chapter we discuss implementation decisions for the algorithm and present the experimental results of our algorithm on different datasets.

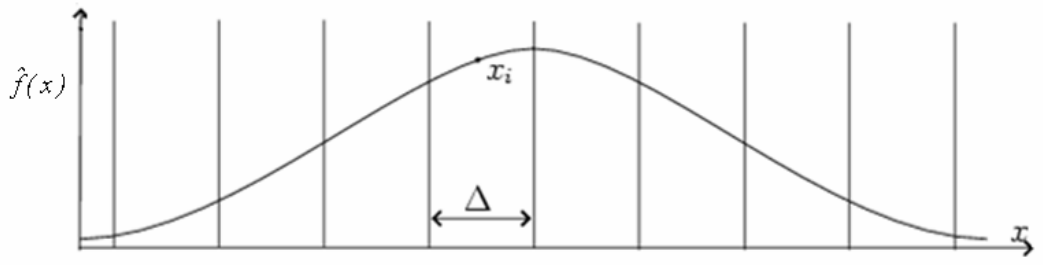


Figure 4.8: Resolution issue with Shannon type measures.

5 ANALYSIS AND RESULTS

We begin this section with important implementation decisions on each of the building blocks for the proposed CVM algorithm. We discuss our algorithm and justify our choice of methods before we present analysis results on intensity images, range images and 3D mesh models.

5.1 Implementation Decisions on the Building Blocks

We have acquired the data and are ready for shape analysis with the CVM. We use triangle mesh datasets as our input. Since our algorithm is a curvature-based algorithm, our first task is to compute curvature at the vertices on the mesh. In Section 5.1.1 we discuss various curvature estimation methods with analysis results on the effectiveness of these measures for surface description. We use curvedness, which is a function of principal curvatures, to perform segmentation. We then perform "region growing" to identify the regions and create a mapping of the vertex and the region to which it belongs. We use curvature at each of these vertices in a particular region to compute the CVM. In short, our CVM algorithm is a three pass algorithm; the first pass is for the estimation of curvature and curvedness, the second one to map vertices to smooth patches (segmentation) and the third one to compute the surface variation measure that we represent in a region adjacency graph.

5.1.1 Analysis of Curvature Estimation Methods

We recall the mathematical definition of triangle meshes as a set of vertices and a list of triangles connecting these vertices. We would like to define more specific terms before we discuss the implementation of curvature estimation methods. A vertex v_i is considered as an immediate neighbor of vertex v if edge vv_i belongs to the mesh. We denote the set of neighboring vertices by $\left[v_i\right]_{i=0}^{n-1}$ and the set of the triangles containing the vertex v by $\left[T_i\right]_{i=0}^{n-1}$ where

$$T_i^v = Triangle(\ v_i\ v\ v_{(i+1)\bmod n}\ ), 0 \le i \le n-1 \tag{5.1}$$

We define N_v as the normal of surface S at a vertex v. We compute the normal at a vertex using the normals of the triangles that contain the vertex. The normal of a triangle is the normal of the plane that fits the three points and is given by Equation 5.2. We compute the vertex normal as the average of these normals weighted by area of the triangles involved.

$$N_i^v = \frac{(v_i - v) \times (v_{(i+1) \bmod n} - v)}{\| (v_i - v) \times (v_{(i+1) \bmod n} - v) \|} \tag{5.2}$$

$$\overline{N}_v = \frac{1}{n} \sum_{i=0}^{n-1} N_i^v \;\; ; \;\; \overline{N}_v = \frac{\overline{N}_v}{\| N_v \|} \tag{5.3}$$

We show a small section of a triangle mesh in Figure 5.1 to understand the definitions better. The blue colored point in the middle is the vertex at which we would like to compute the curvature. Points in red are the neighbor points and the lines connecting the vertex v and its neighbors are the triangles that determine the surface. N_v is the normal at the vertex that we have defined in Equation 5.3.

The paraboloid fitting method [Kresk, 1998] at each vertex is computed by translating the vertex under consideration to the origin and its neighbors are rotated so that the vertex normal coalesces with the z axis. The osculating paraboloid of the form $z = ax^2 + bxy + cy^2$ is assumed to contain these transformed points. The coefficients a, b, c are found by solving a least square fit to v and the neighboring vertices $[v_i]_{i=0}^{n-1}$. The total and mean curvatures are computed using the formula in Equation 5.4.

$$\kappa = 4ac - b^2 \;\; ; \;\; H = a + c \tag{5.4}$$

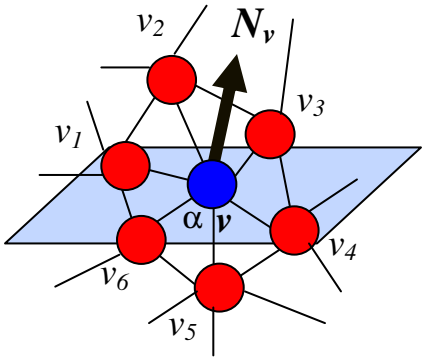


Figure 5.1: Neighborhood of a vertex in a triangle mesh.

Gauss-Bonnet approach [Lin, 1982] makes use of the angle α_i at v and two successive edges. The reduced form of Gauss-Bonnet theorem for polygonal meshes is given in terms of the loss of angle as Equation 5.5.

$$\iint_A KdA = 2\pi - \sum_{i=0}^{n-1} \alpha_i \tag{5.5}$$

Assuming that K is a constant in that neighborhood, Gaussian and mean curvature is computed as

$$\kappa = \frac{2\pi - \sum_{i=0}^{n-1} \alpha_i}{\frac{A}{3}}; H = \frac{\frac{1}{4}\sum_{i=0}^{n-1} || e_i || \beta_i}{\frac{A}{3}} \tag{5.6}$$

where A is the accumulated area of all the triangles that contain vertex v and $|| e_i || \beta_i$ is the measure of angle deviation between the normal at vertex v and its neighbor. Desbrun et al. [Desbrun et al., 1999] reduces the normal deviation as the sum of the cotangents of the angle formed at the neighbor vertex. Taubin [Taubin, 1995] defines a symmetric matrix using the integral formula involving the normal curvature. Assuming that vertex normals at each vertex have been computed, a matrix M_v is approximated with a weighted sum over the neighbor vertices v where T_i is the unit length normalized projection of vector $(v_i - v)$ onto the tangent plane at v.

$$M_v = \sum_{i=0}^{n-1} w_i \kappa_n(T_i) T_i T_i^t \tag{5.7}$$

$$T_i = \frac{\left[I - N_v N_v^t\right]\left[v - v_i\right]}{|| \left[I - NvN_v^t\right]\left[v - v_i\right] ||} \tag{5.8}$$

The weights w_i in Equation 5.7 are selected proportional to the sum of surface areas of the triangles incident to both vertices v and v_i. The matrix M_v is restricted to the tangent plane and its Eigen values correspond to the principal values of curvature.

We compare different approaches to choose one of these for our CVM algorithm. We make our decision based on a few experiments. We have chosen a saddle surface for which we can compute the analytical curvature. We show the surface Gaussian curvature also as a 3D mesh in Figure 5.2 because the variation of curvature along the surface can be visualized better. We have also presented a simple multi-resolution experiment in Figure 5.2. We have sampled the saddle surface so that each surface mesh model is made up of 161 vertices, 961 vertices and 10000 vertices respectively. We have computed curvature based on each of the methods discussed in the previous paragraphs. We observe that the curvature estimate of the Gauss-Bonnets approach and

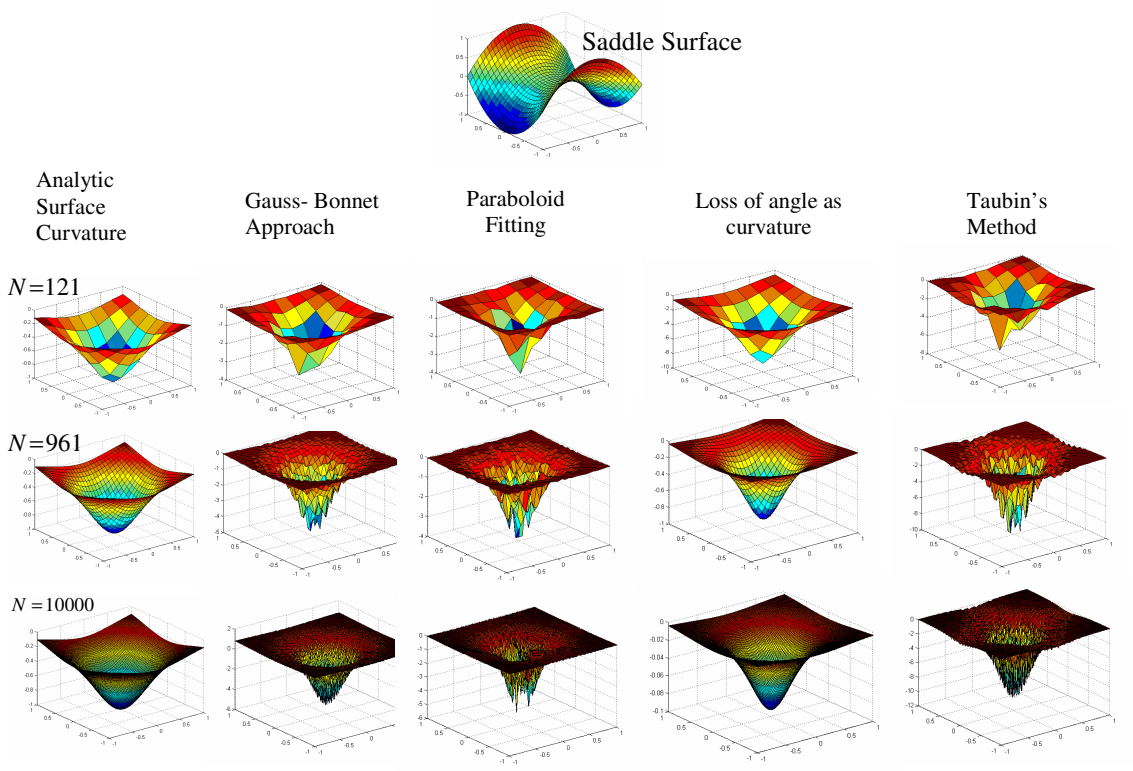


Figure 5.2: Curvature analysis – Multi-resolution error analysis experiment with four different approaches to curvature estimation on triangle meshes.

loss of angle approach are two methods that give us good estimates of Gaussian curvature in comparison with the paraboloid and Taubin methods. We also observe that as the resolution of the data increases Taubin's method also improves drastically. The paraboloid fitting method appears to have the sampled version of the analytical curvature. Since we have fit an analytical surface at each vertex to compute curvature around it, the error in this method seems to have accumulated throughout the mesh.

We next performed the curvature analysis on the unit sphere whose Gaussian curvature estimate should be equal to the reciprocal of the radius. We show how each one of these methods has behaved with the sphere at different resolutions in Figure 5.3. We would like to reiterate the large error in the paraboloid fitting methods at low resolutions. Since we are interested in a scheme that is consistent at all resolutions we need to make a choice between Gauss-Bonnet, loss of angle and Taubin's methods.

We have created synthetic surfaces such as the spherical cup, saddle and a monkey saddle. Visually and analytically the monkey saddle surface has the maximum variation in curvature. We decided to choose the method that categorically shows the variation. We call the variation as the span for curvature and plot it against each surface for the four methods in Figure 5.4. We conclude that Taubin and Gauss-Bonnet's approach for curvature estimation yields accurate results. We have used Taubin's method to compute principal curvatures and the Gauss-Bonnet approach for the Gaussian curvature for our implementation.

We have combined the simplicity of the Harvard mesh library (written by X. Gu) and speed of Triangle mesh library (written by Michael Roy) for our triangle mesh processing. Both the libraries are open source implementations of the half-edge data structure in C++. We have used the Microsoft Developer Environment (Microsoft Visual C++7.0) as our programming platform. For graphs and plots however we have used MATLAB.

5.1.2 Density Estimation for Information Measure

We would like to document our experience with the bandwidth optimization methods. Before incorporating it into our algorithm we have used the MATLAB (implementation of Christian Beardah's) toolbox on kernel density estimation. With ground truth normal density, we have concluded that cross validation methods give us accurate results. We have compared least squares cross validation, smoothed cross validation, likelihood cross validation, biased cross validation, distribution scale methods and the plug-in method. With large data cross validation though accurate was the most time consuming. Cross validation is a $O(N^2)$ complex algorithm in the worst case and had convergence problems with our real data. Sometimes cross validation methods result in monotonic cost functions that output the lower limit as the optimal bandwidth. We use the plug-in method. The plug-in method is a multi pass paradigm

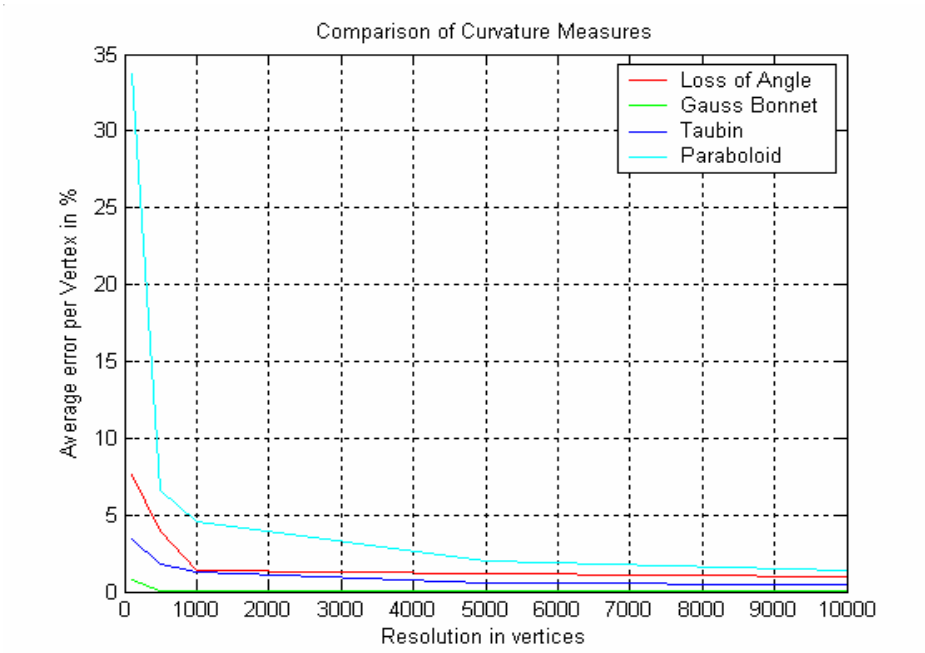


Figure 5.3: Curvature analysis – Error in curvature of a sphere at multiple resolutions.

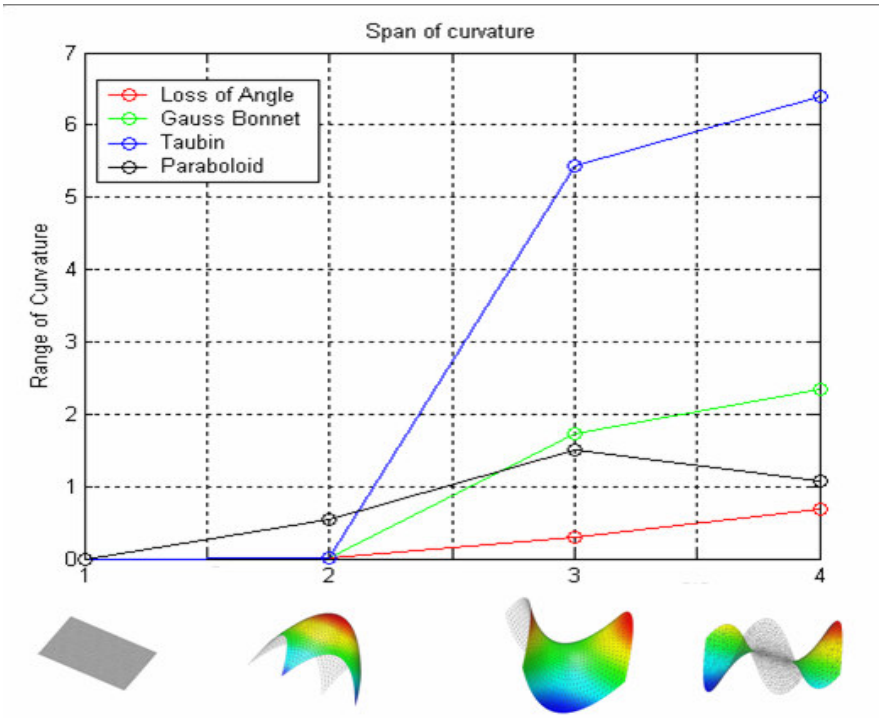


Figure 5.4: Curvature analysis – Variation in curvature for surface description.

that makes use of an equation involving quartiles to output a single number as the optimal bandwidth. We have observed that it sometimes gives us under-smoothed data compared to the cross validation methods. We have decided to use the plug-in method for bandwidth optimization because we want our algorithm to be fully automatic without us having to interfere. Another important parameter with the density distribution that decides accuracy of the estimate is the number of points at which we calculate the density.

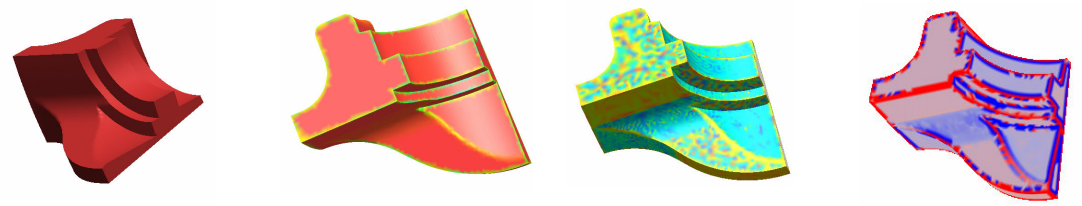
Another small but significant implementation issue that we would like to throw light upon is the difference between continuous random variables and discrete random variables. The discrete density function is not a sampled form of the continuous density function. We note that the density at each point of a discrete random variable is less than or equal to one and the sum of the densities is unity.

Since some values of the density function estimated are possibly zero and since we are using a logarithmic information measure, we should get around the zero points of the density function. We do not compute entropy at the zero points.

5.2 State-of-the-Art Shape Descriptors

The analysis in this section is the backbone of our CVM algorithm. We have implemented a few state-of-the-art algorithms to better understand the process of shape extraction from triangle meshes and also to know about the existing curvature-based metrics. Now that we have accurate measures of principle curvature and Gaussian curvature, we are able to identify curvedness and shape index used by Dorai in her "COSMOS" framework for shape recognition on range images.

In Figure 5.5 we show curvedness, shape index and the Gaussian curvature color coded models of the fan disk. (Model source of the fan disk: Hughes Hoppe, Microsoft Research.) By color coding we mean we have attributed color in the RGB spectrum to each vertex of that model. The cosine color coding is proportional to the value of the parameter that we have computed at that vertex. For example, in Figure 5.5(d) each vertex is color coded to Gaussian curvature. We have chosen the fan disk model because it is the one that has a combination of flat and curved surfaces and is not too simple or too complex.



Fan Disk Model

Curvedness

$$K = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}}$$

Shape Index

$$\eta = \frac{1}{2} - \frac{1}{\pi} tan^{-1} \frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}$$

Gaussian Curvature

$$\kappa = \kappa_1 \kappa_2$$

(a) (b) (c) (d)

Figure 5.5: Curvature-based descriptors.

We would like to make the following conclusions from Figure 5.5. Curvedness proves to be a good descriptor that detects abrupt change in curvature. Curvedness is consistent at low resolutions but with bad triangulation it produces erroneous results. We attribute this result however to the curvature estimation method that assumes good and uniform triangulation. While on the other hand, we see how shape index is colorful indicating surface variation along the flat surface facing us in the diagram. The definition of shape index assumes uniform topology of meshes as in range images. That is why the shape index of a spherical cap and a spherical cup which look the same visually possess different shape indices. We see in Figure 5.5(d) that the Gaussian curvature clearly shows variation in curvature in each of the surface patches and no or very little variation in the flat surfaces.

We have also implemented a recent method for shape classification and description called Shape Distributions [Osada, 2002]. The approach represents shapes as histograms. Randomly sampled points on the surface of a triangle mesh is used to extract several features such as the centroidal profile, distance between two points, angle between three random points. These features are binned into a histogram. This histogram is used for object detection and classification. Results show similar shapes having similar feature histograms. This algorithm was implemented for shape searching and retrieval on the web. We have tested this algorithm with our automotive parts and have come to realize that several 1D features cannot represent completely the 3D information in an object. We show our implementation of Shape Distributions in Figure 5.6 and our experience in representing automotive components in Figure 5.7. We demonstrate the lack of uniqueness in description with the fan disk, disc brake and muffler models. These models have the same bounding box but we see that the disc brake and the fan disk though extremely different in shape have a similar histogram, while mufflers though similar in shape have noticeable amount of variation.

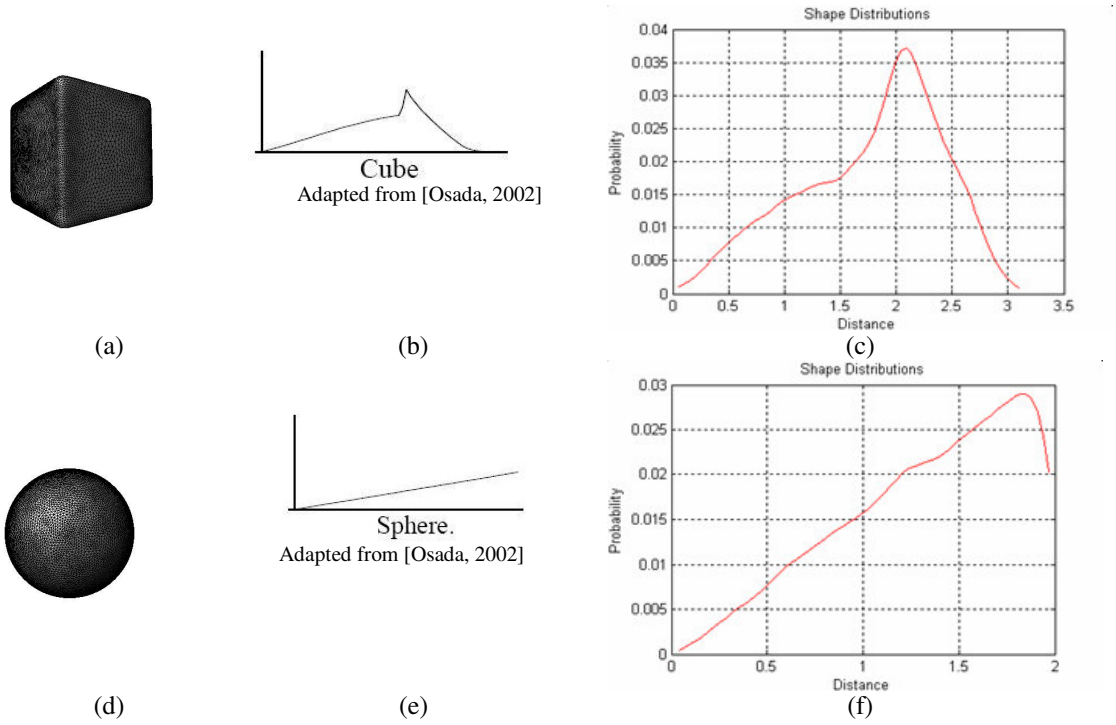


Figure 5.6: Implementation of Shape Distributions. (a) Wire frame model of a cube. (b) Shape Distribution result from the paper [Osada,2002] for a cube. (c) Result of our implementation on the cube. (d) Wire frame model of a sphere. (e) Shape Distribution result from the paper [Osada, 2002] for a sphere. (f) Result of our implementation on the sphere.

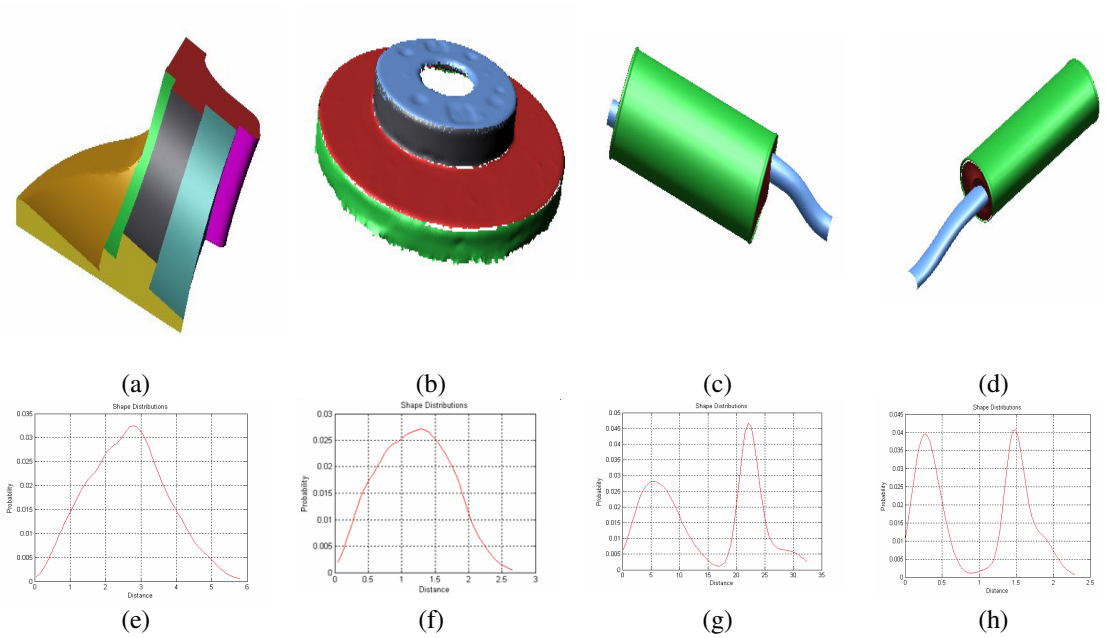


Figure 5.7: Shape Distributions and its uniqueness in description. (a) Model of a fandisk. (b) Model of a disc brake. (c) Model of a Toyota muffler. (d) Model of a Volvo muffler. (e) Shape Distribution of model in (a). (f) Shape Distribution of model in (b). (g) Shape Distribution of model in (c). (h) Shape Distribution of model in (d).

5.3 Results of our Informational Approach

In Chapter 4 we discussed our CVM algorithm that quantifies surface shape complexity. We compute curvature based on the method suggested by [Abidi, 1995] and measure boundary complexity as the Shannon's entropy of curvature on 2D contours. We have presented these results in [Page et al., 2003b]. We discuss some important results on X-ray and range images. We also analyze some limitations of using Shannon entropy and the need for a normalized information measure before discussing the results of our graph representation on automotive components.

5.3.1 Intensity and Range Images

In Figure 5.8(a) we show results on simple curves. We have made a few important assumptions with these curves. These curves are of the same resolution and are uniformly sampled. We have computed the Shannon entropy of the turning angle at each point on the boundary as the shape complexity measure (SCM). We note that SCM and CVM are similar measures but are not equivalent.SCM inspired the development of CVM, and CVM represents the evolution of SCM from lessons learned on scaling and resolution We would like to emphasize in these results on how shape information behaves with symmetry and how important the assumption on size and resolution turns out to be. We would also like to note that the shape information from the Shannon's measure cannot be compared if the two images are not at the same resolution and comparable size. Hence for the real data we have normalized the segmented region of interest for size and resolution and then computed the curvature-based measure on the normalized boundary contour. We would like to recall from Chapter 2 and note that our method falls under the boundary-based description methods. In Figure 5.8(b) we show an example with an X-ray image of a baggage. The bag contains a few objects that we have segmented manually. We take each of the segmented objects and then compute the shape information on each of these contours. Our measure categorizes complex objects and simple ones with satisfactory ease. Next, we show some results on range images in Figure 5.8(c).We believe that we will be able to distinguish between the man-made structures that have flat and nice edges like the building in Figure 5.8 (c) and natural vegetation that has rugged boundaries.

5.3.2 Surface Ruggedness

In terms of resolution we would like to present some results on synthetic DEMs (Digital Elevation Maps) of the same resolution. The Shannon's entropy of curvature gives a consistent ruggedness measure of the surface. But we still face inconsistency with resolution. We formulate our algorithm on the heuristic that the variation in the shape characteristics of surfaces is mathematically the variation of curvature. We define shape information as the entropy of the curvature density of the surface under consideration.



(a)



X-ray image SCM = 1.227 SCM = 2.3458 SCM=3.4050 SCM = 0.891

(b)



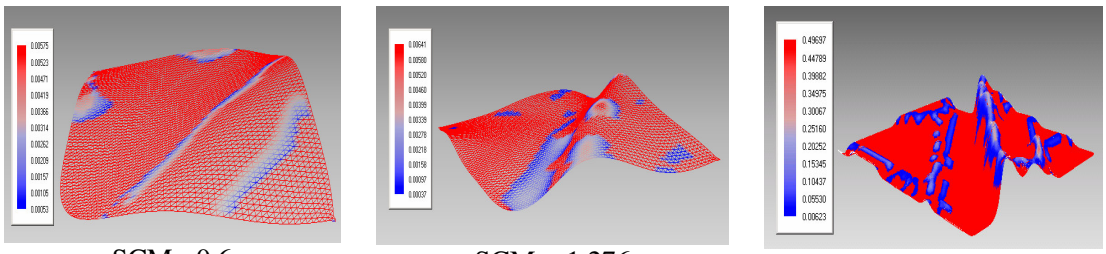
Range Image



SCM = 0.5 SCM = 1.7

(c)

Figure 5.8: Shape complexity measure– using Shannon's definition of information. (a) Results on simple curves. (b) Results on segmented objects from the X-ray image of a baggage. (c) Results on segmented contours from a range image.



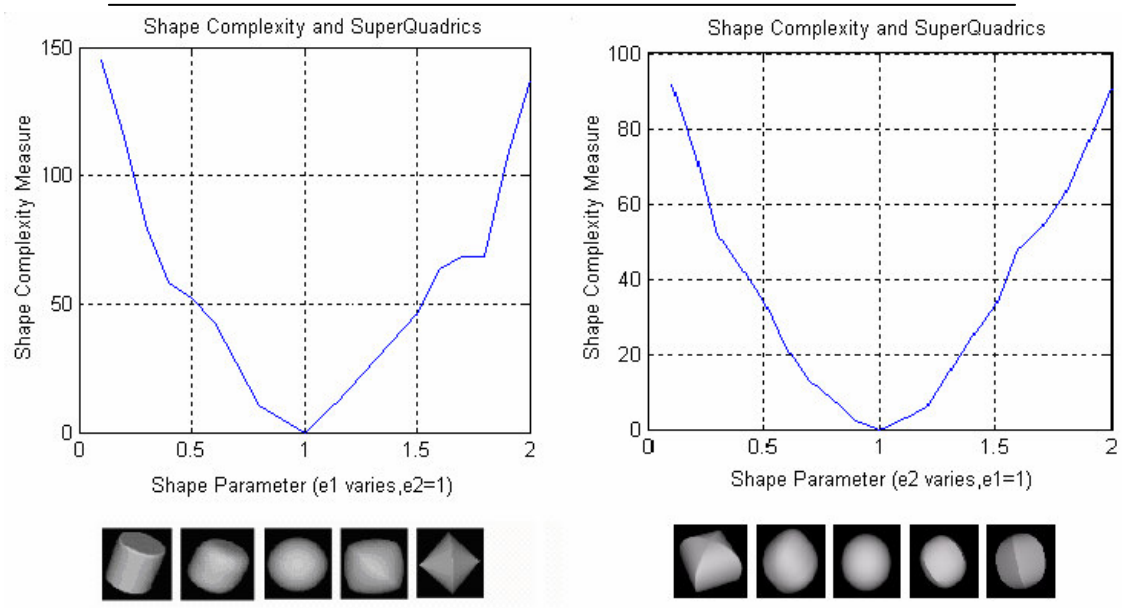
SCM =0.6 (a) SCM = 1.276 (b) SCM =2.2 (c)

Figure 5.9: Shape information and surface ruggedness. (a) Shape information measured on a DEM of a plain terrain. (c) Shape information on a plateau terrain. (d) Shape information on a mountainous terrain.

In Figure 5.9 we show three surfaces. Figure 5.9(a) can be considered to represent a plain while Figure 5.9(b) and 5.9(c) represent a plateau and a mountainous region, respectively. We have color coded each of these surfaces by the scale that we show in the picture. In agreement to our perceptual thinking we observe that the information that the total curvature conveys about each of these surfaces is well quantified by SCM.

5.3.3 3D Mesh Models

We see that the CVM algorithm behaves as expected in Figure 5.9 (a) – (c) but still not robust because of the assumption on resolution and sampling. We counter the problem of resolution as lack of information. We compensate with the information that for a contour its circumscribing circle of the same resolution, a plane of the same resolution for a surface and a circumscribing sphere for 3D models would have the least shape information. We use the circumscribing reference because it is easy to determine from the characteristics of the model and we might lose an important length dimension if we make use of an inscribed reference. The inscribed sphere, for instance, on a cylinder could turn out to be too small compared to the size of the cylinder and might not be a good reference. We measured shape complexity as the shape information distance between the two datasets and used the KL divergence measure (Equation 4.24) on super quadrics of varying shape factors. We present those results in Figure 5.10.



$$\left[\left(\frac{x}{a_1}\right)^{\frac{2}{e_1}}+\left(\frac{y}{a_2}\right)^{\frac{2}{e_2}}\right]^{\frac{e_2}{e_1}}+\left(\frac{z}{a_3}\right)^{\frac{2}{e_1}}=1$$

Figure 5.10: Shape information divergence from the sphere – Experimental results on super quadrics.

We observe that the results with the super quadrics are interesting. We chose super quadrics to perform our experiments because they provide us a scheme of slowly varying shapes (that can be controlled by a parameter) and smoothly varying curvature. Though we are unable to cluster or classify shapes based on a single number, we would like to point out the success with super quadrics. We however had to deal with another major problem. The asymptotic behavior of the divergence measure as the resolution tended to infinity. We end up with an impulse function on the sphere in the continuous case as a reference to a curvature density of another 3D model. Though the divergence measures are defined for the continuous random variables, our shape complexity measure becomes unstable with resolution approaching the continuous case. We also are not able to justify what it means for two completely different shapes having the same measure. The magnitude of the measure though can be understood as the number of bits that are required to describe the shape complexity of the object; it is not very convincing for the application of shape classification or clustering.

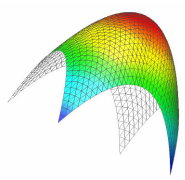
Our focus is to make the CVM independent of resolution and support it with theoretical consistency. We hence decided to change the reference from the sphere that represented the object with least information to the abstract, most complex object at that resolution. We have extended the Shannon's definition to a resolution normalized entropy form as shown in Equation 5.9.

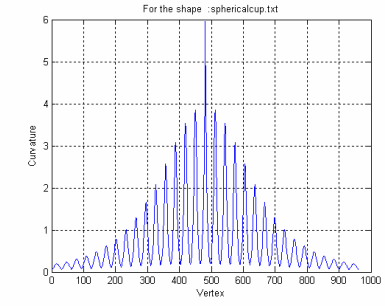
$$H_{\Delta}(X) = -\sum p(x) \log_R p(x) \tag{5.9}$$

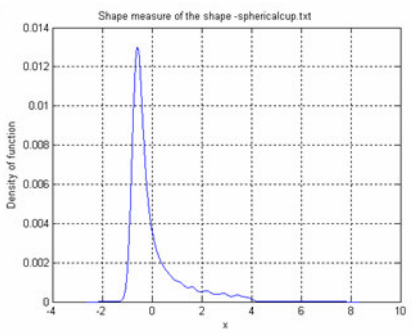
where R is the resolution of the datasets under consideration and $p(x)$ is the probability density of the curvature. We achieve two things with this measure of information. The measure is normalized. It has a minimum value of zero and a maximum value of one. The measure is in a logarithmic scale and is resolution independent. In Figure 5.11 we show how curvature on surfaces acts a descriptor with the spherical cup, saddle and the monkey saddle surfaces. We would like to point out that the broader the probability density function the higher the complexity. In Figure 5.12 we perform a multi-resolution experiment with our CVM shape signature. We resample the monkey saddle without obvious change in shape to show that our measure is now independent of resolution. N refers to the number of vertices in that surface and F is the number of faces.

We recollect the experience with the curvature-based descriptors. Curvature alone is not a sufficient feature for shape description because we have lost more than two dimensions of description in trying to represent 3D into a 1D function of curvature. However, now that we have verified the surface description capabilities of our measure we propose to describe objects that can be broken down into surface patches. We make use of curvedness for this task. We identify the sharp edges and creases and use it for segmentation of the triangle meshes.

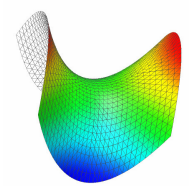
Surface	Curvature at each vertex	Density of curvature

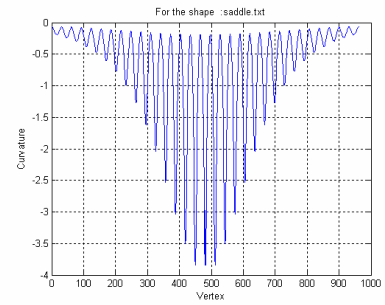


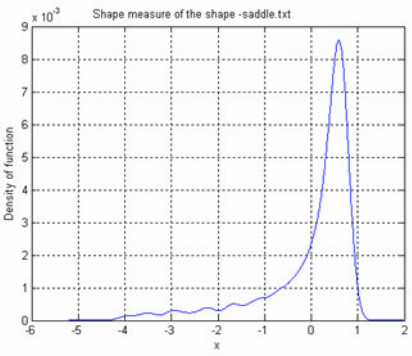




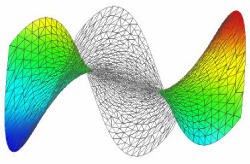
(a)

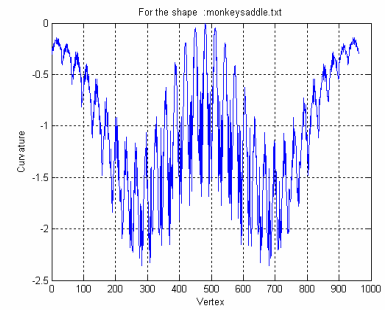


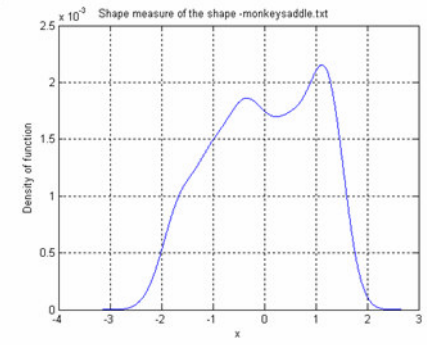




(b)







(c)

Figure 5.11: Surface description results - surface, curvature and density of curvature of (a) Spherical cap (b) Saddle (c) Monkey saddle

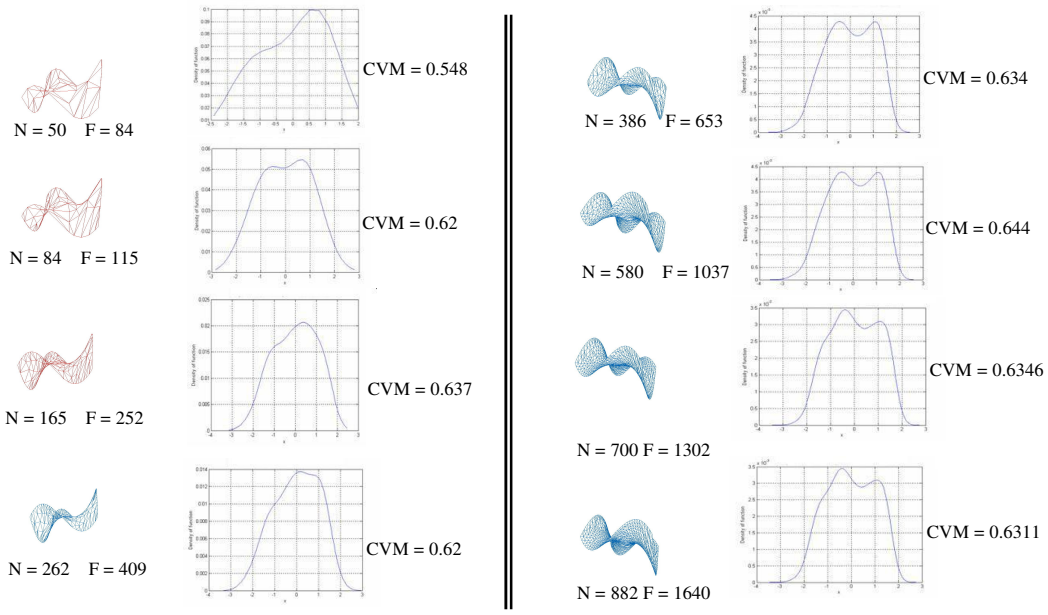


Figure 5.12: Multi resolution experiment on the monkey saddle – The surface, its curvature density and the measure of shape information.

We present the results of the shape description proposed in this thesis in Figures 5.13 and 5.14. We start with the description of the simple cube in Figure 5.13 (a). We show how the six faces of a cube are interconnected in the graph and since each of these faces is planar they convey no shape information. We would like to emphasize that all cuboids will also have the same description. This can be distinguished only with scale information along with the graph. With the fan disk example, we show the graph complexity that we will face with more and more complex parts. Figure 5.13(c), 5.14(a)–(c) are our experimental results on automotive components. Since our assumptions about man-made components go well with the informational signature that we have proposed our results are good. We show the result of applying our measure on the real scene that we acquired before we conclude this section in Figure 5.15. We show the scene, the segmented muffler from the scene and its description that looks very similar to the muffler results in Figure 5.13(c). We consider this as our first step towards object detection. However for the algorithm to be fully automated for object detection we need an implementation of partial graph matching. We also have to address occlusion problems and representation issues with more and more complex components.

This section concludes our experimental results for the CVM algorithm. We have presented the evolution of the algorithm in this chapter with results and analysis at each stage of the development. We now move to the final section of this thesis where we draw conclusions from these results and then discuss future directions for our research.

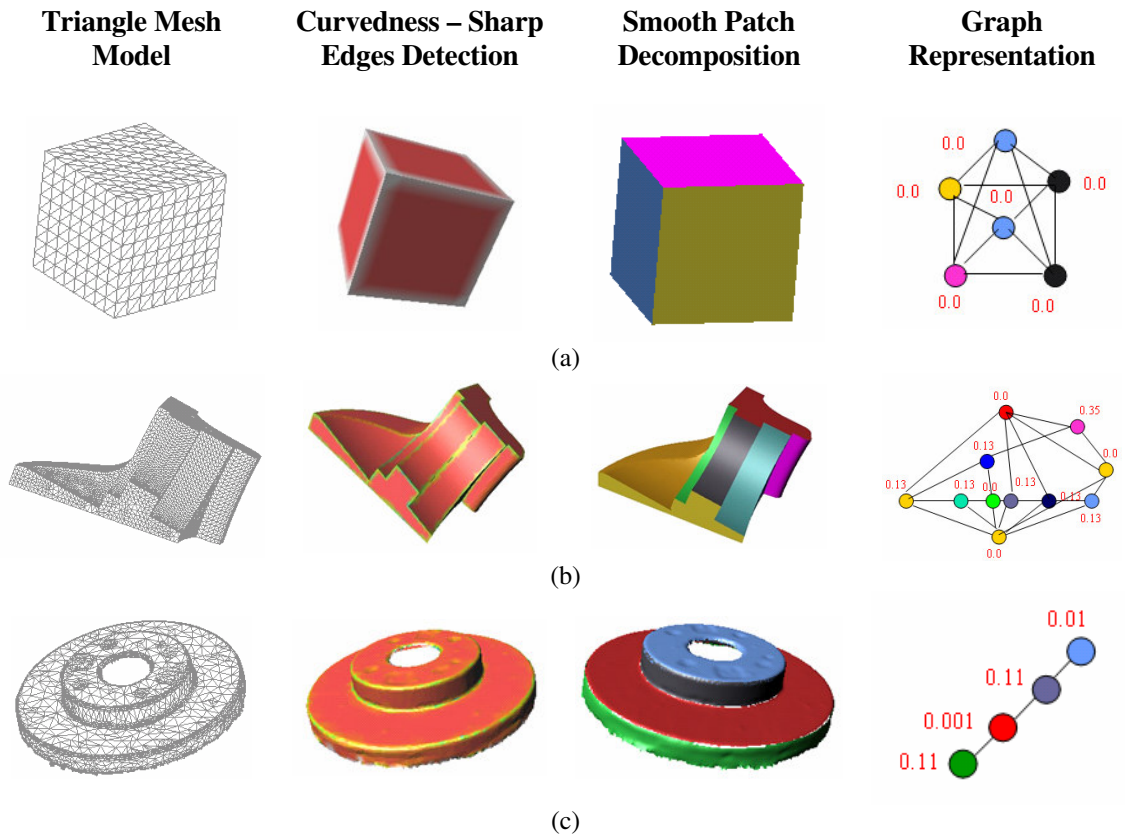


Figure 5.13: CVM graph results on simple mesh models: curvedness-based edge detection, smooth patch decomposition and graph representation. (a) Cube (b) Fan disk. (c) Disc Brake.

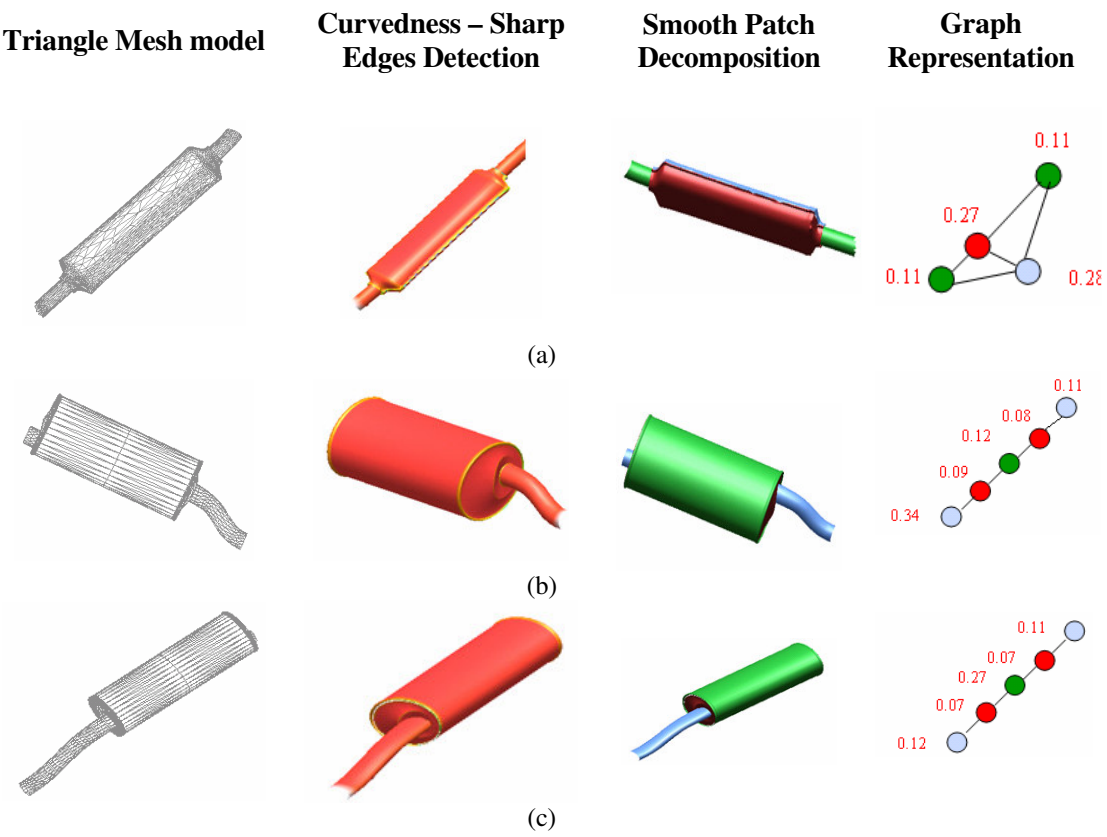
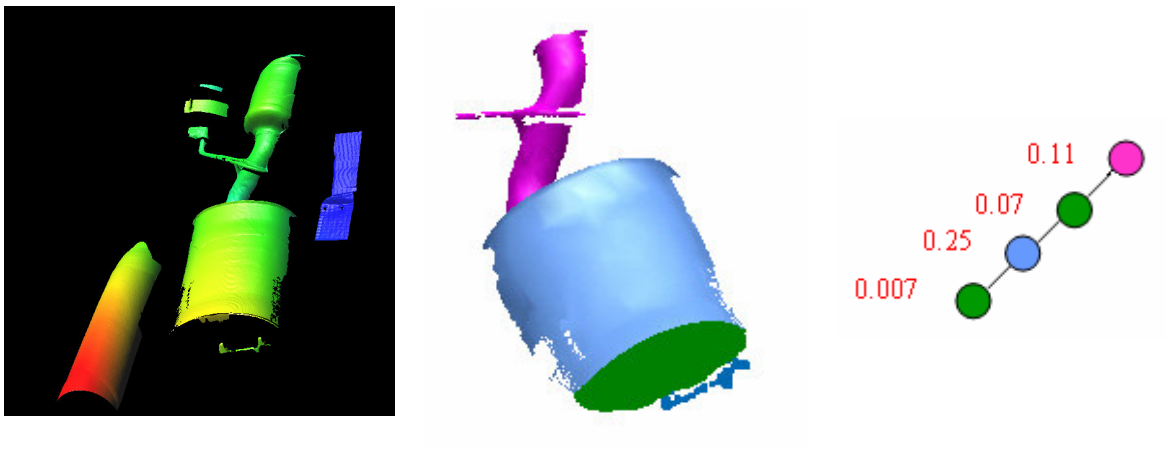


Figure 5.14: CVM graph results on automotive parts: curvedness-based edge detection, smooth patch decomposition and graph representation of (a) catalytic converter, (b) Volvo muffler and (c) Toyota muffler.



(a) (b) (c)

Figure 5.15: CVM graph results on an under vehicle scene. (a) Under vehicle scene. (b) Segmented muffler model. (c) Shape description of the muffler.

6 CONCLUSIONS

In this thesis, we have described a pipeline for real-time imaging and 3D modeling of automotive parts and a representation scheme that would simplify the task of threat detection for vehicle inspection. This research relies heavily on a heuristic, which we call CVM, which is based on curvature of surfaces and their contribution to describing surface complexity. In the previous chapters, we have reviewed research in the computer vision literature similar to our algorithm as a context for our contributions and presented the supporting theory along with experimental results. We have also discussed certain implementation issues of the algorithm. We now conclude with a brief summary of the contributions and a short discussion on future directions.

6.1 Contributions

Our research efforts were focused on the construction of a scanning mechanism that would be able to create 3D models of automotive components. We have used the sheet-of-light active range imaging technique for the data acquisition task and extended its capability to extract geometry of an automotive scene. We have outlined our design efforts towards data collection and followed it up with results on 3D model creation and analysis of objects. We have also presented experimental results of an information theory-based surface shape description algorithm on the laser scanned 3D models. The 3D data acquisition process to generate a dense point cloud of a particular view of an object, multiple view fusion and surface graph representation (comparable to CAD) of the models is our implementation of a pipeline that aids reverse engineering and inspection.

Based on our survey and implementation of the state-of-the art algorithms for curvature estimation on triangle meshes, we have presented a rigorous analysis on key methods. Our comparison sheds light on the errors in magnitude of curvature and also the effect of factors such as resolution and effectiveness in describing visual complexity.

We hence would like to summarize the quintessence of the thesis as the definition of CVM as the informational approach to shape description. We have used curvature estimates at each vertex to generate probability distribution curves. With these curves, we have formulated an information theoretic based on entropy to define surface shape

complexity. In the spirit of Claude Shannon's definition of information, this measure reflects the amount of shape information that an object surface possesses. Objects and scenes with nearly constant curvature contain relatively low values of shape information, while other objects and scenes with significant variation in curvature exhibit fairly large values. Though our idea of using curvature as a feature for description is not new, our attempt to quantify the perceptual complexity of a surface using information theory is. Since we are describing surfaces using our approach, occluded scenes such as the one that was obtained real-time can also be represented with a good degree of confidence towards object detection.

6.2 Directions for the Future

We feel that the process of creating 3D models has scope for improvement. Using our system design, it takes nearly six hours to scan, fuse and integrate multiple views into its complete 3D triangle mesh model. We did not consider optimizing views for minimizing scan time. We can approach view planning as a sensor placement problem for better efficiency. The solution to the problem will also enhance the under vehicle scene modeling. Our system design also has a lot of scope for improvement towards vehicle inspection. Instead of using a conveyer belt that houses the range sensor, it would be better to have a calibrated setup to control the relative motion. We would also like to have the scanning mechanism redesigned to be robot mountable to automate the scanning process.

In the typical context of aligning range scans of an object in order to create a complete model of that object we would like to point out the possibility of application of our algorithm to surface registration. Surface registration is a feature dependent process. More features improve registration. By features, here we mean unique geometric information. We believe that representing a range scan as a cluster of shape measures around a neighborhood would help us recover the rigid transformation from another view of the same object that has some common information (overlap). A multi-scale hierarchical informational approach should be a good start for this process.

Object recognition is an extremely difficult task with most current solutions limited to a very constrained and restricted problem domain. Although we do not claim contributions in terms of recognition as yet, we are encouraged by the results in this thesis that it might serve as a first step in the recognition pipeline. Shapiro and Stockman [Shapiro, 2001] suggest commonly used paradigms for object recognition where the method chosen depends heavily on the application. They discuss two paradigms that use part (region) relationships to move away from a geometric definition of an object to a more symbolic one. Our algorithm benefits the creation of such a symbolic graph representation from a mesh representation. We would like to perform rigorous experiments on partial graph matching for threat detection and model-based

object matching before we claim confidence and robustness. We also would like to experiment the effect of segmentation on our algorithm. More robust segmentation methods based on the minima rule and boundary refinement can substantially enhance the performance when our algorithm is used for the object detection and recognition.

6.3 Closing Remarks

In the first chapter of this document, we proposed to use the part-based human perception model for shape analysis. Though our implementation does not completely capture the perceptual power of the human mind or its coordination with the eye, the concepts presented in this thesis are a first step, though a very small one towards extending the state of the art in 3D computer vision.

BIBLIOGRAPHY

[Abidi, 1995] B. R. Abidi, "Automatic Sensor placement for volumetric object characterization," *PhD Thesis*, University of Tennessee, Knoxville,1995.

[Ankerst et al., 1999] M. Ankerst, G. Kastenmüller, H. P. Kriegel, T. Seidl, "3D Shape Histograms for Similarity Search and Classification in Spatial Databases," *Lecture Notes in Computer Science*, 1999, Volume 1651, Springer, pp. 207-226.

[Arman and Aggarwal, 1993] F. Arman and J. Aggarwal, "Model-based object recognition in dense-range images—A review," *ACM Computing Surveys*, 1993, Volume 25,Issue 1,pp. 5-43.

[Bernardini et al., 1999] F. Bernardini, C. L. Bajaj, J. Chen and D. R. Schikore, "Automatic reconstruction of 3D CAD models from digital scans," *International Journal on Computational Geometry and Applications*, 1999, Volume 9, Issue 4/5, pp. 327-369.

[Besl and Jain, 1986] P. J. Besl, and R. C. Jain, "Invariant surface characteristics for 3D object recognition in range images," *Journal of Computer Vision, Graphics and Image Processing*, 1986, Volume 33, pp. 33-80.

[Besl, 1988] P. J. Besl, "Surfaces in range image understanding," Springer-Verlag New York, Inc., New York, NY, 1988.

[Besl and McKay, 1992] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Transaction on Pattern Analysis and Machine Intelligence*,1992, Volume 14, Issue 2, pp. 239-256.

[Besl ,1995] P. Besl, "Triangles as a primary representation: Object Recognition in Computer Vision," *Lecture notes in Computer Science*, 1995, pp. 191-206.

[Biermann, 2001] H. Biermann, D. Kristjansson and D. Zorin, "Approximate Boolean Operations on Free-form Solids," *In the Proceedings of SIGGRAPH 2001*,Los Angeles, California, August 2001, pp. 185-194.

[Belongie et al., 2002] S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, April 2002, Volume 24, Issue 4, pp. 509-522.

[Beretti et al., 2000] S. Berretti, A. D. Bimbo and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing," *IEEE Transactions on Multimedia*, 2000, Volume 2, Issue 4, pp. 225-239.

[Bimbo and Pala, 1997] A. D. Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, Volume 19, Issue 2, pp. 121-132.

[Blum, 1967] H. Blum, "A transformation for extracting new descriptors of shape," *Models for the Perception of Speech and Visual Forms*, 1967, MIT Press, Cambridge, MA, pp. 362–380.

[Campbell and Flynn, 2001] R. J. Campbell and P. J. Flynn, "A Survey of Free-Form Object Representation and Recognition Techniques," *Journal on Computer Vision and Image Understanding*, February 2001, Volume 81, Issue 2, pp. 166-210.

[Cardone et al., 2003] A. Cardone, S. K. Gupta, and M. Karnik, "A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications," *ASME Journal of Computing and Information Science in Engineering*, 2003, Volume 3, Issue 2, pp. 109-118.

[Carmo, 1976] M. P. do Carmo, "Differential Geometry of Curves and Surfaces," *Prentice Hall Inc.*, Englewood Cliffs, NJ, 1976.

[Champleboux et al., 1992] G. Champleboux, S. Lavallee, P. Sautot and P. Cinquin, "Accurate calibration of cameras and range imaging sensor: the NPBS method," *In the Proceedings of the IEEE International Conference on Robotics and Automation*, California 1992, Volume 2, pp. 1552-1557.

[Chen and Schmitt, 1992] X. Chen and F. Schmitt, "Intrinsic surface properties from surface triangulation," *In the Proceedings of the European Conference on Computer Vision*, Italy, 1992, pp. 739-743.

[Chakrabarti et al, 2000] K. Chakrabarti, M. O. Binderberger, K. Porkaew and S. Mehrotra, "Similar shape retrieval in MARS," *In the Proceedings of IEEE International Conference on Multimedia and Expo*, New York, USA, 2000, , Volume 2, pp. 709-712.

[Chellappa and Bagdazian, 1984] R. Chellappa and R. Bagdazian, "Fourier coding of image boundaries, *IEEE Transactions on Pattern Analysis and Machine Intelligence,*1984 Volume 6,Issue 1,pp. 102-105.

[Chung, 1997] F. R. Chung, "Spectral Graph Theory," American Mathematical Society, 1997.

[Corney et al., 2002] J. Corney, H. Rea, J. Clark, J. Pritchard, M. Breaks and R. MacLeod, "Coarse Filters for Shape Matching," *IEEE Transactions on Computer Graphics and Applications*, 2002, Volume 22, Issue 3, pp. 65-74.

[Cybenko et al., 1997] G. Cybenko, A. Bhasin and K. Cohen, "Pattern Recognition of 3D CAD Objects," *Smart Engineering Systems Design*, 1997, Volume 1, pp.1-13.

[Cyr and Kimia, 2001] C. M. Cyr and B. B. Kimia, "3D object recognition using shape similarity-based aspect graph," *In the Proceedings of the International Conference on Computer Vision*, 2001, pp. 254-261.

[Davies, 1997] E. R. Davies, "Machine Vision: Theory, Algorithms, Practicalities," Academic Press, New York, 1997, pp. 171-191.

[Davis and Chen, 2001] J. Davis and X. Chen, "A Laser Range Scanner Designed for Minimum Calibration Complexity," *In the Proceedings of the Third International Conference on 3D Digital Imaging and modeling* , 2001.

[Desbrun et al., 1999] M. Desbrun, M.Meyer, P. SchrÄoder and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," *In Computer Graphics Proceedings (SIGGRAPH '99)*, 1999, pp. 317-324.

[Delingette, 1999] H. Delingette, "General object reconstruction based on simplex meshes," *International Journal of Computer Vision*, 1999, Volume 32, Issue 2, pp.111-146.

[Dorai, 1996] C. Dorai, "COSMOS: A framework for the representation and recognition of free form objects," *PhD Thesis*, Michigan State University, 1996.

[Duda and Hart, 1973] R. M Duda. and P. E Hart, "Pattern Classification and Scene Analysis," John Wiley and Sons, New York, 1973.

[Dudek and Tsotsos, 1997] G. Dudek and J. K Tsotsos, "Shape representation and recognition from multi-scale curvature," *Journal of Computer Vision and Image Understanding*, 1997, Volume 68, Issue 2, pp. 170-189.

[Elad et al., 2001] M. Elad, A. Tal and S. Ar, "Content-Based Retrieval of VRML Objects – an iterative and interactive approach," *Eurographics Multimedia Workshop*, 2001, pp.97-108.

[Elinson et al., 1997] A. Elinson., D. Nau, and W. C. Regli, "Feature-based Similarity Assessment of Solid Models," *In the Proceedings of 4th ACM/SIGGRAPH Symposium on Solid Modeling and Applications*,1997, Atlanta, pp. 297-310.

[Fix and Hodges, 1951] E. Fix, J. L. Hodges, "Discriminatory analysis, nonparametric discrimination consistency properties," *Technical Report 4*, Randolph Filed, Texas, US Air Force, 1951.

[Flynn and Jain, 1989] P. J. Flynn and A. K. Jain, "On reliable curvature estimation," *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1989, pp. 110-116.

[Freeman and Saghri, 1978] H. Freeman and A. Saghri, "Generalized chain codes for planar curves," *In the Proceedings of the Fourth International Joint Conference on Pattern Recognition*, Kyoto, Japan, November 1978, pp. 701–703.

[Fu, 1974] K. S. Fu, "Syntactic Methods in Pattern Recognition," Academic Press, New York, 1974.

[Gonzalez and Woods, 1992] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 1992, Addison-Wesley, Reading, MA, pp. 502-503.

[Gotsman et al., 2003] C.Gotsman, X. Gu, A.Sheffer, "Fundamentals of spherical parameterization for 3D meshes," *In the Proceedings of ACM SIGGRAPH*, 2003, pp.358-363.

[Gourley, 1998] C. S. Gourley, "Pattern vector based reduction of large multimodal data sets for fixed rate interactivity during visualization of multi-resolution models," *PhD thesis*, University of Tennessee, Knoxville, TN,1998.

[Goshtasby, 1985] A. Goshtasby, "Description and discrimination of planar shapes using shape matrices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1985, Volume 7, pp. 738-743.

[Groskey and Mehrotra, 1990] W. I. Groskey, R. Mehrotra, "Index-based object recognition in pictorial data management," *Journal of Computer Vision, Graphics and Image Processing*, 1990, Volume 52, pp. 416-436.

[Groskey et al., 1992] W. I. Groskey, P. Neo and R. Mehrotra, "A pictorial index mechanism for model-based matching," *Data Knowledge Engineering*, 1992, Volume 8, pp. 309-327.

[Guillaume et al., 2004] L. Guillaume, D. Florent and B. Atilla, "Curvature Tensor Based Triangle Mesh Segmentation with Boundary Rectification," *In the Proceedings of Computer Graphics International*, June 2004, Crete, Greece, pp. 10-17.

[Hetzel et al., 2001] G. Hetzel, B. Leibe , P. Levi , B. Schiele, "3D Object Recognition from Range Images using Local Feature Histograms," *In the Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2001,pp. 394-399.

[Henderson et al., 1993] M. R. Henderson., G. Srinath, R. Stage, K. Walker, and W. Regli, "Boundary Representation-based Feature Identification," *In Advances in Feature Based Manufacturing*, Elsevier-North Holland Publishers, Amsterdam,1993.

[Hilaga et al., 2001] M. Hilaga, Y. Shinagawa , T. Kohmura, and T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3d Shapes," *In the Proceedings of SIGGRAPH, ACM Press*, 2001, pp. 203-212.

[Hoppe et al., 1992] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Suetzle, "Surface reconstruction from unorganized points," *In the Proceedings of ACM SIGGRAPH*, 1992, pp. 71-78.

[Horn et al., 1998] B. Horn, H. Hilden and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of Optical Society of America (Optics and Image Science)*, 1998, Volume 5, Issue 7, pp. 1127-1135.

[Hu, 1962] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, 1962, Volume 8, pp.179-187.

[Huttenlocher, 1992] D. P. Huttenlocher, W. J. Rucklidge, "A multi-resolution technique for comparing images using the Hausdorff distance," *Technical Report, TR-92-1321, Department of Computer Science*, Cornell University, 1992.

[Jeannin, 2000] S. Jeannin (Editor), "MPEG-7 Visual part of experimentation model version 5.0," *ISO/IEC JTC1/SC29/WG11/N3321*, Nordwijkerhout, March, 2000.

[Johnson and Hebert, 1999] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, Volume 21, Issue 5, pp. 433-449.

[Jones et al., 1996] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *Journal of the American Statistical Association*, 1996, Volume 91,Issue 433, pp. 401-407.

[Joshi and Chang, 1988] S. Joshi and T. C Chang, "Graph-based Heuristics for Recognition of Machined Features from a 3D Solid Model," *Computer-Aided Design Journal*, 1988, Volume 20, Issue 2, pp. 58-66.

[Kazhdan et al., 2003] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," *In the Proceedings of ACM/Euro graphics Symposium on Geometry Processing*, 2003, pp. 167-175.

[Khotanzad and Hong, 1990] A. Khotanzad and Y. H. Hong, "Invariant Image Recognition by Zernike Moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, Volume 12, Issue 5, pp. 489-497.

[Kliot and Rivlin, 1998] M. Kliot and E. Rivlin, "Invariant-based shape retrieval in pictorial databases," *Journal of Computer Vision and Image Understanding*, 1998, Volume 71, Issue 2, pp. 182-197.

[Kortgen et al., 2003] M. Kortgen, G. J. Park, M. Novotni and R. Klein, "3D Shape Matching with 3D Shape Contexts," *In the proceedings of 7th Central European Seminar on Computer Graphics*, April 2003.

[Kresk et al., 1998] P. Krsek, C. Lukacs, and R. R. Martin, "Algorithms for computing curvatures from range data.," *In The Mathematics of Surfaces VIII, Information Geometers*, 1998, pp. 1-16.

[Kriegel et al., 2003] H. P. Kriegel, P. Kröger, Z. Mashael, M. Pfeifle, M. Pötke and S. Seidl, "Effective Similarity Search on Voxelized CAD Objects," *In the Proceedings of 8th International Conference on Database Systems for Advanced Applications*,2003, Kyoto, Japan, pp. 27-36.

[Leibowitz et al., 1999] N. Leibowitz, Z. Y. Fligelman , R. Nussinov , and H. J. Wolfson, "Multiple Structural Alignment and Core Detection by Geometric Hashing," *in Proceedings of the 7th International Conference on Intelligent Systems in Molecular Biology*, 1999, Heidelberg, Germany, pp. 169-177.

[Levoy et al., 2000] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," *In the Proceedings of the ACM SIGGRAPH*, 2000, pp. 131-144.

[Lin and Perry, 1982] C. Lin and M. J. Perry, "Shape description using surface triangulation," *In Proceedings of the IEEE Workshop on Computer Vision: Representation and Control*, 1982, pp. 38-43.

[Lu and Sajjanhar, 1999] G. J. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Journal of Multimedia Systems*, 1999, Volume 7, Issue 2, pp. 165-174.

[Mangan and Whitaker, 1999] A. P. Mangan and R. T. Whitaker, "Partitioning 3D surface meshes using watershed segmentation," *IEEE Transactions on Visualization and Computer Graphics*, 1999, Volume 5, Issue 4, pp. 308-321.

[McWherter et al., 2001] D. McWherter, M. Peabody, W. C Regli and A. Shoukofandeh, "Solid Model Databases: Techniques and Empirical Results," *ASME Journal of Computing and Information Science in Engineering*, 2001, Volume 1, Issue 4, pp. 300-310.

[Mehrotra and Gary, 1995] R. Mehrotra, J. E. Gary, "Similar-shape retrieval in shape data management," *IEEE Transactions on Computing*, 1995, Volume 28, Issue 9, pp. 57-62.

[Meyer, 2002] M.Meyer, M. Desbrun and P. Alliez, "Intrinsic Parameterizations of Surface Meshes," *Eurographics 2002*, Volume 21, Issue 2, 2002.

[Morse, 1994] B. S. Morse, "Computation of object cores from grey-level images," *Ph.D. Thesis*, University of North Carolina at Chapel Hill, 1994.

[Mukai et al., 2002] S. Mukai, S. Furukawa, M. Kuroda, "An Algorithm for Deciding Similarities of 3D Objects," *In the Proceedings of the ACM Symposium on Solid Modelling and Applications 2002*, Saarbrücken, Germany, June 2002.

[Oddo, 1992] L. A. Oddo, "Global shape entropy: A mathematically tractable approach to building extraction in aerial imagery," *In the Proceedings of the 20th SPIE AIPR Workshop*, 1992, Volume 1623, pp. 91-101.

[Ohbuchi et al., 2003] R. Ohbuchi, T. Minamitani and T. Takei, "Shape Similarity Search of 3D Models by using Enhanced Shape Functions," *In the Proceedings of Theory and Practice in Computer Graphics*, June 2003, Birmingham, U.K

[Osada et al., 2002] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions," *ACM Transactions on Graphics*, October 2002,Volume 21, Issue 4, pp. 807-832.

[Pap, 2002] E. Pap, "A handbook on measure theory," Elsevier North Holland Press, 2002.

[Page et al., 2001] D. L. Page, Y. Sun, A. F. Koschan, J. Paik and M. A. Abidi, "Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting," *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2001, Volume 1, pp. 162-167.

[Page et al., 2003a] D. L. Page, A. F. Koschan, Y. Sun, and M. A. Abidi, "Laser-based Imaging for Reverse Engineering," *Sensor Review, Special issue on Machine Vision and Laser Scanners*, July 2003, Volume 23, Issue 3, pp. 223-229.

[Page et al., 2003b] D. L. Page, A. F. Koschan, S. R. Sukumar, B. Abidi, and M. A. Abidi, "Shape analysis algorithm based on information theory," *In the Proceedings of the International Conference on Image Processing*, Barcelona, Spain, September 2003,Volume 1, pp. 229-232.

[Parui et al, 1986] S. Parui, E. Sarma and D. Majumder, "How to discriminate shapes using the shape vector," *Pattern Recognition Letters*, 1986, Volume 4, pp. 201-204.

[Parzen, 1962] E. Parzen , "On estimation of a probability density and model," *Annals of Mathematical Statistics*, 1962, pp. 1065-1076.

[Pavlidis, 1982] T. Pavlidis, "Algorithms for Graphics and Image Processing," Computer Science Press, Rockville, MD, 1982.

[Peura and Ivarinen, 1997] M. Peura and J. Ivarinen, "Efficiency of simple shape descriptors," *In the Proceedings of the Third International Workshop on Visual Form*, Capri, Italy, May, 1997, pp. 443-451.

[Reza,1961] F.M. Reza, "An Introduction to Information Theory," McGraw-Hill, 1961.

[Rosenblatt, 1956] M. Rosenblatt, "Remarks on some non-parametric estimates of a density function," *Annals of Mathematical Statistics*, 1956, pp. 642-669.

[Rucklidge, 1997] W. J. Rucklidge, "Efficient locating objects using Hausdorff distance," *International Journal of Computer Vision*, 1997, Volume 24, Issue 3, pp. 251-270.

[Safar et al., 2000] M. Safar, C. Shahabi and X. Sun, "Image retrieval by shape: a comparative study," *In the Proceedings of IEEE International Conference on Multimedia and Expo*, New York, USA, 2000, Volume 1, pp. 141-144.

[Shannon, 1948] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, 1948, Volume 27, pp. 379-423.

[Shum et al., 1996] H. Shum, M. Hebert, and K. Ikeuchi, "On 3D shape similarity," *In the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996, pp. 526-531.

[Silverman, 1986] B. W. Silverman, "Density Estimation for Statistics and Data Analysis," Chapman and Hall, London, 1986.

[Sonka et al., 1993] M. Sonka, V. Hlavac and R. Boyle, "Image Processing, Analysis and Machine Vision," Chapman & Hall, London, UK, NJ, 1993, pp. 193-242.

[Squire and Caelli, 2000] D. M. Squire, T. M. Caelli, "Invariance signature: characterizing contours by their departures from invariance," *Journal of Computer Vision and Image Understanding*, 2000, Volume 77, pp. 284-316.

[Surazhsky et al., 2003] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin , "A Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes," *In the Proceedings of International Conference on Robotics and Automation*, Taiwan, September 2003, pp. 1021-1026.

[Stankiewicz, 2002] B. J. Stankiewicz, "Models of the Perceptual System," *In the Encyclopedia of Cognitive Science*, Macmillan Press, 2002.

[Stokely and Wu, 1992] E. M. Stokely, S. Y. Wu, "Surface Parameterization and Curvature Measurement of

Arbitrary 3D Objects: Five Practical Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,1992, Volume 14,Issue 8, pp. 833-840.

[Suk and Bhandarkar, 1992] M. Suk and S. M. Bhandarkar, "Three-Dimensional Object Recognition from Range Images," Springer-Verlag, Tokyo,1992.

[Takatsuka et al., 1999] M. Takatsuka, A. W. Geoff, S. Venkatesh and T. M. Caelli, "Low cost interactive monocular range finder," *In the Proceedings of Computer Vision and Pattern Recognition*, Colorado, June 1999,Volume 1, pp. 1444-1451.

[Taza and Suen, 1989] A. Taza and C. Suen, "Discrimination of planar shapes using shape matrices," *IEEE Transactions on Systems, Man and Cybernetics*, 1989, Volume 19, pp. 1281-1289.

[Taubin and Cooper, 1991] G. Taubin and D. B. Cooper, "Recognition and positioning of rigid objects using algebraic moment invariants," *SPIE Conference on Geometric Methods in Computer Vision*, Volume 1570, University of Florida, Florida, USA 1991, pp. 175-186.

[Taubin and Cooper, 1992] G. Taubin, D. B. Cooper, "Object recognition based on moment," *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, MA, 1992, pp. 375-397.

[Taubin, 1995] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," *In the Proceedings of the Fifth International Conference on Computer Vision*, 1995, pp. 902-907.

[Teague, 1980] M. R. Teague, "Image analysis via the general theory of moments," *Journal of Optical Society of America*, 1980, Volume 70, Issue 8, pp. 920-930.

[Thompson et al., 1999] W. B. Thompson, J. C. Owen and H. J. Germain, "Feature-base reverse engineering of mechanical

parts," *IEEE Transactions on Robotics and Automation*, 1999, Volume 15, pp. 57-66.

[Turlach, 1996] B. A. Turlach, "Bandwidth Selection in Kernel Density Estimation: A Review," C.O.R.E and Institute de Statistique, University catholique de Louvain,Belgium,1996.

[Trucco and Verri, 1998] E. Trucco and A. Verri, "Introductory Techniques for 3D Computer Vision," Prentice Hall, 1998.

[Vranic and Saupe, 2001] D. V. Vranic and D. Saupe, "3D Shape Descriptor Based on 3D Fourier Transform," *In the Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services,* Hungary, September 2001, pp. 271-274.

[Vranic, 2003] D. V. Vranic, "An Improvement of Rotation Invariant 3D Shape Descriptor Based on Functions on Concentric Spheres," *In the Proceedings of the IEEE International Conference on Image Processing*, September 2003, Volume 3,Barcelona, Spain, pp. 757-760.

[Wu and Lin, 1996] Wu and Lin, "Information bound for bandwidth selection in Kernel Density Estimators," *Statistica Sinica* ,1996, Volume 6, pp. 129-145.

[Wand, 1995] M. P. Wand and M. C Jones, "Kernel Smoothing," Chapman and Hall, London, 1995.

[Yang et al., 1998] H. S. Yang, S. U. Lee and K. M. Lee, "Recognition of 2D object contours using starting-point-independent wavelet coefficient matching," *Journal of Visual Communication and Image Representation*,1998,Volume 9, Issue 2, pp. 171-181.

[Zhang and Hebert, 1999] D. Zhang and M. Hebert, "Harmonic Maps and Their Applications in Surface Matching," *In the Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[Zhang and Chen, 2001] C. Zhang, and T. Chen, "Efficient Feature Extraction for 2D/3D Objects in Mesh Representation," *In the Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, 2001.

[Zhang, 2002] D. S. Zhang, "Image retrieval based on shape," *Ph.D. Thesis*, Monash University, Australia, March, 2002.

[Zhang and Lu, 2002] D. S. Zhang and G. Lu, "Generic Fourier descriptor for shape-based image retrieval," *In the Proceedings of IEEE International Conference on Multimedia and Expo*, Volume 1, Lausanne, Switzerland, August 2002, pp. 425-428.

[Zhang and Lu, 2004] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, January 2004, Volume 37, Issue 1, pp. 1-19.

VITA

Sreenivas Rangan Sukumar was born in Chennai, India on the 16th of May, 1981. He graduated as the department topper of his college with a Bachelors Degree in Electronics and Communication Engineering from the University of Madras, India in 2002. Programming experience at the National Institute of Information Technology, India paved his way into Pentamedia Graphics Limited, India, where he was part of the research team that implemented a data compression framework for archiving multimedia on the web. It infused his interest in image processing and information theory to pursue his Masters Degree at the Imaging, Robotics and Intelligent Systems Lab at the University of Tennessee, Knoxville, U.S.A. He wishes to pursue his academic career with a PhD degree before he can be ready to contribute to the society. He spends his leisure time listening to Carnatic music and wishes for more time with the "veena" (a south Indian musical instrument).