12-2002

Surface Modeling and Analysis Using Range Images: Smoothing, Registration, Integration, and Segmentation

Yiyong Sun
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Yiyong Sun entitled "Surface Modeling and Analysis Using Range Images: Smoothing, Registration, Integration, and Segmentation." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Dr. Mongi A. Abidi, Major Professor

We have read this dissertation and recommend its acceptance:

Dr. Michael J. Roberts, Dr. Hairong Qi, Dr. Daniel B. Koch, Dr. Conrad Plaut

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

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

To the Graduate Council:

I am submitting herewith a dissertation written by Yiyong Sun entitled "Surface Modeling and Analysis Using Range Images: Smoothing, Registration, Integration, and Segmentation." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Dr. Mongi A. Abidi, Major Professor

We have read this dissertation
and recommend its acceptance:

Dr. Michael J. Roberts

Dr. Hairong Qi

Dr. Daniel B. Koch

Dr. Conrad Plaut

Accepted for the Council:

Dr. Anne Mayhew

Vice Provost
and Dean of Graduate Studies

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

Surface Modeling and Analysis Using Range Images: Smoothing, Registration, Integration, and Segmentation

A Dissertation
Presented for the
Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Yiyong Sun

December 2002

Dedication

This dissertation is dedicated to my wife, Li Yang, and to my parents, who have supported me and encouraged me to travel this far.

Acknowledgments

First and foremost, I extend my deepest appreciation to Dr. Mongi Abidi for his advice and leadership during my graduate program. His broad and in-depth knowledge of computer vision and image processing guided me to explore unknown research fields.

I am grateful to the other members on my dissertation committee: Dr. Michael Roberts, Dr. Daniel Koch, Dr. Hairong Qi, and Dr. Conrad Plaut for their support and dedication to this work. The inspirations for my research also came from the discussions in my classes with Dr. Ross Whitaker, Dr. Alexandre Freire, and Dr. Hamed Sari-Sarraf.

I would like to thank Dr. Joonki Paik, Dr. Andreas Koschan, Dr. Jeffery Price, and Dr. Christophe Dumont who provided help and suggestions on my research during my four years at the Imaging, Robotics, and Intelligent Systems Laboratory at The University of Tennessee. I acknowledge Mark Mitckes and Vicki Courtney-Smith who proofread all my publications. It has been my great pleasure to work with all the students in the laboratory. Among them, special thanks goes to David Page who scanned all the small parts using the IVP 3D profiling system and provided the data to my experiments.

Abstract

This dissertation presents a framework for 3D reconstruction and scene analysis, using a set of range images. The motivation for developing this framework came from the needs to reconstruct the surfaces of small mechanical parts in reverse engineering tasks, build a virtual environment of indoor and outdoor scenes, and understand 3D images.

The input of the framework is a set of range images of an object or a scene captured by range scanners. The output is a triangulated surface that can be segmented into meaningful parts. A textured surface can be reconstructed if color images are provided. The framework consists of surface smoothing, registration, integration, and segmentation.

Surface smoothing eliminates the noise present in raw measurements from range scanners. This research proposes area-decreasing flow that is theoretically identical to the mean curvature flow. Using area-decreasing flow, there is no need to estimate the curvature value and an optimal step size of the flow can be obtained. Crease edges and sharp corners are preserved by an adaptive scheme.

Surface registration aligns measurements from different viewpoints in a common coordinate system. This research proposes a new surface representation scheme named point fingerprint. Surfaces are registered by finding corresponding point pairs in an overlapping region based on fingerprint comparison.

Surface integration merges registered surface patches into a whole surface. This re-

search employs an implicit surface-based integration technique. The proposed algorithm can generate watertight models by space carving or filling the holes based on volumetric interpolation. Textures from different views are integrated inside a volumetric grid.

Surface segmentation is useful to decompose CAD models in reverse engineering tasks and help object recognition in a 3D scene. This research proposes a watershed-based surface mesh segmentation approach. The new algorithm accurately segments the plateaus by geodesic erosion using fast marching method.

The performance of the framework is presented using both synthetic and real world data from different range scanners. The dissertation concludes by summarizing the development of the framework and then suggests future research topics.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This introductory chapter first presents the motivation for the research and then introduces the proposed framework and its key components. The challenges in developing this framework are discussed, and the original contributions of this research are presented. Finally, the organization of the remainder of this dissertation is outlined at the end of the chapter to guide the reader to explore the details of the complete research.

1.1 Research Motivation

3D image modeling and analysis has important applications in many areas. In robotic vision, it can help a robot to function in a hazardous environment more precisely by providing an accurate 3D mapping of the surrounding scene. By seeing 3D geometry instead of 2D images, the robot is able to plan paths, grab objects, and avoid obstacles. In virtual reality, generating an accurate 3D environment is essential for a human walkthrough. For example, a real world 3D mapping looks more realistic than a synthetic scene in a driving simulator. In reverse engineering, 3D reconstruction is able to generate a CAD model from a real object. By using 3D FAX, a digitized model can be transferred to another location where a new model is quickly replicated. In image guided surgery, 3D models reconstructed from CTs and MRIs can help a doctor see the patient's anatomic structure, and 3D analysis such as segmentation will help make an accurate diagnosis.

3D models are reconstructed from a set of measurements in 3D space. Laser range sensing and stereo vision are two popular methods for 3D measurement. Although stereo vision devices are much cheaper than laser range scanners, they are limited by measurement accuracy and range. Stereo vision relies on finding the corresponding points on two spatially separated images and using triangulation to get the 3D measurement. This process is sensitive to illumination, and it is difficult to get dense measurements. The requirement of a large base line for long range measurements is impractical in many cases. This research uses laser range scanners to digitize the 3D space. Scanners based on time-of-flight are used to capture long range 3D information. Scanners based on laser triangulations are used to measure small objects.

The motivation of this research is to build a framework for surface modeling and analysis using the range data captured by range scanners. This dissertation makes contributions to solving several difficult problems in building such a framework.

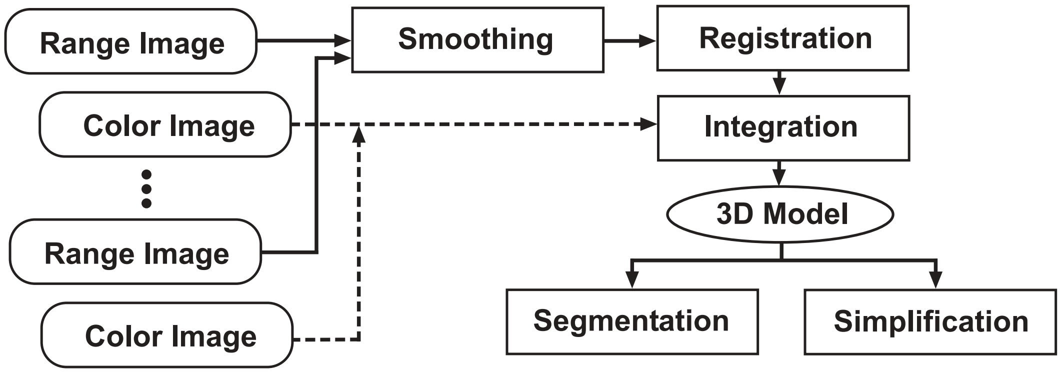


Figure 1.1: A framework for 3D reconstruction and analysis.

1.2 A Framework for 3D Reconstruction and Analysis

Several steps are involved in reconstructing a 3D model and further scene analysis using range images. The framework used in this research is illustrated in Fig. 1.1. After a set of range images are acquired by the range scanners, they are preprocessed to eliminate the sampling noise. In the second step, the range images are registered into a global coordinate system. Then the registered surfaces are integrated into a whole 3D model. When color images are available, they are also integrated to get a textured model. In 3D scene analysis, the model is segmented into different meaningful parts. Each step is essential for a complete framework of surface modeling and analysis. The framework, proposed in [108], segments the point cloud before performing integration. However, segmenting a surface is more robust than segmenting a cloud of points because a surface carries more geometric information. In the following text, a more detailed description of these steps and expected challenges is presented.

A range scanner acquires the distance between itself and an object. However, the range map is always contaminated with noise. The smoothing step can be considered to be a restoration process accomplished by local deformation in 3D space. Many techniques developed in 2D image processing cannot be applied to surface processing. Common problems in surface smoothing are loss of details and shrinkage. A successful surface smoothing algorithm should be able to efficiently suppress noise while making the deformation faithful to the original measurement and preserving the features on the surface.

A full 3D reconstruction requires scanning from different views. The surface measured from each scan is recorded in a local coordinate system centered at the scanner. In order to integrate different views, all the surfaces must be registered into a global coordinate system. Although the pose of the range scanner sometimes can be approximately obtained from the servo system of a carrier such as a robot, an automatic registration from the measurement is still crucial. A human is capable of understanding the environment without knowing the exact coordinate transformation when he is walking around. It is still not clear how to simulate this human ability on the computer. However, efforts in the field of computer vision are currently focusing on estimating the pose mathematically by matching the surfaces. The challenges in automatic registration are how to efficiently find correspondences from two partially overlapped surfaces and robustly deal with the noise and different surface sampling resolutions.

Integrating surfaces from different views is a fusion process that weaves a whole surface from a set of overlapping surface patches. The integration process should be robust to surface noise and registration error. It should also handle other modalities such as color images which generate a textured model. Usually, it is impossible to scan the whole object, so holes are left at regions where the laser cannot reach. These holes can be filled by space carving during the integration or by an automatic hole filling process after integration. Choosing space carving or leaving holes during the integration depends on the completeness of the data. After the surface is reconstructed, post-processing is necessary to further smooth the surface and remove the outliers. Sometimes geometric compression is needed for a real time rendering of the reconstructed surface.

The reconstructed surface may contain a number of parts. In many cases, a segmentation or decomposition of the surface gives a better interpretation of the surface. Surface segmentation is a higher level of surface processing which partitions the surface into different meaningful parts. Segmenting a surface is different from segmenting a 2D image in that the segmentation is guided by geometric variation instead of gray level variation and a surface is not defined on a regular grid. However, 2D image segmentation methods can be extended to surface segmentation. A good segmentation algorithm should be efficient, robust to surface noise, and close to human perception.

1.3 Contributions

The contributions of this research are summarized as follows.

Area-Decreasing Flow for Surface Smoothing [86, 89, 91]

This research proposes to smooth the surface using area-decreasing flow, which is theoretically identical to the mean curvature flow. A popular smoothing method has been local deformation of surfaces based on mean curvature flow. The problem with mean curvature flow is that the curvature value is difficult to estimate on a discrete surface and there is no way to explicitly compute the step size of the flow. Using the area-decreasing flow to smooth a surface, there is no need to estimate the curvature value and an optimal step size of the flow can be obtained. A rigidity term is included to make the deformation faithful to the original measurements. Smoothing is designed to be adaptive to preserve the crease edges and sharp corners. The proposed smoothing algorithm is able to efficiently smooth large triangle meshes and preserve the geometric details.

Point Fingerprint for Surface Matching [84, 90]

This research proposes a new surface representation scheme based on a signature of a 3D point, called point fingerprint. The fingerprint of a 3D point is defined as a set of 2D contours obtained by projecting the geodesic circles of different radii on the tangent plane. Surfaces are registered by finding corresponding point pairs in the overlapping region based on fingerprint comparison. Compared to other representation schemes in previous works, point fingerprint uses the accurate geodesic distance that is intrinsic to the surface. Fingerprint comparison is based on a set of 1D signal correlations, which are more efficient than 2D image correlations. The fingerprint representation functions like a one-to-one mapping and is able to carry other information such as color and curvature. This research also proposes an alternative method for geodesic distance computation on a triangle mesh and a fast approach based on irregularity of the point fingerprint to extract feature points from a surface. The point fingerprint can be applied to surface registration.

Implicit Surface-Based Integration [86, 88, 89]

This research extends previous work on implicit surface-based integration by combining the textures fusion. Within a volumetric grid, each surface patch generates a signed distance field. All signed distance fields are fused by weighted average. The whole surface is extracted using a polygonization algorithm. The method can generate watertight models using space carving. It can also work without space carving and fill the holes by volumetric interpolation. The proposed algorithm is employed to reconstruct 3D models using range images captured by various range scanners.

Geodesic Erosion in 3D Watershed Segmentation [87]

This research proposes a watershed-based surface mesh segmentation approach. Based on eigen analysis of surface normals inside a geodesic neighborhood, the approach robustly estimates the edge strength of each vertex on the surface mesh on which the watershed segmentation is applied. Compared with the previous works on watershed-based mesh segmentation, this research first applies the geodesic erosion on a triangle mesh to generate a lower complete image, which enables a more accurate segmentation of the plateaus. The proposed algorithm is successfully applied to segmenting surfaces reconstructed from real range images.

1.4 Document Organization

A thorough background of the related research is presented in Chapter 2. The background information includes the literature review in surface smoothing, registration, integration, and segmentation.

Chapter 3 first presents the area-decreasing flow and its relationship with the mean curvature flow. This chapter proceeds to describe the application of area-decreasing flow to calibrated range image smoothing and the adaptive scheme based on 2D image edge detection. The extension to arbitrary surface mesh smoothing and the adaptive scheme based on tensor voting are then presented.

Chapter 4 begins with an introduction of the exponential map. A new method to compute geodesic distance on a triangle mesh, the definition of point fingerprint, a scheme to select candidate points, and the point matching approach are then introduced. Application of point fingerprint to surface registration is discussed.

Chapter 5 introduces the surface integration approaches based on mesh zippering and implicit surface fusion. Automatic hole filling and post-processing by mean curvature flow are presented. Chapter 6 presents the watershed-based mesh segmentation approach which includes minima detection, geodesic erosion, finding the swiftest descending path, and region merging.

Literature reviews are included in each chapter. Experimental results for a variety of synthetic and real data are presented in Chapter 7, followed by conclusions and suggestions for future work in Chapter 8.

Chapter 2

Background

This work makes contributions in surface smoothing, registration, integration, and segmentation. The literature reviews of the related research are presented in Section 2.1 to 2.4.

2.1 Surface Smoothing

As explained in the Appendix A, raw data acquired by range scanners are always corrupted by noise. Smoothing the corrupted surfaces is essential to build a robust surface modeling framework. The technique also applies to surfaces digitized using other techniques such as stereo vision. Successful surface smoothing can greatly improve the visual appearance of a 3D object, and at the same time, can feed improved data to successive processes, such as matching, surface segmentation, and mesh simplification.

If a range value of a range image is considered equivalent to an intensity value of a two-dimensional (2D) image, 2D lowpass filtering can serve as a simple surface smoothing method. For an arbitrary surface, 2D image processing algorithms such as spatial or frequency domain lowpass filterings, however, cannot provide promising results because they do not take 3D parameters into consideration. For this reason, the surface smoothing problem has been tackled in the literature using different approaches, including regularization, surface fairing, and surface evolution using the level set method. The approach proposed in this research is based on regularization.

Regularization has been used for surface interpolation from sparse range data and for restoring noisy surfaces. Regularization performs smoothing operations by minimizing an energy function

$$f(\mathbf{x}) = g(\mathbf{x}) + \lambda h(\mathbf{x}) \quad \mathbf{x} \in \mathbf{R}^3 \tag{2.1}$$

that includes data compatibility term $g(\mathbf{x})$ and smoothing term $h(\mathbf{x})$. Minimization of $g(\mathbf{x})$ involves the compatibility of the solution to the original observation, and minimization of $h(\mathbf{x})$ incorporates prior knowledge. λ is called the regularization parameter which determines the weight of minimization between $g(\mathbf{x})$ and $h(\mathbf{x})$. The result of minimization is a trade-off between remaining close to the given observed data and avoiding a bumpy, coarse surface.

Most existing regularization works consider the surface as a height map. For a height map $z(x, y)$ the energy function can be written as

$$f(\mathbf{z}) = g(\mathbf{z}) + \lambda h(\mathbf{z}) \quad \mathbf{z} \in \mathbf{R}. \tag{2.2}$$

Although this assumption is quite limited, the idea of treating smoothing as a regularization process is the basis of the approach developed in this research. Different definitions of the smoothing term have been proposed in the literature. Blake and Zissermen [10] introduced the membrane and plate model. Using the membrane model, the smoothing term for a height map z is

$$h = \iint |\nabla z|^2 dxdy \tag{2.3}$$

The membrane has intrinsic resistance to creasing. In order to fit to crease discontinuities, a plate model should be used. The smoothing term for a plate model is the Quadratic Variation

$$h = \iint (z_{xx}^2 + 2z_{xy}^2 + z_{yy}^2) dxdy \tag{2.4}$$

or the Square Laplacian

$$h = \iint (z_{xx} + z_{yy})^2 dxdy \tag{2.5}$$

Stevenson and Delp [83] chose h as the sum square integral of the two principal curvatures of points on a surface in the form of a height map.

$$h = \iint (k_1^2 + k_2^2) dx dy \tag{2.6}$$

where k_1 and k_2 are principal curvatures. Eq. (2.6) came from the analysis of a plate of elastic material. Smoothly varying surface can be modeled as an ideal thin flexible plate of elastic material. The potential energy density of a thin plate is given by

$$A(\frac{k_1^2 + k_2^2}{2}) + Bk_1k_2 \tag{2.7}$$

where A and B are constants of the material. To simplify the equation, let $A = 1$ and $B = 0$, and Eq. (2.6) is obtained. The equation still models a valid thin plate. Various approximations to the stabilizing term were discussed to overcome the computational complexity. A two-stage reconstruction algorithm that would form an approximately invariant surface was proposed. The first stage forms a piecewise planar approximation to the surface that is invariant to the coordinate system. The piecewise planar surface is then used to construct an approximate parameterization of the reconstructed surface that can be used to make a valid approximation to the invariant functional. The minimization problem was solved using finite element methods. The experiment only showed the reconstruction of surfaces with simple shapes. The method involves computing the second order derivatives on the surface, which are sensitive to the surface noise.

Yi and Chelberg [107] proposed a simple first order smoothing term because first order models entail significantly less computational efforts than second order models. The volume between two surfaces normalized by the surface area was used as an invariant quantitative measure for comparing surface reconstruction results. This measure is invariant with respect to rotations and translations of coordinate systems. The algorithm for surface reconstruction consists of three steps: an initial reconstruction, partial derivative estimates from the initial reconstruction result, and then a second reconstruction which uses the estimated derivatives. The estimated derivatives are inserted as constants into an approximately invariant energy functional which make it convex. In order to estimate the derivatives, the input surface is reconstructed using a simple membrane regularization technique. The final reconstructed results depend on the reasonable derivative estimates. The stabilizing term is defined as

$$h = \iint (\sqrt{1 + z_x^2 + z_y^2} - 1) dx dy \tag{2.8}$$

where z_x and z_y are the partial derivatives of the height map. Then (2.8) is approximated by a convex function. The reason for using (2.8) was not clearly stated, but as shown by the analysis in Section 3.2.1, it is similar to minimizing the surface area of a height map. The 3D objects used in the experiments have simple shapes.

Because the above methods assume that the surface is a height map, they cannot be applied to smooth arbitrarily defined surfaces. Another category of surface smoothing methods, which is known as discrete surface fairing, directly process the surface mesh and are able to smooth arbitrary surfaces. Taubin [93] applied a weighted Laplacian smoothing on the surface mesh. Shrinkage is avoided by alternating the scale factors of opposite signs. Vollmer *et al.* [99] proposed another modified Laplacian smoothing. Ohtake *et al.* [60] showed that Laplacian smoothing tends to develop unnatural deformations. They applied the mean curvature flow to smooth the surface and the Laplacian flow to improve the mesh regularity. Page *et al.* [64] smooth and simplify a triangle mesh simultaneously using the surface normal voting approach.

Mean curvature flow originated from generating minimal surfaces in mathematics and material sciences [77] and has been applied to implicit surface smoothing by Whitaker [101], Zhao *et al.* [112], and Gomes and Faugeras [31]. The surface, represented by a zero crossing field in the volumetric grid, is deformed according to the magnitude of the mean curvature using the level set methods.

Mean curvature flow is mathematically equivalent to surface area minimization, as shown in Section 3.1. However, direct area-decreasing flow better fits the discrete surface smoothing than mean curvature flow because the curvatures are not well defined on a discrete surface [29, 62, 92] where the area can be explicitly computed.

In this research, a new regularized 3D image smoothing method based on locally adaptive minimization of the surface area is proposed. The approach is first applied to range image smoothing. Since range values are optimally estimated along the ray of measurement, there is no overlapping data problem. The method is then extended to surface fairing by processing arbitrary surfaces represented by the triangle mesh. The position of each vertex is adjusted along the surface normal to minimize the simplex area. The optimal, unique magnitude of the adjustment can be obtained. Crease edges are preserved by adaptive smoothing according to the edge strength of each vertex on a triangle mesh, which is computed by fusing the tensor voting and the orientation check of the normal vector field inside a geodesic window.

2.2 Surface Registration

Surface matching has two direct important applications in the area of computer vision. The first is three-dimensional (3D) image registration [33, 45, 94, 105, 110], which is also known as pose estimation. When 3D images are taken at different viewpoints and data fusion is necessary, the rigid transformation between each view needs to be computed. Horn [41] showed that, given three or more pairs of non-coplanar corresponding 3D points, the rigid transformation between the point pairs has a closed form solution. Thus the pose estimation problem becomes a surface point matching problem. The other application is object recognition [16, 20, 25, 82, 109]. The model library stores the surface features of each object, and the corresponding scene object is found by comparing those features.

Discrete surface matching is difficult because two surfaces may have self occlusions, different sampling resolutions, and are only partly overlapped, which makes statistically based features such as moments [67] difficult to apply. Because the local surface geometry characteristic is insensitive to the sampling resolution, previous works in this area tried to use this information in the encoded form for point matching, especially with different surface representation schemes which convert the problem of 3D point matching into 1D or 2D feature matching [20, 45, 82, 105, 110].

More specifically, previous works encoded the point's local surface geometry characteristics using either a contour on the surface or a 2D image of neighborhood near the encoded point. Stein and Medioni [82] used the notion of Splash to represent the normals along a geodesic circle of a center point, which is the local Gauss map, for 3D object recognition tasks. The geodesic circle is parameterized by the angle θ from 0 to 2π. On each point, a local orthogonal coordinate system is defined on the tangent plane with the normal $\mathbf{n}$ as the z axis. Each point on the geodesic circle is encoded by three angles θ, ϕ_θ and ψ_θ. For the point $\mathbf{p}$ on the geodesic circle, ϕ_θ and ψ_θ are defined as $\phi_\theta = \arccos(\mathbf{n} \cdot \mathbf{n}_\theta)$ and $\psi_\theta = \arccos(\mathbf{x} \cdot \mathbf{n}_{\theta(z=0)})$, where $\mathbf{x}$ is a vector along the x axis and $\mathbf{n}_\theta$ is the normal vector at $\mathbf{p}$. In the coordinate system created by θ, ϕ_θ and ψ_θ, the geodesic circle is encoded by another 3D curve and approximated by polygons. For every polygonal approximation, a 3D super segment is computed. The starting point of a 3D super segment is defined as the point with the maximal distance from the θ axis. The 3D super segment is also encoded using various attributes. All 3D super segments serve as keys into a database. Models containing similar codes as the Splashes appearing in the scene are extracted.

Chua and Jarvis [20] proposed a point feature named Point Signature (PS) for 3D object recognition. For a given point $\mathbf{p}$, a sphere of radius r centered at $\mathbf{p}$ is intersected with the surface and creates a 3D space curve C. The curve's orientation is defined by an orthonormal frame formed by a normal vector $\mathbf{n}_1$, a reference vector $\mathbf{n}_2$ and their cross product. $\mathbf{n}_1$ is defined as the unit normal vector of a plane P fitted through the space curve C. The distance between C and P can be plotted as a function, which serves as the feature of the center point and is called the Point Signature. Due to the simple representation, matching Point Signatures is efficient.

Johnson and Hebert [45] proposed the Spin Image for surface registration. To create a Spin Image, a local 2D basis is computed at an oriented point that is a 3D point with surface normal. The coordinates of the other points on the surface with respect to the basis are then used in a voting procedure to create the descriptive Spin Image for the point. With the surface point $\mathbf{p}$ and its normal $\mathbf{n}$, two coordinates α and β of a given point $\mathbf{x}$ are computed. α represents the distance from $\mathbf{x}$ to the tangent plane at $\mathbf{p}$ and β is the distance from $\mathbf{x}$ to $\mathbf{n}$. Next, a bin is determined by discretizing (α, β) and then used to match the corresponding points. The concept of Spin Images was developed from Geometric Hashing [50], but Spin Images use the image to describe the feature instead of performing lookup in a hash table. The term Spin Image came from the cylindrical symmetry of the representation. The Spin Image was also applied to 3D object recognition by Carmichael *et al.* [18].

Yamany and Farag [105] proposed a modified version of Spin Image, which is called the Surface Point Signature (SPS), where α and β are differently defined from the Spin Image. α represents the distance between the center point and every surface point, and β is the angle between the normal of the center point and the segment created by the center point and every surface point. SPS was applied to 3D image registration [105] and 3D object recognition [106].

Ashbrook and Fisher [4] proposed a histogram-based method to find corresponding facets between two surfaces represented by a triangle mesh. A Pairwise Geometric Histogram (PGH) is constructed for each triangle in a given mesh which describes its pairwise relationship with other surrounding triangles within a prespecified distance. For a pair of triangles t_i and t_j, the two parameters of the histogram are the angle between the triangle normals and the range of perpendicular distances from the plane, in which t_i lies, to all points on t_j. Corresponding triangles are found by comparing the histograms.

An application of harmonic maps to surface registration was reported by Zhang and Hebert [110]. A surface patch enclosed by the geodesic circle is mapped to a unit disk by the harmonic map. Curvature values of the vertices are textured onto the harmonic image to generate a Harmonic Shape Image (HSI). Corresponding points are found by comparing the HSIs. In implementation, boundary mapping needs to be assigned in order to compute the unit disk's interior mapping. A method of boundary mapping is proposed, which keeps the ratio of mapped angles. The method, however, makes the mapping no longer strictly one-to-one since the angle along the geodesic circle does not always change monotonically. A comparison between two HSIs is conducted by cross correlation. This method was also applied to 3D object recognition [109].

Rather than using the local geometry, some representation schemes use the surface's global properties. Hebert and Ikeuchi [35] created a Spherical Attribute Image (SAI) by mapping all the surface points to a sphere. A regular mesh is first created by deforming an initial geodesic dome onto the object surface. Each node on the geodesic dome stores the corresponding surface point's curvature value. The surfaces are matched by aligning the curvature map on the geodesic domes. SAI only works for objects with spherical topology. Dorai and Jain [25] proposed the Curvedness Orientation Shape Map On Sphere (COSMOS) by combining the local and global geometric information. An object is characterized by a set of maximally sized surface patches of constant shape index and their orientation-dependent mapping onto the unit sphere. Constant Shape Maximal Patch (CSMP) is defined as a group of points with the same shape index. The average normal of each CSMP is used to generate the Gauss map. The surface area and the connectivity list of CSMPs are recorded. All these descriptors are used for object recognition. Generally, the representations using global geometry are less flexible in dealing with arbitrary topology and occlusion. A detailed survey of free-form object representation can be found in [15].

In this research, a new surface representation scheme called 3D point fingerprint [84] and its application to surface registration are proposed. The proposed point fingerprint is a set of 2D contours that are the projections of geodesic circles onto the tangent plane. Point fingerprint is so named because it looks like human fingerprints and can be used as a discriminating feature for the surface point. It can carry more information than the existing schemes using only one contour or 2D histogram. The computation

Table 2.1: Comparison of different surface representation schemes

Features	Mapping	Type	Measure	Carries Information
Splash [82]	Gaussian map of surface normals along a geodesic circle	Local	Geodesic	No
SAI [35]	Spherical mapping of surface from deformation	Global	Euclidean	Yes
COSMOS [25]	Spherical mapping of orientation of CSMPs	Global	Euclidean	No
Spin-Image [45]	2D histogram of distance to tangent plane and surface normal	Local	Euclidean	No
PS [20]	Signed distance to a plane of a contour	Local	Euclidean	No
PGH [4]	2D histogram of angle and distances between triangles	Local	Euclidean	No
SPS [105]	2D histogram of distance and angle with surface normal	Local	Euclidean	No
HSI [110]	Harmonic map of underlying surface onto a unit disk	Local	Geodesic	Yes
Fingerprint [84], the proposed	Projected contours of geodesic circles on the tangent plane	Local	Geodesic	Yes

is cheaper than 2D image representation-based schemes. Since the projection on the tangent plane is not a one-to-one mapping, some projections of geodesic circles are possibly overlapped. Each geodesic circle projection can, however, be traced back to the corresponding geodesic circle, and therefore, the method can have the same advantage of the one-to-one mapping. The above mentioned representation schemes are summarized in Table. 2.1 in chronological order, identifying whether the mapping is using local geometry, computing geodesic distance and being able to carry features. This table is an updated version of the summary published in [109].

2.3 Surface Integration

Surface reconstruction is a step that extracts a surface from 3D measurements. The measurements are from multiple views of range images that have been registered together. For a calibrated range scanner, each range image can be converted to a surface patch. The registered surface patches may overlap in the 3D space and may not cover the whole object due to incomplete scans. The surface reconstruction algorithm has to deal with the partly redundant and partly incomplete data, surface sampling noise, and registration error.

Previous works in surface reconstruction from multi-view range images can be classified into three groups: reconstruction from unorganized points [2, 3, 6, 13, 28, 40], mesh integration [66, 74, 75, 81, 85, 96], and implicit surface integration [21, 36, 101].

Reconstruction from Unorganized Points

Surface Reconstruction algorithms using unorganized points are flexible because only a cloud of points are needed. These approaches estimate the neighborhood relations between points. The Euclidean distance between measurements is used as the basis for establishing the adjacency on the surface. Neighborhood relations enable approximation of surface topology and continuity. Boissonnat [13] describes a method for Delaunay triangulation of a set of points in 3D space. Edelsbrunner and Mucke [28] proposed the α-shape which is a parameterized construction that associates a polyhedral shape with an organized set of points. For $\alpha = \infty$, the α-shape is identical to the convex hull. As α decreases, the α-shape shrinks by gradually developing cavities. α-shapes were applied for surface reconstruction.

Hoppe *et al.* [40] developed an algorithm to reconstruct surfaces from unorganized points using the concept of the implicit surface. A signed distance field is estimated from the point cloud. The isosurface is then extracted using a variation of the marching cubes algorithm [1]. The key ingredient in defining the signed distance function is to associate a tangent plane with each of the data points. These tangent planes serve as local linear approximations to the surface and are used to define the signed distance function to the surface.

Amenta *et al.* [2] proposed an algorithm that reconstructs surfaces with provable guarantees. The output is guaranteed to be topologically correct and convergent to the original surface as the sampling density increases. The algorithm is based on the 3D

Voronoi diagram and Delaunay triangulation. It produces a set of triangles, called crust of the sample points. All vertices of crust triangles are sample points.

Amenta *et al.* introduced the power crust in [3]. The power crust is a construction which takes a sample of points from the surface of a three-dimensional object and produces a surface mesh and an approximate medial axis. The approach is to first approximate the medial axis transform (MAT) of the object, and then use an inverse transform to produce the surface representation from the MAT.

Bernardini *et al.* [6] developed a system named ball-pivoting based on α-shapes while avoiding the computation of the Voronoi diagram. The ball-pivoting algorithm computes a triangle mesh interpolating a given point cloud. Three points form a triangle if a ball of a user-specified radius ρ touches them without containing any other point. Starting with a seed triangle, the ball pivots around an edge until it touches another point, forming another triangle. The process continues until all reachable edges have been tried, and then starts from another seed triangle, until all points have been considered. The process can then be repeated with a ball of larger radius to handle uneven sampling densities.

Zhao *et al.* [112] introduced a minimal surface like model and its variational and partial differential equation formulation for surface reconstruction from an unorganized data set. The data set can include points, curves, and surface patches. In the formulation, only distance to the data set is used as the input. To find the final shape, they continuously deform an initial surface following the gradient flow of an energy functional. An offset of the distance function to the data set is used as the initial surface. The level set method [77] is used in the numerical computation.

Although the algorithms reconstructing surfaces from unorganized points are able to compute a surface model using only point information, they discard useful information such as surface normal and reliability estimates. These algorithms make three assumptions [81]. First, they assume that the K nearest surface neighbors of a point can be estimated by finding its K nearest 3D neighbors. Second, these methods assume that the density of data points is reasonably uniform over the surface to be modeled. Finally, they assume that the points are measured with the same accuracy. These assumptions are too restrictive for integrating multiple range images.

Mesh Integration

Surface reconstruction methods by merging the triangle meshes assume a surface mesh can be easily obtained from a range image. This is true for most range scanners. Range scanners usually acquire range images on a rectangular grid, where the triangulation process is straightforward.

Turk and Levoy [96] integrate the surface by zippering triangle meshes from different views. Merging begins by converting two meshes that may have considerable overlap into a pair of meshes that just barely overlap along portions of their boundaries. This is done by simultaneously eating back the boundaries of each mesh that lie directly on top of the other mesh. Next, the meshes are zippered together. The triangles of one mesh are clipped to the boundary of the other mesh, and the vertices on the boundary are shared. Once all the meshes have been combined, the final position of a vertex is found by taking an average of nearby positions from each of the original range images.

Rutishauser *et al.* [74] described an algorithm to merge triangle meshes. They consider the accuracy of the 3D position as defined by an anisotropic Gaussian error model. Next they do a mutual approximation of the two meshes in an area where they overlap. Finally a retriangulation is completed to merge the two meshes.

Soucy and Laurendeau [81] introduced an approach for measuring the level of redundancy in a set of range images through the use of the Venn diagram. This diagram may also be viewed as a multi-view connectivity graph. The Venn diagram allows a piecewise estimation of the integrated model by a set of local surface triangulations modeling its canonical subsets. The set of non-redundant triangulations is then connected in a final step to yield a global integrated triangulation.

Pito [66] described a method of mesh integration based on co-measurements. All of the triangles from either mesh which have sampled the same surface patch of the object are identified as co-measurements, and only the most confidently acquired one is kept. Co-measurement identification is based on the position and orientation of the range scanner when each triangle was sampled. Once the redundant triangles have been removed, what remains is a patchwork of unconnected non-intersecting meshes which cover the sampled areas of the object. Neighborhood relationships established between the edge points of each patch are used to seam them together.

Mesh integration-based algorithms use the information of surface normals and measurement confidence. However, in order to detect the overlapping regions, back projection of 3D points to the image plane is needed. These algorithms are not robust to registration error and may fail in areas of high curvature.

Implicit Surface Integration

Surface reconstruction by implicit surface integration is a volume-based approach. A simple but coarse volume-based method to generate 3D models from range images is a binary reconstruction based on space carving. The 3D space is partitioned into small cubes, which are called voxels. A voxel is considered either empty or occupied. The volume between the measured 3D points and the scanner is regarded as empty space and carved out. The skin of the remaining volume after carving from different views is the final result. Pulli *et al.* [69] used the octree to improve the space efficiency. The binary reconstruction results in a blocky-looking surface. Thus the continuous volumetric functions were introduced.

Whitaker [101] let the volumetric function be the surface likelihood. The strategy is to use a maximum a posteriori approach: find a surface which is the most likely, given the data and some prior knowledge about the application domain.

Curless and Levoy [21] used the cumulative weighted signed distance function. Working with one range image at a time, they first scan-convert it to an implicit surface based on the distance from the voxel to the measured surface, then combine this with the data already acquired using a simple additive scheme. The surface is extracted from the zero crossing of the signed distance field using the marching cube algorithm. This scheme is able to generate a watertight surface. Run length encoding and resampling of the range images are implemented to achieve space and time efficiency.

Instead of computing the signed distance along the measurement direction [21], Hilton *et al.* [36] constructs the implicit surface using the true distance between the voxel and the surface. Because this approach is not based on back projection, the holes cannot be filled. Three techniques for nearest point computation are discussed. The integrated surface is extracted using the marching triangle algorithm [37].

Implicit surface-based algorithms are robust to surface sampling noise and registration error. The quality of the extracted surface mesh is guaranteed. To have an accurate reconstruction, a high resolution of the volumetric grid takes a large amount of memory.

A variation of the mesh integration-based algorithm is implemented in this research [85], which is similar to [66]. The implicit surface-based algorithm used in this research is a combination of [21] and [36], which computes the true signed distance field and is able to generate watertight surfaces [89]. Beside the geometric fusion, texture integration is introduced into both methods.

2.4 Surface Segmentation

Subdividing an image into its constituent parts or objects, segmentation is usually the first step in image analysis. Autonomous segmentation is one of the most difficult tasks in image processing [32].

In this research, the interest is in segmenting arbitrary surfaces represented by a triangle mesh. The triangle mesh is one of the most popular formats in representing complex 3D objects. Most existing surface reconstruction algorithms [5, 6, 21, 36, 112] export the result as a triangle mesh. Other polygon meshes can also be easily translated into a triangle mesh. Triangle mesh segmentation can be used as a general tool for 3D surface segmentation tasks.

Segmenting a 3D object can partition it into constituent parts, which may have important applications in CAD and computer graphics. Segmentation of a 3D scene can extract 3D objects from the scene, which may help scene understanding and 3D object recognition.

Most work on 3D segmentation has been focused on segmenting range images [8, 39, 44, 111] obtained by different types of range scanners. 3D coordinates can be recovered from the range image if the calibration of the scanner is known. However, a single range image does not completely represent the scanned object due to occlusions. For this reason, the range image is often called 2.5D image. Although this research is not interested in segmenting range images, the proposed method can also be applied to range image segmentation because a triangulated surface can be easily obtained from a single range image.

Yu *et al.* [108] segmented a cloud of 3D points using a normalized cut algorithm and then reconstructed the surfaces from each segmented cluster of points. In this research, the surface is first reconstructed and then the segmentation is applied. The segmentation is easier and more reliable when it is applied on a surface instead of a cloud of unorganized points.

Woo *et al.* [102] used a volume-based approach to segment a point cloud. Initial grids containing the points are generated based on a bounding box. When the standard deviation of point normals in a cell is larger than a threshold, the cell is subdivided. When the size of 3D cells becomes very small, these cells correspond to the edges. Removal of these cells separates the point cloud into several regions by leaving gaps between different regions.

Huang and Menq [42] segmented a point cloud in three steps. In the first step, a mesh surface domain is reconstructed to establish explicit topological relation among the discrete points. In the second step, curvature-based border detection is applied on the irregular mesh to extract both sharp borders with tangent discontinuity and smooth borders with curvature discontinuity. Finally, the mesh patches separated by the extracted borders are grouped together.

Li *et al.* [53] presented a mesh segmentation approach based on space sweeping. The first step is the skeletonization of the 3D object. Skeletal edges are extracted by mesh simplification. A skeleton tree is obtained to determine a traversal order by adding virtual edges to connect disjointed skeletal edges. In the course of sweeping plane movement along skeletal edges, the geometric and topological functions are computed and analyzed. When a consecutive pair of critical points is found, the part of the polygon mesh that is swept is extracted as a component.

Wu and Levine [103] introduced simulated electrical charge distributions for surface mesh segmentation based on the physical fact that electrical charge on the surface of a conductor tends to accumulate at a sharp convexity and vanish at a sharp concavity. From an initial triangle which is the local minimum of the charge density, the algorithm proceeds to the neighbor with the lowest charge density. The tracing process detects the object part boundaries denoted by the sharp concavity. This approach can be regarded as edge-based segmentation. Similar to other 2D edge-based segmentation methods, the segmentation fails when the boundary is not connected due to noise.

The mesh segmentation problem was first formally defined by Mangan and Whitaker [55]. In their work, total curvature values at each vertex are estimated, and the surface is segmented into patches based on the 3D watershed, where each patch has a relatively consistent curvature throughout. This approach can be classified as region-based segmentation. Although the watershed segmentation algorithm in Mangan's work is the first extension from a 2D image to a 3D surface, the extension is primitive. Some key features that have been popular in 2D watershed-based segmentation, such as solving the plateau problem, are ignored.

The watershed segmentation method can be classified as a region-based segmentation approach [72], which is more robust than an edge-based approach. The watershed segmentation method is also chosen in this work. The fast marching watershed algorithm proposed in this research solves the plateau problem, which has been ignored in Mangan's work, and is a more complete extension of watershed-based segmentation from 2D image to 3D surface.

Next a brief review of watershed-based segmentation algorithms for 2D images is given. A detailed review can be found in [12, 72]. Watersheds for image segmentation are described in the classic work of Serra [76] on mathematical morphology. 2D gray-scale image can be segmented according to the watersheds of the image, where the watersheds are the domains of attraction of rain falling over the region. A typical watershed-based segmentation has three steps. First, regional minima are detected and uniquely labeled. Then other pixels get the labels of regional minima that the swiftest descending paths (SDP) lead to. The final segmentation is obtained by region merging.

The plateau problem means that the SDP is undefined for pixels on a plateau. The usual solution is transforming the image to a lower complete image [12] so that no plateau exists and the SDP is defined everywhere except at the regional minima. Lower completion is accomplished by raising the plateau according to the geodesic distance to the lower boundary of the plateau. Vincent and Soille [98] used a first-in-first-out (FIFO) queue-based breadth-first algorithm [79] to propagate the label from the plateau boundary to the inside. However, the breadth-first algorithm can only find the shortest path on the unweighted graphs such as a 2D regular grid for an image. For the weighted graphs such as a 3D triangle mesh, a priority queue-based algorithm is necessary. The geodesic distance computed by Dijkstra's algorithm [79], which is priority queue-based, is not accurate because the shortest path is restricted on triangle edges. The proposed fast marching watershed algorithm computes a more accurate geodesic distance using Sethian's fast marching method [77] and precisely partitions the plateaus.

Rettmann *et al.* [70] proposed a mesh segmentation approach similar to this work. It is applied to extract regions of cortical surface that surround sulci. Gyral and sulcal regions are initially labeled based on its Euclidean distance to a shrink wrap surface that is a deformable surface fitted to the original cortical surface. The height map is obtained by computing the geodesic distance from sulcal regions to gyral regions. The watershed segmentation for an accurate extraction of sulcal regions is an extension of [98] to the 3D surface mesh. The fast marching method is used to segment the plateaus. However, when the geodesic distance is computed in a plateau, the fast marching is applied as many times as the number of the surrounding regions. The method proposed in this research is more efficient because there is only one marching process on the plateaus.

Chapter 3

Surface Smoothing Based on Area-Decreasing Flow

This chapter proposes a new surface smoothing method based on area-decreasing flow, which can be used for preprocessing raw range data or postprocessing reconstructed surfaces. Although surface area minimization is mathematically equivalent to the mean curvature flow, area-decreasing flow is far more efficient for smoothing a discrete surface on which the mean curvature is difficult to estimate. A general framework of regularization based on area-decreasing flow is proposed and applied to smoothing range data and arbitrary triangle meshes. Crease edges are preserved by adaptively changing the regularization parameter. The edge strength of each vertex on a triangle mesh is computed by fusing the tensor voting and the orientation check of the normal vector field inside a geodesic window. Experimental results show the proposed algorithm provides successful smoothing for both raw range data and surface meshes.

The literature review of range image smoothing and surface smoothing is given in Section 2.1. In Section 3.1, the regularized energy function is formulated for surface smoothing based on area-decreasing flow. Section 3.2 derives the area-decreasing stabilizer for calibrated range data and shows how to preserve the edges. The stabilizer for the triangle mesh is introduced in Section 3.3 along with the discussion of edge preservation based on the tensor voting approach. Experimental results are presented in Section 7.1.

3.1 Regularization Based on Area-Decreasing Flow

The 3D image smoothing problem corresponds to a constrained optimization problem. Regularization is the most widely used method to solve practical optimization problems with one or more constraints. Major advantages of regularization include: (i) simple and intuitive formulation of the objective or energy function, and (ii) flexibility in incorporating one or more constraints into the optimization process.

When regularization is applied to surface smoothing, $\mathbf{x}$ is replaced by $\mathbf{X}$, a parameterized surface which represents a differentiable map from an open set $U \subset \mathbf{R}^2$ into $\mathbf{R}^3$, that is $\mathbf{X} : U \subset \mathbf{R}^2 \to \mathbf{R}^3$. Given a bounded domain $D \subset U$ and a differentiable function $l : \overline{D} \to \mathbf{R}$, where $\overline{D}$ represents the union of the domain D and its boundary, the variation of $\mathbf{X}(\overline{D})$ along normal $\mathbf{n}$ is given as

$$\phi : \overline{D} \times (-\varepsilon, \varepsilon) \to \mathbf{R}^3, \tag{3.1}$$

$$\phi(u, v, t) = \mathbf{X}(u, v) + t\, l(u, v)\mathbf{n}(u, v), \tag{3.2}$$

where $(u, v) \in \overline{D}$ and $t \in (-\varepsilon, \varepsilon)$.

For the map $\mathbf{X}^t(u, v) = \phi(u, v, t)$, the data compatibility term $g(\mathbf{X})$ is chosen to be the distance between the original surface and the smoothed surface, such as

$$g(\mathbf{X}) = \|\mathbf{X} - \mathbf{X}^t\|^2, \tag{3.3}$$

where $\| \cdot \|$ denotes a norm. The smoothness is assumed to be the prior knowledge of the surface. From the frequency analysis point of view, the smoothing or stabilizing term $h(\mathbf{X})$ should reflect the high frequency energy. From the observation that noisy surfaces usually have larger area than smooth surfaces, $h(\mathbf{X})$ is chosen to be the surface area A_s. This is closely related to the mean curvature flow that has been applied to 3D image processing [22, 60, 101, 112].

The area $A_s(t)$ of $\mathbf{X}^t(\overline{D})$ is obtained as

$$A_s(t) = \int_{\overline{D}} \sqrt{1 - 4\, t\, l\, H + R} \sqrt{EG - F^2} du\, dv, \tag{3.4}$$

where $lim_{t \to 0}(R/t) = 0$, H represents the mean curvature and E, F and G are the coefficients of the first fundamental form. The derivative of $A_s(t)$ at $t = 0$ is

$$A'_s(0) = -\int_{\overline{D}} 2\, l\, H \sqrt{EG - F^2} du\, dv. \tag{3.5}$$

The area is always decreasing if the normal variation is set as $l = H$, which is called the mean curvature flow. This flow will generate a minimal surface whose mean curvature vanishes everywhere. The details of minimal surface theory can be found in [23]. Finding a minimal surface that spans a given boundary is called the plateau problem. The discrete plateau problem was solved by Dziuk and Hutchinson [26, 27].

A minimal surface may take the form of a plane, catenoid, Enneper's surface, Scherk's surface, etc. Under mean curvature flow, a smooth cylinder can deform into a catenoid or two planes. To apply the mean curvature flow to regularized surface smoothing, the compatibility term $g(\mathbf{X})$ should be used to generate variation near the original surface.

The curvature on discrete surfaces, such as for the range data and surface meshes, is difficult to compute because it is defined on an infinitesimal area. For this reason, direct surface area minimization is better than mean curvature flow for smoothing discrete surfaces. The triangle mesh is used to represent surfaces because most types of surface meshes can be easily translated into a triangle mesh.

3.2 Range Data Smoothing

In Section 3.2.1, the area-decreasing flow is applied to range data smoothing. Edge preservation by adaptive regularization is discussed in Section 3.2.2.

3.2.1 Area-Decreasing Stabilizer for Range Data

In previous works [10, 80, 83, 97, 107], the surface was considered as a graph $z(x, y)$ and represented as z_{ij} over a rectangular grid. Given the observed data c_{ij} on the same rectangular grid, the viewpoint-invariant surface reconstruction can be performed by minimizing the regularized energy function as

$$f = \sum_{ij} (z_{ij} - c_{ij})^2 / \sigma_{ij}^2 + \lambda h, \tag{3.6}$$

where $1/\sigma_{ij}$ denotes the confidence of the measurement. In practice, $1/\sigma_{ij}$ approximates the surface slant, $\cos \zeta$, with respect to the incident laser. The larger the angle ζ between the surface normal and the direction of measurement is, the smaller the confidence becomes. Because $(z_{ij} - c_{ij})/\sigma_{ij}$ is also the perpendicular distance between the estimated and the real surfaces, this distance is viewpoint invariant. The stabilizing function h can take several different forms. For example, a first order term is used in [107]; while a second order term is employed in [83].

Estimating the elevation z_{ij} is feasible for sparse data. But in dense range images from a range scanner with a spherical coordinate system, $z(x, y)$ is no longer a graph and estimating the elevation may result in overlap in range measurement. Therefore, the range r_{ij} is estimated instead of z_{ij} so that all refinement takes place along the line of measurement.

For the PERCEPTRON range scanner [65], the range value of each pixel R_{ij} is converted to (x_{ij}, y_{ij}, z_{ij}) in Cartesian coordinates. The spherical coordinate system-based calibration model described in [38] is adopted here:

$$\begin{cases} x_{ij} = dx + r\sin\alpha \\ y_{ij} = dy + r\cos\alpha\sin\beta \\ z_{ij} = dz - r\cos\alpha\cos\beta \end{cases} \tag{3.7}$$

$$\begin{cases} \alpha = \alpha_0 + H_0(col/2 - j)/N_0 \\ \beta = \beta_0 + V_0(row/2 - i)/M_0 \end{cases} \tag{3.8}$$

$$\begin{cases} r_1 = (dz - p_2)/\delta \\ r_2 = \sqrt{dx^2 + (p_2 + dy)^2}/\delta \\ r = (R_{ij} + r_0 - r_1 - r_2)/\delta \end{cases} \tag{3.9}$$

$$
\begin{cases}
dx = (p_2 + dy)\tan\alpha \\
dy = dz\tan(\theta + 0.5\beta) \\
dz = -p_1(1.0 - \cos\alpha)/\tan\gamma
\end{cases}
\tag{3.10}
$$

where $\{p_1, p_2, \gamma, \theta, \alpha_0, \beta_0, H_0, V_0, r_0, \delta\}$ represents the set of calibration parameters of the scanner, and (M_0, N_0) refers to the image size.

For estimating r, the following parameterization can be used

$$
X(\alpha, \beta) = (r\sin\alpha, r\cos\alpha\sin\beta, -r\cos\alpha\cos\beta), \tag{3.11}
$$

and ignore small values denoted by dx, dy and dz in the analysis. The coefficients of the first fundamental form, which will be used shortly in the computation of the surface area, are given, in the basis of $\{\mathbf{X}_\alpha, \mathbf{X}_\beta\}$, as

$$
\begin{cases}
E = \mathbf{X}_\alpha \cdot \mathbf{X}_\alpha = r^2 + r_\alpha^2 \\
F = \mathbf{X}_\alpha \cdot \mathbf{X}_\beta = r_\alpha r_\beta \\
G = \mathbf{X}_\beta \cdot X_\beta = r^2\cos^2\alpha + r_\beta^2
\end{cases}
\tag{3.12}
$$

where

$$
\mathbf{X}_\alpha = \frac{\partial \mathbf{X}}{\partial \alpha}, \quad \mathbf{X}_\beta = \frac{\partial \mathbf{X}}{\partial \beta}, \quad r_\alpha = \frac{\partial r}{\partial \alpha}, \quad \text{and} \quad r_\beta = \frac{\partial r}{\partial \beta}.
$$

c is denoted as the observed value of r. Range data smoothing can then be performed by minimizing the following energy function

$$f = \sum_{ij} (r_{ij} - c_{ij})^2 / \sigma_{ij}^2 + h. \tag{3.13}$$

Let the stabilizing function h be the surface area, which can be calculated as

$$h = A_s = \iint_D \sqrt{EG - F^2}\, d\alpha\, d\beta, \tag{3.14}$$

where D represents the domain of (α, β).

The stabilizing function used by Yi [107] has the same effect of minimizing surface area, but the assumption is made that the surface is a graph in Cartesian coordinates. By using the height map $z(x, y)$, the coefficients of the first fundamental form are obtained as

$$\begin{cases} E = 1 + z_x^2 \\ F = z_x z_y \\ G = 1 + z_y^2 \end{cases}. \tag{3.15}$$

Accordingly, the stabilizing function is obtained as

$$h = \iint_D \sqrt{1 + z_x^2 + z_y^2}\, dx\, dy, \tag{3.16}$$

which is similar to the function used in [107].

As (3.14) is not easily minimized due to the square root operation, minimization is applied on

$$h = \iint_D (EG - F^2)\, d\alpha\, d\beta. \tag{3.17}$$

That minimizations of (3.14) and (3.17) are equivalent is justified in the Appendix B. From (3.12), (3.13) and (3.17), the finally regularized energy function is given as

$$f = \sum_{ij} (r_{ij} - c_{ij})^2/\sigma_{ij}^2 + \sum_{ij} \lambda_{ij}(r_{ij}^4 \cos^2\alpha + r_{ij}^2 r_\beta^2 + r_\alpha^2 r_{ij}^2 \cos^2\alpha), \tag{3.18}$$

Among various optimization methods, the simple gradient descent method is adopted to minimize (3.18). The estimation r'_{ij} of each measurement r_{ij} is given as

$$r'_{ij} = r_{ij} - w\frac{\partial f}{\partial r_{ij}}, \tag{3.19}$$

where w represents the iteration step size and

$$\begin{aligned} \frac{\partial f}{\partial r_{ij}} &= 2(r_{ij} - c_{ij})/\sigma_{ij}^2 + \lambda_{ij}\Big\{4r_{ij}^3 \cos^2\alpha \\ &+ [2r_{ij}(r_{i+1,j} - r_{ij})^2 - 2r_{ij}^2(r_{i+1,j} - r_{ij}) + 2r_{i-1,j}^2(r_{ij} - r_{i-1,j})]\left(\frac{1}{d\beta}\right)^2 \\ &+ [2r_{ij}(r_{i,j+1} - r_{ij})^2 - 2r_{ij}^2(r_{i,j+1} - r_{ij}) + 2r_{i,j-1}^2(r_{ij} - r_{i,j-1})]\left(\frac{\cos\alpha}{d\alpha}\right)^2\Big\}. \end{aligned} \tag{3.20}$$

In calculating the derivative of f in (3.20), the following forward difference approximations were used:

$$r_\alpha = \frac{r_{i,j+1} - r_{ij}}{d\alpha} \tag{3.21}$$

and

$$r_\beta = \frac{r_{i+1,j} - r_{ij}}{d\beta}. \tag{3.22}$$

Alternatively, the central difference approximation can also be used, such as

$$r_\alpha = \frac{r_{i,j+1} - r_{i,j-1}}{2d\alpha} \tag{3.23}$$

and

$$r_\beta = \frac{r_{i+1,j} - r_{i-1,j}}{2d\beta}. \tag{3.24}$$

The derivative of f is then differently obtained as

$$\begin{aligned}\frac{\partial f}{\partial r_{ij}} &= 2(r_{ij} - c_{ij})/\sigma_{ij}^2 + \lambda_{ij}\Big\{4r_{ij}^3 \cos^2\alpha \\ &+ [2r_{ij}(r_{i+1,j} - r_{i-1,j})^2 - 2r_{i+1,j}^2(r_{i+2,j} - r_{ij}) + 2r_{i-1,j}^2(r_{ij} - r_{i-2,j})]\left(\frac{1}{2d\beta}\right)^2 \\ &+ [2r_{ij}(r_{i,j+1} - r_{i,j-1})^2 - 2r_{i,j+1}^2(r_{i,j+2} - r_{ij}) + 2r_{i,j-1}^2(r_{ij} - r_{i,j-2})]\left(\frac{\cos\alpha}{2d\alpha}\right)^2\Big\}.\end{aligned} \tag{3.25}$$

The experiment shows that the central difference approximation makes the convergence faster.

3.2.2 Edge Preservation for Range Data

Incorporation of the regularizing term in the regularized energy function, as shown in (3.18), tends to suppress local change in the range image. Although the smoothing function is good for suppressing undesired noise, it also degrades important feature information such as edges, corners, and segment boundaries. Using an additional energy term to preserve discontinuity [10], however, generally makes the minimization very difficult. Instead, the results of a 2D edge detection operation are used to adaptively change the weight of the regularization parameter λ so that edges are preserved during the regularization. Although there are various simple edge enhancement filters, the edge enhancer in [17] is used, which guarantees both good detection and localization. Let $J_x(i,j)$ and $J_y(i,j)$ be the gradient component of the Gaussian filtered version of r_{ij} in the horizontal and the vertical directions, respectively. Then the edge strength image can be obtained as

$$e_{ij} = \sqrt{{J_x}^2(i,j) + {J_y}^2(i,j)}. \tag{3.26}$$

The regularizing term in (3.18) can then be adaptively weighted, as in [46], using

$$\lambda_{ij} = \frac{\rho}{1 + \kappa\, e_{ij}^2}, \text{ for } 0 < \kappa < 1, \tag{3.27}$$

where κ represents a parameter that determines the sensitivity of edge strength, and ρ is a prespecified constant. The selection of ρ generally depends on the desired data compatibility as well as the level of noise.

3.3 Surface Mesh Smoothing

In order to apply the proposed regularized range image smoothing algorithm, the accurate calibration model of the scanner must be known. However, for some scanners, such as the RIEGL system LMS-Z210 [71] used in this work, the calibration parameters are used by the manufacturer and not released. It is also desirable to smooth an arbitrary surface instead of single view range data. In Section 3.3.1, the area-decreasing flow is extended to process arbitrary surfaces represented by a triangle mesh. Adaptive smoothing is discussed in Section 3.3.2, where edge strength is computed based on the tensor voting approach.

3.3.1 Area-Decreasing Stabilizer for Surface Mesh

For an umbrella neighborhood [49] with I triangles on a triangle mesh, the position of the center vertex $\mathbf{v}$ is adjusted along the normal direction $\mathbf{n}$, as shown in Fig. 3.1. The superscript k represents the k-th adjustment. The original position of the center vertex is denoted as $\mathbf{v}^{(0)}$. In the k-th adjustment, the center vertex moves from $\mathbf{v}^{(k)}$ to $\mathbf{v}^{(k+1)}$

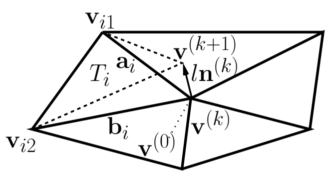


Figure 3.1: Umbrella operation: In the k-th iteration, the vertex moves from $\mathbf{v}^{(k)}$ to $\mathbf{v}^{(k+1)}$ by $l\,\mathbf{n}^{(k)}$.

by the length l in the direction of $\mathbf{n}^{(k)}$, such as

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + l\,\mathbf{n}^{(k)}. \tag{3.28}$$

An adjustment is made to minimize the area of the umbrella and at the same time be compatible to the original measurement. The local energy function is defined as

$$f(l) = \sum_{i=1}^{I} 4[S_i^{(k+1)}]^2 + \lambda\,\|\Delta\mathbf{v}^{(k+1)}\|^2, \tag{3.29}$$

where $S_i^{(k+1)}$ is the area of the triangle $\mathbf{v}_{i1}\mathbf{v}_{i2}\mathbf{v}^{(k+1)}$, denoted by T_i, and $\Delta\mathbf{v}^{(k+1)}$ is defined as

$$\Delta\mathbf{v}^{(k+1)} = \mathbf{v}^{(k+1)} - \mathbf{v}^0 = \Delta\mathbf{v}^{(k)} + l\,\mathbf{n}^{(k)}. \tag{3.30}$$

For computational convenience, $\sum S_i$ is replaced by $\sum S_i^2$. Similar to the justification shown in the Appendix B, the replacement can be justified using the Cauchy-Schwarz inequality, such as

$$\sum_{i=1}^{I} S_i \leq \sqrt{I \sum_{i=1}^{I} S_i^2}. \tag{3.31}$$

$S_i^{(k+1)}$ is computed as

$$S_i^{(k+1)} = \frac{1}{2} \|(\mathbf{v}_{i1} - \mathbf{v}^{(k+1)}) \wedge (\mathbf{v}_{i2} - \mathbf{v}^{(k+1)})\|. \tag{3.32}$$

For notational simplicity,

$$\mathbf{a}_i = \mathbf{v}_{i1} - \mathbf{v}^{(k)} and \mathbf{b}_i = \mathbf{v}_{i2} - \mathbf{v}^{(k)} \tag{3.33}$$

are defined. The energy function defined in (3.29) can then be rewritten as

$$\begin{aligned} f(l) &= \sum_{i=1}^{I} \|(\mathbf{a}_i - l\,\mathbf{n}^{(k)}) \wedge (\mathbf{b}_i - l\,\mathbf{n}^{(k)})\|^2 + \lambda\,\|\Delta \mathbf{v}^{(k+1)}\|^2 \qquad (3.34) \\ &= l^2 \Big\{ \sum_{i=1}^{I} \{\|\mathbf{a}_i - \mathbf{b}_i\|^2 - [(\mathbf{a}_i - \mathbf{b}_i) \cdot \mathbf{n}^{(k)}]^2\} + \lambda \Big\} \\ &\quad + 2\,l\,\{\sum_{i=1}^{I} [(\mathbf{a}_i \cdot \mathbf{n}^{(k)})\mathbf{b}_i - (\mathbf{b}_i \cdot \mathbf{n}^{(k)})\mathbf{a}_i] \cdot (\mathbf{a}_i - \mathbf{b}_i) + \lambda\,(\Delta \mathbf{v}^{(k)} \cdot \mathbf{n}^{(k)})\} \\ &\quad + \sum_{i=1}^{I} \|\mathbf{a}_i\|^2 \|\mathbf{b}_i\|^2 - (\mathbf{a}_i \cdot \mathbf{b}_i)^2 + \lambda \|\Delta \mathbf{v}^{(k)}\|^2. \end{aligned}$$

The optimum value of l, which minimizes f, can be obtained by solving $\frac{\partial f}{\partial l} = 0$, that is

$$l = \frac{A - \lambda\,\Delta\mathbf{v}^{(k)} \cdot \mathbf{n}^{(k)}}{B + \lambda}, \tag{3.35}$$

where

$$A = \sum_{i=1}^{I} \{(\mathbf{b}_i \cdot \mathbf{n}^{(k)})\mathbf{a}_i - (\mathbf{a}_i \cdot \mathbf{n}^{(k)})\mathbf{b}_i\} \cdot (\mathbf{a}_i - \mathbf{b}_i),$$

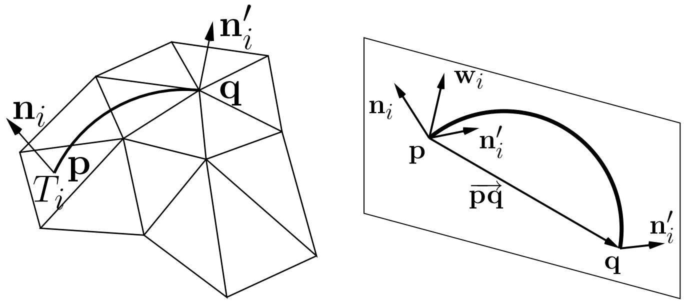
and

$$B = \sum_{i=1}^{I} \|\mathbf{a}_i - \mathbf{b}_i\|^2 - \{(\mathbf{a}_i - \mathbf{b}_i) \cdot \mathbf{n}^{(k)}\}^2.$$

The surface is iteratively deformed in the sense of minimizing the area of the umbrella according to (3.28).

3.3.2 Edge Preservation for Surface Mesh

Similar to the process used for 2D images, edge detection on a triangle mesh is performed by operation in a local window that is usually called the neighborhood. From a small neighborhood, such as an umbrella, it is difficult to determine if the vertex is from noise or near an edge. A large neighborhood is necessary for detecting edges on a surface with strong noise. The irregular connections on the triangle mesh make window selection not as straightforward as with 2D images. This research proposes a new 3D edge detection algorithm using the geodesic window instead of the neighborhood defined



(a) (b)

Figure 3.2: Surface normal voting. (a) $\mathbf{n}'_i$ is the voted normal by T_i's normal $\mathbf{n}_i$. (b) The normal $\mathbf{n}_i$ at $\mathbf{p}$ is transported through the arc $\widehat{\mathbf{pq}}$ producing the voted normal $\mathbf{n'}_i$ at $\mathbf{q}$.

by the Euclidean measure in previous works. The geodesic window is a small surface patch whose boundary has the same geodesic distance to the center vertex. Details for computing the geodesic distance on a triangle mesh is introduced in Section 4.2.1.

Medioni *et al.* [57] introduced a tensor voting approach that can signal the presence of a salient structure, a discontinuity, or an outlier, at any location. Page *et al.* [62, 63] applied tensor voting to robustly estimate the principal curvatures and principal directions. In this work, tensor voting and the orientation check are combined inside the geodesic window to detect the crease edges on a triangle mesh.

The tensor voting method for detecting crease edges can simply be considered as the eigen analysis of the surface normal vector field. For a certain vertex $\mathbf{q}$, the votes are cast by the neighboring triangles, as shown in Fig. 3.2(a). The voted tensor cast by the triangle T_i at vertex $\mathbf{q}$ is $\mu_i \mathbf{n}'_i \mathbf{n}'^T_i$, where $\mathbf{n}'_i$ is the voted normal by T_i's normal $\mathbf{n}_i$

and μ_i is the weight of the vote. The new tensor collected at $\mathbf{q}$ is

$$\mathbf{T} = \sum_{i=1}^{M} \mu_i \mathbf{n}'_i \mathbf{n}'^T_i, \tag{3.36}$$

where M is the number of triangles inside the geodesic window of $\mathbf{q}$ and $M > I$. $\mathbf{n}'_i$ is obtained by transporting $\mathbf{n}_i$ through a sector of arc connecting $\mathbf{p}$ and $\mathbf{q}$ where $\mathbf{p}$ represents the centroid of T_i. Fig. 3.2(b) illustrates the voting process. The arc is on the plane defined by two vectors $\mathbf{n}_i$ and $\overrightarrow{\mathbf{pq}}$. The normals at two terminals of the arc are $\mathbf{n}_i$ and $\mathbf{n}'_i$. $\mathbf{n}'_i$ is computed as

$$\mathbf{n}'_i = 2(\mathbf{n}_i \cdot \mathbf{w}_i)\mathbf{w}_i - \mathbf{n}_i, \tag{3.37}$$

where

$$\mathbf{w}_i = \frac{(\overrightarrow{\mathbf{pq}} \wedge \mathbf{n}_i) \wedge \overrightarrow{\mathbf{pq}}}{\|(\overrightarrow{\mathbf{pq}} \wedge \mathbf{n}_i) \wedge \overrightarrow{\mathbf{pq}}\|}. \tag{3.38}$$

The weight μ_i exponentially decreases according to the geodesic distance d between $\mathbf{p}$ and $\mathbf{q}$, such as

$$\mu_i = e^{-(d/\tau)^2}, \tag{3.39}$$

where τ controls the decaying speed and depends on the scale of the input triangle mesh.

The singular value decomposition is applied to the new tensor $\mathbf{T}$ as

$$\mathbf{T} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} \nu_1 & 0 & 0 \\ 0 & \nu_2 & 0 \\ 0 & 0 & \nu_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix}. \tag{3.40}$$

where $\nu_1 \geq \nu_2 \geq \nu_3$. In [57], $\mathbf{T}$ is rewritten as

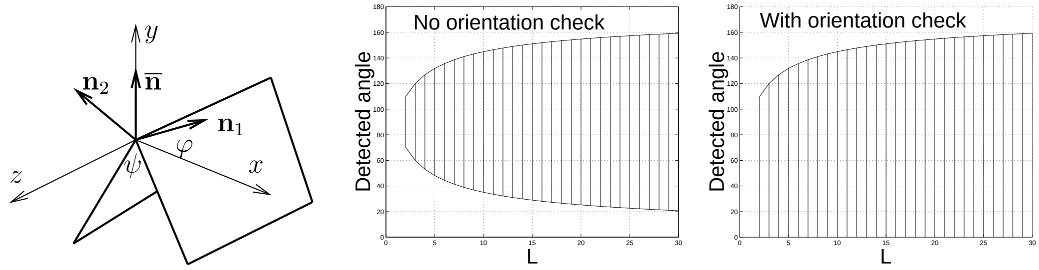
$$\mathbf{T} = (\nu_1 - \nu_2)\mathbf{e}_1\mathbf{e}_1^T + (\nu_2 - \nu_3)(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T) + \nu_3(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T + \mathbf{e}_3\mathbf{e}_3^T), \tag{3.41}$$

where $\mathbf{e}_1\mathbf{e}_1^T$ describes a stick, $\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T$ describes a plate and $\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T + \mathbf{e}_3\mathbf{e}_3^T$ describes a ball. Here the plate component is of special interest. $\nu_2 - \nu_3$ is related to the strength of the planar junction. The junction is detected if

$$\nu_2 - \nu_3 > \nu_1 - \nu_2 \quad \text{and} \quad \nu_2 - \nu_3 > \nu_3. \tag{3.42}$$

However, this only works for junctions near 90°. To explain this situation, assume that a crease edge is parallel to the z axis and two planes generating the edge are symmetric according to the y axis. The normals of the two surfaces are

$$\mathbf{n}_1 = (\cos\varphi, \sin\varphi, 0)^T \tag{3.43}$$



(a) (b) (c)

Figure 3.3: Crease edge detection. (a) A crease edge with angle ψ. (b) Angle ψ that can be detected without an orientation check. Edges with sharp angles are missing. (c) Angle ψ that can be detected with an orientation check. L is the factor in (3.48).

and

$$\mathbf{n}_2 = (-\cos\varphi, \sin\varphi, 0)^T, \tag{3.44}$$

as shown in Fig. 3.3(a).

According to (3.36), if μ_i is set to 1, the collected tensor at the vertex on the edge is given as

$$\mathbf{T} = \begin{bmatrix} M\cos^2\varphi & 0 & 0 \\ 0 & M\sin^2\varphi & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.45}$$

And the eigen values are simply obtained as

$$\nu_1 = M\cos^2\varphi,\ \nu_2 = M\sin^2\varphi, \text{ and } \nu_3 = 0, \text{ for } 0^\circ < \varphi \leq 45^\circ \tag{3.46}$$

or

$$\nu_1 = M \sin^2 \varphi, \, \nu_2 = M \cos^2 \varphi, \text{ and } \nu_3 = 0, \text{ for } 45^\circ < \varphi < 90^\circ. \tag{3.47}$$

From (3.42), where

$$\nu_1 < 2\nu_2,$$

and $\psi = 2\varphi$, a limited range of detectable edge angle is obtained, such as

$$70.53^\circ < \psi < 109.47^\circ.$$

This research proposes a method to solve this problem. Initially, it is obvious that if the edge detection condition is set as

$$\nu_1 < L\nu_2 \quad \text{and} \quad \nu_2 > 2\nu_3, \tag{3.48}$$

Crease edges can be detected with

$$2\, tan^{-1} \frac{1}{\sqrt{L}} < \psi < 2\, tan^{-1} \sqrt{L}, \tag{3.49}$$

which is depicted by the shaded region in Fig. 3.3(b). Crease edges with large ψ can be detected by appropriately increasing L. However, (3.49) is still not complete because of a limit on sharp edges. This limit stems from the covariance matrix $\mathbf{T}$ that contains no orientation information. In other words, $\mathbf{v}$ and $-\mathbf{v}$ result in the same $\mathbf{T}$. Let

$$\overline{\mathbf{n}} = \sum_{i=1}^{M} \mu_i \mathbf{n}'_i, \tag{3.50}$$

as shown in Fig. 3.3(a). Observing that $\overline{\mathbf{n}}$ tends to align with $\mathbf{e}_1$ if $\psi > 90°$ or to be perpendicular with $\mathbf{e}_1$ if $\psi < 90°$,

$$|\overline{\mathbf{n}} \cdot \mathbf{e}_1| < \delta \tag{3.51}$$

is used as an additional edge detection condition, where δ is a positive threshold. In the experiments, $\delta = 0.3$ is used. Condition (3.48) and the orientation check in (3.51) provide a reasonable range of detectable crease edge angles as shown in Fig. 3.3(c).

The crease edge strength is defined as

$$s = \begin{cases} 1, & \text{if } |\overline{\mathbf{n}} \cdot \mathbf{e}_1| < \delta \\ 1, & \nu_3 > \alpha(\nu_1 - \nu_2) \\ & \text{and } \nu_3 > \beta(\nu_2 - \nu_3) \\ (\nu_2 - \nu_3)/\nu_1, & \text{otherwise} \end{cases}, \tag{3.52}$$

such that $0 \le s \le 1$. The definition of edge strength is divided by three conditions. The second and the third condition correspond to the tensor voting theory [57]. Vertices on the corners are considered to have high edge strength, described by the second condition.

The third condition provides a continuous crease edge strength approximation in $[0, 1]$. The first condition is to detect sharp edges where the traditional tensor voting method fails.

Eq. (3.28) is then modified by inserting an additional control factor based on the crease edge strength, such as

$$\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + e^{-5s}\, l\, \mathbf{n}^{(k)}, \tag{3.53}$$

to realize the edge-adaptive smoothing.

The crease edge strength for each vertex is computed only once before deforming the surface. Assume the triangle mesh has N vertices, and there are M triangles inside the geodesic window on average. The computational complexity of crease edge strength is $O(NM \log M)$. The complexity for each iteration of smoothing is $O(N)$.

The iteration is stopped if the following condition is satisfied

$$|Z^{(k+1)} - Z^{(k)}|/Z^{(k+1)} < \epsilon, \tag{3.54}$$

where

$$Z^{(k)} = \sum^{N} \|\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}\|,$$

and ϵ is a threshold chosen as 0.1 in our experiment. Smaller ϵ results in more iterations of the smoothing.

Surface smoothing using area-decreasing flow has been successfully applied to real data from different range scanners. Experimental results are shown in Section 7.1.

Chapter 4

Surface Registration by Point Fingerprint

In this chapter, a new and efficient surface representation method for surface matching and its application to 3D image registration are proposed. A feature carrier for the surface point, which is a set of 2D contours that are the projections of geodesic circles onto the tangent plane, is generated. The carrier is named point fingerprint because its pattern is similar to human fingerprints and discriminating for each point. Corresponding points on surfaces from different views are found by comparing their fingerprints. Rigid transformation is computed from the point correspondences.

The literature review of surface registration is given in Section 2.2. In Section 4.1 the exponential map that is the theoretical basis of the proposed point fingerprint scheme is introduced. In Section 4.2 methods for generating geodesic circles and defining 3D point fingerprint are presented. Section 4.3 introduces a novel method to select the candidate points. In Section 4.4 the feature matching method is proposed. The application of point fingerprint scheme to surface registration is discussed in Section 4.5. Experimental results of 3D image registration are presented in Section 7.2.

4.1 The Exponential Map

Point correspondence-based surface matching methods generate a feature map for each surface point by mapping from a surface patch to the 1D or 2D domain. We propose that the mapping should have the following four properties.

1. **View-Invariance**

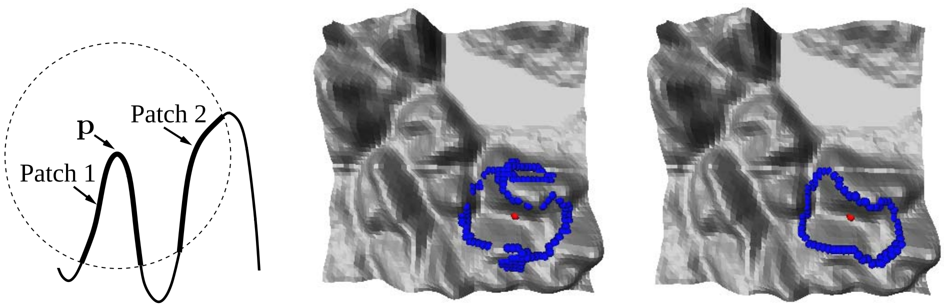
   The features for surface matching must be view-invariant because they are used to match the points from different views.

2. **One-to-One mapping**

   Local one-to-one mapping allows the map to carry meaningful features such as curvature, normal, and color. If the mapping is not one-to-one, each pixel may correspond to several surface points so that this pixel cannot carry specific information.

3. **Continuity**

   There are no exact correspondences between vertices on two surfaces due to discrete sampling and surface noise. Although two maps generated from different



(a) (b) (c)

Figure 4.1: Geodesic vs Euclidean. (a) Surface patch 1 is the geodesic neighborhood of **p**; Euclidean neighborhood erroneously includes surface patch 2. (b) An ambiguous Euclidean contour. (c) A clearly defined geodesic contour.

views cannot be exactly the same, they should be similar to a certain degree for the robustness of the feature map.

4. **Localization**

   The mapping should reflect the local geometry, which is more flexible in dealing with surfaces of arbitrary topology than global geometric information.

The view invariant 2D feature of a point $\mathbf{p}$ on a surface S should be defined on a plane that is common to different views. The tangent plane $T_{\mathbf{p}}(S)$ is an option and can be easily obtained. In previous works, Splash [82], Spin Image [45], and SPS [105] are defined on the tangent plane. Some works use the geodesic measure while others use the Euclidean measure. The problem with using the Euclidean measure is that the neighborhood of a surface point is sometimes ambiguous. In Fig. 4.1(a), surface patch 1 is the geodesic neighborhood of $\mathbf{p}$, while the Euclidean neighborhood erroneously includes surface patch 2. In previous works, only Splash [82] and HSI [110] considered the geodesic measure. PS [20] uses the Euclidean measure because a contour is generated by the intersection between a sphere and the surface. Point fingerprint uses geodesic contour. The difference is illustrated by generating both Euclidean and geodesic contours on a surface. The Euclidean contour in Fig. 4.1(b) is ambiguous, which makes the PS generated later on problematic. However, the geodesic contour in Fig. 4.1(c) is clearly defined.

4.1.1 Definition of the Exponential Map

The concept of a point fingerprint was inspired from the exponential map, which was initially considered as a possible alternative to the harmonic map used in [110]. The concept of the exponential map is briefly introduced next.

For surfaces in $\mathbf{R}^3$, the geodesics can be characterized as those curves $\mathbf{c}(s)$, where s represents arc length, for which the acceleration $\mathbf{c}''(s)$ in $\mathbf{R}^3$ is perpendicular to the surface, i.e., the acceleration of $\mathbf{c}$ from the viewpoint of the surface is zero. A geodesic minimizes arc length for points sufficiently close. In addition, if a curve minimizes the arc length between any two of its points, it is a geodesic. S represents the surface, and $T_{\mathbf{p}}(S)$ represents the tangent plane at the point $\mathbf{p}$. The following theorem indicates the uniqueness of the geodesic in the closed neighborhood of a surface point.

Theorem 1 *Given a point $\mathbf{p} \in S$ and a vector $\mathbf{v} \in T_{\mathbf{p}}(S)$, $\mathbf{v} \neq \mathbf{0}$, there exist an $\varepsilon > 0$ and a unique parameterized geodesic $\gamma : (-\varepsilon, \varepsilon) \rightarrow S$ such that $\gamma(0) = \mathbf{p}$, $\gamma'(0) = \mathbf{v}$.*

The proof can be found in [23].

To indicate the dependence of the geodesic on the vector $\mathbf{v}$, it is convenient to denote this by $\gamma(t, \mathbf{v})$. The exponential map is defined as $exp_{\mathbf{p}}(\mathbf{v}) = \gamma(1, \mathbf{v})$, with $exp_{\mathbf{p}}(\mathbf{0}) = \mathbf{p}$. Geometrically, the construction corresponds to laying off a length equal to $\|\mathbf{v}\|$ along the geodesic that passes through $\mathbf{p}$ in the direction of $\mathbf{v}$, and the point of S thus obtained is denoted by $exp_{\mathbf{p}}(\mathbf{v})$. The exponential map establishes a one-to-one correspondence between a point's surface neighborhood and its tangent plane on which a feature map can be obtained.

The $exp_{\mathbf{p}}$ is important in that it is always defined and differentiable in some neighborhood of $\mathbf{p}$. The following theorem states this fact.

Theorem 2 *$exp_{\mathbf{p}} : B_\varepsilon \subset T_{\mathbf{p}}(S) \rightarrow S$ is a diffeomorphism in a neighborhood $U \subset B_\varepsilon$ of the origin of $T_{\mathbf{p}}(S)$.*

The proof can be found in [23].

The exponential map establishes a one-to-one correspondence between the neighborhood of a point and its tangent plane on which a feature map can be obtained. It satisfies four properties of a mapping proposed earlier. From the exponential map, local coordinate systems can be introduced. The most commonly used are normal coordinates and geodesic polar coordinates. The geodesic polar coordinates are used here, which correspond to polar coordinates (ρ, θ) in the tangent plane $T_{\mathbf{p}}(S)$, where ρ refers to the geodesic distance and θ refers to the departure angle of vector $\mathbf{v}$ relative to a reference vector.

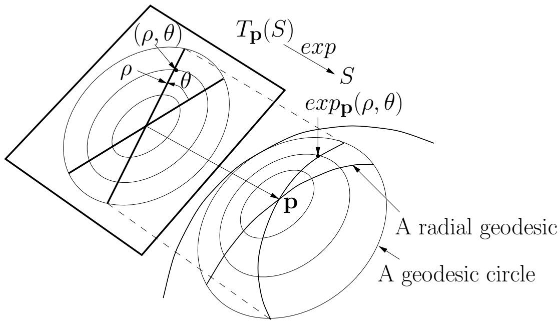


Figure 4.2: Definition of the exponential map. The geodesic circle is formed by all the points that have the same geodesic distances ρ to the center point $\mathbf{p}$. θ is the departure angle of the radial geodesic.

Fig. 4.2 shows the definition of the exponential map on a surface patch. The exponential map is a map from the tangent plane to the surface patch. When the feature on the tangent plane is generated, the inverse exponential map is actually used. But for convenience, the term exponential map will continue to be used.

To use the exponential map to generate the feature image of one point $\mathbf{p}$, ρ and θ must be computed for each vertex in a neighborhood of $\mathbf{p}$. The method for computing ρ will be presented later. In the following subsection the difficulty involved in computing the departure angle θ is discussed.

4.1.2 Computation of the Departure Angle

Let M be a Riemannian manifold and $\mathbf{p} \in M$. If $exp_{\mathbf{p}}$ is defined at $\mathbf{v} \in T_{\mathbf{p}}(M)$ and $\mathbf{w} \in T_{\mathbf{v}}(T_{\mathbf{p}}(M))$, then the differential of $exp_{\mathbf{p}}$ is

$$(\mathrm{d}exp_{\mathbf{p}})_{\mathbf{v}}\mathbf{w} = \frac{\partial f}{\partial s}(1, 0), \tag{4.1}$$

where f represents a parameterized surface as

$$f(t,s) = exp_{\mathbf{p}} t\mathbf{v}(s), \text{ for } 0 \leq t \leq 1, \text{ and } -\varepsilon \leq s \leq \varepsilon, \tag{4.2}$$

and $\mathbf{v}(s)$ is a curve in $T_{\mathbf{p}}(M)$ with $\mathbf{v}(0) = \mathbf{v}$, $\mathbf{v}'(0) = \mathbf{w}$. $\|(\mathrm{d}exp_{\mathbf{p}})_{\mathbf{v}}(\mathbf{w})\|$ denotes the rate of spreading of the geodesics $t \to exp_{\mathbf{p}} t\mathbf{v}(s)$ which start from $\mathbf{p}$. Now consider the field

$$(\mathrm{d}exp_{\mathbf{p}})_{t\mathbf{v}}(t\mathbf{w}) = \frac{\partial f}{\partial s}(t,0) \tag{4.3}$$

along the geodesic $\gamma(t) = exp_{\mathbf{p}}(t\mathbf{v}), 0 \leq t \leq 1$. It can be proven that $\frac{\partial f}{\partial s}$ satisfies a differential equation, which is called the Jacobi equation,

$$\frac{D^2 J}{dt^2} + K(\gamma'(t), J(t))\gamma'(t) = 0, \tag{4.4}$$

where $J(t) = \frac{\partial f}{\partial s}(t,0)$, K is the curvature, and $\frac{D}{dt}$ is the covariant derivative operator that is the orthogonal projection of the usual derivative onto the tangent plane. A detailed study of the Jacobi field can be found in [24].

The difficulty in solving the departure angle θ originates from the covariant derivative. No existing numerical method that computes the departure angle on a discrete surface has been found, and the computation involving the second order derivative is usually sensitive to noise. In the next section, the 3D point fingerprint based on the exponential map is introduced.

4.2 Fingerprint of the Surface Point

Fig. 4.2 indicates that geodesic circles carry the geometry information of the surface. To simplify feature comparison, the geodesic circles are projected onto the tangent plane to obtain a set of 2D contours, which are called the point fingerprint due to their similar appearance to human fingerprints. Point fingerprint is view-invariant because it is defined on the tangent plane. It also has continuity and localization properties. Although the orthogonal projection is not a one-to-one mapping, the fingerprint functions like a one-to-one mapping. This concept will be discussed later. In this research, point fingerprint is used to find the corresponding points.

The geodesic circle is formed by the points that have the same geodesic distance to the center point. The problems now are how to compute geodesic distance on a triangulated surface and generate the 3D point fingerprint.

4.2.1 Geodesic Circle Generation

The geodesic distance on the discrete surface must be computed, especially on a triangulated surface, which is the most popular representation of 3D objects. Dijkstra's algorithm [79] is widely used for finding the shortest path on a network with prescribed weights for each link between nodes, which was the case in generating HSI [110]. The problem of this algorithm is the inconsistency with the underlying continuous problem. For this reason this research proposes a modified version of Kimmel's work [48] to compute the geodesic distance. Kimmel developed two methods to compute geodesic distance on surfaces. These methods are based on Sethian's level set method and the fast marching method [78], respectively. The basic concept of Kimmel's methods is to evolve a geodesic circle with unit speed from a starting point **p** on the surface. The contour evolution on the surface can be described as solving an Eikonal equation such as

$$|\nabla \rho| F = 1, \tag{4.5}$$

with $\rho = 0$ at the initial location and F represents the magnitude of the evolution vector.

Kimmel's first method [47] is for a surface that is a height map $z(x, y)$ on a rectangular grid, in which the level set method is used. The evolution of the geodesic circle is obtained by evolving its corresponding projection on the xy plane. Kimmel's second method [48] is for an arbitrary surface in the form of a triangle mesh, which is a more general approach. The fast marching method is used because it is more efficient than the level set method for solving the Eikonal equation.

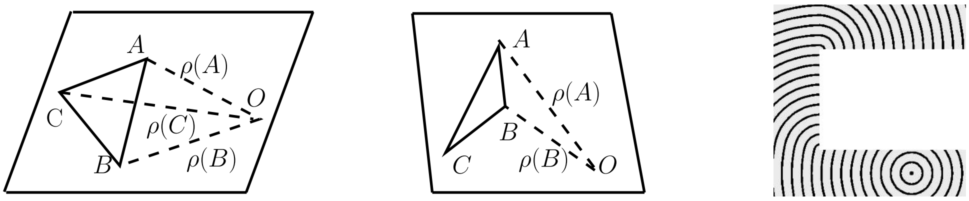
When the geodesic circle passes a 3D point, the reaching time, that is, the geodesic distance, is stored for that point. After the evolution stops, the geodesic distance between every point within the final geodesic circle and the starting point is known. Since Kimmel's second method computes the geodesic distance on a triangle mesh more accurately than Dijkstra's algorithm, it begins to draw much attention from different research areas. This method has been used to solve surface matching problems [100, 104] and accurately compute the curvature on a discrete surface [62].

The geodesic distance computation using the fast marching method is as follows.

Algorithm 4.1 (Computing Geodesic Distance [48]) *The center point is tagged as* Inside*, all the neighboring points of the center point are tagged as* Front*, and all the other surface points are tagged as* Outside*. The geodesic distance to the center point is denoted as ρ, and the following loop is then executed until ρ of a certain point exceeds a prespecified value.*

1. *Change the tag of the* Front *point with the smallest ρ to* Inside*.*
2. *Tag all neighboring points of this new* Inside *point as* Front*.*
3. *Recompute ρ of these neighboring points, using only values of points that are* Inside*. ρ is updated only if the recomputed result is smaller.*
4. *Go back to step 1.*

Step 1 in Algorithm 4.1 guarantees that the marching process is always from the point with the smallest geodesic distance to the point with the largest distance. Kimmel and Sethian [48] proposed a scheme for updating ρ in step 3 on an arbitrary triangle mesh. This scheme was extended from the marching on a regular triangulated planar domain and derived for the triangle mesh with only acute triangles. The obtuse triangles need to be split into acute triangles first.



(a) (b) (c)

Figure 4.3: Geodesic distance update. (a) Updating ρ inside the triangle. (b) Updating ρ outside the triangle. (c) Geodesic circles under occlusion.

This research proposes a simple new scheme to update ρ. Assume the geodesic distances at A and B have already been calculated as $\rho(A)$ and $\rho(B)$, i.e., A and B are *Inside*, as shown in Fig. 4.3(a). In order to update $\rho(C)$, a virtual triangle OAB is created with the lengths of the two edges equal to $\rho(A)$ and $\rho(B)$. Point O is the virtual center point in the same plane as the triangle ABC. If the update occurs inside the triangle, the new $\rho(C)$ is assigned as the length of the segment OC as

$$\rho(C) = \sqrt{|AC|^2 + \rho^2(A) + 2|AC|\rho(A)\cos(\alpha + \theta)}, \tag{4.6}$$

where

$$\alpha = \arccos \frac{|AB|^2 + |AC|^2 - |BC|^2}{2|AC||AB|}$$

and

$$\theta = \arccos \frac{\rho^2(A) + |AB|^2 - \rho^2(B)}{2|AB|\rho(A)}.$$

If the update occurs outside the triangle as shown in Fig. 4.3(b), $\rho(C)$ is assigned as

$$\rho(C) = \rho(B) + |BC|. \tag{4.7}$$

Usually, the vertex may be updated inside the triangle from at least one direction. The algorithm degenerates to Dijkstra's algorithm for the vertex that has not been updated inside a triangle. Locally, the computation of the distance on the surface is changed to a simple calculation in the planar triangles. The validity of this method is that the length of the curve on the surface is the integration along the tangent direction and the tangent surfaces are locally isometric to the planes [23].

Although the geodesic measure is more natural than the Euclidean measure in differential geometry, this measure has two shortcomings. First, the geodesic distance is more sensitive to surface sampling noise. To overcome this, the data should be smoothed in the preprocessing stage. Second, when the geodesic circles are generated on the self occluded objects, the calculated geodesic distance may be different from the exact value. As shown in Fig. 4.3(c), when a part of the plane is occluded, the geodesic circles are no longer concentric circles. One method to solve this problem is that the marching process is stopped whenever a step discontinuity is encountered.

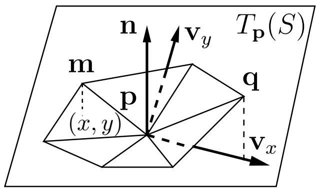


Figure 4.4: Local coordinate system.

4.2.2 Definition of Point Fingerprint

Before projecting the geodesic circles onto the tangent plane, the local coordinate system should be defined first, as shown in Fig. 4.4. The normal vector $\mathbf{n}$ at the point $\mathbf{p}$ defines one coordinate axis, which is computed as the average normal of the neighboring triangles. By arbitrarily choosing one of the neighbor points, $\mathbf{q}$, the other two axes can be defined as

$$\mathbf{v}_y = \mathbf{n} \times \overrightarrow{\mathbf{pq}} / \|\overrightarrow{\mathbf{pq}}\| \tag{4.8}$$

and

$$\mathbf{v}_x = \mathbf{v}_y \times \mathbf{n}. \tag{4.9}$$

The projection of a certain point $\mathbf{m}$ onto the tangent plane generated by $\mathbf{v}_x$ and $\mathbf{v}_y$ can be computed as

$$x = ((\mathbf{n} \times \overrightarrow{\mathbf{pm}}) \times \mathbf{n}) \cdot \mathbf{v}_x \tag{4.10}$$

and

$$y = ((\mathbf{n} \times \overrightarrow{\mathbf{pm}}) \times \mathbf{n}) \cdot \mathbf{v}_y. \tag{4.11}$$

Fig. 4.5(a) and 4.5(b) show geodesic circles on two surface patches of a synthetic head model. The corresponding point fingerprints are illustrated in Fig. 4.5(c) and 4.5(d). The fingerprint in Fig. 4.5(c) is not complete because of the step discontinuity caused by self occlusion. Fig. 4.5(e) plots the radius variation of the third pair of contours, in which both signals are periodic and one is a translated version of the other. Similarly the normal variation is plotted in Fig. 4.5(f), which is the dot product of normal vectors between the center point and points on the geodesic circle.

Not only are the fingerprints discriminating themselves, but they also can carry other features. The key is to use contours instead of a 2D image. The projection on the tangent plane, which is not a one-to-one mapping, may cause many surface points to be mapped to the same pixel in the fingerprint. In that case, each pixel cannot be allowed to carry features of different surface points. When the information on contours is stored, each point in a certain contour corresponds to one surface point, although the contours may intersect each other on the tangent plane. Thus the points in the contours can be made to carry features of the surface points. This is one advantage of using fingerprints rather than some previous works that project the whole surface patch on the tangent plane. The other advantage of the proposed fingerprint is that the comparison of several contours is much more efficient than the comparison of a pair of

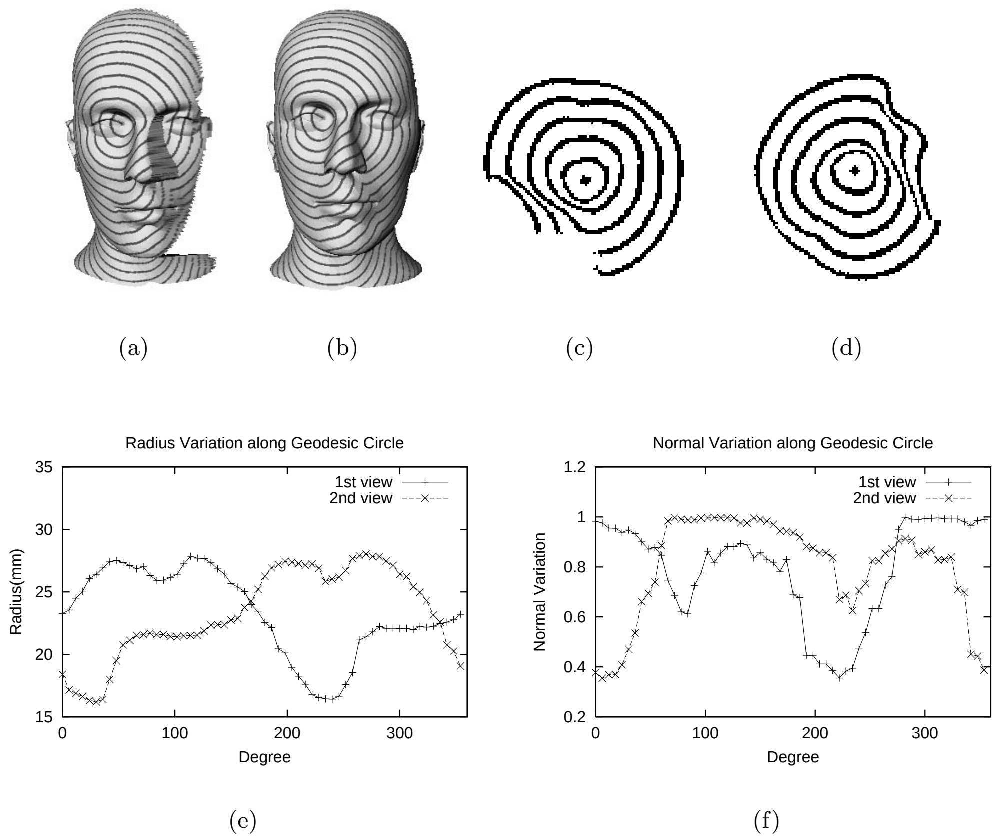


Figure 4.5: Definition of point fingerprint. (a)(b) Geodesic circles around the same point on two surface patches. (c)(d) Corresponding point fingerprints. (e)(f) Radius and normal variations from 0 to 360° along the third pair of contours.

images.

4.3 Candidate Point Selection

Because it is time consuming to compare all pairs of points in two surfaces, and points in the flat area whose fingerprints are like concentric circles provide little information in the point matching, we need to choose a meaningful set of points to compare. Various previous works [20] argued that all point pairs should be compared in the case of free-form surface matching [9], where the surface may not have easily detectable landmark features such as edges and vertices. In this work, some feature points can be extracted even for the free-form surfaces as long as a sufficiently large neighborhood is considered. To operate locally on the triangle mesh, most previous works only considered the neighborhood as a simplex [109], or some nearest points obtained by KD-tree implementation based on the Euclidean measure. It is suggested that using the geodesic measure to define the neighborhood is a better way because the resulting neighborhood is independent of the surface sampling resolution.

Although the most popular feature point extraction method is to find high curvature points, the estimation of stable and accurate curvature values on the discrete surface is difficult. In this research, a novel method is proposed to efficiently extract candidate points.

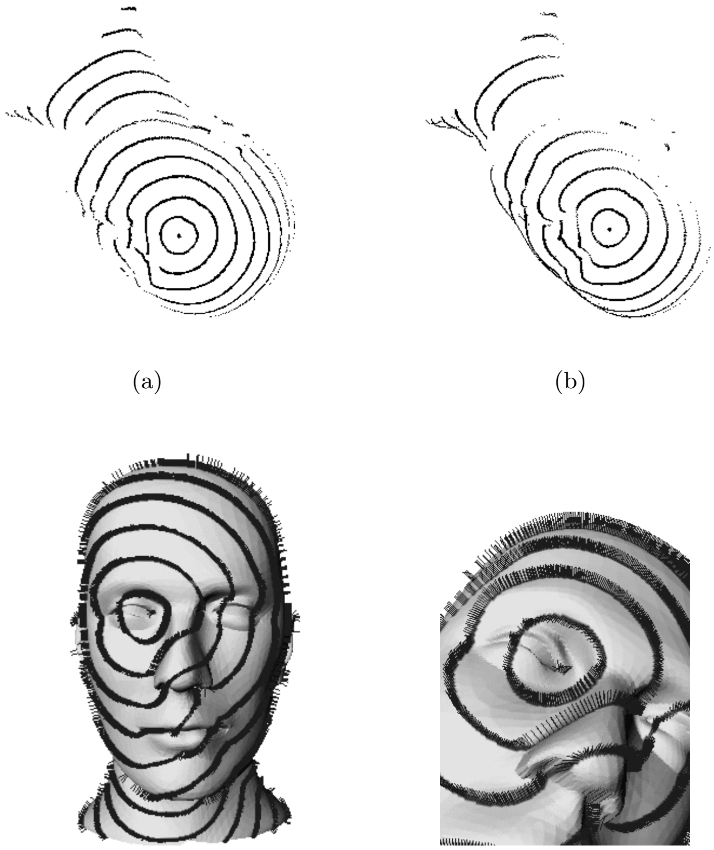
Considering that a point-of-interest has discriminating fingerprints, candidate points that result in irregular contour shapes in the fingerprints are found. The irregularity measure is defined by the ratio of the maximum radius to the minimum radius for a certain contour in its fingerprint. The irregularity measures for points on a planar region or a sphere are close to 1. On the other hand, points of interest have irregularity measures much greater than 1. Only one contour is used, and the size of the geodesic circle generating the contour depends on the type and size of the surfaces under matching. It is reasonable that relatively large contours are used for free-form surfaces or large scale surfaces. The candidate point is labeled if the irregularity measure is larger than a prespecified value. The fingerprints of the extracted candidate points are then compared to find correspondences.

The complexity of candidate point selection is $O(N_1 N_2 \log(N_2))$, where N_1 is the number of points in the surface mesh and N_2 is the number of neighboring points considered for each point. Typically, the geodesic radius of the neighborhood is three to five times larger than the average edge length of the triangle mesh.

4.4 Feature Matching

Some candidate points are located near the surface boundary. Although the fingerprint contours of boundary points may not be closed, they still contain useful information. Therefore, boundary points are not discriminated from other points in the matching process. In our work, the whole surface is marched to create the fingerprint for each candidate point. Fig. 4.6(a) and 4.6(b) show examples of two fingerprints.

From various features a fingerprint can carry, we exclusively use the contour radius



(a) (b)

(c) (d)

Figure 4.6: Global fingerprint and normal variation. (a) and (b) show the same 3D point fingerprint on a head model from two different views, which were obtained by marching the whole surface. (c) Normal variation on the geodesic circles. (d) A zoom view of (c).

variation and the normal variation for surface matching. Fig. 4.6(c) and 4.6(d) plot the normal variation along the geodesic circles. On each contour of the fingerprint, we sample with an incremental angle of $2\pi/K$ to represent the whole contour. Because each surface may have L candidate points and each candidate point fingerprint may have M contours, we used a three dimensional $(L \times M \times K)$ data structure to store the information for each surface. In the experiments, $L \approx 100$, $M < 20$, and $K = 30$ were used.

The fingerprints of an identical point from different views match with a 2D rotation, and the samples along each contour are periodic. The following formula is used to compute R_{ij} which is the dissimilarity measure between the ith candidate point on the first surface and the jth point on the second surface. The formula is similar to the form of cross correlation:

$$R_{ij} = \min_{l=1}^{K} \Big[\sum_{m=1}^{M} \sum_{k=1}^{K} (\mathbf{n}_{1,i,m,k} \cdot \mathbf{n}'_{1,i} - \mathbf{n}_{2,j,m,k+l} \cdot \mathbf{n}'_{2,j})^2 \Big], \tag{4.12}$$

where $\mathbf{n}_{1,i,m,k}$ is the normal at the kth point on the mth contour of the ith fingerprint from the first surface and $\mathbf{n}'_{1,i}$ is the normal at the center point of the ith fingerprint from the first surface, and similarly for $\mathbf{n}_{2,j,m,k}$ and $\mathbf{n}'_{2,j}$ from the second surface. The ith candidate point in the first surface and the jth candidate point in the second surface correspond if

$$j = arg \, \underset{k}{min} R_{ik}, \tag{4.13}$$

and R_{ij} is below a threshold. The contour radius variation is similarly used to confirm the correspondences.

4.5 Application to Surface Registration

For a given pair of surfaces, the algorithm for surface registration using point fingerprint works as follows.

Algorithm 4.2 (3D Registration by Point Fingerprint)

1. *Extract candidate points in both surfaces.*

2. *Generate fingerprint for every candidate point.*

3. *Find corresponding points by fingerprint matching.*

4. *Compute a coarse rigid transformation using Horn's method.*

5. *Apply Iterative Closest Point (ICP) to get a refined transformation.*

Previous sections present the first three steps. Coarse registration and ICP refinement are discussed in this section.

4.5.1 Coarse Registration by Fingerprint Comparison

After point correspondences are established, a coarse registration between surfaces can be solved. Horn [41] discussed how to compute the coordinate transformation from N pairs of corresponding points based on quaternion and orthonormal matrix respectively, which is also called the 3D-3D absolute orientation problem. This problem was also discussed by several other works [34], but with the same results. Assume N pairs of points are represented by $\mathbf{p}_1, ..., \mathbf{p}_N$ and $\mathbf{q}_1, ..., \mathbf{q}_N$, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and the translation vector. $\mathbf{p}_n$ and $\mathbf{q}_n$ are related by

$$\mathbf{p}_n = \mathbf{R}\mathbf{q}_n + \mathbf{t}, \quad n = 1, ..., N. \tag{4.14}$$

To determine $\mathbf{R}$ and $\mathbf{t}$, a constrained least-squares problem is set up. The function to be minimized is $\sum_{n=1}^{N} \|\mathbf{p}_n - (\mathbf{R}\mathbf{q}_n + \mathbf{t})\|^2$ subject to the constraint that $\mathbf{R}$ is a rotation matrix. Once $\mathbf{R}$ is known, the translation can be obtained directly as

$$\mathbf{t} = \overline{\mathbf{p}} - \mathbf{R}\overline{\mathbf{q}}, \tag{4.15}$$

where

$$\overline{\mathbf{p}} = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \mathbf{p}_n \text{ and } \overline{\mathbf{q}} = \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \mathbf{q}_n.$$

Let

$$\mathbf{B} = (\mathbf{b}_1\mathbf{b}_2\mathbf{b}_3), \tag{4.16}$$

where

$$\mathbf{b}_k = \sum_{n=1}^{N} (p_{nk} - \overline{p}_k)(\mathbf{q}_n - \overline{\mathbf{q}}). \tag{4.17}$$

If the singular-value decomposition of $\mathbf{B}$ is

$$\mathbf{B} = \mathbf{UDV}, \tag{4.18}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal and $\mathbf{D}$ is diagonal matrix, then

$$\mathbf{R} = \mathbf{V}'\mathbf{U}', \tag{4.19}$$

where $\mathbf{U}'$ and $\mathbf{V}'$ are the transpose of $\mathbf{U}$ and $\mathbf{V}$.

4.5.2 ICP Refinement

Due to the nature of the discrete sampling, the corresponding points cannot be exactly the same. Therefore, the registration almost always has a certain amount of error which may affect the appearance of the surface reconstruction. The ICP algorithm can be used to refine the registration results. The ICP algorithm is very effective to register two surfaces with fairly good initial pose estimation. This algorithm has been widely used to refine the coarse registration result obtained from either manual point matching or automatic point matching. Since the ICP algorithm was first introduced by Besl and McKay [7] and Chen and Medioni [19], many variants have been proposed in the literature [52, 56, 68, 95, 96]. A recent review and comparison of these variants was done in [73]. The basic ICP algorithm is given in the following.

Algorithm 4.3 (Basis ICP Algorithm) *The following loop is executed until a pre-*

specified condition is satisfied, for example the pose estimation change between two iterations is smaller than a certain threshold.

1. *Select a set of points in one surface.*

2. *Find their nearest corresponding points in the other surface.*

3. *Compute the pose estimation based on these correspondences.*

4. *Go back to Step 1.*

Variants of the basic algorithm reside in:

1. How to select the set of points, such as sampling randomly or uniformly.

2. How to find corresponding points, such as using a KD-tree search or projection.

3. Using different rejection criteria to remove certain correspondences to increase robustness.

4. Using different error metrics to estimate the pose.

The ICP algorithm in this work is similar to the one used by Turk and Levoy [96]. The point correspondences for the points on the mesh boundary are discarded, and a threshold is used to reject matched points which are far apart. Experimental results show that the modified ICP algorithm works well and demonstrates a significant improvement on the coarse registration using fingerprint methods.

Experimental results of point fingerprint-based surface registration are presented in Section 7.2.

Chapter 5

Surface Reconstruction from Multi-View Range and Color Images

This chapter presents research on surface reconstruction using range and color images from multiple viewpoints. A mesh-based algorithm and a volume-based algorithm for surface reconstruction are implemented and compared. This research introduces texture fusion into these two methods to generate textured 3D models. Surface reconstruction with or without space carving is discussed. In the model post-processing stage, volumetric smoothing driven by mean curvature flow, and hole filling by volumetric interpolation are presented. Automatic hole filling makes it possible to generate watertight models when the surfaces of the object are not completely scanned.

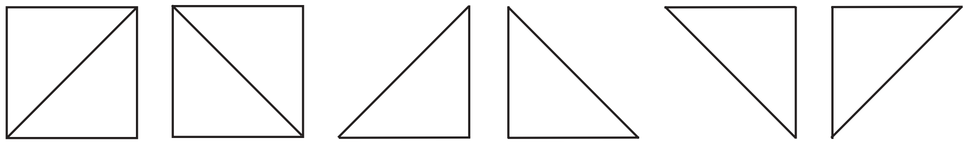


Figure 5.1: Triangulation of a single range image. There are six possible configurations for the creation of triangles from four neighboring points.

The literature review of multi-view surface reconstruction using range images is given in Section 2.3. In Section 5.1, a mesh integration approach is presented. Section 5.2 discusses the implicit surface integration approach and model post-processing methods. Experimental results are presented in Section 7.3.

5.1 Mesh-Based Surface Integration

The mesh-based surface integration includes three steps: single view triangulation, removing less confident triangles, and linking the gaps.

5.1.1 Triangulating a Single Range Image

Most laser range scanners employ a spherical coordinate system, and the viewing volume is restricted by the horizontal and vertical limits. The range measurements are stored as a 2D grayscale image, from which the 3D coordinates can be recovered when the calibration parameters are known. The initial triangulation considers four neighboring points and the six possible connections [74] as shown in Fig. 5.1.

When two neighboring range measurements differ by more than a threshold, there

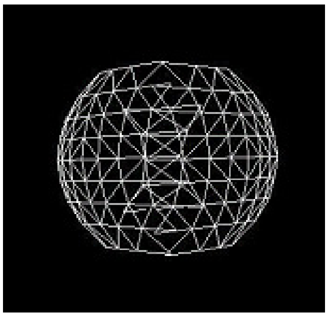


Figure 5.2: Two registered and overlapping meshes.

is a step discontinuity. The threshold is determined by the average range value and the sampling resolution. If a discontinuity is present, a triangle should not be created. Triangles created across step discontinuities usually have very small internal angles, which cause problems in searching for neighboring triangles and identifying overlapping regions. Among the four points, only the ones that are not along discontinuities are considered. If three of the four points satisfy this condition, a triangle will be created, such as one of the last four cases shown in Fig. 5.1. If none of the four are along a discontinuity, two triangles will be created and the common edge will be the one with the shortest 3D distance, as illustrated by the first two cases shown in Fig. 5.1.

5.1.2 Removing Triangles in Overlapping Regions

Fig. 5.2 shows two registered and overlapping meshes from simulated range shots of a sphere. The overlapping region detection is based on back projection. Knowing the calibration model, the 3D points can be projected back to a 2D reference frame. Given a new triangle mesh, each triangle of the existing mesh is projected onto the new 2D

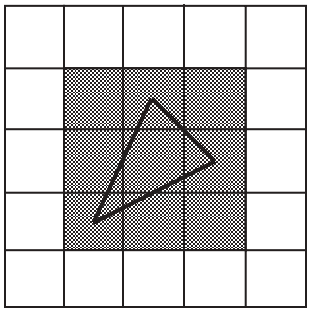


Figure 5.3: Bounding box of a triangle.

reference frame, which is the image plane of the new range image. If the 2D projection of the triangle is out of the reference frame, the triangle is not in the viewport of the new range shot and will be left unchanged. If the projection is inside the new reference frame, it is necessary to check whether this triangle overlaps the new mesh. A test is performed to see whether the triangle is facing the viewpoint of the new range shot. If the dot product of the triangle normal with one of the three measurement rays (i.e., the rays from the viewpoint to each of the triangle vertices) is positive, the triangle is front-facing. The bounding rectangle of the projected triangle is computed, as shown in Fig. 5.3. To check if a front-facing triangle in the existing mesh overlaps any triangle in the new mesh, the triangles in the new mesh whose projections are in the bounding rectangle are considered. In Fig. 5.4, conditions of 2D triangle intersection, which can be detected by edge intersection, are illustrated. An efficient algorithm for checking 2D line intersection is described by O'Rourke [61].

When all the triangles in the bounding rectangle have been checked and there is overlapping, either the triangle in the existing mesh or all the overlapping triangles in

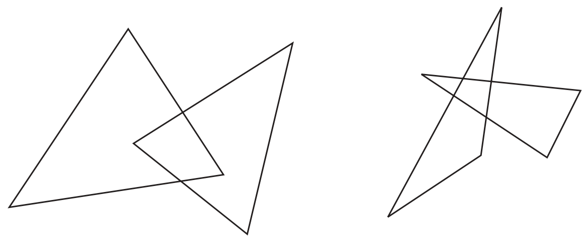


Figure 5.4: Intersecting triangles.

the new mesh are deleted. To keep the best measurements, a measurement confidence is computed for each triangle. Similar to previous works [21, 66, 96], the confidence is defined as the dot product of the normal of the triangle and the normalized viewing direction. This concept matches the range scanners' working principle: the measurement accuracy depends on the incident angle. The average confidence of all the overlapping triangles in the bounding rectangle is computed. If this average is larger than that of the triangle from the existing mesh, the triangle in the existing mesh is deleted. Otherwise, all the overlapping triangles in the bounding box are deleted. Overlapping in 2D does not always imply overlapping in 3D. A threshold is set to determine whether two patches overlapping in 2D are from the same area of the object. If the distance between two triangles is smaller than the threshold, they are considered to be the representations of the same surface patch. The threshold is set according to the accuracy of the range scanner and the measured distance.

Since there is always registration error and noise in the range data, registered surface patches cannot be aligned perfectly. The triangles may not be overlapping in one view while they are in another view. This case is illustrated in Fig. 5.5. As the overlap

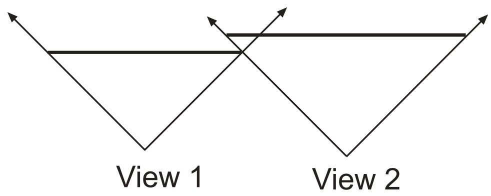
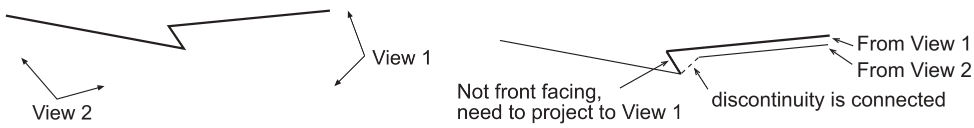


Figure 5.5: View-dependent overlapping. From View 1 there is overlapping, but not from View 2.



(a) (b)

Figure 5.6: Checking overlap for back-facing triangles.

detection is view dependent, the test is performed not only in the new viewport, but also in the existing viewports.

In general, triangles in the existing mesh that are not front-facing do not need to be checked for overlapping. However, a special case must be considered, as shown in Fig. 5.6. When the step discontinuity is smaller than the threshold, two points along a discontinuity are connected. But when the real surface is measured, the connection may need to be removed. For example, in Fig. 5.6(a) a surface is measured from two different views. The dashed line shown in Fig. 5.6(b) is created from View 2, which is not correct if it is seen from View 1. Therefore, the triangles indicated by the dashed line should be removed.

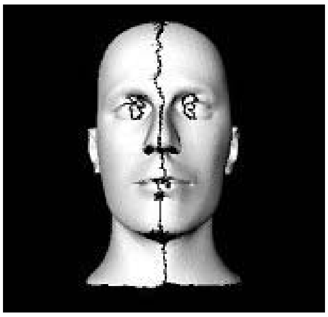


Figure 5.7: Gap between surfaces after deleting overlapping triangles.

In Fig. 5.7, a two-view image of a head model is shown after deleting overlapping triangles. Two views are taken, one at each side of the model. The most confident measurements are kept.

5.1.3 Linking the Mesh Patches

To link the gaps between the mesh patches, candidate triangles are labeled to combine with other points for building new triangles. These candidate triangles, which are on the mesh boundaries, are called active triangles. If one of a triangle's neighbors has been deleted, it is marked as an active triangle.

An active triangle may have one, two, or even three active edges that need to find a point to build a new triangle. For one active edge, some neighboring points are found as candidates. A KD-tree [58] is employed for candidate point searching. The validity of each candidate is then checked, and the best one is chosen to create a new triangle that does not intersect the existing triangles. For all the valid candidate points, the one that faces the active edge with the largest angle is the best. If the new triangle has a common edge with any existing triangle, both triangles will update the neighboring information. After linking all the gaps, a global mesh representation of the surface is obtained.

5.1.4 Texture Mapping

To produce a realistic scene, the color images should be fused with the range images as a texture map. Generally the texture map can be of any type, such as color or thermal images. In the simulations, both range images and color images are captured from the exact same view (and are therefore automatically registered). Each triangle in the complete mesh is associated with the texture image corresponding to the range image from which it was generated. The triangles seaming the meshes are associated with the texture image corresponding to the range image where two of the three triangle vertices lie. Each triangle is projected onto its 2D reference frame to find the 2D texture coordinates. The final result is a 3D textured scene.

In the experimentation, the mesh integration method performs well on the synthetic data. However, the zippering process is not as robust as the volumetric integration approach against sampling noise and registration error which always exist in real range data. The assumption of the knowledge of the calibration parameters is another limitation of this approach.

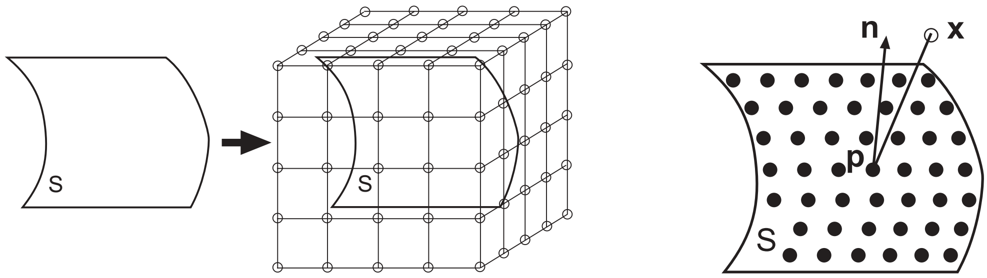
5.2 Volume-Based Surface Integration and Post-Processing

The implicit surface integration method and surface post-processing techniques presented in this section are volume-based.

5.2.1 Implicit Surface Integration

The proposed implicit surface integration approach is an extension of Hilton's work [36]. Signed distance fields are generated in a volumetric grid. The value at each voxel is computed as the signed distance to the surface mesh and updated when a new mesh is integrated. Curless and Levoy [21] also integrated multiple surfaces based on the fusion of implicit surfaces. In their work, space carving is implemented to generate watertight models. The advantages of space carving are watertight model reconstruction and the ability to remove outliers. However, Curless' method is not proper for the reconstruction tasks when the scene contains a lot of deep step discontinuities and when complete scanning of the scene is impossible. Space carving also assumes to know the range scanner's calibration model because the carving is based on back projecting each voxel onto the image plane. In this context, Hilton's method is more flexible because it assumes no knowledge about the scanner, and the only inputs to the algorithm are triangle meshes that are not even necessarily from range scans. Curless computes the signed distance by approximation along the viewing direction. Hilton's method computes the true distance from the voxel to the surface mesh.

This research incorporates color image integration into Hilton's method to generate



(a) (b)

Figure 5.8: Generating signed distance field in a volumetric grid. (a) Construction of a volumetric grid containing the surface. (b) Calculation of signed distance field at a voxel $\mathbf{x}$ by $\overrightarrow{\mathbf{xp}} \cdot \mathbf{n}$, where $\mathbf{p}$ is the closest points on the surface to $\mathbf{x}$.

texture integrated models. The algorithm is described as follows.

Algorithm 5.1 (Implicit Surface Integration)

1. *A volumetric grid is initialized to contain the region of interest, as illustrated in Fig. 5.8(a).*

2. *The voxels near the surface mesh are located, and their indices are put into a queue.*

3. *A KD-tree data structure is built for all the vertices of the mesh.*

4. *For each voxel* $\mathbf{x}$ *in the queue, its nearest point* $\mathbf{p}$ *on the surface mesh is obtained from the query of the KD-tree. The signed distance between the voxel and the surface mesh is computed as the dot product of* $\overrightarrow{\mathbf{xp}}$ *with* $\mathbf{n}$*, where* $\mathbf{n}$ *is the normal vector at* $\mathbf{p}$*, as illustrated in Fig. 5.8(b). If the nearest point is on the boundary of the mesh, the signed distance is discarded due to lack of information. The*

2D texture coordinate of $\mathbf{x}$ *is set the same as that of* $\mathbf{p}$*, and the pointer to the corresponding texture file is stored. When a new surface is integrated, the signed distance is updated by the weighted average according to the confidence of the measurement.*

5. *The final surface mesh is extracted using the marching cube algorithm [54]. The texture coordinate of each vertex on the extracted mesh is set the same as its nearest voxel.*

If it is necessary to generate a watertight model, a space carving process still can be added as long as the range scanner calibration is known.

5.2.2 Volumetric Postprocessing by Mean Curvature Flow

After reconstruction, the surfaces are post-processed by mean curvature flow using the level set method [77] in a volumetric grid. In this post-processing stage, mean curvature flow is chosen instead of the area-decreasing flow because the mean curvature can be easily computed for the implicit surface, and volumetric deformation can effectively remove the outliers. Similar mean curvature flow implementations can be found in [101, 112].

The mean curvature flow process also involves five steps in the volumetric integration process except that there is only one mesh involved and the signed distance field is modified according to the mean curvature value. The signed distance field value ψ at each voxel is updated as

$$\psi^{(k+1)} = \psi^{(k)} + \epsilon H, \tag{5.1}$$

where ϵ controls the speed of surface deformation, and the mean curvature H is estimated as

$$H = \frac{1}{(\psi_x^2 + \psi_y^2 + \psi_z^2)^{3/2}} \begin{bmatrix} (\psi_{yy} + \psi_{zz})\psi_x^2 + (\psi_{xx} + \psi_{zz})\psi_y^2 \\ +(\psi_{xx} + \psi_{yy})\psi_z^2 \\ -2\psi_x\psi_y\psi_{xy} - 2\psi_x\psi_z\psi_{xz} - 2\psi_y\psi_z\psi_{yz} \end{bmatrix}, \tag{5.2}$$

where ψ_x, ψ_y, and ψ_z are the first order derivatives along three coordinate axes, and ψ_{xx}, ψ_{yy}, ψ_{zz}, ψ_{xy}, ψ_{yz}, and ψ_{xz} are the second order derivatives. These derivatives are estimated in a $5 \times 5 \times 5$ window.

5.2.3 Automatic Hole Filling

Although it is easy to change the pose of an object and scan most of the surface in small parts reconstruction, sometimes it is impossible for the laser to cover every corner due to the complexity of the object. This results in holes in the reconstructed model. Most of these holes can be automatically filled. Filling the holes on a triangle mesh is conducted by filling the holes on the implicit surface in the volumetric grid. Fig. 5.9 shows a slice of the volumetric grid in which a curve with a gap represents a surface with a hole. In the fourth step of the integration process, the signed distance values

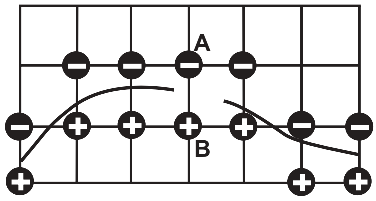


Figure 5.9: Hole filling illustrated in a slice of volumetric grid. The hole can be filled if the signed distance values at voxels A and B are computed.

of the voxels whose nearest points are on the mesh boundary are discarded. However, these values are useful in the hole filling process. In Fig. 5.9, voxels A and B have their nearest neighbors on the mesh boundary. The signed distance field has a gap between A and B. The gap can be filled by computing the signed distance values at A and B as if the surfaces are extended. Once the signed distance field describing the surface is complete, the extracted triangle mesh is also complete with the holes filled.

The reconstructed surface usually uses a lot of triangles, making model rendering and further operation extremely slow. The number of triangles are reduced by using a mesh simplification algorithm. Among the various simplification algorithms, Garland's algorithm [30], which is based on the Quadric Error Metrics, is implemented.

The volume-based approach is robust to surface noise and registration error. There is no assumption of the calibration parameters. It is also convenient to post-process the surface in a volumetric grid. However, the volume initialized for reconstruction is not easy to change during the integration process. Overall, the volume-based approach is more suitable for surface reconstruction from range data than are the mesh-based approaches.

Chapter 6

Surface Mesh Segmentation

This chapter presents a watershed-based approach to segmenting the surfaces represented by triangle meshes. The proposed approach includes a robust method for the edge strength computation at each vertex and an accurate segmentation method based on fast marching watershed. Edge strength constructs a piecewise continuous height map on the surface, which is used for watershed segmentation. Compared to previous watershed-based mesh segmentation approaches, the proposed algorithm is able to segment the surface more accurately by generating a lower complete image using the geodesic erosion.

The literature review of surface segmentation and watershed-based segmentation is given in Section 2.4. This chapter starts with an introduction of a typical watershed algorithm in Section 6.1. Section 6.2 introduces the edge strength map on a triangle mesh. The fast marching watershed is proposed in Section 6.3. Experimental results are presented in Section 7.4.

6.1 A Typical Watershed Algorithm

As a primary tool of mathematical morphology for image segmentation, watershed-based segmentation has been studied for over twenty years [12]. A 2D grey scale image can be considered as a 3D landscape with the third dimension being the grey level. This image is defined on a 2D regular grid, which is an unweighted graph. The image can be segmented by the watershed of the landscape. A typical watershed algorithm is as follows.

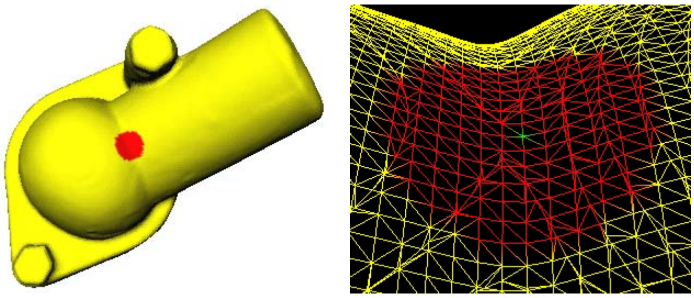
Algorithm 6.1 (A Typical Watershed Algorithm)

1. *(Minima Detection) Find local minima and assign a unique label to each minimum.*
2. *(Descending) Allow all unlabeled vertices to descend and join to labeled regions.*
3. *(Region Merging) Merge regions whose watershed depths are below a preset threshold, and finally, relabel all the regions.*

Vincent and Soille [98] proposed another version of this algorithm by performing the first two steps based on the immersion simulation.

6.2 Construct a Height Map of Edge Strength

To apply watershed segmentation on a triangle mesh, a height map based on the high pass filtering needs to be computed. The edge strength height map in this research is



(a) (b)

Figure 6.1: Geodesic neighborhood on a waterneck model. (a) The neighborhood contains 200 vertices. (b) The zoomed view of (a).

more robust to noise than Mangan's total curvature height map [55]. Mangan computed the total curvature value at each vertex only based on an umbrella neighborhood, so the result is very sensitive to the surface noise and mesh resolution.

In many cases, 3D surfaces are represented by very dense triangle meshes. The transition of geometric features is not obvious in a small neighborhood. For robust edge detection, the neighborhood size often needs to be very large. In Fig. 6.1, a reasonably large neighborhood that is able to detect weak crease edges contains 200 vertices! This neighborhood is much larger than the umbrella neighborhood used in [55] and in most computer graphics applications.

The edge strength computation in this work, which was presented in Section 3.3.2, involves a geodesic neighborhood with flexible size and eigen analysis of surface normals. The robust edge strength estimation makes the watershed segmentation work well on noisy and dense triangle meshes.

6.3 Applying the Fast Marching Watershed on a Triangle Mesh

The swiftest descending path (SDP) from a pixel (x, y) is defined as a finite succession of connected pixels such that each pixel is not higher than its predecessor and is one of its lowest neighbors. The step of descending in Algorithm 6.1 is a process of finding the SDP.

The watershed-based algorithm has the plateau problem, where the SDP is undefined for pixels inside the plateau. A common approach to plateau elimination transforms the image into a lower complete image so that the SDP is defined on every pixel. The transformation raises the plateau according to the geodesic distance from a pixel to the plateau boundary. The pixel in the center of the plateau is higher than the pixel near the boundary. An alternative method is geodesic erosion by directly extending the SDP from the boundary pixels inside the plateau. The plateau is eroded starting from the boundary with the same speed until the plateau disappears. The SDP is in the opposite direction of the erosion process.

Geodesic erosion was introduced into image analysis by Lantuejoul and Maisonneuve [51]. They proposed important concepts: geodesic zone of influence and skeleton by influence zones. Vincent and Soille [98] used the breadth-first algorithm for geodesic erosion on a 2D hexagonal grid. Bleau [11] described an algorithm based on idempotent geodesic transform implemented on square or hexagonal grids of any dimensions. However, the geodesic distance computed by using these methods is not accurate because the shortest path is restricted along the edges.

Geodesic erosion for partitioning the plateaus is also necessary on a 3D triangle mesh. In [55], Mangan and Whitaker first reported the research on watershed-based segmentation of triangle meshes. However, their algorithm classifies a whole plateau into a neighboring region instead of segmenting it. The breadth-first algorithm can be used for finding the shortest path only on an unweighted graph. For a weighted graph, such as a 3D triangle mesh, a priority queue-based algorithm must be used. This research applies the fast marching method for geodesic erosion on a triangle mesh because it is more accurate than Dijkstra's algorithm [79].

Let a graph $G = (V, E, H)$ denote a triangle mesh consisting of a set V of vertices, a set E of edges, and a height map H defined on the vertices. $N_G(v)$ represents the set of vertices that are in the umbrella neighborhood of v. $v_{.d}$ and $v_{.s}$ are used to represent the geodesic distance and status at a vertex v, used for geodesic erosion. The status of each vertex may have one of three conditions. *Inside* vertex represents a vertex inside the current geodesic neighborhood. *Front* vertex represents a vertex on the propagation front. *Front* vertices are stored in a priority queue Q using a heap data structure and keyed by the geodesic distance. The status of other vertices is *outside*. $v_{.n}$ points to the next vertex in the SDP. $v_{.l}$ is the label denoting the region to which the vertex belongs. $v_{.h}$ is the associated height map value at v.

The fast marching watershed algorithm has four steps as summarized in the following algorithm.

Algorithm 6.2 (Fast Marching Watershed)

1. *(Minima Detection) Extract flat regions. Assign unique labels to minima. Vertices on ramp and plateau boundaries get $v_{.n}$.*

2. *(Geodesic Erosion) Propagate $v_{.n}$ from plateau boundaries toward inside.*

3. *(Descending) Label non-minima vertices, directed by $v_{.n}$.*

4. *(Region Merging) Merge all shallow regions into neighboring regions.*

The four steps in Algorithm 6.2 are introduced in Sections 6.3.1 to 6.3.3.

6.3.1 Minima Detection

All vertices are initially considered on the ramp and denoted by -1. From a vertex v, a set of connected vertices with the same height map value as $v_{.h}$ are extracted and stored in a vector A. A FIFO queue-based breadth-first algorithm, which has been used for minima detection in 2D watershed-based segmentation in [72], is applied. Flat regions are extracted and classified into minima and plateaus. A flat region is considered as a minimum if all adjacent vertices have height map values greater than or equal to that of the region. Otherwise, the flat region becomes a plateau if it contains more than one vertex. Minima regions are assigned unique labels starting from 0. Vertices on the plateaus are assigned a -2 label. Vertices on ramp and plateau boundaries obtain their $v_{.n}$ simply by looking for the vertex with the smallest height map value in $N_G(v)$. The vertices on the plateau boundaries are put into a vector *bound* used for geodesic erosion.

A vector structure has three operations:

- $vector\_init(A)$ initializes a vector A.
- $vector\_add(A, v)$ adds a vertex v into A.
- $vector\_size(A)$ returns the size of A.

A FIFO queue structure has four operations:

- $fifo\_init(Q)$ initializes a FIFO queue Q.
- $fifo\_add(Q, v)$ adds a vertex v into Q.
- $fifo\_delete(Q)$ gets the first element of Q.
- $fifo\_empty(Q)$ checks if Q is empty.

The detailed algorithm for minima detection is as follows.

Algorithm 6.3 (Minima Detection)

#define Inside *0*
#define Outside *2*
for all $v \in G$ ***do***
    $v_{.s} \leftarrow 2$; $v_{.l} \leftarrow -1$; $v_{.n} \leftarrow null$
end for
$L = 0$
for all $v \in G$ ***do***
    if $v_{.l} = -1$ ***then***
        $vector\_init(A)$; $adj_{min} \leftarrow v_{.h}$; $v_{cur} \leftarrow v$; $fifo\_init(Q)$; $v_{.s} \leftarrow 0$
        loop
            $vector\_add(A, v_{cur})$; $min \leftarrow v_{cur.h}$
            for all $v_i \in N_G(v_{cur})$ ***do***
                if $v_{i.h} < min$ ***then***
                    $min \leftarrow v_{i.h}$; $v_{cur.n} \leftarrow v_i$

```
            end if
            if v_{i.h} < adj_{min} then
                adj_{min} ← v_{i.h}
            end if
            if v_{i.s} ≠ 0 and v_{i.h} = v_{cur.h} then
                fifo_add(Q, v_i); v_{i.s} ← 0
            end if
        end for
        if v_{cur.n} ≠ null then
            vector_add(bound, v_{cur})
        end if
        if fifo_empty(Q) then
            BREAK
        else
            v_{cur} ← fifo_delete(Q)
        end if
    end loop
    if adj_{min} ≥ v_{.h} then
        for all v_i ∈ A do
            v_{i.l} ← L
        end for
        L ← L + 1
    else if adj_{min} < v_{.h} and vector_size(A) > 1 then
        for all v_i ∈ A do
            v_{i.l} ← −2
        end for
    end if
    for all v_i ∈ A do
        v_{i.s} ← 2
    end for
  end if
end for
```

Fig. 6.2(a) shows a surface whose height map has four flat regions. A 2D slice of the height map is shown in Fig. 6.2(b). The flat regions are extracted by FIFO queue-based flooding from vertices A, B, C, and D. Solid arrows denote that the vertices have found $v_{.n}$. In this step, only the vertices on the ramp and plateau boundaries found $v_{.n}$, and the labeling result is shown in Fig. 6.2(b).

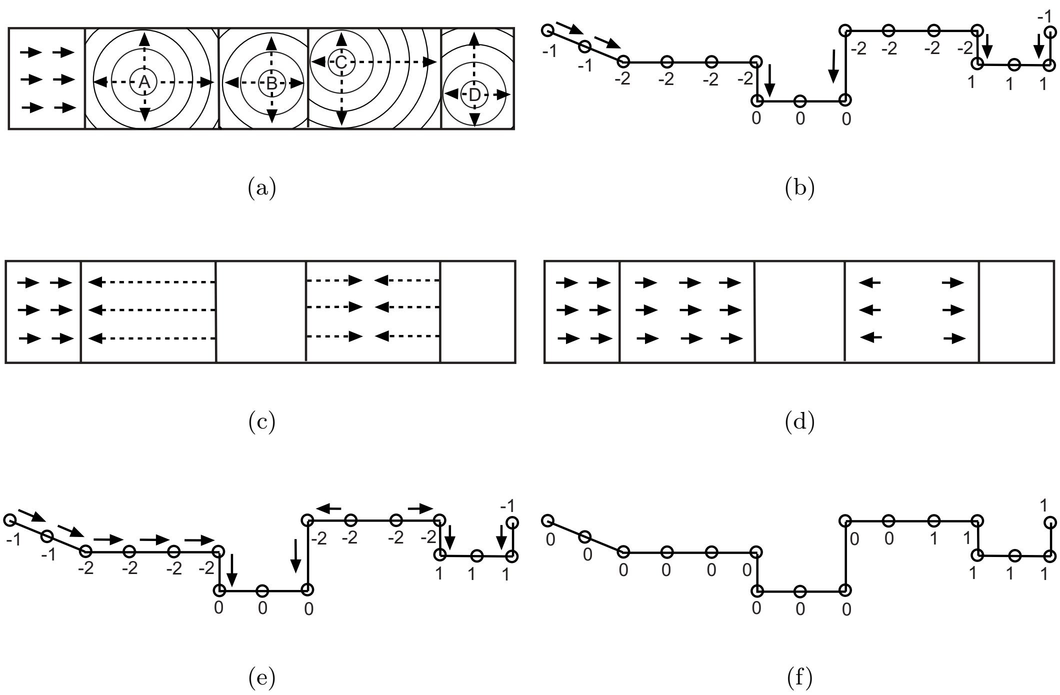


Figure 6.2: Fast marching watershed. (a) A surface patch whose flat regions are extracted by FIFO queue-based flooding. (b) A 2D slice of the height map, on which the minima are uniquely labeled and plateaus are marked as -2. Vertices on ramp and plateau boundaries get $v_{.n}$ represented by solid arrow. (c) Propagate $v_{.n}$ from the plateau boundary by geodesic erosion. (d) $v_{.n}$ at vertex on the plateau is opposite to the erosion direction. (e) All non-minima vertices get $v_{.n}$. (f) All vertices are labeled by tracing $v_{.n}$.

6.3.2 Geodesic Erosion

This step assigns $v_{.n}$ for vertices inside the plateaus. From the previous step, a list of vertices that are on the plateau boundaries are recorded in a vector *bound*. These vertices have been assigned $v_{.n}$. A geodesic erosion process propagates $v_{.n}$ from the plateau boundary toward the inside.

A priority queue-based heap structure used in the geodesic erosion has five operations:

- *heap_init*(Q) initializes a heap Q.
- *heap_insert*(Q, v) adds a vertex v into Q.
- *heap_delete*(Q) gets the first element of Q.
- *heap_changekey*(Q, v) re-sorts the queue after the key of a component v is changed.
- *heap_empty*(Q) checks if Q is empty.

The detailed algorithm is as follows.

Algorithm 6.4 (Geodesic Erosion)

#define Inside *0*
#define Front *1*
#define Outside *2*
if *vector_size(bound)=0* ***then***
    EXIT
end if
heap_init(Q)
for all $v \in$ *bound* ***do***
    $v_{.d} \leftarrow 0$; $v_{.s} \leftarrow 0$; *heap_insert*(Q, v)

```
end for
v_cur ← heap_delete(Q)
loop
   v_cur.s ← 0
   for all v_i ∈ N_G(v_cur) do
      if v_i.s ≠ 0 and v_i.h = v_cur.h then
         old_v_i.d ← v_i.d
         compute v_i.d
         if v_i.d < old_v_i.d then
            v_i.n ← v_cur.n
         end if
         if v_i.s = 2 then
            v_i.s ← 1; heap_insert(Q, v_i)
         else
            heap_changekey(Q, v_i)
         end if
      end if
   end for
   if heap_empty(Q) then
      BREAK
   else
      v_cur ← heap_delete(Q)
   end if
end loop
```

The dashed arrows in Fig. 6.2(c) represent the directions of geodesic erosion. The erosion seems to proceed in parallel on all plateaus because all the vertices on the plateau boundaries are put in one priority queue. Compared with the algorithm in [70], this proposed approach is more efficient because there is only one marching process. Solid arrows on the plateaus in Fig. 6.2(d) are opposite to the erosion direction, and they represent that the vertices inside plateaus obtain $v_{.n}$ after geodesic erosion. For a vertex v in the plateau, the geodesic distance $v_{.d}$ to the plateau boundary is computed from $v_i \in N_G(v)$, and $v_{.n}$ is defined as v_i that generates the smallest $v_{.d}$.

Geodesic erosion enables plateau segmentation. The surface shown in Fig. 6.3 is a

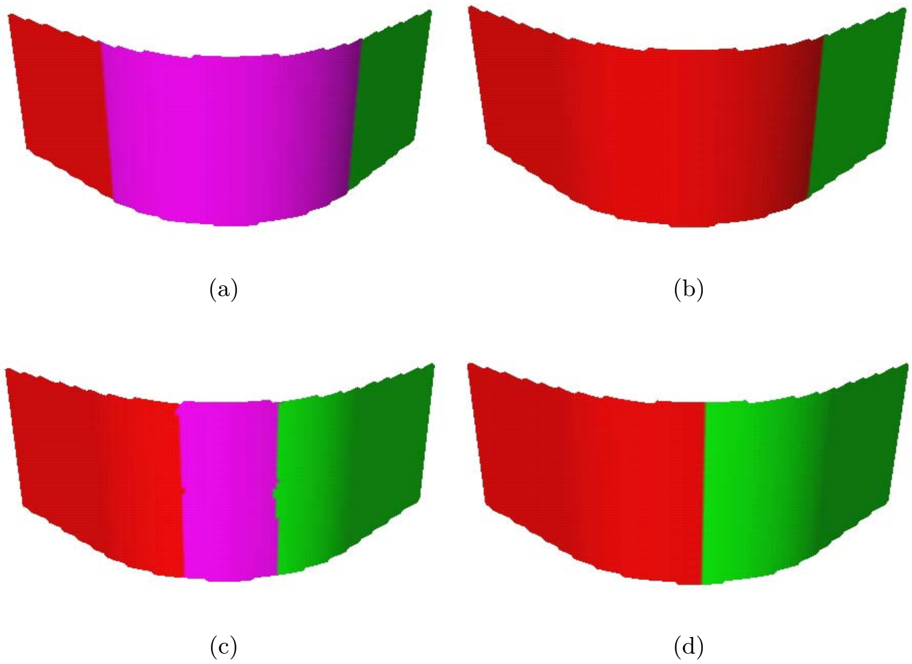


Figure 6.3: Segmenting a plateau area. (a) Initially labeled regions on a rounded corner, minima are in red and green and plateau is purple. (b) Mangan's algorithm groups the plateau into a neighboring region. (c) Geodesic erosion in progress by fast marching watershed. (d) Fast marching watershed equally divides the plateau region.

rounded edge appearing in many CAD models, formed by a piece of cylinder and two tangent planes. The height map on the curved region constructs a plateau. The labeling result after minima detection is shown in Fig. 6.3(a), where two planes are labeled as minima and shown in red and green. The plateau is the purple region. Mangan's algorithm [55] gives the result shown in Fig. 6.3(b) where the whole plateau is merged into a neighboring region. The fast marching watershed algorithm erodes the plateau from the boundary as shown in Fig. 6.3(c) and generates the correct segmentation in Fig. 6.3(d), where the boundary is exactly in the middle of the plateau.

6.3.3 Descending and Region Merging

Labeling vertices by descending is straightforward after $v_{.n}$ is defined on every non-minima vertex. This work applies a similar method of region merging used in Mangan's work [55]. Region merging is essentially a graph problem with the node being the individual region. The process is independent from the triangle mesh segmentation.

In the proposed algorithm, a larger region is favored over a smaller region. Although the real surface area is more accurate, the vertices number is simply used as an area measure. It was reported in [55] that the area-based metric penalizes the small area too much. This research confirms that report when the metric is used for all regions. However, if the metric is only applied on those relatively small areas, the area-based metric is very effective in avoiding over-segmentation, as shown in the experimental results.

In the surfaces reconstructed from range scanners, sometimes there exist outliers that have only a few vertices. These outliers can not be merged into other regions no matter what area penalization is applied because they are separated in 3D space. One way to avoid over-segmentation caused by outliers is to remove them before segmentation. Another way, which is used in this research, is to discard the regions whose areas are smaller than a given threshold after segmentation.

The proposed segmentation algorithm is applied to segment the surfaces reconstructed from the range data. Experimental results will be given in Section 7.4.

Chapter 7

Experimental Results

This chapter presents the experimental results of surface smoothing, registration, integration, and segmentation from Section 7.1 to 7.4. An application of the whole surface modeling and analysis framework is shown in Section 7.5.

7.1 Surface Smoothing

Fig. 7.1 shows raw data captured by the PERCEPTRON laser range scanner. The size of the original range image is 1024 by 1024 pixels. The PERCEPTRON scanner is able to scan objects in a range from 2 to 20 m. Besides random noise, measurement accuracy is also sensitive to the surface material. Fig. 7.2 shows the corresponding nonadaptive regularization results that are much smoother than the raw surfaces shown in Fig. 7.1.

Fig. 7.3(a) shows the result of a 3 by 3 median filtering conducted twice, which does not produce sufficiently smoothed surface. Additional median filtering provides

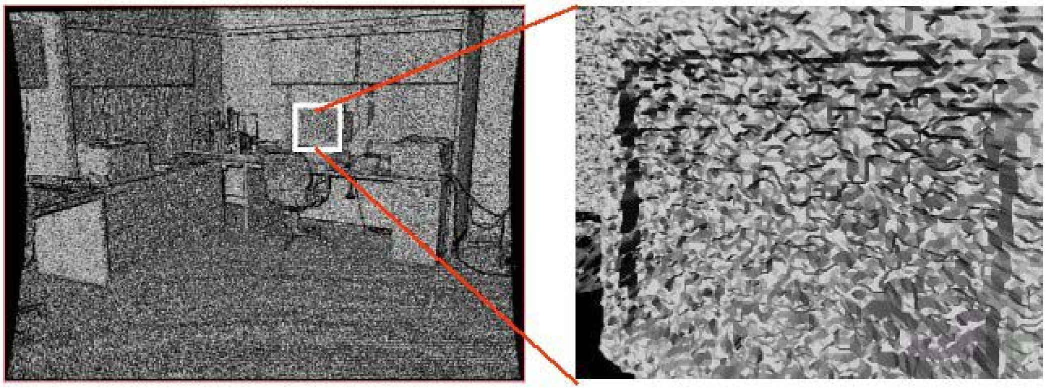


Figure 7.1: Raw range data (left) and zoomed portion (right). The image was taken by the PERCEPTRON range scanner. The size of the original range image is 1024 by 1024. The 3D model has 1,996,958 triangles.

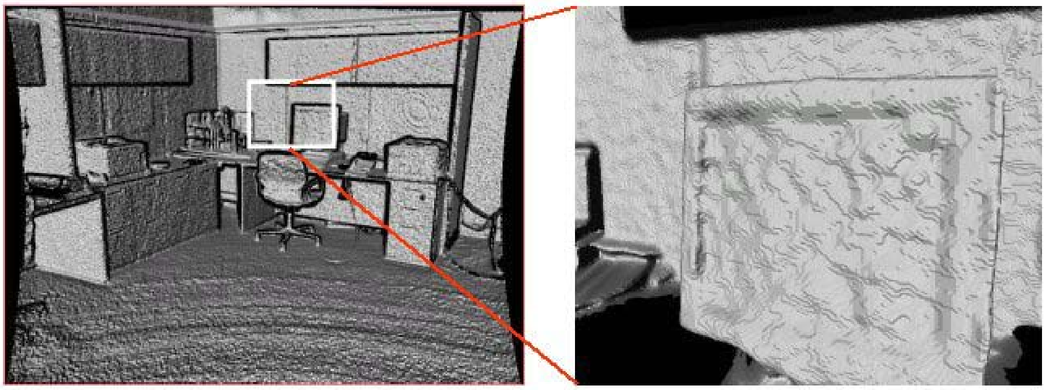
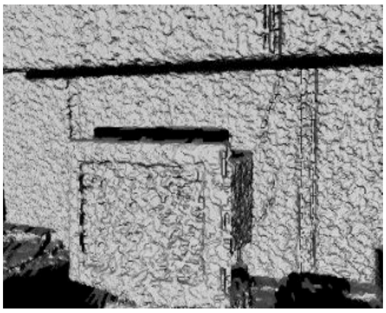


Figure 7.2: Range data regularization result (left) and zoomed portion (right). The smoothed image is obtained by 50 iterations of nonadaptive regularization using area-decreasing flow.

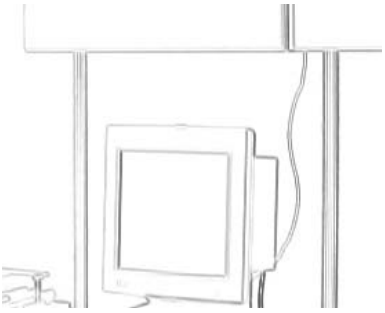
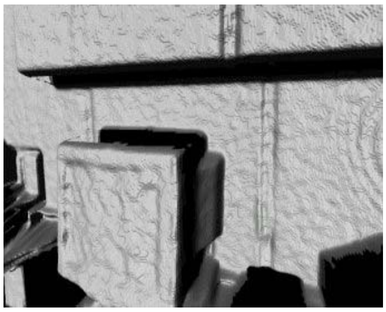
no discernible improvement. For fair comparison, results with a larger median filtering window are not included because the proposed algorithm is based on operation with a 3 by 3 window. Fig. 7.3(b) shows the regularization result using the simple 2D Laplacian smoothing term. Unstable results along edges are obtained, which coincide with the results reported in [10]. The edge map and 50 iterations of nonadaptive and adaptive regularization results are shown in Fig. 7.3(c)-(e), respectively. In the regularization, $w = 10^{-5}$, $\rho = 0.01$ and $\kappa = 0.5$ were selected. Note in Fig. 7.3(e), the wires on the cubicle wall behind the monitor which are preserved by the adaptive regularization. The adaptive regularization technique gives much better results than the median filtering method. In the experiments, central difference approximation makes the minimization more robust, and the regularization factor can be set to a large value to speed up the convergence.

Fig. 7.4 shows nonadaptively smoothed results of the surface mesh for synthetic data. The blocky-looking surfaces in Fig. 7.4(a) and (c) are caused by binary reconstruction using the marching cube algorithm [54]. Binary reconstruction means the voxel's status is either empty or occupied. The aliasing artifacts are caused by the discontinuous transition of the status. Fig. 7.4(b) shows the nonadaptively smoothed result of Fig. 7.4(a) after 6 iterations. Fig. 7.4(d) shows the nonadaptively smoothed result of Fig. 7.4(c) after 7 iterations.

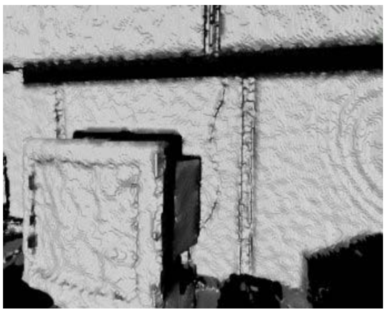
Fig. 7.5 shows the result of surface mesh smoothing using area-decreasing flow applied on a waterneck model scanned by the IVP RANGER Profiling System. Figs. 7.5(a)

(a) (b)

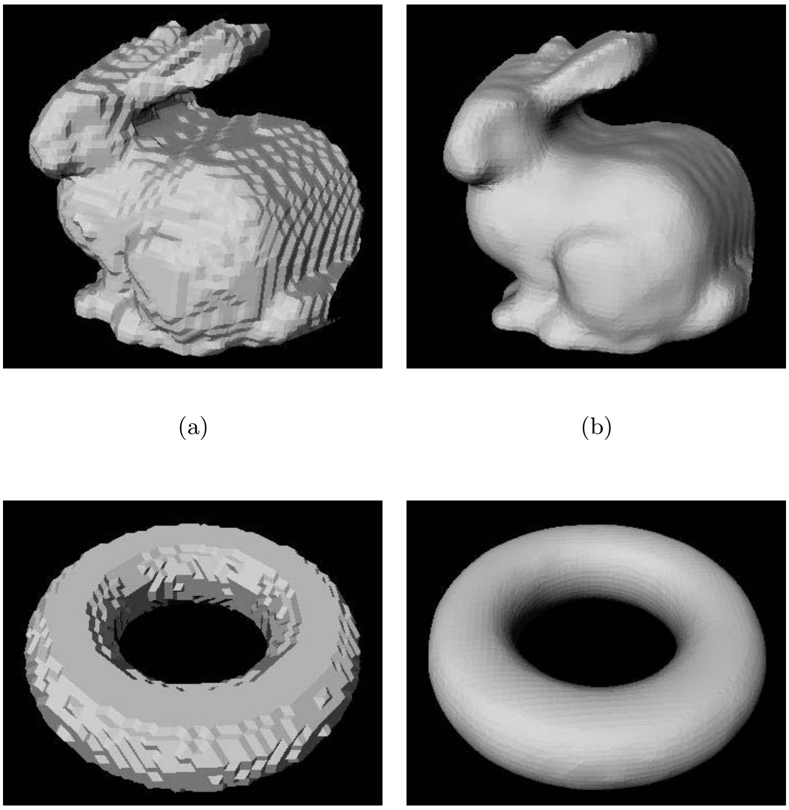
 

(c) (d)



(e)

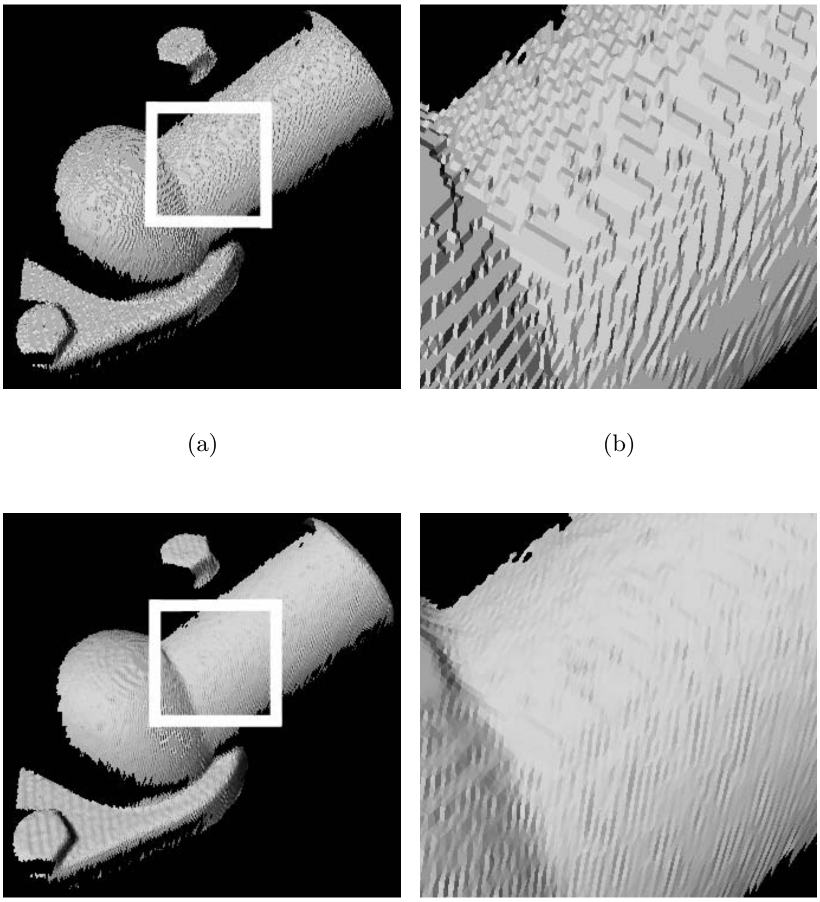
Figure 7.3: Results of median filtering, nonadaptive and adaptive regularization of range data. (a) Result from 3 by 3 median filtering conducted twice. (b) Result from regularization using Laplacian smoothing term. Note the instability along edges. (c) The edge map. (d) Result from nonadaptive regularization. (e) Result from adaptive regularization. Note the wire on the wall preserved by the adaptive method.



(a) (b)

(c) (d)

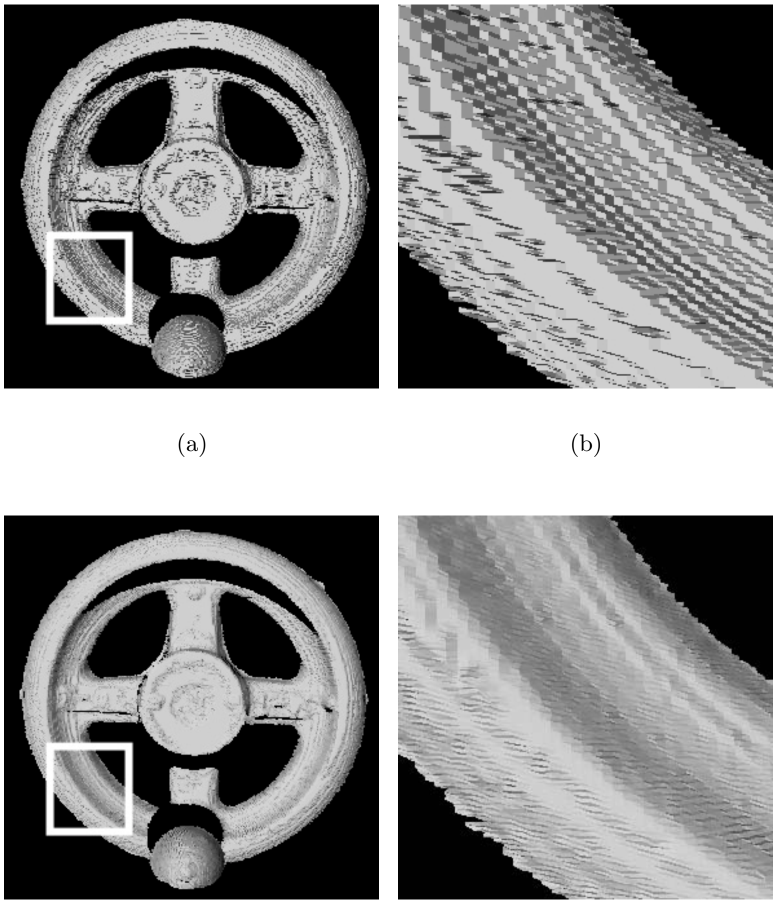
Figure 7.4: Surface smoothing of synthetic data. (a) Surface mesh of Stanford bunny model generated by binary reconstruction, 15,665 triangles. (b) 6 iteration, nonadaptive smoothed result of (a). (c) Synthetic surface mesh of a torus model generated by binary reconstruction, 14,604 triangles. (d) 7 iteration, nonadaptive smoothed result of (c).



(a) (b)

(c) (d)

Figure 7.5: Smoothing surfaces from scans of a waterneck captured by the IVP RANGER Profiling System. (a)(b) Surfaces from a range scan of a waterneck and zoomed window. (c)(d) Smoothed surfaces after 2 iterations and zoomed window.



(a) (b)

(c) (d)

Figure 7.6: Smoothing surfaces from scans of a crank captured by the IVP RANGER Profiling System. (a)(b) Surfaces from a range scan of a crank and zoomed window. (c)(d) Smoothed surfaces after three iterations and zoomed window.

(a) (b)

Figure 7.7: Smoothing surfaces captured by the RIEGL System. (a) Raw surface with 99,199 triangles. (b) 6 iteration, nonadaptive smoothed result of (a).

and 7.5(b) show the raw surface; Figs. 7.5(c) and 7.5(d) show the smoothed results that are obtained after two iterations. Similar results for a crank model are shown in Fig. 7.6 using three iterations.

Fig. 7.7(a) shows the raw surface captured by the RIEGL laser mirror scanner LMS-Z210 [71], with 99,199 triangles. The scanner is able to capture range images and color images simultaneously in a range from 2 up to 350 m. The standard deviation of the measurement error is 2.5 to 5 cm. Fig. 7.7(b) shows the corresponding nonadaptively smoothed result after 6 iterations with $\lambda = 0.01$, in which noise is effectively suppressed.

Fig. 7.8 shows the experimental results of adaptive smoothing on the triangle mesh. Fig. 7.8(a) and Fig. 7.8(b) show the raw surface captured by the RIEGL scanner with and without texture. The sampling noise can be observed from the zoomed portion of the window. The size of the original range image is 524 by 223 pixels. The building is approximately 50 m away from the scanning position. No data were obtained behind

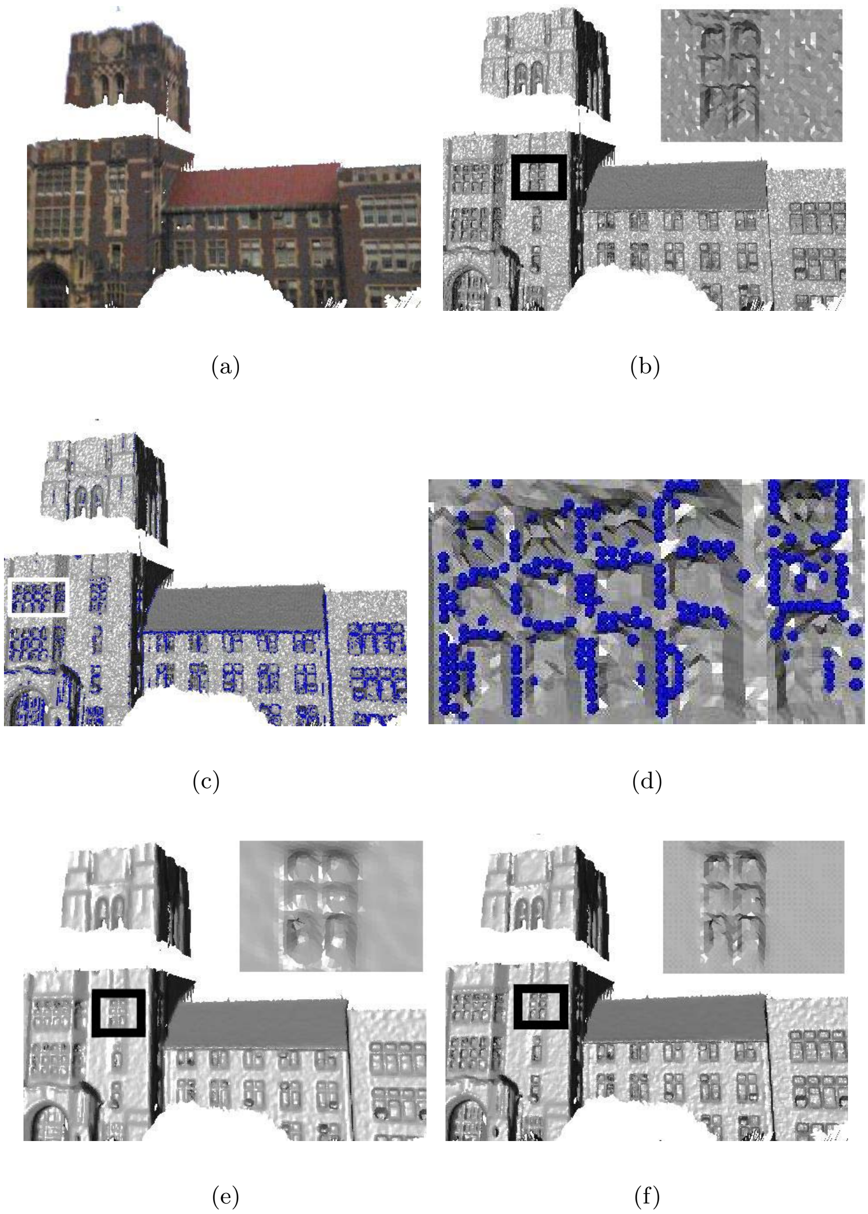


Figure 7.8: Adaptive smoothing of a surface mesh. (a) Raw textured surface captured by the RIEGL laser range scanner. (b) Raw surface without texture, 139,412 triangles. (c) Edge detection. Vertices on the edges are marked by small spheres. (d) Zoomed window frame portion of (c). (e) 5 iterations of nonadaptively smoothed result of (b). (f) 5 iterations of adaptively smoothed result of (b). Note the well preserved window frame structures.

the building where the distance is greater than the scanner's capturing range. The tower is separated due to self occlusion. Trees in front of the building are removed to highlight the smoothing on the building surface. The 3D model has 139,412 triangles. Crease edge detection on the triangle mesh is shown in Fig. 7.8(c), where each vertex on the crease edge is marked by a small sphere. The window frame portion is zoomed and shown in Fig. 7.8(d). Fig. 7.8(e) and Fig. 7.8(f) show the nonadaptively and adaptively smoothed results, respectively, after 5 iterations with $\lambda = 0.01$. The geometric details such as window frames, as seen from the zoomed portion, are well preserved by the adaptive smoothing.

The algorithm relies on adjusting the vertex along the normal direction. When the surface is so noisy that the normal estimation is no longer stable, the algorithm fails because the smoothing will cause mesh self-intersection. The algorithm works well for all tested real range data. The noise effect is tested using a digital elevation map by adding Gaussian noise. The smoothing fails when the signal-to-noise ratio reaches 8.1 dB. This problem can be solved using Laplacian flow by improving the mesh regularity [60].

7.2 Surface Registration

The point fingerprint-based surface registration scheme was tested on both synthetic and real range data. Synthetic range images were obtained from a range scanner simulator that reads the depth buffer [59] and recovers the range values of the rendered 3D object.

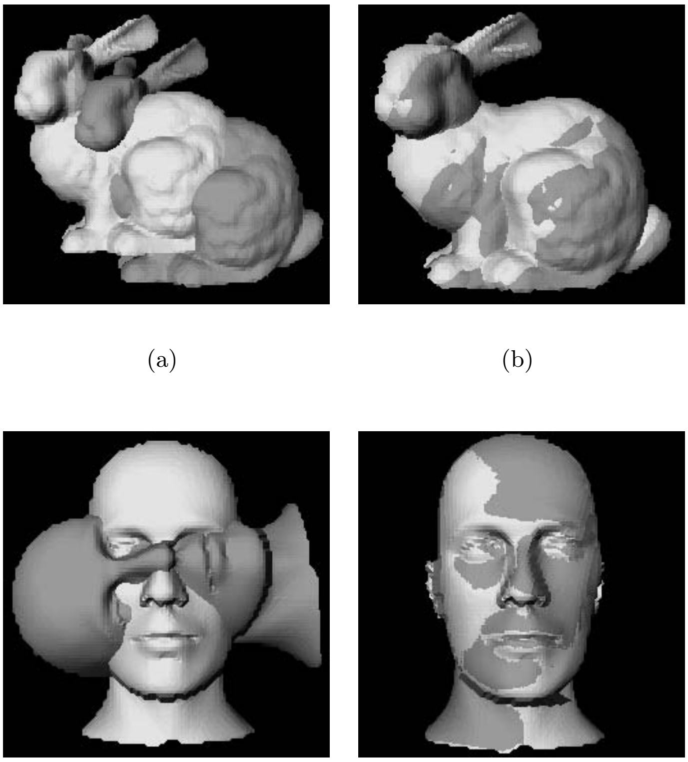
The real range images were scanned using the IVP RANGER Profiling System [43], where objects scanned were located on a conveyor belt.

Fig. 7.9(a) shows the misaligned surfaces from a synthetic bunny model, and 7.9(b) shows the registered result. The surfaces were misaligned by a translation before registration. Fig. 7.9(c) shows the misaligned surfaces from a synthetic head model, and 7.9(d) shows the registered result. The surfaces were misaligned by a translation and a rotation before registration.

This method was also applied to align USGS DEM data. Fig. 7.10(a) shows two misaligned surfaces, and Fig. 7.10(b) shows the registered result. Note that the two data sets only overlap in some area and the proposed method successfully found the corresponding point pairs in the overlapping area.

Fig. 7.11(a) and 7.11(b) show extracted points on two surfaces which were scanned from a brain model using the Minolta 700 range scanner [14]. After point matching by fingerprint comparison, corresponding points are obtained and displayed in Fig. 7.11(c) and 7.11(d). A coarse registration based on point correspondences is computed, and registered surfaces are shown in Fig. 7.11(e). Fig. 7.11(f) shows the registered surfaces after ICP refinement. Similarly, experimental results on a face model [14] are illustrated in Figs. 7.12(a) to 7.12(f).

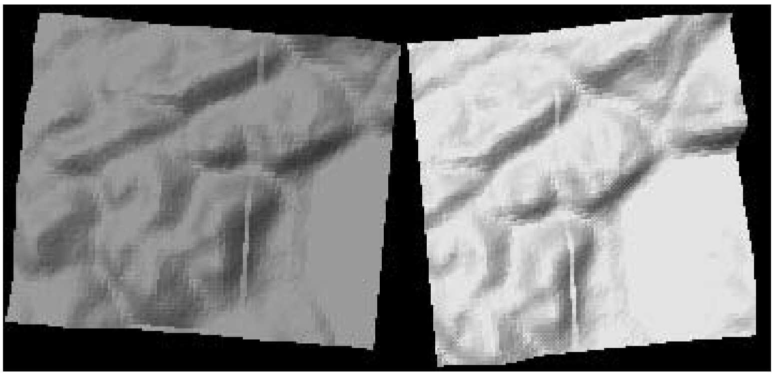
A pair of surfaces in Fig. 7.13 were scanned from a mannequin using the IVP RANGER profiling system [43] and used for occlusion testing. The measurements near the nose were incomplete due to self occlusions. Experimental results of extracted



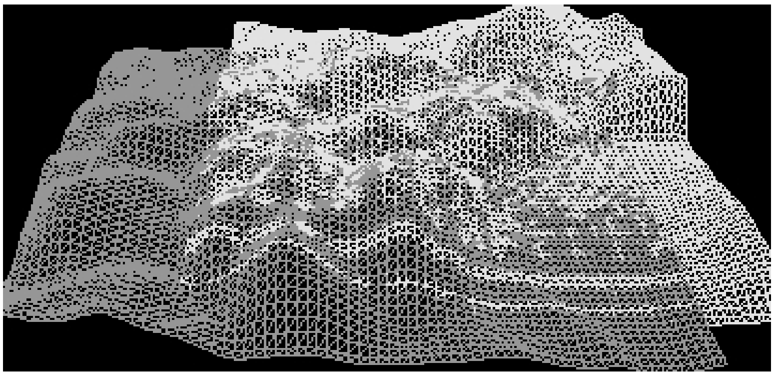
(a) (b)

(c) (d)

Figure 7.9: 3D registration of synthetic range data. (a) and (c) are unregistered synthetic surfaces from a bunny model and a head model. The bunny model was from Stanford University Computer Graphics Laboratory and available at http://graphics.stanford.edu/data/3Dscanrep/. The head model was reconstructed by Hugues Hoppe and available at ftp://ftp.research.microsoft.com/users/hhoppe/data/thesis/. (b) and (d) are surface registration results of (a) and (c) by point fingerprint matching.

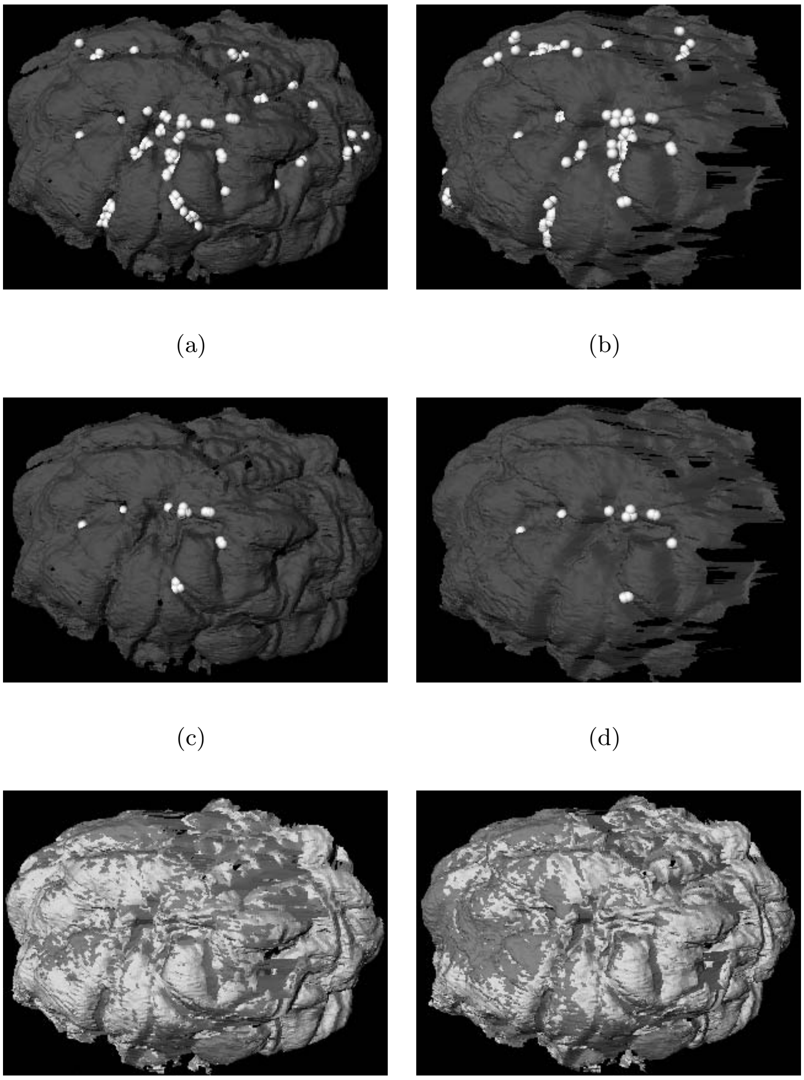


(a)



(b)

Figure 7.10: Registration of DEM data. (a) Misaligned surfaces from USGS DEM data with only a partially overlapping region between each. (b) Registration results of (a) by point fingerprint matching, shown in wireframe.



(a) (b)

(c) (d)

(e) (f)

Figure 7.11: Matching surfaces of a brain model downloaded from http://sampl.engr.ohio-state.edu/∼sampl/database.htm [14]. (a)(b) Extracted feature points on two surfaces. (c)(d) Corresponding points by fingerprint matching. (e) Surface registration. (f) Refined registration using ICP.

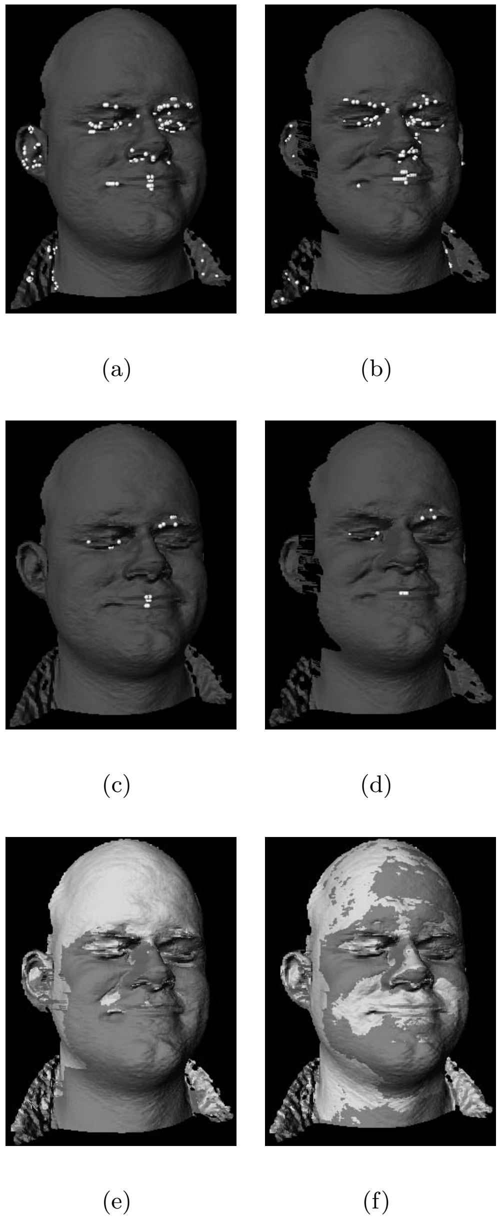
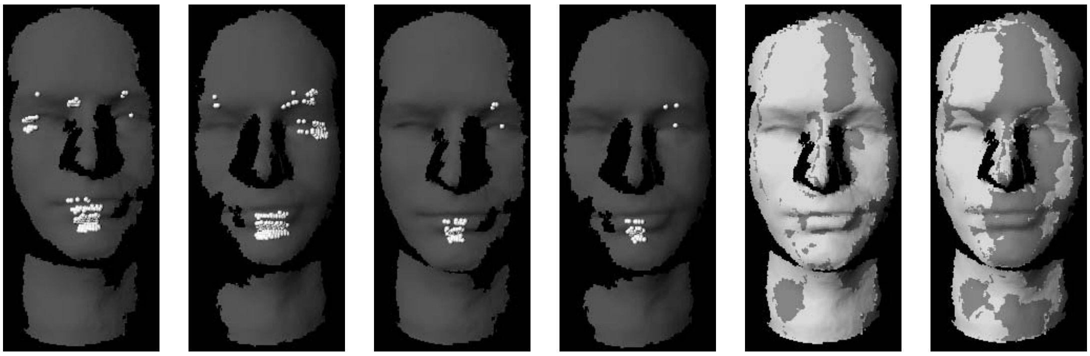


Figure 7.12: Matching surfaces of a face model downloaded from http://sampl.engr.ohio-state.edu/∼sampl/database.htm [14]. (a)(b) Extracted feature points on two surfaces. (c)(d) Corresponding points by fingerprint matching. (e) Surface registration. (f) Refined registration using ICP.



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

Figure 7.13: Matching surfaces of a mannequin face. (a)(b) Extracted feature points on two surfaces. (c)(d) Corresponding points by fingerprint matching. (e) Surface registration. (f) Refined registration using ICP.

points, point correspondences, coarse registration, and refined registration are shown in Figs. 7.13(a) to 7.13(f).

Surfaces in Fig. 7.14(a) and 7.14(b) are from the USGS Digital Elevation Model (DEM) with an overlapping region. Zero-mean Gaussian noise is superimposed on the original surfaces. The signal-to-noise ratio (SNR) of surfaces in Fig. 7.14(c)-7.14(d), 7.14(e)-7.14(f), 7.14(g)-7.14(h), and 7.14(i)-7.14(j) are 31.63, 22.08, 17.65, and 11.63 dB, respectively. Obtained corresponding points are displayed on the surfaces. Results show that fingerprint matching is robust against noise. With a SNR lower than 11.63 dB, the matching failed.

Surfaces [14] in Fig. 7.15 are used in the experiment of handling different surface sampling resolutions. The surface in Fig. 7.15(a) has 28,964 triangles. The surface in

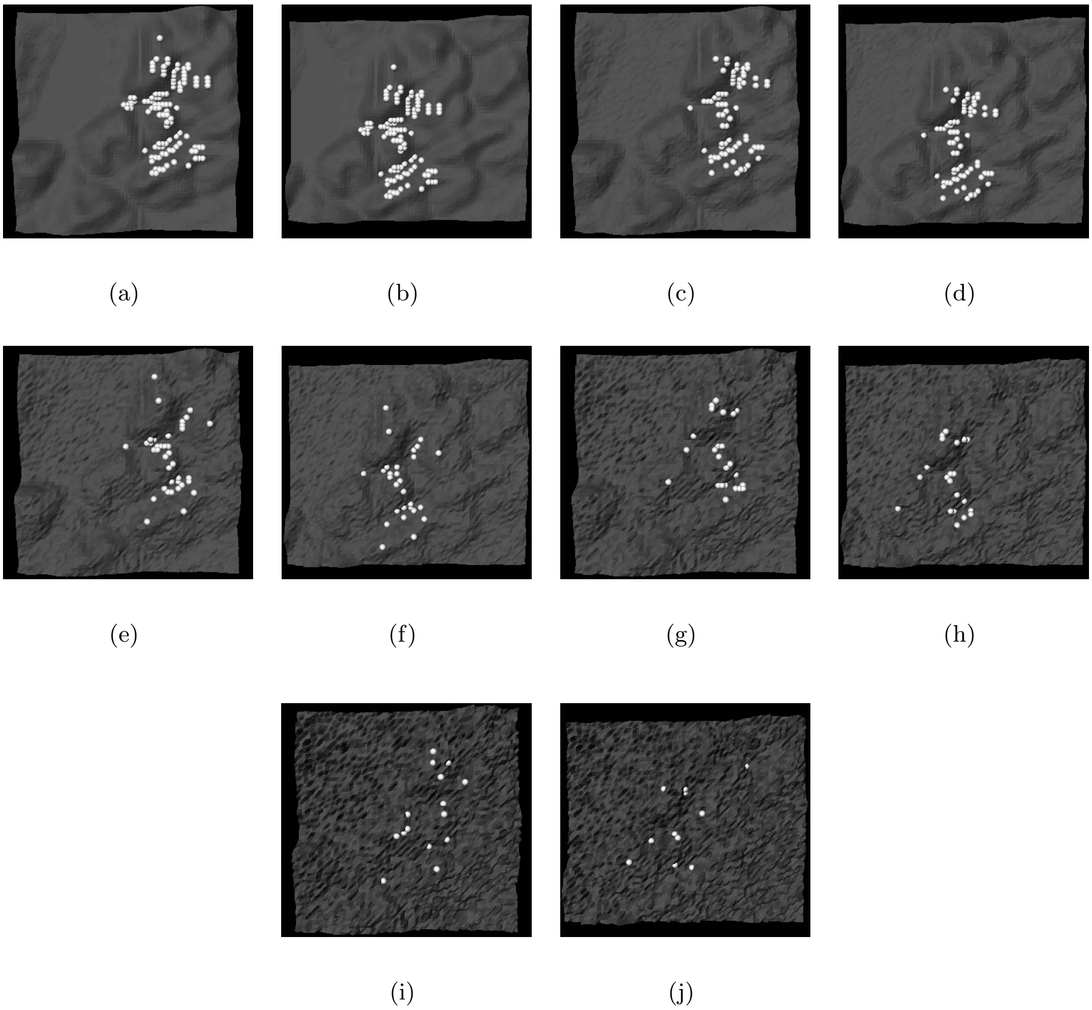


Figure 7.14: Finding corresponding points on surfaces with noise. (a)(b) Surfaces without noise. (c)(d) Surfaces with 31.63 dB SNR. (e)(f) Surfaces with 22.08 dB SNR. (g)(h) Surfaces with 17.65 dB SNR. (i)(j) Matching fails on surfaces with 11.63 dB SNR.

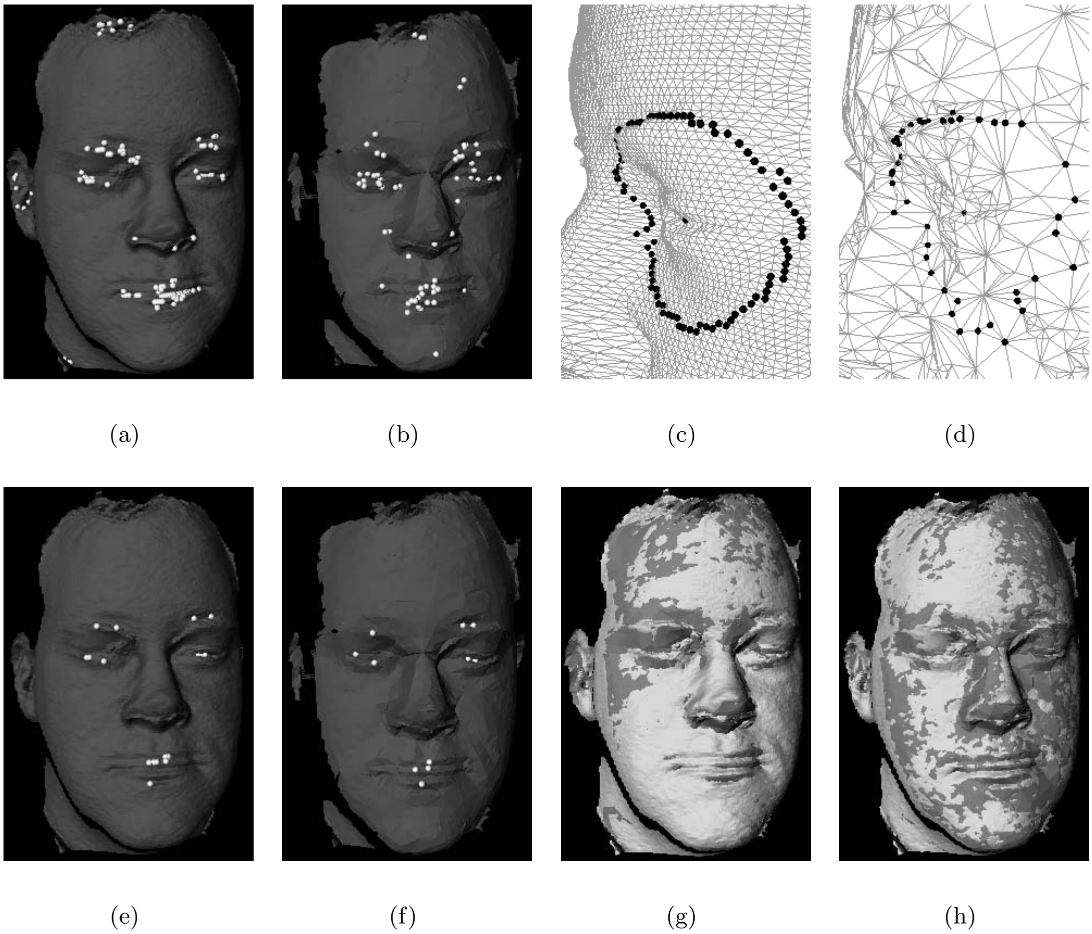
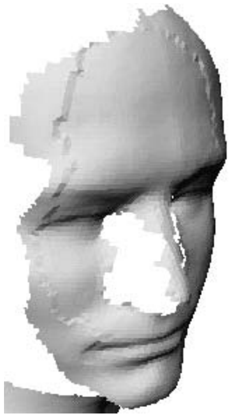
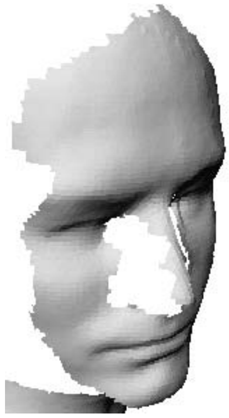


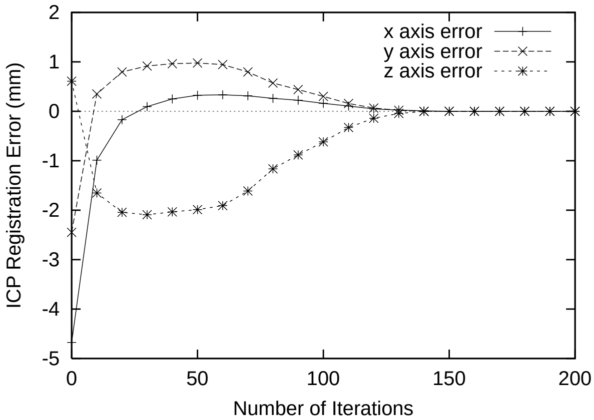
Figure 7.15: Matching surfaces with different resolutions. Original range images were downloaded from http://sampl.engr.ohio-state.edu/∼sampl/database.htm [14]. (a)(b) Extracted feature points on two surfaces with 28,964 and 5,000 triangles respectively. (c)(d) Geodesic contours on two surfaces. (e)(f) Corresponding points by fingerprint matching. (g) Surface registration. (h) Refined registration using ICP.



(a)



(b)



(c)

Figure 7.16: ICP refinement. (a) Surface integration result without using the ICP refinement. Note the seam on the forehead. (b) Smooth surface integration result after using the ICP algorithm. (c) Translation errors along three axes during ICP iterations.

Fig. 7.15(b) has 5,000 triangles, reduced from 28,893 triangles. Extracted candidate points are shown on the surfaces. Figs. 7.15(c) and 7.15(d) illustrate the geodesic contours of the same radius on two surfaces shown in wireframe. Corresponding points, coarse registration, and refined registration are shown in Figs. 7.15(e) to 7.15(h).

The improvement from using ICP refinement can be seen from the two-view merged surfaces shown in Figs. 7.16(a) and 7.16(b), which respectively represent the integration results with and without ICP refinement. After ICP refinement, the integrated surface becomes smoother without the appearance of having seams. Fig. 7.16(c) shows the error convergence. Because the rotation refinement is very small and can be ignored, only the convergence of translation errors along three axes was plotted. The transformation after 200 iterations is regarded as the ground truth. The errors become stable after 150 iterations.

7.3 Surface Reconstruction

Fig. 7.17(a) shows a synthetic 3D model with texture, and Fig. 7.17(b) shows the corresponding wireframe model. Using a simulated range scanner, both range and color information from the rendered 3D scene can be captured. Fig. 7.17(c) shows the surface from one range scan. Figs. 7.17(d) to 7.17(f) show the two-, three-, and four-view integration results using the mesh zippering method. The reconstructed model consists of 424,495 triangles.

Fig. 7.18 shows the two-view integration of the synthetic room model using the

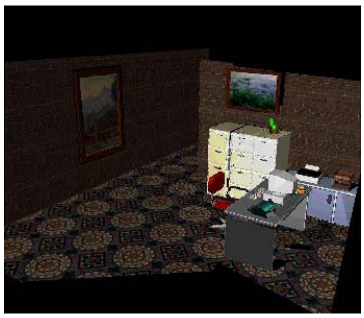
(a)

(b)

(c)

(d)

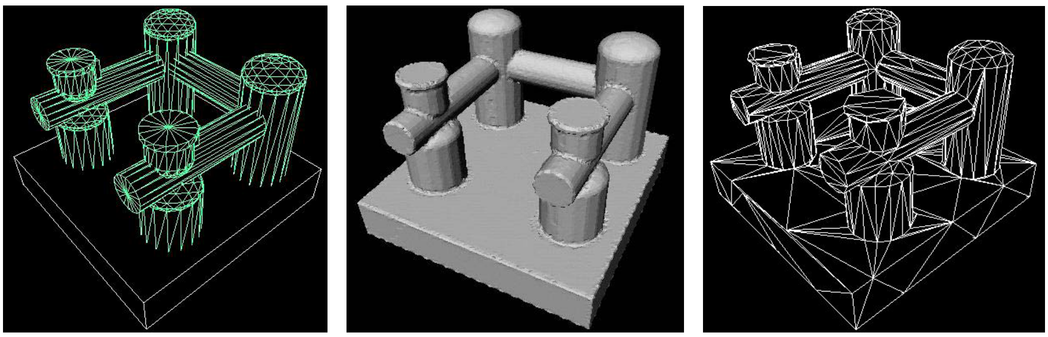
(e)



(f)

Figure 7.17: Surface reconstruction using mesh zippering. (a) A synthetic 3D office model, downloaded from http://www.cowhouse.com. (b) Wireframe of (a). (c) Reconstructed surface from one simulated range scan of (a), 126,517 triangles. (d)-(f) Two-, three-, and four-view integration, with 218,283, 327,816, and 424,495 triangles.

Figure 7.18: Two-view integration by implicit surface fusion with 150,351 triangles.



(a) (b) (c)

Figure 7.19: Implicit surface-based reconstruction of a synthetic object. (a) Original synthetic model. (b) 44-view reconstructed surface with 119,911 triangles. (c) Simplified surface with 800 triangles.

Table 7.1: Small parts 3D reconstruction

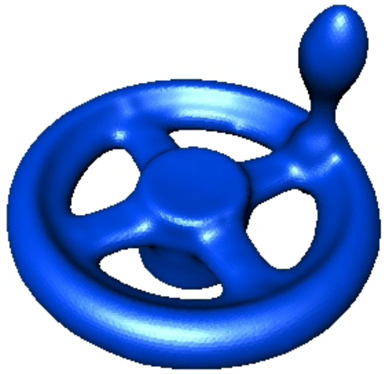
Model	Number of Views	Number of Triangles
Crank	35	93,572
Disk brake	10	73,553
Waterneck	28	117,564
Distributor cap	16	117,036
Racecar	9	109,823
Mannequin	10	80,148

implicit surface-based fusion. Both the geometric and texture integrations are similar to the one obtained using mesh zippering. However, the back projection is not necessary.

Fig. 7.19(a) shows a synthetic model composed of different parts that are intersected with each other. The 44-view reconstruction without space carving is shown in Fig. 7.19(b) with 119,911 triangles. Fig. 7.19(c) shows the simplified mesh with 800 triangles. The reconstruction process is able to convert a rendered model of any structure into a whole sheet of triangle mesh.

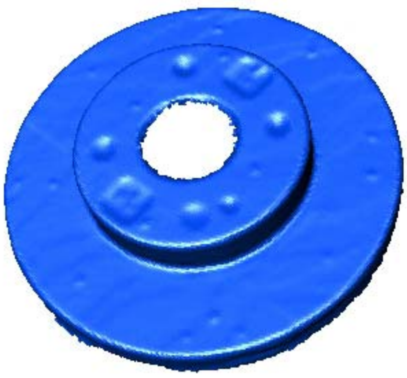
Figs. 7.20 to 7.22 illustrate small object 3D reconstruction results using the IVP Profiling System, by showing the photos of the objects and screen shots of the 3D reconstructions. The objects include a crank, disk brake, waterneck, distributor cap, racecar, and a mannequin. Multiple views are scanned for a full 3D reconstruction. The number of views and number of triangles used for reconstruction of each object are listed in Table 7.1. The reconstructions (without space carving) of the disk brake, distributor cap, and racecar do not have bottoms because only the top of the objects were scanned. Using a number of views to cover the whole object, the reconstruction is

(a)



(b)

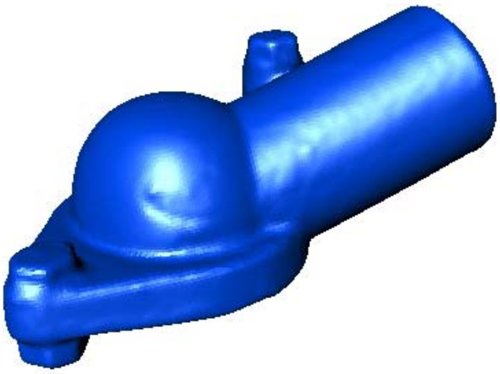
(c)



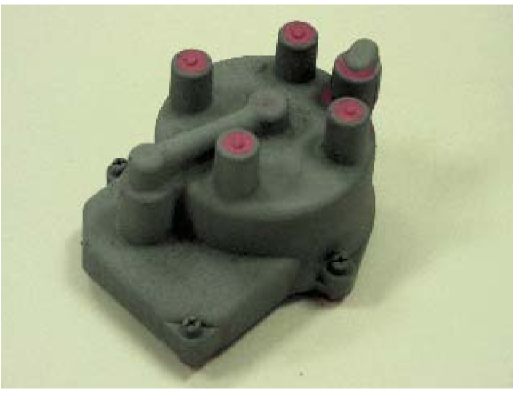
(d)

Figure 7.20: Surface modeling of a crank and a disk brake using the RANGER System. (a) Photo of a crank. (b) 35-view 3D reconstruction of (a) with 93,752 triangles. (c) Photo of a disk brake. (d) 10-view 3D reconstruction of (c) with 73,553 triangles.

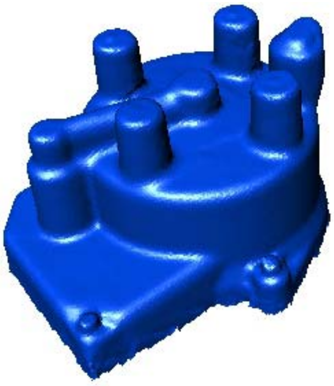
(a)



(b)

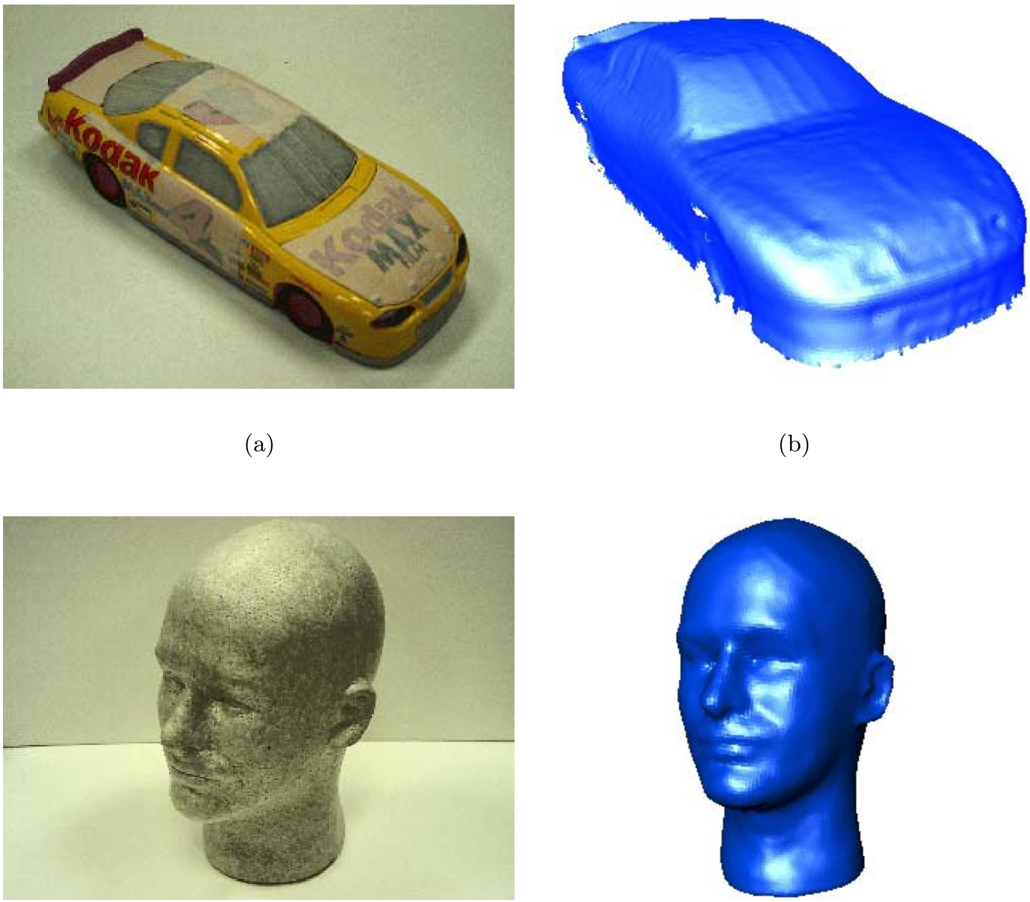


(c)



(d)

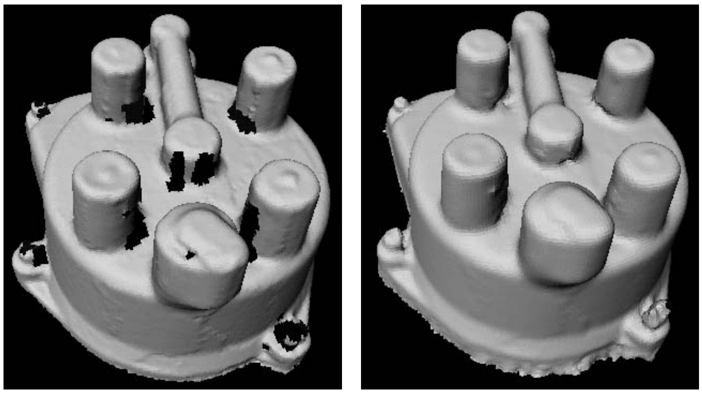
Figure 7.21: Surface modeling of a waterneck and a distributor cap using the RANGER System. (a) Photo of a waterneck. (b) 28-view 3D reconstruction of (a) with 117,564 triangles. (c) Photo of a distributor cap. (d) 16-view 3D reconstruction of (c) with 117,036 triangles.



(a) (b)

(c) (d)

Figure 7.22: Surface modeling of a racecar and a mannequin using the RANGER System. (a) Photo of a racecar. (b) 9-view 3D reconstruction of (a) with 109,823 triangles. (c) Photo of a mannequin. (d) 10-view 3D reconstruction of (c) with 80,148 triangles.



(a) (b)

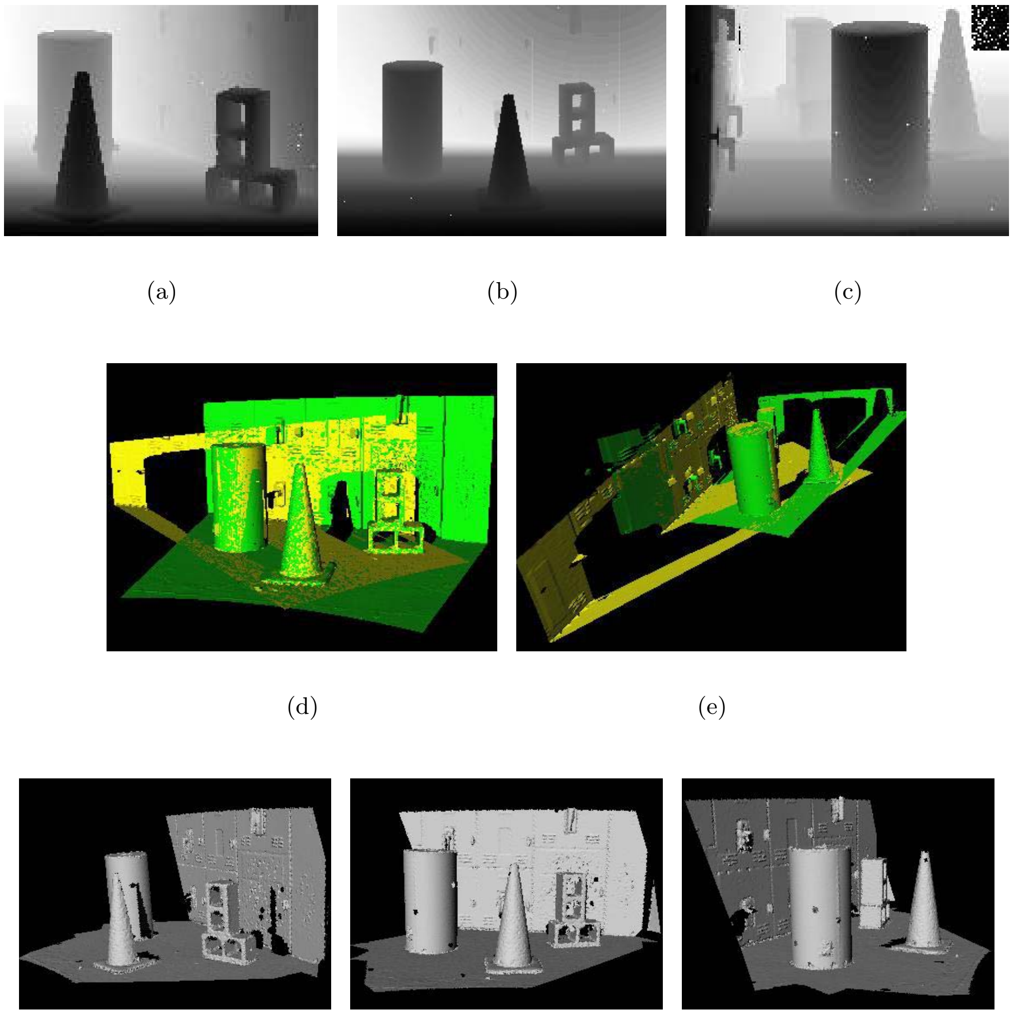
Figure 7.23: Automatic hole filling. (a) Reconstructed surface with holes. (b) Surface after automatically filling holes.

watertight for the crank, the waterneck, and the mannequin.

Fig. 7.23(a) shows the original reconstructed surface of the distributor cap. The surface contains holes where the laser could not reach. By volumetric processing and applying the hole filling algorithm, most holes can be automatically filled, as shown in Fig. 7.23(b).

Fig. 7.24 shows the 3-view reconstruction results using range data captured by the Coleman scanner. Three range images are displayed in Figs. 7.24(a) to 7.24(c), and the registered pairs of surfaces are illustrated in Figs. 7.24(d) and 7.24(e). The reconstructed surfaces displayed from three different viewpoints are shown in Figs. 7.24(f) to 7.24(h).

Implicit surface-based integration of geometry and texture using real data is illustrated in Fig. 7.25. Figs. 7.25(a) and 7.25(b) show a pair of range images taken from



(a) (b) (c)

(d) (e)

(f) (g) (h)

Figure 7.24: Surface modeling using the COLEMAN scanner. (a)-(c) Range images from three different views. (d) Registered surfaces of (a) and (b). (e) Registered surfaces of (a) and (c). (f)-(h) Reconstructed surfaces displayed from three different viewpoints.

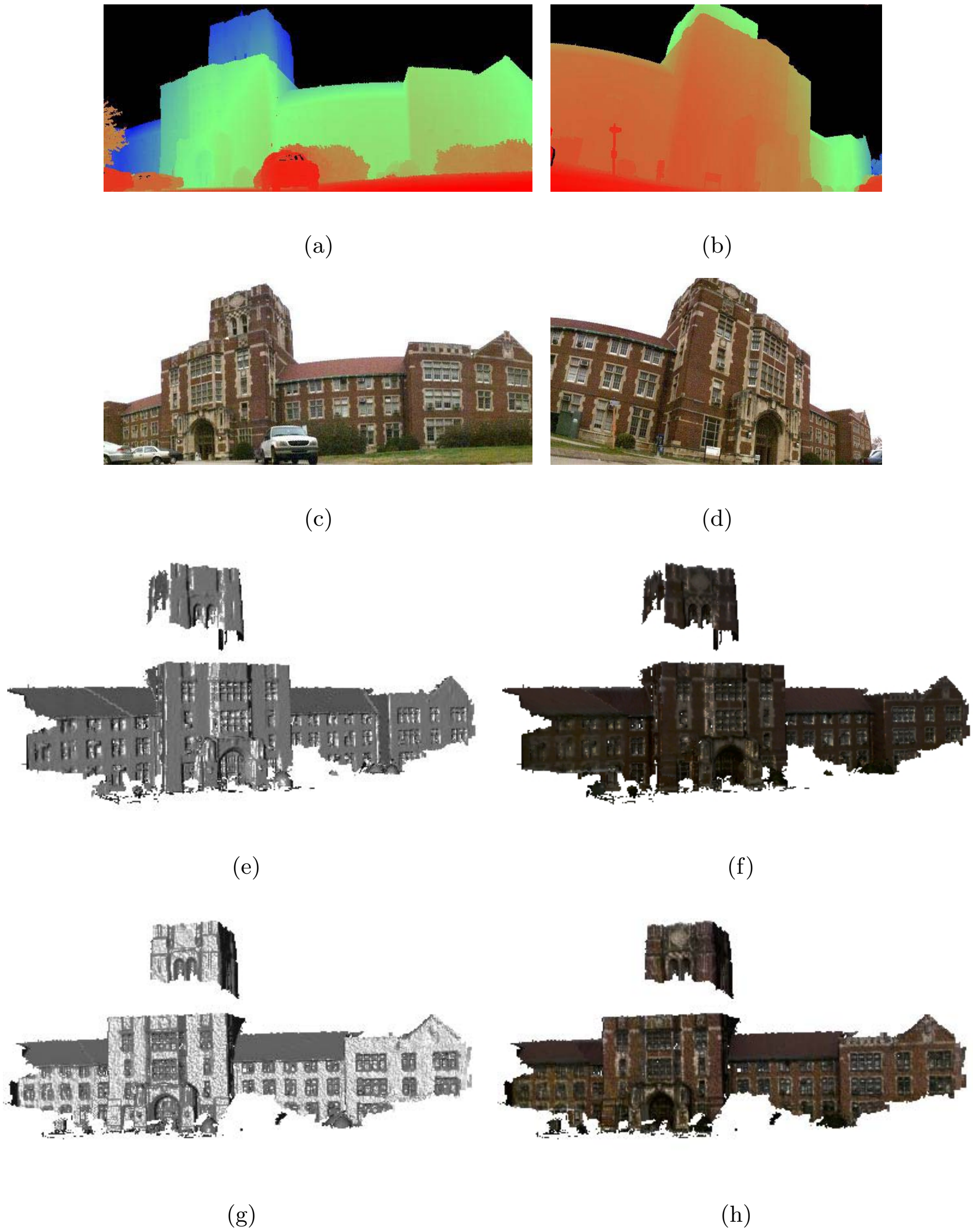
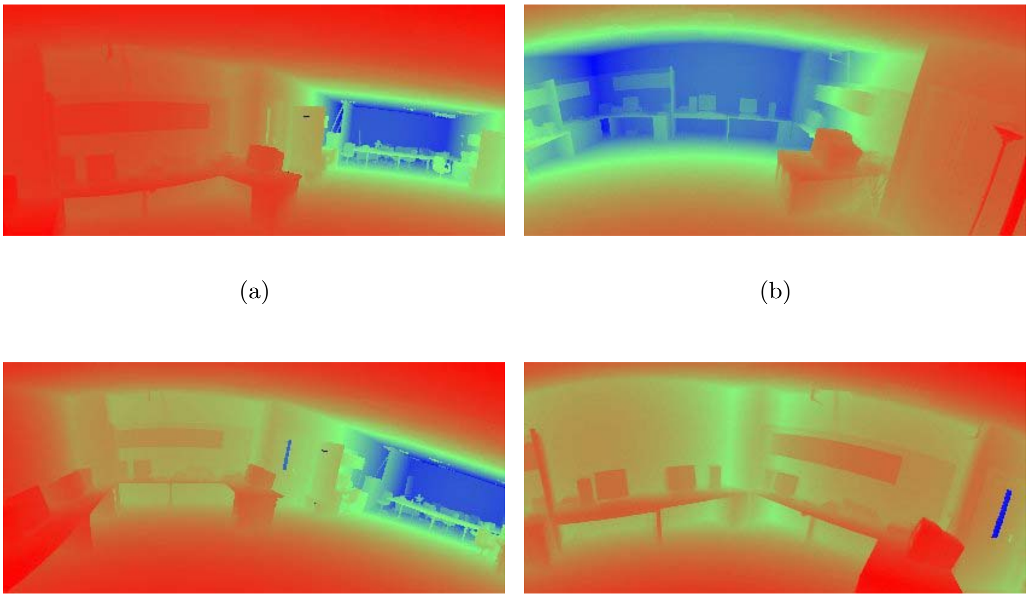


Figure 7.25: Surface modeling using the RIEGL scanner. (a)(b) A pair of range images captured by the RIEGL scanner. (c)(d) Corresponding color images. (e) 2-view 3D reconstruction from (a) and (b) with 159,677 triangles. (f) Reconstructed surface with texture fusion. (g)(h) Reconstructed surface displayed from another viewpoint.



(a) (b)

(c) (d)

Figure 7.26: Four indoor range images scanned by the RIEGL scanner.

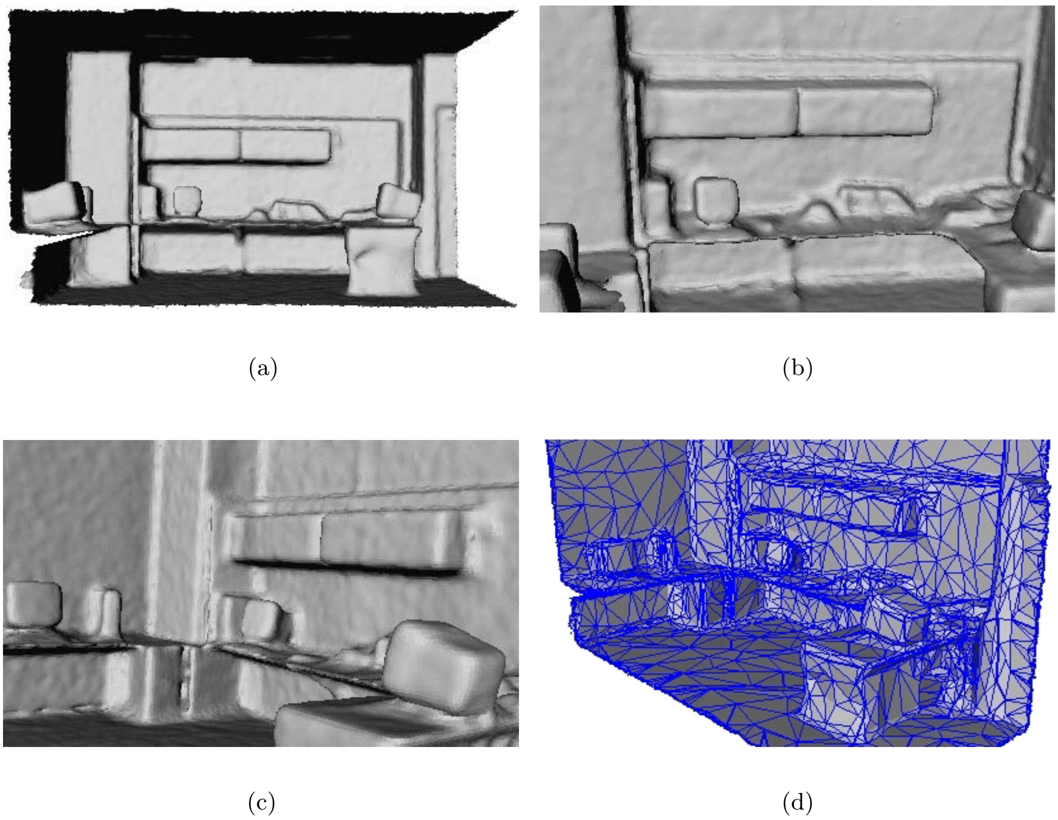


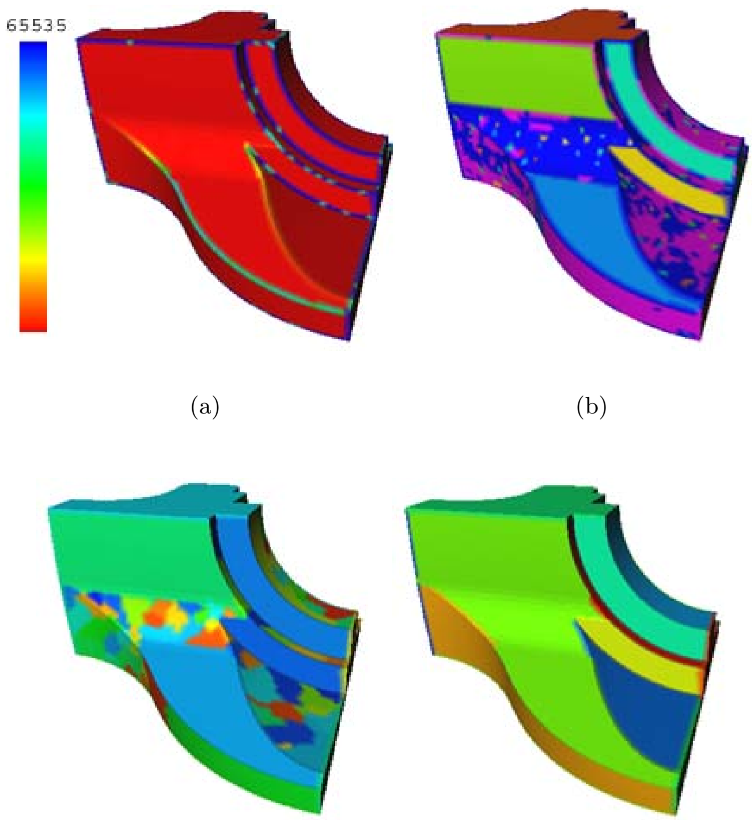
Figure 7.27: Four-view reconstruction using the RIEGL scanner. (a)-(c) Four-view reconstruction using the RIEGL scanner, with 341,639 triangles, displayed from different viewpoints. (d) Simplified model with 5,000 triangles.

two different views in front of Ayres Hall using the RIEGL scanner. Figs. 7.25(c) and 7.25(d) are the corresponding color images. Fig. 7.25(e) shows the 3D reconstruction with 159,677 triangles. Fig. 7.25(f) shows the results with texture fusion. Figs. 7.25(g) and 7.25(h) display the reconstructed surface from another viewpoint.

Reconstruction by space carving is shown in Figs. 7.26 to 7.27 using range images captured by the RIEGL scanner. Four range images of a room are shown in Fig. 7.26(a) to 7.26(d). The region of interest is a corner of the room. The reconstructed surface has 341,639 triangles and is displayed in Figs. 7.27(a) to 7.27(c) from four different viewpoints. Due to the noise introduced by the scanner, the surface was heavily smoothed using the mean curvature flow in a volumetric grid. The simplified model with 5,000 triangles is shown in Fig. 7.27(d).

7.4 Surface Segmentation

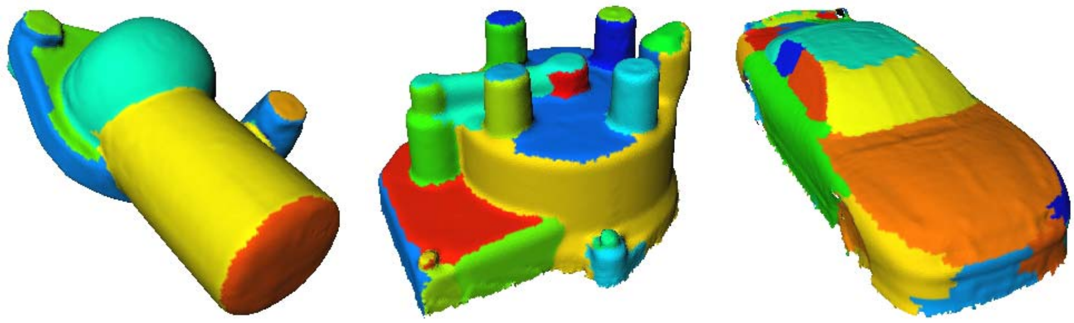
Fig. 7.28 shows the process of segmenting a synthetic fandisk model. Fig. 7.28(a) shows the color-coded edge strength of each vertex. Piecewise continuous edge strength definition results in smooth color transition along the crease edges. Fig. 7.28(b) shows the labeling after minima detection. Minima are uniquely labeled and painted by random colors. Vertices in purple are on plateaus, and vertices in blue are on ramps. Fig. 7.28(c) shows the labeling result after geodesic erosion. Plateaus are segmented and distributed into neighboring regions, and all vertices are labeled. Fig. 7.28(d) shows the final segmentation result after region merging.



(a) (b)

(c) (d)

Figure 7.28: Segmentation process. (a) Color-coded edge strength on a fandisk model. (b) Labeling after minima detection. Plateau regions are in purple, and vertices on the ramp are in blue. Local minima are labeled in random colors.(c) Labeling after geodesic erosion. (d) Final segmentation after region merging.



(a) (b) (c)

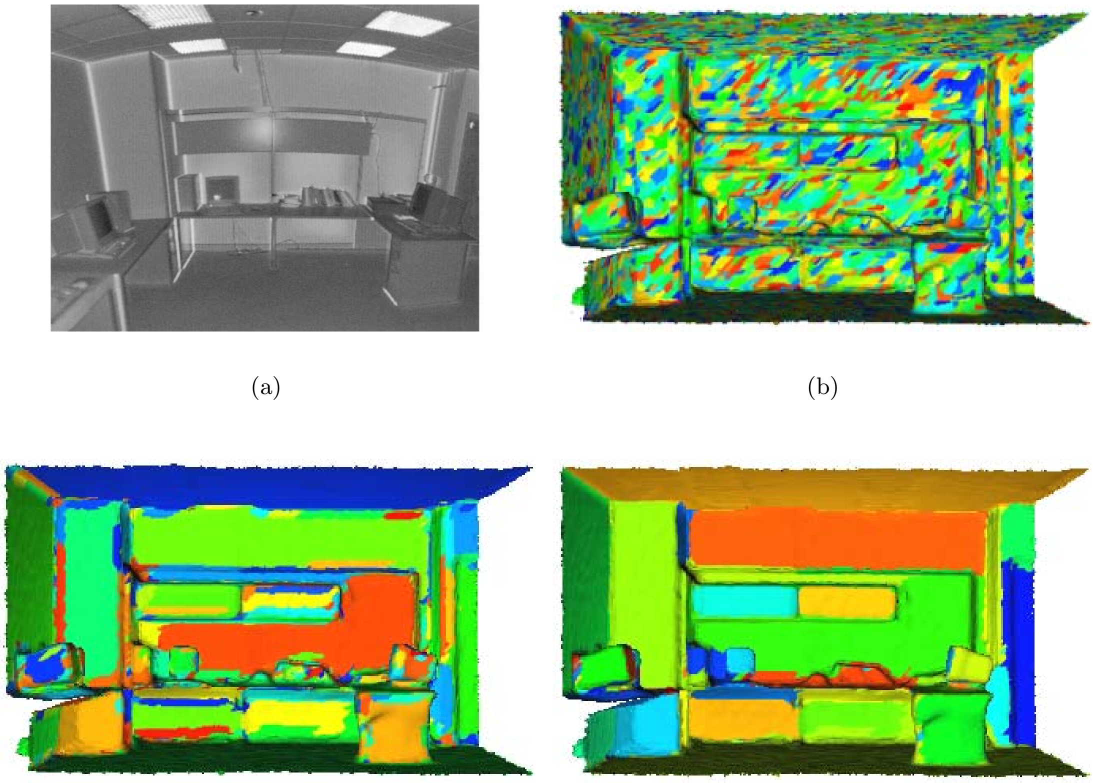
Figure 7.29: Surface segmentation of small parts. (a) Waterneck. (b) Distributor cap. (c) Racecar model.

Fig. 7.29 show the segmentation results of the waterneck, distributor cap, and racecar, which were reconstructed using the data from the IVP profiling system.

Figs. 7.30(a) to 7.30(d) show the picture of the room, the surface before region merging (8,851 regions), the surface after region merging (451 regions), and the surface with area penalization (49 regions), respectively. Area penalization is very effective to avoid over-segmentation. By deleting the small regions caused by outliers, the final segmentation has 30 regions.

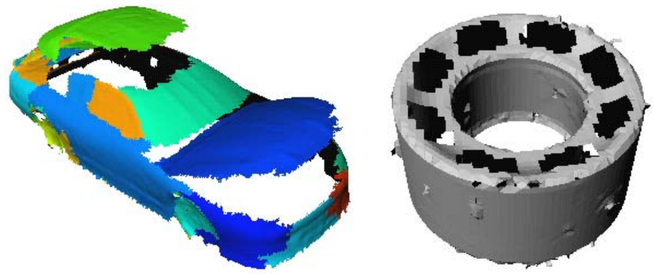
The segmented parts can be manipulated by rotation or translation in 3D space. Fig. 7.31(a) shows that the hood is open and the top is displaced. A pulley model in Fig. 7.31(b) is decomposed and shown in Fig. 7.31(c).

Table 7.2 shows the segmentation time spent in each step for seven models used in the experiment. All models except the fandisk are reconstructed from the real data. The

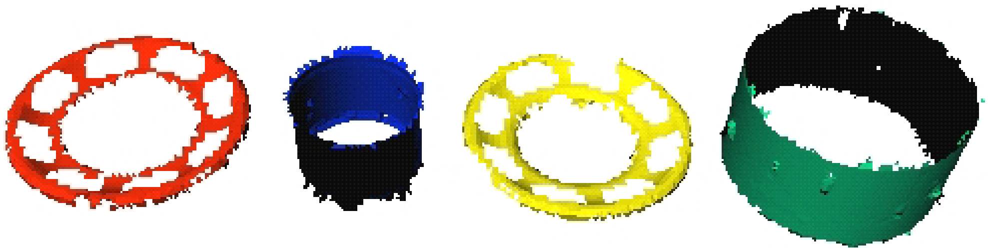


(a) (b) (c) (d)

Figure 7.30: Surface segmentation of a scene. (a) A photo of a room's corner reconstructed from 4-view range scans with 341,639 triangles. (b) Segmentation result before region merging with 6,381 regions. (c) Segmentation result after region merging with 936 regions. (d) Final segmentation result after penalizing small areas with 160 regions.



(a) (b)



(c)

Figure 7.31: Manipulating segmented parts. (a) The hood is open and the top is moved. (b) A 3D pulley model. (c) The decomposition of (b).

Table 7.2: Performance of the fast marching watershed on six 3D models

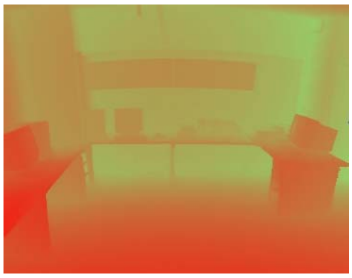
Model	Number of Triangles	Minima (ms)	Erosion (ms)	Descend (ms)	Merge (ms)	Number of Regions
Fandisk	12,936	99.3	37.5	1.4	6.5	18
Pulley	11,366	118.7	58.9	2.3	106.1	4
Racecar	109,823	1182.9	610.0	35.8	1463.4	30
Distributor cap	117,036	1187.0	659.0	36.3	4820.3	40
Waterneck	117,564	1174.1	668.5	38.4	3259.6	13
Room	341,639	3493.4	1976.9	108.1	4551.0	30

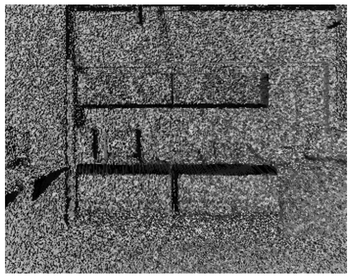
time is measured in milliseconds on an SGI Octane. Time spent in each step depends on the size of the triangle mesh, the geometric complexity of the models, and the surface noise level. The times for minima detection and for geodesic erosion are approximately proportional to the number of triangles. However, for smooth or synthetic surfaces, the flat regions are often large and minima detection takes more time than for noisy surfaces. On the other hand, synthetic surfaces often have sharp edges and the plateau regions are small. Therefore, geodesic erosion on a synthetic model is often faster than on a real model. The number of regions in Table 7.2 represents the segmentation result after penalizing small regions and deleting small regions caused by outliers.

7.5 The Frame of Surface Modeling and Analysis

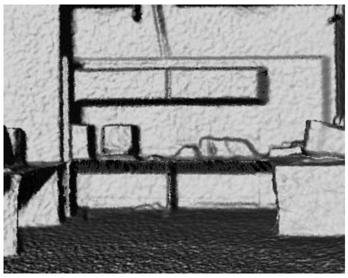
Through an application of indoor 3D mapping using the laser range scanner, this section explains how the whole surface modeling and analysis framework works.

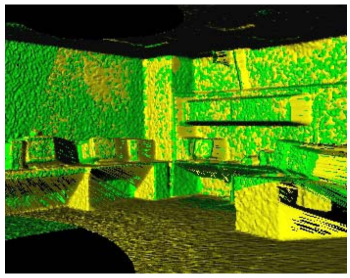
Fig. 7.32(a) is a range image acquired in a room using the RIEGL scanner. The



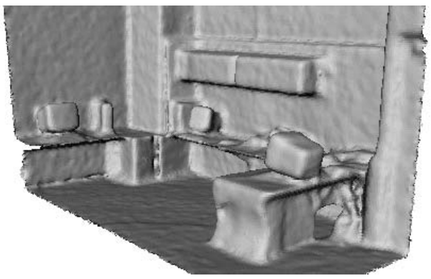


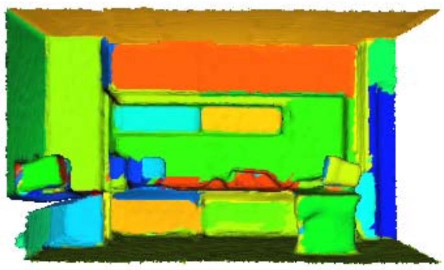
(a) (b)





(c) (d)





(e) (f)

Figure 7.32: Application of the surface modeling framework. (a) A range image captured by the RIEGL scanner. (b) The surface reconstructed from the raw range data. (c) Smoothed surface. (d) Two registered surfaces. (e) Four-view integrated surface. (f) Segmented surfaces.

surface reconstructed from the raw data is shown in Fig. 7.32(b). Fig. 7.32(c) displays the smoothed surface using the area-decreasing flow. Registered by matching the point fingerprints, two surfaces are shown in Fig. 7.32(d). By integrating the surfaces from four different views, the final reconstruction is obtained and illustrated in Fig. 7.32(e), which has been post-processed by volumetric mean curvature flow. Fig. 7.32(f) shows the surface segmentation results using the fast marching watershed algorithm.

Chapter 8

Conclusions and Future Work

This chapter summarizes the contributions of this research. It concludes by describing the opportunities for future work.

8.1 Conclusions

This section summarizes the contributions of this research in building a framework of surface modeling and analysis.

Area-Decreasing Flow

For surface smoothing, area-decreasing flow instead of mean curvature flow is proposed. Despite their mathematical equivalence, area minimization generates a more efficient algorithm for discrete surface smoothing. The problems with mean curvature flow are that the curvature is difficult to estimate on a discrete surface, and there is no easy way to choose a proper flowing step size. The advantages of the proposed algorithm are as follows.

1. Curvature estimation is eliminated. Surface area can be easily formulated on a triangle mesh.

2. An optimal flowing step size can be computed.

A typical problem for surface smoothing is shrinkage. Previous works avoided this problem by preserving the volume of the surface being smoothed. The new algorithm incorporates a rigidity term in the energy function to prevent the shrinkage problem.

An adaptive term is added into the smoothing scheme based on the edge strength at each vertex. Edge strength is robustly estimated using tensor voting on a triangle mesh. Adaptive smoothing effectively preserves the crease edges and sharp corners while achieving the same smoothing result elsewhere.

Experimental results show the proposed algorithm is able to efficiently smooth both calibrated range images and large meshes generated by different range scanners.

Point Fingerprint

A new surface representation scheme, called point fingerprint, based on a set of geodesic circles generated on the triangle mesh, is presented. The projections of geodesic circles on the tangent plane form a discriminating feature, which is similar to human fingerprints and can be used to match surface points. The concept of point fingerprint originated from the exponential map that is well defined in differential geometry.

The fingerprints of points of interest from a pair of surfaces are compared to find the corresponding points.

There are four major advantages of the point fingerprint scheme:

1. Only HSI [110] and point fingerprint, based on a one-to-one mapping, are able to carry additional information such as curvature and color to improve matching accuracy. Spin Images [45] and SPS [105] are based on 2D histograms and cannot carry additional information.

2. HSI finds corresponding points by 2D image correlation, which is more computationally expensive than point fingerprint matching, which is based on a set of 1D signal correlations.

3. Only Splash [82], HSI, and point fingerprint use geodesic measure. However, the geodesic distance computed in point fingerprint is more accurate due to use of the fast marching method instead of Dijkstra's algorithm.

4. Both point fingerprint and PS [20] use contours around a point. The contours of PS, obtained by intersecting a sphere with a surface, are sometimes ambiguous. However, the contours in point fingerprint are clearly defined using the geodesic measure.

A simple alternative method is proposed to compute the geodesic distance on a triangle mesh, based on the fast marching method. To speed up the matching process, this work employs a novel candidate point selection approach, which identifies the points of interest based on the shape irregularity of their fingerprints.

The point fingerprint was successfully applied to automatic registration of partially overlapped surfaces obtained from real range data. Experimental results demonstrated that the method can provide a good initial pose estimation for further ICP refinement. As an efficient point representation scheme, point fingerprint may also be applied to 3D object recognition tasks.

Multi-View Surface Reconstruction

The surface reconstruction algorithm employed in this research is implicit surface-based. Registered surface meshes from different views are put in a volumetric grid. Signed distances from each voxel to registered surfaces are computed and fused together. The reconstructed surface is extracted from the fused signed distance field.

This research incorporated fusion of color images in the volumetric grid to generate a textured surface. The automatic hole-filling algorithm is able to generate a watertight 3D model when the range data are incomplete due to self occlusions.

Depending on the completeness of the range data, the algorithm either carves the empty space to generate a watertight model directly, or faithfully reconstructs the surface by leaving holes that can be filled in the post-processing stage.

The algorithm is applied to reconstruct surfaces using various range scanners. It is adapted to small parts reverse engineering, indoor 3D mapping, and outdoor 3D scene reconstruction.

Fast Marching Watershed

This dissertation describes an approach to segmenting surfaces represented by triangle meshes, which is based on the robust edge detection using tensor voting and a fast marching watershed process.

Edge strength at all vertices defines a piecewise continuous height map on the triangle mesh. A watershed-based segmentation approach is applied to partition the surface based on the height map. One problem associated with watershed segmentation on triangle mesh is how to find the swiftest descending path on plateaus. A popular method in 2D watershed-based segmentation is partitioning plateaus by geodesic erosion from plateau boundaries. The fast marching watershed method extends the geodesic erosion to watershed-based segmentation of 3D triangle mesh. On a plateau, the descending path is traced back from the boundary to the inside of the plateau. The geodesic erosion guarantees the accurate segmentation of plateaus. The breadth-first algorithm cannot be used for geodesic erosion on the triangle mesh because a triangle mesh is a weighted graph. The geodesic distance on the triangle mesh is computed using the fast marching method, which is more accurate than Dijkstra's algorithm.

The experimental results show successful segmentation of various 3D models reconstructed from multi-view range scans of real objects. The segmentation makes it possible to manipulate and animate partitioned surfaces in 3D space and simplifies the 3D object recognition tasks.

8.2 Future Research

There are many opportunities to improve the whole framework. The most promising opportunity for new research is to make the surface registration more robust. Current research uses a simple threshold to sift corresponding point pairs. However, a risk of a simple threshold is to introduce false correspondences. To obtain the same number of corresponding points, the dissimilarity threshold needs to be larger for a noisy surface than for a smooth surface. How to automatically set such a threshold becomes another problem.

In most cases, a pair of corresponding points with the smallest dissimilarity measure match correctly. More corresponding point pairs can be confirmed by considering the geometric constraints from the known correspondences. For example, the second pair of corresponding points should be within approximately the same distance from the first pair of corresponding points. If they are not, other corresponding pairs can be inspected. Whenever a new correspondence is confirmed, the geometric constraint will be updated for further inspection.

This process of finding corresponding points is independent of the proposed point fingerprint scheme. However, it may significantly improve the matching result and the registration accuracy.

Future work also aims at applying point fingerprint to 3D object recognition.

Bibliography

Bibliography

[1] E. L. Allgower and P. H. Schmidt. An algorithm for piecewise linear approximation of an implicitly defined manifold. *SIAM Journal of Numerical Analysis*, 22:322–346, 1985.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH*, pages 415–420, 1998.

[3] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proc. of 6th ACM Symposium on Solid Modeling*, pages 249–260, 2001.

[4] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Finding surface correspondences for object recognition and registration using pairwise geometric histograms. In *Proc. European Conf. on Computer Vision*, volume II, pages 674–680, 1998.

[5] F. Bernardini, I. M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Trans. Visualization and Computer Graphics*, 7(4):318–332, 2001.

[6] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Visualization and Computer Graphics*, 5(4):349–359, 1999.

[7] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):239–256, 1992.

[8] P. J. Besl. *Surfaces in Range Image Understanding.* Springer-Verlag, 1988.

[9] P. J. Besl. The free-form surface matching problem. *Machine Vision for Three-Dimensional Scenes, H. Freeman, Ed.*, pages 25–71, 1990.

[10] A. Blake and A. Zisserman. *Visual Reconstruction.* Cambridge, Mass.: MIT Press, 1987.

[11] A. Bleau, J. D. Guise, and A. R. Leblanc. A new set of fast algorithm for mathematical morphology, I. idempotent geodesic transforms. *Computer Vision and Image Understanding*, 56(2):178–209, 1992.

[12] A. Bleau and L. J. Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77:317–370, 2000.

[13] J. D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, 1984.

[14] R. Campbell and P. Flynn. A WWW-accessible 3D image and model database for computer vision research. In *IEEE Computer Society Workshop on Empirical Evaluation Methods in Computer Vision*, pages 148–154, 1998.

[15] R. Campbell and P. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.

[16] R. J. Campbell and P. J. Flynn. Eigenshapes for 3D object recognition in range data. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 505–510, 1999.

[17] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 8(6):679–698, 1986.

[18] O. Carmichael, D. Huber, and M. Hebert. Large data sets and confusing scenes in 3-D surface matching and recognition. In *Proc. Int'l Conf. on Recent Advances in 3D Digital Imaging and Modeling*, pages 358–367, 1999.

[19] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 2724–2729, 1991.

[20] C. Chua and R. Jarvis. Point signatures: a new representation for 3D object recognition. *Int'l J. Computer Vision*, 25(1):63–85, 1997.

[21] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, pages 303–312, 1996.

[22] M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. SIGGRAPH*, pages 317–324, 1999.

[23] M. DoCarmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[24] M. DoCarmo. *Riemannian Geometry*. Birkhauser, 1992.

[25] C. Dorai and A. K. Jain. COSMOS – a representation scheme for 3D free-form objects. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(10):1115–1130, October 1997.

[26] G. Dziuk and J. E. Hutchinson. The discrete plateau problem: Algorithm and numerics. *Math. Comp.*, 68(225):1–23, 1999.

[27] G. Dziuk and J. E. Hutchinson. The discrete plateau problem: Convergence results. *Math. Comp.*, 68(226):519–546, 1999.

[28] H. Edelsbrunner and E. P. Mucke. Three dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[29] P. J. Flynn and A. K. Jain. On reliable curvature estimation. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 110–116, 1989.

[30] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. SIGGRAPH*, pages 209–216, 1997.

[31] J. Gomes and O. D. Faugeras. Level sets and distance functions. In *Proc. European Conf. on Computer Vision*, pages 588–602, 2000.

[32] R. C. Gonzalez. *Digital Image Processing.* Addison-Wesley, 1992.

[33] E. Guest, E. Berry, R. A. Baldock, M. Fidrich, and M. A. Smith. Robust point correspondence applied to two- and three-dimensional image registration. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(2):165–179, Feb. 2001.

[34] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume II. Addison-Wesley Publishing Company, 1993.

[35] M. Hebert, K. Ikeuchi, and H. Delingette. A spherical representation for recognition of free-form surfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(7):681–690, July 1995.

[36] A. Hilton, A. Stoddart, J. Illingworth, and T. Windeatt. Implicit surface-based geometric fusion. *Computer Vision and Image Understanding*, 69:273–291, 1998.

[37] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt. Marching triangles: Range image fusion for complex object modeling. In *Proc. IEEE Int'l Conf. on Image Processing*, pages 381–384, 1996.

[38] A. Hoover. *The Space Envelope Representation for 3D Scenes.* PhD thesis, University of South Florida, 1996.

[39] A. Hoover, G. J. Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(7):673–689, 1996.

[40] H. Hoppe, T. Derose, and T. Duchamp. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH*, volume 26, pages 71–78, 1992.

[41] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal-of-the-Optical-Society-of-America-A-(Optics-and-Image-Science)*, 5(7):1127–1135, July 1988.

[42] J. Huang and C. Menq. Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points. *IEEE Trans. Robotics and Automation*, 17(3):268–279, 2001.

[43] Integrated Vision Products, Sweden. *User Documentation: MAPP Ranger System*, 2000. Version 1.6.

[44] X. Jiang, H. Bunke, and U. Meier. High-level feature based range image segmentation. *Image and Vision Computing*, 18:817–822, 2000.

[45] A. E. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proc. Int'l Conf. on Recent Advances in 3D Digital Imaging and Modeling*, pages 12–15, 1997.

[46] A. K. Katsaggelos. Iterative image restoration algorithms. *Optical Engineering*, 28(7):735–748, 1989.

[47] R. Kimmel, A. Amir, and A. M. Bruckstein. Find shortest paths on surfaces using level sets propagation. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(6):635–640, June 1995.

[48] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. In *Proc. National Academy of Sciences*, pages 8431–8435, July 1998.

[49] L. Kobbelt, S. Campagna, J. Vorsatz, and H. P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. SIGGRAPH*, pages 105–114, 1998.

[50] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 238–249, 1988.

[51] C. Lantuejoul and F. Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern Recognition*, 17(2):177–187, 1984.

[52] D. Laurendeau, G. Roth, and L. Borgeat. Optimization algorithms for range image registration. In *Proc. of Vision Interface*, pages 141–151, 1996.

[53] X. Li, T. Woon, T. Tan, and Z. Huang. Decomposing polygon meshes for interactive applications. In *ACM Symp. on Interactive 3D Graphics*, pages 35–42, 2001.

[54] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH*, pages 163–169, 1987.

[55] A. P. Mangan and R. T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. Visualization and Computer Graphics*, 5(4):308–321, 1999.

[56] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, 1995.

[57] G. Medioni, M. S. Lee, and C. K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier Science B.V., 2000.

[58] D. M. Mount. *ANN Programming Manual*. Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, Maryland.

[59] J. Neider, T. Davis, and M. Woo. *OpenGL Programming Guide*. Addison-Wesley Publishing Company, 1993.

[60] Y. Ohtake, A. Belyaev, and I. Bogaevski. Polyhedral surface smoothing with simultaneous mesh regularization. In *Proc. Geometric Modeling and Processing*, pages 229–237, 2000.

[61] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, second edition, 1998.

[62] D. L. Page, A. Koschan, Y. Sun, J. K. Paik, and M. A. Abidi. Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 162–167, 2001.

[63] D. L. Page, Y. Sun, A. Koschan, J. K. Paik, and M. A. Abidi. Normal vector voting: crease detection and curvature estimation on large, noisy meshes. To appear in Graphical Models, 2003.

[64] D. L. Page, Y. Sun, A. Koschan, J. K. Paik, and M. A. Abidi. Simultaneous mesh simplification and noise smoothing of range images. In *Proc. IEEE Int'l Conf. on Image Processing*, volume III, pages 821–824, 2002.

[65] Perceptron Inc., 23855 Research Drive, Farmington Hills, Michigan 48335. *LASAR Hardware Manual*, 1993.

[66] R. Pito. Mesh integration based on co-measurements. In *Proc. IEEE Int'l Conf. on Image Processing*, pages 397–400, 1996.

[67] R. J. Prokop and Anthony P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438–460, September 1992.

[68] K. Pulli. Multiview registration for large data sets. In *Proc. Int'l Conf. on Recent Advances in 3D Digital Imaging and Modeling*, pages 160–168, 1999.

[69] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle. Robust meshes from multiple range maps. In *Proc. Int'l Conf. on Recent Advances in 3D Digital Imaging and Modeling*, pages 205–211, 1997.

[70] M. E. Rettmann, X. Han, C. Xu, and J. L. Prince. Automated sulcal segmentation using watersheds on the cortical surface. *NeuroImage*, 15:329–344, 2002.

[71] RIEGL Laser Measurement Systems. *Laser Mirror Scanner LMS-Z210, Technical Documentation and User's Instructions*, 2000.

[72] J. R. Roerdink and A. Meijster. The watershed transform: definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2001.

[73] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proc. Int'l Conf. on Recent Advances in 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[74] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 573–580, 1994.

[75] A. D. Sappa and M. A. Garcia. Incremental multiview integration of range images. In *Proc. IEEE Int'l Conf. on Pattern Recognition*, pages 546–549, 2000.

[76] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.

[77] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Sciences.* Cambridge University Press, second edition, 1998.

[78] J. A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999.

[79] C. A. Shaffer. *A Practical Introduction to Data Structure and Algorithm Analysis.* Prentice-Hall, 1997.

[80] J. H. Shin, Y. Sun, J. K. Paik, and M. A. Abidi. Adaptive regularized noise smoothing of dense range image using directional laplacian operators. In *Proc. SPIE, The International Society for Optical Engineering*, volume 4298, pages 119–126, 2001.

[81] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(4):344–358, April 1995.

[82] F. Stein and G. Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):125–145, February 1992.

[83] R. L. Stevenson and E. J. Delp. Viewpoint invariant recovery of visual surface from sparse data. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(9):257–270, September 1992.

[84] Y. Sun and M. A. Abidi. Surface matching by 3D point's fingerprint. In *Proc. IEEE Int'l Conf. on Computer Vision*, volume II, pages 263–269, 2001.

[85] Y. Sun, C. Dumont, and M. A. Abidi. Mesh-based integration of range and color images. In *SPIE's 14th International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, volume 4051, pages 110–117, 2000.

[86] Y. Sun, D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi. Triangle mesh-based surface modeling using adaptive smoothing and implicit texture integration. In *Proc. First Int'l Symposium on 3D Data Processing Visualization and Transmission*, pages 588–597, 2002.

[87] Y. Sun, D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi. Triangle meshes-based edge detection and its application to surface segmentation and adaptive surface smoothing. In *Proc. IEEE Int'l Conf. on Image Processing*, volume III, pages 825–828, 2002.

[88] Y. Sun, J. K. Paik, A. Koschan, and M. A. Abidi. 3D reconstruction of indoor and outdoor scenes using a mobile range scanner. Accepted by IEEE Int'l Conf. on Pattern Recognition 2002.

[89] Y. Sun, J. K. Paik, A. Koschan, and M. A. Abidi. Surface modeling using multi-view range and color images. Accepted by The Journal of Integrated Computer-Aided Engineering.

[90] Y. Sun, J. K. Paik, A. Koschan, D. L. Page, and M. A. Abidi. Point's fingerprint: a new 3D object representation scheme for surface registration. In revision, IEEE Trans. Systems, Man and Cybernetics, Part B.

[91] Y. Sun, J. K. Paik, J. R. Price, and M. A. Abidi. Dense range image smoothing using adaptive regularization. In *Proc. IEEE Int'l Conf. on Image Processing*, volume II, pages 744–747, 2000.

[92] C. Tang and Gerard Medioni. Robust estimation of curvature information from noisy 3D data for shape description. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 426–433, 1999.

[93] G. Taubin. A signal processing approach to fair surface design. In *Proc. SIGGRAPH*, pages 351–358, 1995.

[94] J. Thirion. Extremal points: Definition and application to 3D image registration. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 587–592, 1994.

[95] E. Trucco, A. Fusiello, and V. Roberto. Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letter*, 20(9):889–898, September 1999.

[96] G. Turk and M. Levoy. Zippered polygon meshes form range images. In *Proc. SIGGRAPH*, pages 311–318, 1994.

[97] N. M. Vaidya and K. L. Boyer. Discontinuity-preserving surface reconstruction using stochastic differential equations. *Computer Vision and Image Understanding*, 72(3):257–270, December 1998.

[98] L. Vincent and P. Soille. Watershed in digital spaces: an efficient algorithm based on immersion simulation. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(6):583–598, 1991.

[99] J. Vollmer, R. Mencl, and H. Muller. Improved Laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum(Proc. Eurographics 1999)*, pages 131–138, 1999.

[100] Y. Wang, B. Peterson, and L. Staib. Shape-based 3D surface correspondence using geodesic and local geometry. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume II, pages 663–668, 2000.

[101] R. T. Whitaker. A level-set approach to 3D reconstruction from range data. *Int'l J. Computer Vision*, 29(3):203–231, 1998.

[102] H. Woo, E. Kang, S. Wang, and K. H. Lee. A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture*, 42:167–178, 2002.

[103] K. Wu and M. D. Levine. 3D part segmentation using simulated electrical charge distributions. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(11):1223–1235, 1997.

[104] H. Yahia, E. Huot, I. Herlin, and I. Cohen. Geodesic distance evolution of surfaces: a new method for matching surfaces. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume I, pages 644–651, 2000.

[105] S. M. Yamany and A. A. Farag. Free-form surface registration using surface signatures. In *Proc. IEEE Int'l Conf. on Computer Vision*, pages 1098–1104, 1999.

[106] S. M. Yamany, A. A. Farag, and A. El-Bialy. Free-form surface recognition and recognition using surface signatures. In *Proc. IEEE Int'l Conf. on Image Processing*, pages 457–461, 1999.

[107] J. H. Yi and D. M. Chelberg. Discontinuity-preserving and viewpoint invariant reconstruction of visible surface using a first order regularization. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(6):624–629, June 1995.

[108] Y. Yu, A. Ferencz, and J. Malik. Extracting objects from range and radiance images. *IEEE Trans. Visualization and Computer Graphics*, 7(4):351–364, 2001.

[109] D. Zhang. *Harmonic Shape Images: A 3D Free-Form Surface Representation and Its Application in Surface Matching*. PhD thesis, Carnegie Mellon University, 1999.

[110] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume II, pages 524–530, 1999.

[111] Y. Zhang, Y. Sun, S. Sarraf, J. R. Price, and M. A. Abidi. Impact of intensity edge map on segmentation of noisy range images. In *Proc. SPIE, The International Society for Optical Engineering*, volume 3958, pages 260–269, 2000.

[112] H. K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.

Appendix

Appendix A

Laser Range Scanners

Laser range finders make more accurate measurements than stereo vision-based techniques in digitizing surfaces of real 3D objects. In this research, several laser range scanners are used for 3D reconstruction, including PERCEPTRON Laser System [65], RIEGL-Z210 Laser Mirror Scanner [71], COLEMAN Scanner, and IVP RANGER 3D Profiling System [43]. Most laser range scanners available today are based on time-of-flight and laser triangulation, which are explained as follows.

A.1 Scanners Based on Time-of-Flight

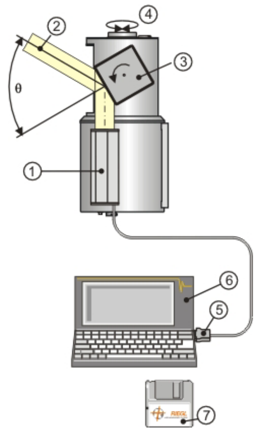
The scanners based on the time-of-flight send out laser beam and detect the reflection. By measuring the light traveling time, the distance between the scanner and the object where the laser hits can be calculated. PERCEPTRON, RIEGL, and COLEMAN scanners fall into this category. Being one of the major scanners employed in this research, the RIEGL system is described here as an example of the scanners based on time-of-flight.

Fig. A.1(a) shows the RIEGL 3D-Laser Mirror Scanner LMS-Z210 [71]. The 3D images are gained by performing a number of independent laser range measurements in different, but well-defined angular directions. These range data together with the associated angles form the basis of the 3D images. The scanner consists of a laser range finder unit and a two axis beam scanning mechanism.

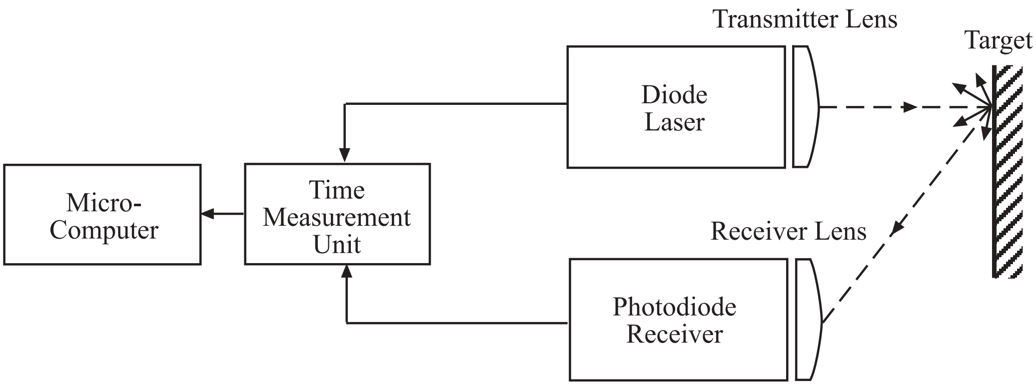
An electrical pulse generator periodically drives a semiconductor laser diode sending out infrared light pulses, which are collimated by transmitter lens. Via the receiver lens, part of the echo signal reflected by the target hits a photodiode, which generates an electrical receiver signal. The time interval between transmitted and received pulses are counted by means of a quartz-stabilized clock frequency. The calculated range value is fed into the internal microcomputer which processes the measured data. Figs. A.1(b) and A.1(c) show the principle of the scanner operation.

The scanner directs the laser beam for range measurement in a precisely defined position. A 3D image is obtained by scanning a number of lines which are composed of a number of pixels. To scan a vertical line, the angular deflection of the laser beam is realized by a rotating polygon mirror wheel. The frame scanner mechanism relies on rotating the optical head together with the fast line scan mechanism, accomplished by mounting both the line scanner mechanism and the optical head on a rotating table. The components one to six in Fig. A.1(b) represent range finder electronics, laser

(a)



(b)



(c)

Figure A.1: Laser range scanners based on time-of-flight. (a) RIEGL LMS-Z210 Laser Mirror Scanner. (b) RIEGL scanner operating principle. (c) Measurement principle of a pulsed range finder.

beam, rotating mirror, optical head, parallel port for data communicator, computer, and software for sensor configuration and data acquisition.

Although the measurement from laser range scanners has a higher accuracy than stereo vision-based systems, the acquired range signals are, however, still corrupted by noise. Noise may come from the error introduced by the motor that drives the rotating table. Even the slightest vibration of the system causes a certain amount of error in the acquired 3D geometry. Both the error in detecting the reflected pulses and the round-up error from the clock in the time measurement unit contribute to the total measurement error. The accuracy degrades for long distance measurement due to the weak echo signal. The accuracy also depends on the target material. Black objects tend to absorb the light, and specular objects tend to reflect the light. In the extreme cases, the echo signal cannot be detected and the measurement fails.

The measurement error of RIEGL LMS-Z210 has a standard deviation of 5 cm for retroreflecting targets in a distance up to 700 m, or for natural targets in a distance up to 450 m. The standard deviation is 2.5 cm for natural targets at a distance up to 350 m. The performance of the scanner also depends on the weather. For example, in bright sunlight, the operational range of the scanner is considerably shorter than under an overcast sky.

A.2 Scanners Based on Laser Triangulation

Many active range imaging techniques use a 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) are the triangulation parameters.

The most common active triangulation methods include illumination with a single spot, a sheet of light, and coded light, as seen in Fig. A.2. The single-spot technique requires advanced mechanics to allow the spot to reach the whole scene. The coded-light system requires a high-intensity projector that can switch between patterns as fast as the sensor can integrate images. In the case of sheet-of-light systems, the projection of the light can be done with one single scanning mirror which is considerably simpler than the projector design for spatially coded light, or the two mirror arrangement for single spot illumination. Actually, in most sheet-of-light systems the sheet of light is not swept at all. Instead the apparatus itself or the scene is moving. For example, the IVP RANGER System used in this research, which is based on sheet-of-light projection as shown in Fig. A.2(b), uses a conveyor belt to move the object so that the whole scene can be reached by the light, as seen in Fig. A.3(a). To make a sheet of light, the sharp laser spot-light passes through a lens and the lens spreads the light into a sheet in one dimension.

The high speed of electromagnetic waves makes time-of-fight methods difficult to use for high accuracy range imaging since small differences in range have to be resolved

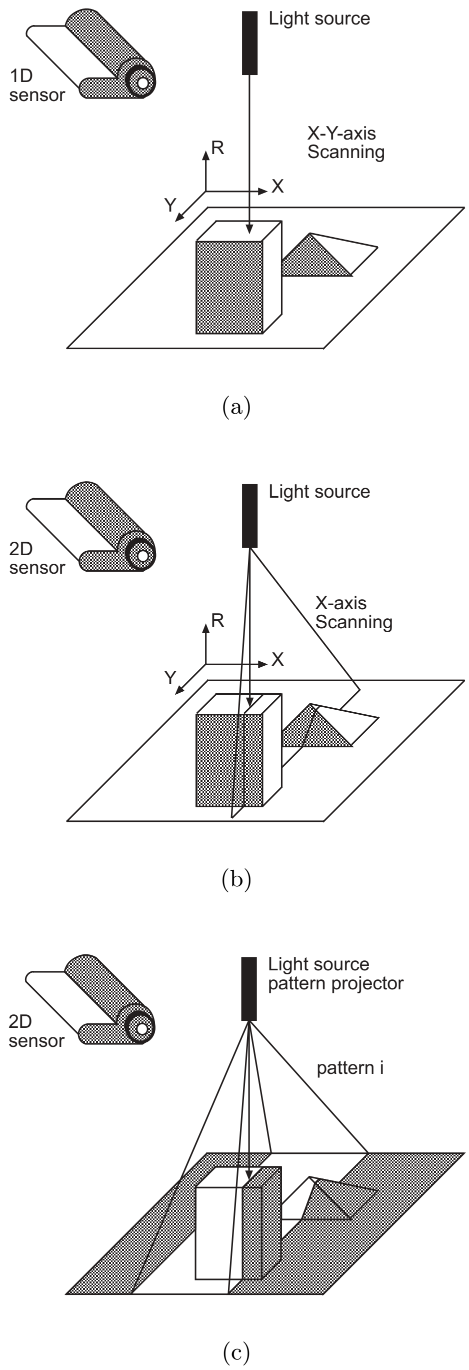
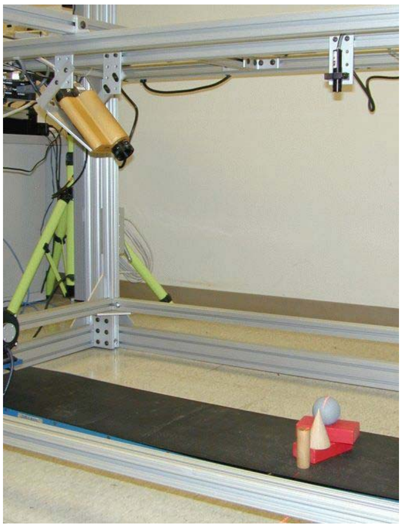
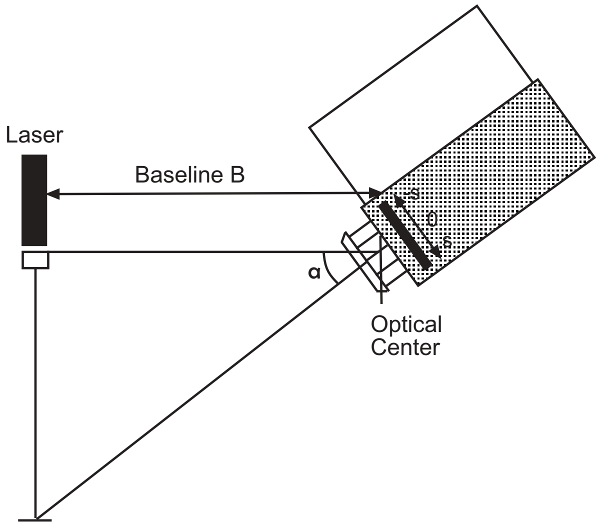


Figure A.2: Laser range scanners based on triangulation. (a) Single-spot range imaging. (b) Sheet-of-light range imaging. (c) Coded-light range imaging.



(a)



(b)

Figure A.3: IVP RANGER 3D Profiling System. (a) IVP Ranger scanner in use. (b) Description of the System's setup.

by extremely fine discriminations in time. Therefore, it is more appropriate to use a triangulation-based scanner to achieve high accuracy for scanning small objects. Actually, range imaging based on triangulation is only effective for short range distances because the baseline should be at least in the same order of magnitude as the range distance, and a large baseline for long distance scan will make the system too big to use. In this research, the RANGER System is used for surface modeling of small parts with millimeter accuracy, and the RIEGL scanner is used for 3D reconstruction of indoor and outdoor scenes with centimeter accuracy.

For triangulation-based scanners, the limited resolution of the sensor limits the ranging accuracy. For a popular setup in Fig. A.3(b), if angle α increases, the accuracy decreases since a small range interval Δr on the object will be projected on a smaller interval Δs on the sensor. Also if the angle α increases, the focus of the line decreases since a larger focal depth is required. To obtain high resolution the laser sheet should cover several pixels, so that an accurate estimate of the peak position can be found. However, a thick laser sheet may cause ambiguity in range determination. Because the system relies on the sensor seeing the sheet of light shed on the object, strong background illumination also affects the measurement accuracy, especially when a filter is not used. Similar to RIEGL scanner, the performance of RANGER System is also sensitive to the target material; accuracy decreases for black or specular objects.

Appendix B

Simplification of the Area-Decreasing Stabilizer

This appendix proves that the minimizer of an area integration of the square-root of a function is equivalent to that of the same area integration of the function without the square root. In other words, justification is shown that the minimizer of (3.14) is equivalent to that of (3.17).

Define

$$\xi_{iN_0+j} = \sqrt{E_{ij}G_{ij} - F_{ij}^2} \quad \text{and} \quad \eta_{iN_0+j} = 1, \tag{B.1}$$

for $i < M_0$, $j < N_0$, and

$$\xi_k = \eta_k = 0 \tag{B.2}$$

for $k > M_0 N_0$. Using the Cauchy-Schwarz inequality, we obtain

$$\sum_{j=1}^{\infty} |\xi_j \eta_j| \leq \sqrt{\sum_{k=1}^{\infty} |\xi_k|^2} \sqrt{\sum_{m=1}^{\infty} |\eta_m|^2}, \tag{B.3}$$

where $\sum_{j=1}^{\infty} |\xi_j|^2 < \infty$ and $\sum_{j=1}^{\infty} |\eta_j|^2 < \infty$ because only a finite number of terms are nonzero. This then yields

$$\sum_{i,j}^{M_0 N_0} \sqrt{E_{ij} G_{ij} - F_{ij}^2} \leq \sqrt{M_0 N_0 \sum_{i,j}^{M_0 N_0} (E_{ij} G_{ij} - F_{ij}^2)}, \tag{B.4}$$

which shows that the minimizer of (3.17) implies that of (3.14).

Vita

Yiyong Sun was born in Beijing, China on November 4, 1972. He graduated from Beijing 15th Middle School in 1990. He entered the Department of Automatic Control at the Beijing Polytechnic University and received a B.S. and an M.S. in Electrical Engineering, in 1995 and 1998, respectively. His research interest was in Sliding Mode Variable Structure Control.

He entered the University of Tennessee, Knoxville, in August 1998. Since then, he has been studying in the Department of Electrical and Computer Engineering and working as a research assistant in the Imaging, Robotics, and Intelligent Systems Laboratory. He received his PhD in Electrical Engineering in December, 2002. His research interest is in computer vision and image processing.