3D Surface Matching from Range Images using Multiscale Local Features

By

Huy Tho Ho

B.E. (First Class Honours)

A thesis submitted in partial fulfillment of the

requirements for the degree of

Master of Applied Science

School of Electrical and Electronic Engineering

The University of Adelaide

Australia

December 2009

Contents

Abstract

Object recognition is one of the most important problems in computer vision. Traditional object recognition techniques are usually performed on optical images that are 2D projections of the 3D world. Information about the depth of objects in the scene is not provided explicitly in these images and thus, it makes 2D object recognition techniques sensitive to changes in illumination and shadowing. As surface acquisition methods such as LADAR or range scanners are becoming more popular, there is an increasing interest in the use of three-dimensional geometric data in object recognition to overcome these limitations.

However, the matching of 3D free-form surfaces is also a difficult problem due to the shape and topological complexity of 3D surfaces. In addition, the problem is further complicated by other issues such as variations in surface sampling resolution, occlusion, clutter and sensor noise. The huge amount of information required to describe a 3D surface is also another challenge that 3D surface matching techniques have to deal with.

This thesis investigates the problems of 3D surface matching that include 3D surface registration and object recognition from range images. It focuses on developing a novel and efficient framework for aligning 3D surfaces in different coordinate systems and from this, recognizing 3D models from scenes with high levels of occlusion and clutter using multi-scale local features.

The first part of the thesis presents two different schemes for extracting salient geometric features from 3D surfaces using surface curvature measures known as the curvedness and shape index. By deriving the scale-space representation of the input surface, surface positions with high local curvature or high local shape variations are selected as features at various degrees of scale. One advantage of the proposed approaches is their applicability to both 3D meshes with connectivity information and unstructured point clouds.

In the second part of the thesis, an application of the multi-scale feature extraction framework to 3D surface registration and object recognition is proposed. A Delaunay tetrahedrization is performed on the features extracted from each input range image to obtain a set of triangles. Possible correspondences are found by matching all possible pairs of triangles between the scene and model surfaces. From these correspondences, possible transformations between the two surfaces can be hypothesized and tested. In order to increase the accuracy and efficiency of the algorithm, various surface geometric and rigidity constraints are applied to prune unlikely correspondences. By finding the match that aligns the largest number of features between the two surfaces, the best transformation can be estimated. In the case of surface registration, this transformation can be used to coarse-align two different views of the same object. In the case of 3D object recognition, it provides information about the possible pose (location and orientation) of the model in the scene surface. Experimental results on a variety of 3D models and real scenes are shown to verify the effectiveness and robustness of the approach.

Declaration

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution to Huy Tho Ho and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis (as listed in the publications page) resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue, the Australasian Digital Thesis Program (ADTP) and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Huy Tho Ho

December 2009

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to my supervisors, Dr. Danny Gibbins and Prof. Douglas A. Gray, for their valuable guidance, support and encouragement throughout my Master research. Completion of the thesis would not have been possible without their encouragement and time.

I would also like to thank the School of Electrical & Electronic Engineering, Dr. Danny Gibbins and Prof. Douglas A. Gray for providing the Master scholarship as well as their support for my travel to conferences.

I am grateful to the institutions and people who made their data and code publicly available, including Dr. Ajmal Mian (3D data), Carnegie Mellon University (Mesh Toolbox), Standford University (3D data), Universität Stuttgart (3D data). I would also like to thank the Defence Science and Technology Organisation (DSTO) for providing the vehicle simulation data used in the experiments.

Finally, I would like to thank my grandmother, my parents and my brother for their unconditional love and support. Last of all, thank you, Thanh, for always being there for me.

List of Publications

The following papers have been written based on the materials presented in this thesis

1. H.T. Ho and D. Gibbins, "Multi-scale Feature Extraction from 3D Models with Applications to Surface Registration", to appear in *IET Computer Vision Journal*, 2009.

2. H.T. Ho and D. Gibbins, "Multi-scale Feature Extraction from 3D Models using Local Surface Curvature", *Proceedings of Digital Image Computing: Techniques and Applications (DICTA'08)*, Canberra, Australia, December 2008.

3. H.T. Ho and D. Gibbins, "Multi-scale Feature Extraction for 3D Surface Registration using Local Shape Variation", *Proceedings of Image and Vision Computing New Zealand (IVCNZ'08)*, Christchurch, New Zealand, November 2008.

List of Tables

List of Figures

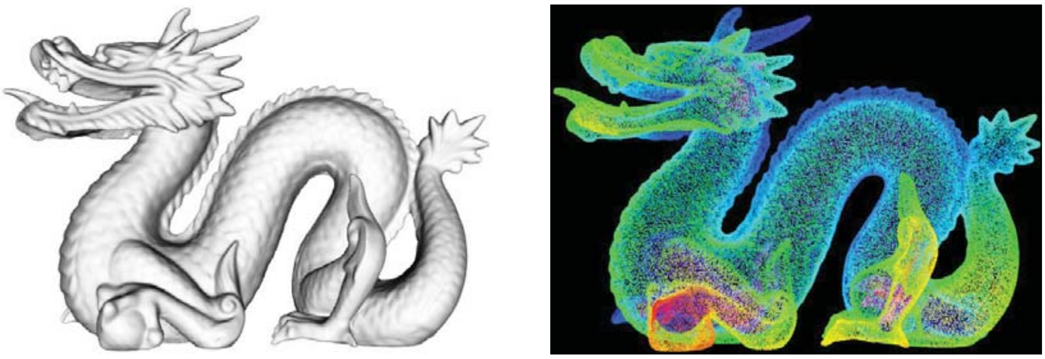
Chapter 1

Introduction

1.1 Motivation

Object recognition is one of the most important problems in computer vision. It is the process of determining the identities, locations and orientations of objects in an image. There are many applications of object recognition including, but not limited to, autonomous navigation and surveillance [22], industrial machine vision [28], content-base image retrieval [42] and medical image analysis [14].

Traditional object recognition techniques are usually performed on optical images that are 2D projections of the 3D world. Information about the depth of objects in the scene is not provided explicitly in these images. As most 2D recognition algorithms use the intensity values of the images to recognize objects, they are very sensitive to changes in illumination and shadowing. This is not the case in 3D object recognition from range input data [46, 56]. A range image is a set of 3D points representing a surface obtained by using specialized sensor hardwares or reconstructed from two or more regular images using multiple view geometry. If the connectivity information between the points is available in the range image, the surface is represented as a mesh with vertices and facets. If the range image does not contain this connectivity information, the surface is represented as a point cloud. Figure



(a) Mesh (b) Point Cloud

Figure 1.1: 3D surface of the ‘Dragon’ model. (a) The surface is represented as a mesh with connectivity information between the vertices. (b) The surface is represented as a point cloud. The point cloud is colour-coded using the z coordinates of the 3D points.

1.1 shows an example of a 3D surface of the ‘Dragon’ model represented by both a mesh and a point cloud. Range images provide more complete information about object geometry and thus allow the geometric relations to be compared between the scene and the training data [5]. As surface acquisition methods such as LADAR (Laser Detection and Ranging) or range scanners are becoming more popular, there is an increasing interest in the use of three-dimensional geometric data in not only object recognition but also other computer vision applications.

The problem of 3D object recognition from range images can be considered as the process of matching the shape of the object’s surface to the shape of the scene’s surface [34]. The way 3D data are represented decides how the recognition system will process this information [54]. There are many different ways to represent the shapes from the information stored in the sensed points of the range images. For instance, surface shapes can be represented in parametric forms such as the nonuniform rational B-spline (NURBS) [41]. The geometry of surfaces can also be characterized globally using deformable shapes [16] or locally using surface curvature [13]. Choosing the appropriate representation for shape in order to compare surfaces efficiently and accurately is still considered an open research question in computer vision [26].

The matching of 3D free-form surfaces is a very difficult problem due to the shape and topological complexity of 3D surfaces. Two different surfaces representing the same object may have different topologies, thus making the global matching between them difficult to conduct [87]. Variations in sampling resolution are another challenge that 3D surface matching techniques have to deal with. Such resolution differences make it difficult to compare two surfaces as it is hard to establish the point-wise correspondences between them. In addition, the problem is further complicated by partial occlusion and the present of other structures such as clutter in the scene. A surface matching algorithm must be able to handle scenes with clutter and occlusion in order to be applicable to the real world.

This thesis investigates the problems of 3D surface matching that include 3D surface registration and object recognition from range images. It proposes a novel and efficient framework for aligning 3D surfaces in different coordinate systems which can be used in recognizing 3D models from scenes with high levels of occlusion and clutter using multi-scale local features. The ability of this novel framework is demonstrated using a standard set of 3D test objects and cluttered scenes.

1.2 Outline of the Thesis

In this section, an overview of the structure of the thesis is presented. In Chapter 2, several techniques for matching 3D surfaces represented by range images are examined. Matching methods using local surface descriptors are particularly considered as they prove to be robust and less sensitive to occlusion and clutter than other approaches. A review of different techniques for extracting local geometric features from 3D surfaces which have application to 3D surface matching is also presented in this chapter.

In Chapter 3, two different schemes for extracting salient features from 3D datasets using local surface curvature are discussed. Two surface curvature measures known as curvedness and shape index are used to create the scale-space representations of

3D surfaces.

Chapter 4 presents a method for registering 3D scans that differ by a rigid body transformation using a proposed multi-scale feature extraction framework. The proposed approach is also compared with the spin-image algorithm [35] in order to evaluate its robustness and effectiveness.

In Chapter 5, results of using multi-scale features in recognizing 3D models from scenes with occlusion and clutter are described. Experimental results on a number of different 3D models and scenes are presented to measure the merits of the proposed approach.

Finally, Chapter 6 summarizes the main contributions of the thesis and discusses possible future directions for this research.

1.3 Contributions

The following is an outline of the key contributions of the research:

- Two novel schemes for extracting salient features from 3D datasets at multiple scales using local surface curvature. Features can be selected as surface positions with either high local curvature or high shape variation. Methods for building the scale-space representations of a 3D surface are also presented. The proposed framework can be applied to both 3D meshes with connectivity information between the vertices and unstructured point clouds. Another advantage of the proposed method is that it does not introduce any shrinking artefact to the input surface unlike other approaches using smoothing filters [82].

- An automatic and unified framework for the registration of both 3D meshes and point clouds using multi-scale local features, assuming that they differ by a rigid body transformation. A Delaunay tetrahedrization is performed on the features extracted from each input range image to obtain a set of triangles. Possible correspondences are found by matching all possible pairs of triangles between the scene and model surfaces. From these correspondences, possible transformations between the two surfaces can be hypothesized and tested. In order to increase the accuracy and efficiency of the algorithm, various surface geometric and rigidity constraints can also be applied to prune unlikely correspondences. By finding the match that aligns the largest number of features between the two surfaces, the best transformation is able to be estimated. The proposed method outperforms the spin-images [35] in terms of accuracy and efficiency when compared on the same datasets. It is also able to provide good results for noisy surfaces and those sampled at different resolutions.

- Automatic 3D object recognition from range images with occlusion and clutter based on the same triangle pair matching approach. Here the quality of the registration fit is used to determine the correct matched object. The proposed approach is able to recognize a 3D model from highly occluded scenes with the presence of other objects. It can also handle scenes with noise and variations in surface resolution.

Chapter 2

Background

2.1 Introduction

This chapter provides an overview of several popular techniques for matching 3D surfaces represented by range images. Many 3D surface matching methods have been proposed in the literature including appearance-based techniques [38, 60, 81, 4], methods that perform the matching using object silhouettes [59, 66], 3D correlation techniques [69, 25], algorithms based on an exhaustive search for corresponding points [9, 11] and approaches that use local surface descriptors [12, 19, 36, 27, 23, 56, 46]. However, none of these techniques could fully meet all the important requirements of 3D surface matching such as:

- Applicability to free-form surfaces represented by either 3D meshes or unstructured point clouds.

- Robustness to noise and surface resolution.

- Accuracy and efficiency.

- Ability to handle occlusion and clutter.

In addition, this chapter also discusses different approaches to the problem of feature extraction from range images which have applications to 3D surface registration and object recognition. These feature extraction algorithms have a common goal of providing a systematic way to obtain a representative set of features from the input surface. The selected features can be used to improve the effectiveness as well as the robustness of any 3D surface matching algorithm.

2.2 3D Surface Matching Algorithms

2.2.1 Appearance-based Approaches

In an appearance-based approach, each individual object view is encoded as a point in one or more multidimensional spaces [5]. From a statistical analysis of the training images, the bases for these spaces can be built using Principle Component Analysis (PCA) [60]. The basis vectors are termed *eigenpictures* or *eigenimages* for 2D imagery and *eigenshapes* for 3D imagery [5]. An unknown view is recognized by projecting this view into the spaces along the basis vectors. A match can be found by finding the nearest projected view of a training image. The advantage of this method is that it is quite robust under changes of illumination, shape, pose and reflectance [67]. However, this approach lacks the ability to handle more than one object in the scene with the possibility of occlusion [30]. In order to handle scenes with multiple objects, the input image needs to be segmented into parts before performing the recognition on these parts to identify the objects as well as their pose [5]. The major disadvantage is that only objects that are reliably segmented can be recognized. Furthermore, the segmentation of 3D scenes is not always a trivial process [29, 86].

2.2.2 Silhouette-based Approaches

Object silhouettes can be used to characterize free-form objects in order to determine their identities and pose [5]. A complete system developed by Mokhtarian [59] based on closed object silhouettes can be used to perform object recognition. This system has the same limitation as the appearance-based approach; it can only work when there is no more than one object in the scene. Another approach proposed by [66] could handle multiple object scenes with a moderate amount of occlusion by using contours of discontinuous surface normals and occluding contours to recognize 3D objects. Another system that also employs object silhouettes to recognize 3D objects is the one proposed by Joshi *et al.* [37]. In this technique, HOT (high-order tangent) curves identified from points on an object's silhouette are used to model a smooth object for recognition. The disadvantage of all of these approaches is that they can only be used in a controlled environment [5].

2.2.3 Correlation Filter Approaches

Correlation techniques have been used extensively for object recognition in 2D imagery [51, 50]. Some research has also been carried out in the area of 3D object recognition using correlation [69, 25]. A technique using three-dimensional correlation of range images to identify an object in a 3D scene was presented in [25]. In this approach, a 3D discrete Fourier Transform algorithm (3D-DFT) is performed on both the scene and reference data volume $\mathbf{r}(x, y, z)$ and $\mathbf{h}(x, y, z)$, respectively. The 3D correlation is the inverse 3D-DFT of the product of these two transforms $R(u, v, w)$ and $H(u, v, w)$:

$$C(x, y, z) = \text{3D-DFT}^{-1}[R(u, v, w)H^*(u, v, w)] \tag{2.1}$$

where H^* is the complex conjugate of H. The position of the peak in the correlation surface $C(x, y, z)$ denotes the position of the object in the scene. However, the main limitations of this approach are the inability to handle pose variations, occlusion and the presence of clutter in the scene.

2.2.4 Exhaustive Search Approaches

In order to align two views with n data points each, it is necessary to find a minimum of three non-collinear corresponding point pairs in order to estimate the rigid transformation. By applying techniques based on an exhaustive search for corresponding points, the best solution is guaranteed to be found. However, even with a medium size dataset, the search space is still very large ($C_3^n C_n^3$), and thus increasing the computational cost of the technique (complexity of $O(n^6)$) [55]. One of the most popular approaches based on an exhaustive search is the RANSAC-based DARCES (Random Sample Consensus based Data Aligned Rigidity Constrained Exhaustive Search) [9].

In the RANSAC-based DARCES, a triangle containing three control points S_p (primary), S_s (secondary) and S_a (auxiliary) is formed from a set of reference points created by uniformly sampling the scene surface. A possible correspondence M_p for the primary point S_p can be chosen from any point in the model data set. Once M_p is selected, a sphere C_s whose center is M_p and radius $d_{ps} = ||S_p - S_s||$ is used to constrain the search for M_s - the corresponding point of S_s. If S_q is the orthogonal projection of S_a onto the line segment S_pS_s and M_q is the corresponding point to S_q, the search for M_a, the corresponding model point of S_a, is limited to a circle centered at M_q with radius $d_{qa} = ||S_q - S_a||$. Figure 2.1 shows the search regions for M_s and M_a used in the approach. After the three corresponding points have been found on the model surface, a rigid transformation $\mathbf{T}_c$ can be determined from the three point pairs. The transformation is then verified by transforming all the scene reference points by T_c and then counting the number of occurrences, N_o, on the model surface. An occurrence is counted if the distance from a transformed scene reference point $S_{r'}$ to the model surface is smaller than a threshold. N_o is called the *overlapping number* and it is calculated for each possible three point correspondences. The transformation $\mathbf{T}_c$ with the largest N_o is selected as the solution. More than 3 control points can be used to improve the efficiency of the algorithm

Figure 2.1: (a) Three control points S_p, S_s and S_a. S_q is the orthogonal projection of S_a onto the line segment S_pS_s. (b) The search region for M_s. (c) The search region for M_a. (Figures taken from [55]).

by finding all their possible candidates before computing the overlapping number.

The main drawback of the RANSAC-based DARCES is its high computational cost as it is based on an exhaustive search. The technique is also sensitive to variations in surface sampling and noise since it uses a point-wise rather than a surface-wise approach [55].

2.2.5 Matching using 3D Surface Descriptors

Matching 3D surface descriptors has become one of the most effective methods in 3D surface registration and object recognition for range images. In this approach, a region of points is characterized in such a way that it can be recognized between different datasets representing the same object. The representations of these regions of points are called 3D surface descriptors and they need to be both robust and discriminative. The key problem lies in finding pose independent 3D surface descriptors that can capture the local geometrical characteristics of surfaces effectively and how to match the surfaces using the selected representation [10].

In the early literature, Nevatia and Binford [61] proposed a method that uses symbolic descriptions derived from a generalized cone part segmentation of range images. This technique could recognize free-form articulated objects (doll, horse, snake, glove, and a ring) in the presence of occlusion. However, this approach only works for objects with very different symbolic descriptions if there are multiple objects in the scene [5]. Another approach developed by Raja and Jain [68] fits a deformable superquadric to range data in order to classify the shape into one of 12 different shape classes. Superquadrics are geometric shapes that are generalizations of quadrics (ellipsoids, hyperboloids of one or two sheets, paraboloids). Their formulae resemble those of quadrics, except that the power on the coordinate does not have to be 2 [1]. This method is very sensitive to noise and discontinuities in rough range images and it cannot handle occlusion in the scene [5]. Another class of methods uses surface curvature-based features for 3D object recognition such as Gaussian curvature [80], mean curvature [2], principal curvatures [19]. In these methods, the radius for collecting surface points to estimate the local curvature is fixed. In other words, as the local surface curvature is estimated at only a single scale, they are very sensitive to noise and also can not capture the geometric features of the surface at other scales.

Methods that rely on global properties of the object are vulnerable to occlusion and clutter in the scene [27], and approaches that are based on using local features to represent free-form surfaces have emerged as solutions to this problem. The *splash features* scheme proposed by Stein and Medioni [76] matches local patches of surfaces by using changes in surface orientation. In this approach, they defined *splash* as the distribution of all the normals on the surface within a distance ρ around a given point $\mathbf{p}$ (Figure 2.2). A spherical coordinate system is used to encode relative information about the normals $N\rho(\theta)$, where the relative orientation of $N\rho(\theta)$ with respect to $\mathbf{p}$'s normal N and the $X(\theta)$ axis is given by the angles $\phi(\theta)$ and $\psi(\theta)$. The $X(\theta)$ axis is chosen so that it is perpendicular to N. It also lies in the plane containing $\mathbf{p}$, N and the point ρ distance away from $\mathbf{p}$ and angle θ from where the encoding started. The values of $\phi(\theta)$ and $\psi(\theta)$ form a 3D curve as θ is varied from 0 to 2π

$$\mathbf{v}(\theta) = \begin{pmatrix} \phi(\theta) \\ \psi(\theta) \end{pmatrix} \tag{2.2}$$

In order to quicker match a pair of curves, the curves are polygonized starting at θ

Figure 2.2: The distribution of all the normals on the surface around a given point is encoded to form the splash feature at that point. $N\rho(\theta)$ is the normal at a distance ρ away from the central point and angle θ from where the encoding started. (Figure taken from [76]).

where $\phi(\theta)$ is maximum. The polygonal curve is finally encoded into a representation called the *3D supersegment* that stores the curvature angles between links κ_i and the torsion angles τ_i as shown in Figure 2.3. Figure 2.4 shows an example of a splash feature descriptor. The distance from the origin of the graph represents the angle κ and the direction, starting north and rotating counter-clockwise, represents the angle τ [76]. Areas of high curvature are the best places to encode splash features as they provide splashes with a rich structural description of the local surface shape of the object. Finally, the matching of splash features is performed using structural indexing, which is a variant on hashing where the indices to the hash table are related to structures formed from the features [5].

Similarly, a representation called *point signatures* formulated by Chua and Jarvis [12] encodes the minimum distances of points on a 3D contour to a reference plane. In order to extract the signature at a point $\mathbf{p}$, a 3D curve C is obtained by intersecting a sphere of radius r centered at $\mathbf{p}$ with the object surface. Next, a plane $\mathbf{P}$ is fitted to the contour C using a principal component analysis of C. The normal vector $\mathbf{n}_1$ of the plane can be considered as an approximation of the surface normal around $\mathbf{p}$. The plane is then translated in the direction of $\mathbf{n}_1$ to the point $\mathbf{p}$, forming

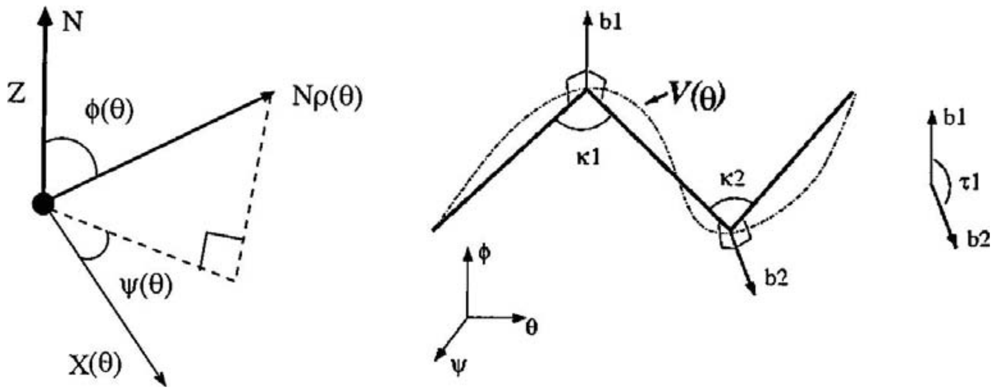


Figure 2.3: Orientation coordinates for splash features. (Figure taken from [5]).

NOTE:
This figure is included on page 13
of the print copy of the thesis held in
the University of Adelaide Library.

Figure 2.4: An example of a splash feature. The distance from the origin of the graph represents the angle κ and the direction, starting north and rotating counter-clockwise, represents the angle τ. (Figure taken from [76]).

a new plane $\mathbf{P}'$. A signed distance profile represented by a 1D parametric curve $d(\theta)$ ($0 \leq \theta \leq 360°$) is formed by projecting C perpendicularly to the plane $\mathbf{P}'$. The 1D curve $d\theta$ is called the point signature of $\mathbf{p}$ and θ is the rotation angle about $\mathbf{n}_1$ from the reference direction $\mathbf{n}_2$ as shown in Figure 2.5 ($\mathbf{n}_2$ is chosen as the unit vector from $\mathbf{p}$ to a projected point of C on $\mathbf{P}'$ that gives the largest positive distance). Point signatures are pose invariant and they provide a compact way of representing the local structure of the surface [5]. Positions on a grid evenly spaced over the scene are used to estimate point signatures during recognition. Partial correspondence search is used to determined the rotation and translation transformation between

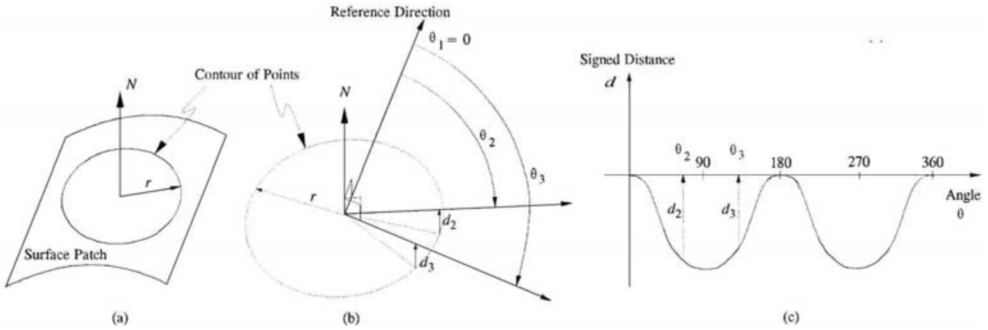


Figure 2.5: (a) Contour of a surface point $\mathbf{p}$ at radius r. (b) Reference direction. (c) The point signature $d(\theta)$ of $\mathbf{p}$. (Figure taken from [12]).

the scene point and model [12]. The main drawback of this approach lies in the extraction of the reference vector $\mathbf{n}_2$. The reference vector $\mathbf{n}_2$ is not robust to noise and also ambiguous in the case of planar and spherical regions [55].

A representation using object-centered local histograms of surface locations known as *Spin-images* was proposed by Johnson and Hebert [36]. This employed point features for 3D object recognition in scenes with occlusion and clutter. Each spin image is a local surface descriptor calculated at an oriented point $(\mathbf{p}, \mathbf{n})$ (3D point with normal vector) by encoding two of the three cylindrical coordinates of its surrounding points (Figure 2.6). The spin image X for a surface point $\mathbf{p}$ is a 2D histogram in which each pixel is a bin that stores the number of neighbours that are a distance α from $\mathbf{n}$ along its tangent plane $\mathbf{P}$ and a depth β from $\mathbf{P}$. Figure 2.7 shows the 'Chef' model [53] and a spin-image extracted at a vertex on its face. Spin images have been proved to be a powerful method for object recognition in range images because they are translation, rotation and pose invariant [18, 7] as long as the local surface normal can be accurately obtained. Carmichael and Hebert extended the method to work with large and non-uniform sampling 3D datasets [7, 6]. Other extensions of this approach including the multi-resolution [18] and spherical [71] spin-image approaches tried to reduce the computational time and automatically select the appropriate parameters such as bin size and histogram width. However, the main disadvantage of the spin-image representation is that it has a low discriminating capability and thus, may lead to many ambiguous

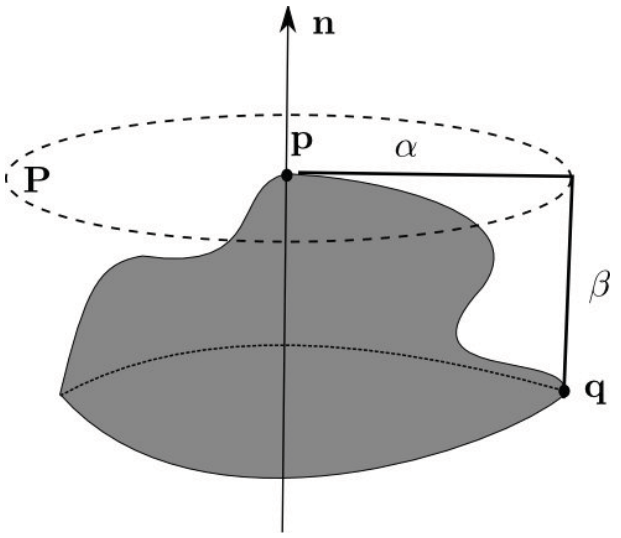


Figure 2.6: The (α, β) coordinate of the surface point $\mathbf{q}$ relative to $\mathbf{p}$.

correspondences [55].

Although most local surface descriptors are defined using a single selected point, a recently proposed matching algorithm uses a tensor-based surface representation defined on pairs of oriented points [56]. The variation of surface position is measured by high dimensional surface histograms (tensors) that include several hundred to several thousand elements in each descriptor. Tensors are matched between the model and the scene by using a modified geometric hashing algorithm [44]. However, this tensor-based approach requires a re-sampling of surfaces before constructing local descriptors. This may incorrectly change the surface topology and thereby reduce the accuracy of the registration result. Another recognition algorithm based on high dimensional surface histograms is the 3D shape context proposed by Frome *et al.* [23]. This algorithm also performs well on the problem of object recognition from range data.

2.2.6 Iterative Closest Point (ICP) Algorithm

The Iterative Closest Point (ICP) algorithm is a fine registration algorithm developed by Besl and McKay [3]. The aim of the algorithm is to make small changes

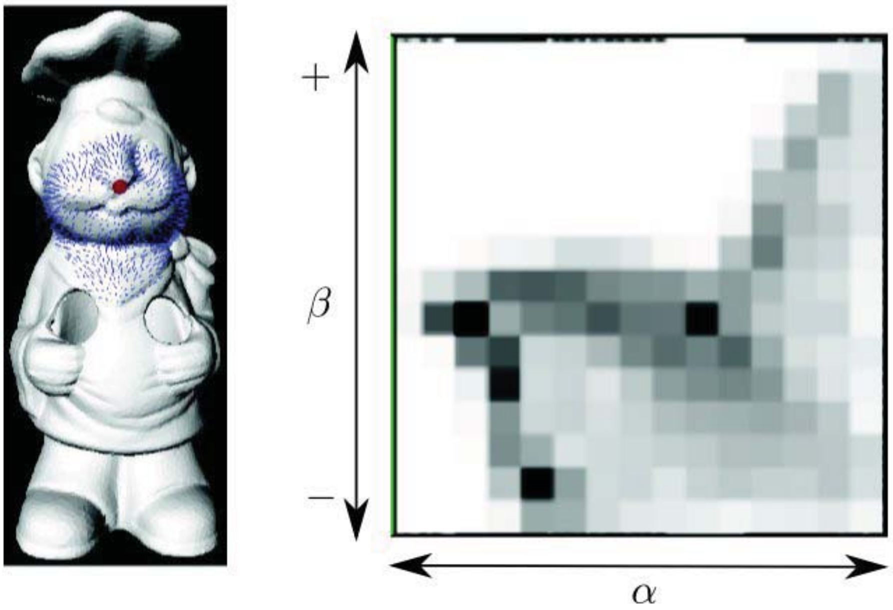


Figure 2.7: A spin-image extracted at a vertice on the face of the ‘Chef’ model. The location of the vertex is marked in red. Surface points contributing to the spin-image are highlighted by blue line segments oriented along the surface normals at these positions.

to the coarse registration results provided by the previously described methods in order to improve the overall quality of the registration. The quality of a registration is usually decided by the distance between points in the corresponding surfaces of two overlapping views [5].

Given two sets of points $\mathbf{P}_1 = \{\mathbf{p}_1^1, \mathbf{p}_2^1, \ldots, \mathbf{p}_{N_1}^1\}$ and $\mathbf{P}_2 = \{\mathbf{p}_1^2, \mathbf{p}_2^2, \ldots, \mathbf{p}_{N_2}^2\}$ where N_1 and N_2 are the number of points in $\mathbf{P}_1$ and $\mathbf{P}_2$, respectively; the ICP algorithm employs a nonlinear optimization procedure to further align the point sets. The algorithm is outlined in [5] and is summarized in Algorithm 1. The transformation matrix $\mathbf{T}(l)$ can be estimated using the quaternion-based algorithm in [3]. The registration error between $\mathbf{P}_1$ and $\mathbf{P}_2(l)$ is calculated as

$$E = \frac{1}{N_2} \sum_{k=1}^{N_2} ||\mathbf{y}_k - \mathbf{p}_k^2(l)||^2 \tag{2.3}$$

The initial transformation of the ICP must be chosen as close to the true value

Algorithm 1 Iterative Closest Point (ICP) Algorithm

```
Data:
$\mathbf{P}_1 = \{\mathbf{p}_1^1, \mathbf{p}_2^1, \ldots, \mathbf{p}_{N_1}^1\}$
$\mathbf{P}_2(0) = \mathbf{P}_2 = \{\mathbf{p}_1^2, \mathbf{p}_2^2, \ldots, \mathbf{p}_{N_2}^2\}$
Algorithm:
repeat
    for every point in $\mathbf{P}_2(l)$ do
        Find the closest point in $\mathbf{P}_1$
    end for
    The closest points form a new point set $\mathbf{Y}(l)$ where the pairs of points
    $\{(\mathbf{p}_1^2, \mathbf{y}_1), (\mathbf{p}_2^2, \mathbf{y}_2), \ldots, (\mathbf{p}_{N_2}^2, \mathbf{y}_{N_2})\}$ form the correspondences between $\mathbf{P}_1$ and
    $\mathbf{P}_2(l)$
    if registration error between $\mathbf{P}_1$ and $\mathbf{P}_2(l)$ is too large then
        Compute the transformation $\mathbf{T}(l)$ between $(\mathbf{P}_2(l), \mathbf{Y}(l))$
        Apply registration $\mathbf{P}_2(l+1) = \mathbf{T}(l) \cdot \mathbf{P}_2(l)$
    else
        STOP
    end if
until $||\mathbf{P}_2(l+1) - \mathbf{P}_2(l)|| > threshold$
```

as possible to avoid converging to a non-optimal solution as the parameter space explored by the algorithm may have many local minima. This is the reason why it is necessary to use coarse registration algorithms such as the RANSAC-based technique or the spin-image method before refining the registration results using ICP.

The search for closest points can be expedited by implementing a space-partitioning data structure called the *k-d tree.* A *k*-d tree is a *binary tree*[1] in which each node is a *k*-dimensional point. A hyperplane at each non-leaf node splits the space into two subspaces. The left sub-tree of that node contains points left to the hyperplane and the right sub-tree includes all points right to the hyperplane. Each hyperplane is chosen so that it is perpendicular to one of the dimension vectors of the k-dimensional space. In the case of N randomly distributed points, the search for the nearest point is an $O(\log N)$ operation [15].

[1] A binary tree is a tree data structure in which each node has at most two children.

2.3 Multi-scale Feature Extraction from 3D Surfaces

2.3.1 Scale-space Representation using Smoothing Kernels

In the 2D image domain, the scale-space representation is a well-established problem [84, 47, 49]. The fundamental idea of the scale-space representation, first introduced in 1983 [84], is to transform an input signal $f(\mathbf{x}) : \mathbf{R}^d \rightarrow \mathbf{R}$ ($\mathbf{x}$ is a d-dimensional vector) at different scales σ as $L(\mathbf{x}, \sigma) : \mathbf{R}^d \times \mathbf{R}_+ \rightarrow \mathbf{R}$. $\mathbf{R}^d \times \mathbf{R}$ denotes the Cartesian product of sets $\mathbf{R}^d$ and $\mathbf{R}$, which is the set of all possible ordered pairs $(\mathbf{x}, \sigma)$ with $\mathbf{x} \in \mathbf{R}^d$ and $\sigma \in \mathbf{R}_+$ ($\mathbf{R}_+$ is the set of all positive real numbers). The Gaussian scale-space representation could be obtained by convolving $f(\mathbf{x})$ with Gaussian kernels G of increasing width σ [47]

$$L(\mathbf{x}, \sigma) = G(\mathbf{x}, \sigma) \otimes f(\mathbf{x}) \tag{2.4}$$

Figure 2.8 shows an example of applying Gaussian kernels of increasing widths to a 2D image in order to create its scale-space representation. After constructing the scale-space representation of the input image, features are usually selected as positions having extremal values in their local neighborhoods in the scale-space representation.

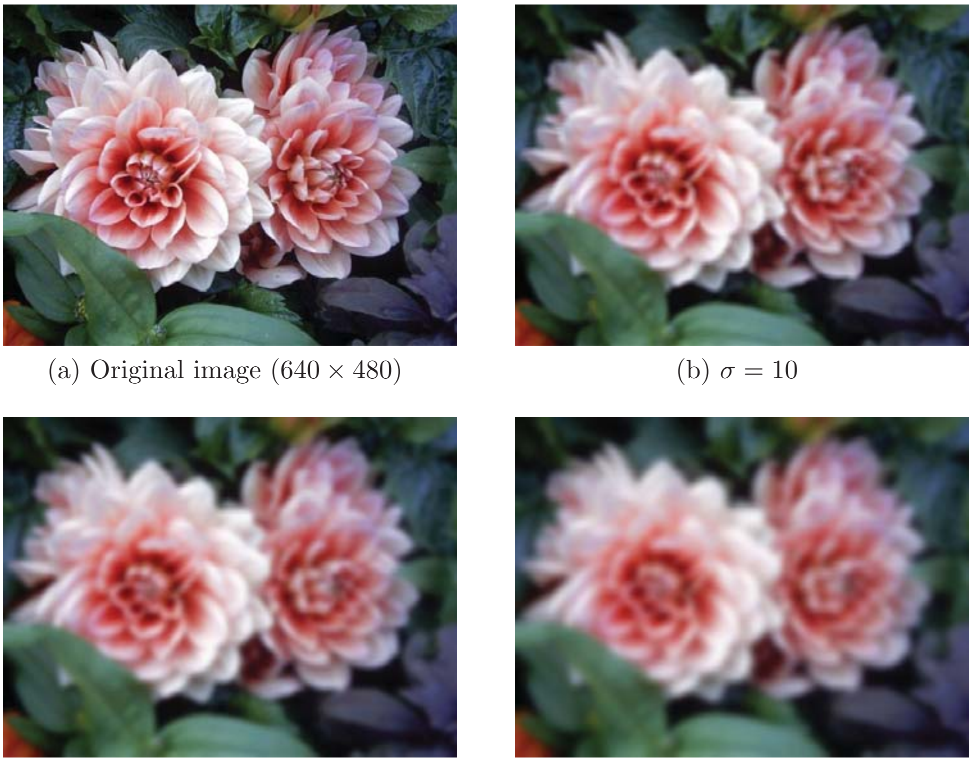
The connection between the Gaussian scale-space and the diffusion equation was obtained in [39]

$$\frac{\partial L(\mathbf{x}, \sigma)}{\partial \sigma} = \triangle L(\mathbf{x}, \sigma) \equiv \sum_{i=1}^{d} \frac{\partial^2 L(\mathbf{x}, \sigma)}{\partial x_i^2} \tag{2.5}$$

where $\triangle$ denotes the Laplacian operator. Successive convolutions of the signal may be obtained by linearizing the diffusion equation with an explicit Euler scheme [49]

$$L(\mathbf{x}, \sigma + \delta\sigma) = L(\mathbf{x}, \sigma) + \delta\sigma \triangle L(\mathbf{x}, \sigma) \tag{2.6}$$

There are many methods that extend the above idea of 2D scale-space representation to point-sampled surfaces. One method is the iterative Laplacian smoothing [78]



(a) Original image (640 × 480)

(b) $\sigma = 10$

(c) $\sigma = 20$

(d) $\sigma = 40$

Figure 2.8: Scale-space representation of a 2D image with increasing Gaussian kernel width.

that uses the graph Laplacian L_g instead of the continuous second order Laplacian operator $\triangle$

$$L_g f(\mathbf{x}_i) = \sum_{j,\{i,j\} \in \mathbf{E}} (f(\mathbf{x}_i) - f(\mathbf{x}_j)) w_{ij} \tag{2.7}$$

where $\mathbf{x}_i$ is a set of vertices of a graph, the summation is over graph edges (i, j) in the edge set $\mathbf{E}$ and w_{ij} are positive edge weights. However, the drawback of this method is that its smoothing kernel often produces surface deformation artefacts such as volume shrinkage and thus may incorrectly change the surface geometry [64, 82].

2.3.2 Feature Extraction using Surface Variation

Another extension of the scale-space representation of 3D images is the surface variation measure, $\sigma_n(\mathbf{p})$, proposed by Pauly *et al.* [64]

$$\sigma_n(\mathbf{p}) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{2.8}$$

where λ_i's are eigenvalues of the sample covariance matrix $\mathbf{C}$ evaluated in a local neighbourhood N_p of the sample point $\mathbf{p}$. If the local neighbourhood contains k points, the 3×3 covariance matrix C is evaluated as

$$\mathbf{C} = \frac{1}{k} \begin{bmatrix} \mathbf{p}_{i_1} - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_{i_k} - \bar{\mathbf{p}} \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{p}_{i_1} - \bar{\mathbf{p}} \\ \dots \\ \mathbf{p}_{i_k} - \bar{\mathbf{p}} \end{bmatrix}, \mathbf{p}_{i_j} \in N_p \tag{2.9}$$

where $\bar{\mathbf{p}}$ is given by

$$\bar{\mathbf{p}} = \frac{1}{k} \sum_{j=1}^{k} \mathbf{p}_{i_j} \tag{2.10}$$

Features are selected as range points with maximum local shape variation. The neighbourhood size for which $\sigma_n(\mathbf{p})$ gets a local extremum is chosen as the scale of the corresponding feature. However, the drawback of this method is that it requires heuristic pre-smoothing procedures to be applied to the surface and $\sigma_n(\mathbf{p})$ is very sensitive to noise [82].

2.3.3 Corner and Edge Detectors

A recent work proposed by Novatnack and Nishino [62, 63] aims at detecting multi-scale corner and edge features from 3D meshes. In this approach, the surface mesh of the model is first parameterised on a 2D plane. A 2D representation of the original surface called the *normal map* is then created by interpolating over the surface normals at each 2D-embedded vertex. A *distortion map* is also computed by encoding the relative change in the model edge lengths. The scale-space representation of the normal map is constructed by convolving the vector field with

Figure 2.9: (a) A 3D model of a hand. (b) The 2D normal map of the 3D model. (c) the 2D distortion map of the 3D model. Iso-contour lines illustrate the various levels of distortion. (Figure taken from [62]).

modified Gaussian kernels of increasing width σ. Figure 2.9 shows a 3D model and its 2D normal map and distortion map.

In this approach, the geodesic distance[2] is used as the distance metric to construct a geometric scale-space that encodes the surface geometry. Given $\mathbf{u}, \mathbf{v} \in \mathbf{R}^2$ be two points on a normal map, the geodesic Gaussian kernel at $\mathbf{v}$ is defined as

$$g(\mathbf{v}, \mathbf{u}, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left[\frac{-d(\mathbf{v}, \mathbf{u})^2}{2\sigma^2}\right] \tag{2.11}$$

where $d : \mathbf{R}^2 \times \mathbf{R}^2 \to \mathbf{R}$ is the geodesic distance between the corresponding 3D surface points $\phi(\mathbf{v})$ and $\phi(\mathbf{u})$ of $\mathbf{v}$ and $\mathbf{u}$, respectively.

Corner and edge detectors are derived using the first and second-order partial derivatives of the normal map in the horizontal (s) and vertical direction (t). The corner response at a point $\mathbf{u} = (s, t)$ of the normal map $\mathbf{N}^\sigma$ at the scale σ is defined as the maximum eigenvalue of the Gram matrix M calculated at that point

$$M(\mathbf{u}; \sigma, \tau) = \sum_{\mathbf{v} \in W} \begin{bmatrix} \mathbf{N}_s^\sigma(\mathbf{v})^2 & \mathbf{N}_s^\sigma(\mathbf{v})\mathbf{N}_t^\sigma(\mathbf{v}) \\ \mathbf{N}_s^\sigma(\mathbf{v})\mathbf{N}_t^\sigma(\mathbf{v}) & \mathbf{N}_t^\sigma(\mathbf{v})^2 \end{bmatrix} g(\mathbf{v}; \mathbf{u}, \tau) \tag{2.12}$$

where σ is the particular scale, W is the local window around $\mathbf{u}$ and τ is the weighting of each point in the Gram matrix. $\mathbf{N}_s^\sigma(\mathbf{v})$ and $\mathbf{N}_t^\sigma(\mathbf{v})$ are the derivative

[2] In graph theory, the geodesic distance between two vertices is the number of edges in the *shortest path* connecting them.

of $\mathbf{N}\sigma(\mathbf{v})$ with respect to s and t, respectively

$$\mathbf{N}_s^\sigma(\mathbf{v}) = \frac{\partial \mathbf{N}^\sigma(\mathbf{v})}{\partial s} \tag{2.13}$$

$$\mathbf{N}_t^\sigma(\mathbf{v}) = \frac{\partial \mathbf{N}^\sigma(\mathbf{v})}{\partial t} \tag{2.14}$$

The corners lying along the 3D edges can be pruned using the second order derivatives of the normal map. Edges can be located using the zero crossing of the Laplacian of the normal map

$$\nabla^2 \mathbf{N}^\sigma = \mathbf{N}_{ss}^\sigma + \mathbf{N}_{tt}^\sigma \tag{2.15}$$

where

$$\mathbf{N}_{ss}^\sigma = \frac{\partial^2 \mathbf{N}^\sigma}{\partial s^2} \tag{2.16}$$

$$\mathbf{N}_{tt}^\sigma = \frac{\partial^2 \mathbf{N}^\sigma}{\partial t^2} \tag{2.17}$$

The disadvantage of this approach is that the 2D normal map can only be created from a 3D mesh with connectivity information between vertices. It also suffers from the same limitation as in [45] that good surface normals must be available in order to construct the scale-space representation.

2.3.4 Feature Extraction using 3D Hessian

Recently, Flint *et al.* [20, 21] proposed the use of a 3D version of the Hessian to measure the distinctiveness of candidate interest points. In this technique, the normalised density map of an input range image whose domain is $\mathbf{I} \subset \mathbb{Z}^3$ is first constructed as

$$D(i,j,k) = \frac{n(B_{ijk})}{argmax_{(i,j,k)\in\mathbf{I}}\{n(B_{ijk})\}} \tag{2.18}$$

where $B = \{B_{ijk}\}_{(i,j,k)\in\mathbf{I}}$ is a set of equal-size boxes distributed regularly in each spatial dimension and $n(B_{ijk})$ is the number of range points inside the box B_{ijk}. The scale-space representation is then constructed similarly to other approaches by convolving the density map with Gaussian kernels of increasing standard deviation σ as in Equation (2.4). In this case, f is the normalised density map D.

The 3D Hessian $\mathbf{H}(\mathbf{x}, \sigma)$ is computed at each sampled location $\mathbf{x}$ for each scale σ

$$\mathbf{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}; \sigma) & L_{xy}(\mathbf{x}; \sigma) & L_{xz}(\mathbf{x}; \sigma) \\ L_{yx}(\mathbf{x}; \sigma) & L_{yy}(\mathbf{x}; \sigma) & L_{yz}(\mathbf{x}; \sigma) \\ L_{zx}(\mathbf{x}; \sigma) & L_{zy}(\mathbf{x}; \sigma) & L_{zz}(\mathbf{x}; \sigma) \end{bmatrix} \tag{2.19}$$

where

$$L_{xx} = \frac{\partial^2 L}{\partial x^2} \quad L_{yy} = \frac{\partial^2 L}{\partial^2 y} \quad L_{zz} = \frac{\partial^2 L}{\partial^2 z} \tag{2.20}$$

$$L_{xy} = \frac{\partial^2 L}{\partial x \partial y} \quad L_{xz} = \frac{\partial^2 L}{\partial x \partial z} \quad L_{yz} = \frac{\partial^2 L}{\partial y \partial z} \tag{2.21}$$

A detailed discussion of the computational cost of estimating the Hessian matrix can be found in [43]. Weak responses are pruned by applying a user defined threshold T_D to the determinant of the Hessian. Features are selected by comparing each voxel with its scale-space neighbours and selecting the local maxima. One disadvantage of this method is the cost of resampling regularly in space throughout the data which may also destroy fine structures relevant to recognition. Another drawback is the stability of the Hessian determinant used for searching local maxima in the presence of noise.

2.4 Summary

In this chapter, a review on previous approaches to the problem of 3D surface matching from range images was presented. Different techniques for extracting local features from 3D data sets were also discussed. These methods can be applied to 3D surface matching algorithms as a preprocessing step to improve their accuracy and efficiency. In the next chapter, two novel schemes for extracting salient features from 3D data sets using local surface curvature will be described.

Chapter 3

Multiscale Feature Extraction

3.1 Introduction

The processing of 3D datasets such as range images is a demanding job due to not only the huge amount of surface information but also the noise and non-uniform sampling introduced by the sensors or the reconstruction process. It is therefore desirable to have a more compact intermediate representation of 3D images that can be used efficiently in computer vision tasks such as 3D scene registration or object recognition.

As mentioned in Section 2.2.5, one of the most popular approaches uses local descriptors, or signatures that describe local surface regions, to represent a surface [23, 36, 52, 56, 63]. This strategy has been proven to be robust to partial occlusion, clutter and intra-class variation [82]. However, in those techniques, the surface locations used for estimating local descriptors are either selected exhaustively at each point, or randomly from the data [23, 36]. In the case of exhaustive selection, this is very inefficient because of the redundancy in areas with little shape variation. In the case of randomly selection, distinctive geometric structures may be missed thus reducing the accuracy of the algorithm [82]. Therefore, it is desirable to have a principled way to sample a representative set of feature points from the surface

[45].

Surface curvature has been used extensively in the literature for mesh simplification and smoothing [17], object recognition [19, 58, 85] and mesh segmentation [33]. However, there is a lack of a systematic approach in extracting local feature points from an input surface using its local curvature information at multiple scales. In this chapter, we address the above issue by proposing two different schemes for multi-scale feature extraction from 3D datasets using two surface curvature measures known as the curvedness and shape index.

3.2 Local Surface Curvature

3.2.1 Shape Operator

Let $M \subset \mathbf{R}^3$ be a regular surface, and let $\mathbf{n}$ be a surface normal to M defined in a neighbourhood of a point $\mathbf{p} \in M$. The shape operator applied to a tangent vector $\mathbf{v_p}$ to M at $\mathbf{p}$ is defined as below [24]

$$S(\mathbf{v_p}) = -\mathbf{D_v n} \tag{3.1}$$

where $\mathbf{D_v n}$ is the directional derivative of $\mathbf{n}$ in the direction of $\mathbf{v_p}$ at the point $\mathbf{p}$

$$\mathbf{D_v n} = \lim_{h \to 0} \frac{\mathbf{n}(\mathbf{p} + h\mathbf{v_p}) - \mathbf{n}(\mathbf{p})}{h} \tag{3.2}$$

The shape operator can be used to measure how the regular surface M bends in $\mathbf{R}^3$ as it estimates how the surface normal $\mathbf{n}$ changes from point to point. It can be seen that the shape operator of a plane is identically zero at all points. It is nonzero for non-planar surface patches as in this case, the surface normal $\mathbf{n}$ twists and turns from point to point.

3.2.2 Principal Curvatures

If $\mathbf{u_p}$ is a tangent vector to a regular surface $M \subset \mathbf{R}^3$ at point $\mathbf{p} \in M$ with $||\mathbf{u_p}|| = 1$, the normal curvature of M in the direction of $\mathbf{u_p}$ is defined as

$$k(\mathbf{u_p}) = S(\mathbf{u_p}) \cdot \mathbf{u_p} \tag{3.3}$$

where $S(\mathbf{u_p})$ is the shape operator as defined above.

The maximum and minimum values, k_1 and k_2 respectively, of the normal curvature $k(\mathbf{u_p})$ are called *principal curvatures* of M at point $\mathbf{p}$. These values measure the maximum and minimum bending of M at $\mathbf{p}$ [40]. Unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$ at which these extreme values occur are called *principal vectors* and the corresponding directions are called *principal directions*. The principal curvatures are related to the two classic shape measures, the Gaussian curvature K and mean curvature H, by

$$k_1 = H + \sqrt{H^2 - K} \tag{3.4}$$

$$k_2 = H - \sqrt{H^2 - K} \tag{3.5}$$

Figure 3.1 shows the principal curvatures for a point $\mathbf{p}$ with the normal vector $\mathbf{n}$ on a surface. The curvature is positive if the curve turns in the same direction as the surface normal $\mathbf{n}$. Otherwise, it will be negative. Figure 3.2 shows the maximum and minimum principal curvatures estimated for the Stanford ‘Buddha’ model [75]. The colormap is shown in Figure 3.2c.

3.2.3 Shape Index and Curvedness

The local surface topology of a 3D model at a point $\mathbf{p}$ can be represented by a single-value, angular measure called the shape index $S_I(\mathbf{p})$ [40]

$$S_I(\mathbf{p}) = \frac{2}{\pi} \arctan \frac{k_1 + k_2}{k_2 - k_1} \tag{3.6}$$

From (3.6), it can be seen that the range of the shape index is $[-1, 1]$. Except for a plane where the shape index is undefined ($k_1 = k_2 = 0$), every distinct surface

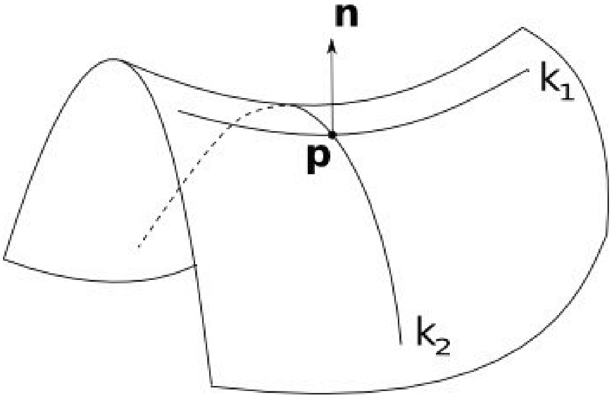


Figure 3.1: Principal curvatures at a point **P** on a surface.

shape corresponds to a unique value of S_I. Whereas both the Gaussian and mean curvatures are necessary to describe the local surface curvature, only a single shape index value is needed for the same task [40].

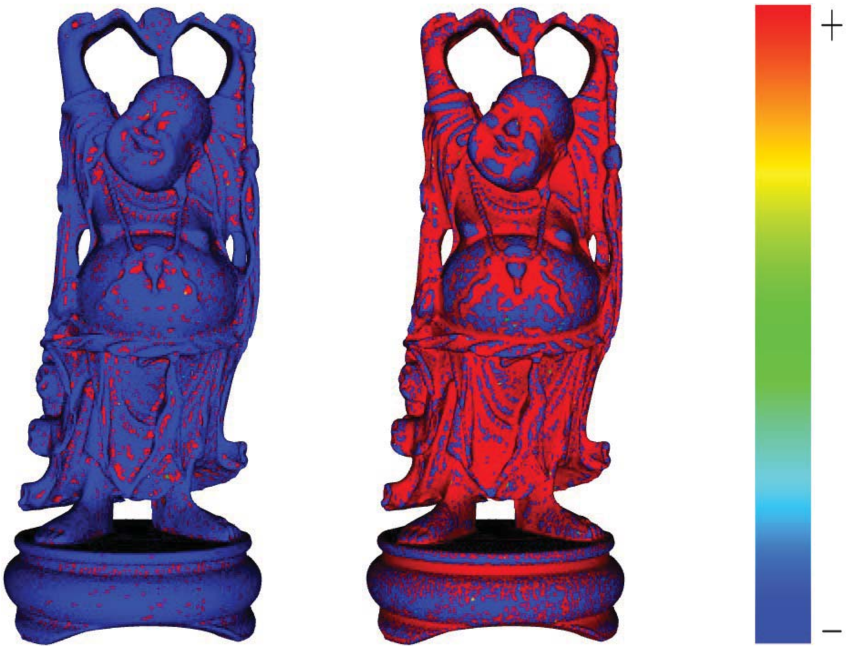
The local geometric aspect of a 3D model can also be defined using a bending energy measure of the surface called the curvedness. The curvedness at a point **p** on a surface can be estimated as [40, 19]

$$c_{\mathbf{p}} = \sqrt{\frac{k_1^2 + k_2^2}{2}} \tag{3.7}$$

The curvedness could be used to indicate how highly or gently curved a surface is [33]. Unlike Gaussian curvature which vanishes on parabolic surfaces such as ruts and cylindrical ridges, the curvedness is zero only for planar patches as $k_1 = k_2 = 0$ in this case. Both the shape index and curvedness are invariant to rotation as well as translation. Furthermore, the shape index is also independent of scale [40]. Figure 3.3 plots the shape index and curvedness of the Stanford 'Dragon' model [75]. The colormap is the same as in Figure 3.2c.

3.2.4 Local Differential Properties of 3D Surfaces

There are many methods to estimate the local curvature information such as Gaussian, mean or principal curvatures for 3D surfaces [65]. One of the most popular approaches that provides quite accurate approximations is using the estimation of



(a) k_1 (b) k_2 (c) Colormap

Figure 3.2: Principal curvatures of the 'Buddha' model.

the differential properties of a smooth manifold - a curve or a surface in 3D [8]. This method fits the local representation of the manifold using a truncated Taylor expansion called the *jet*. All local geometric quantities such as normal, curvatures, extrema of curvatures are encoded in the jets [8].

It has been proven that any regular embedded smooth curve or surface can be locally written as the graph of a bi-variate function termed the *height function* [74]

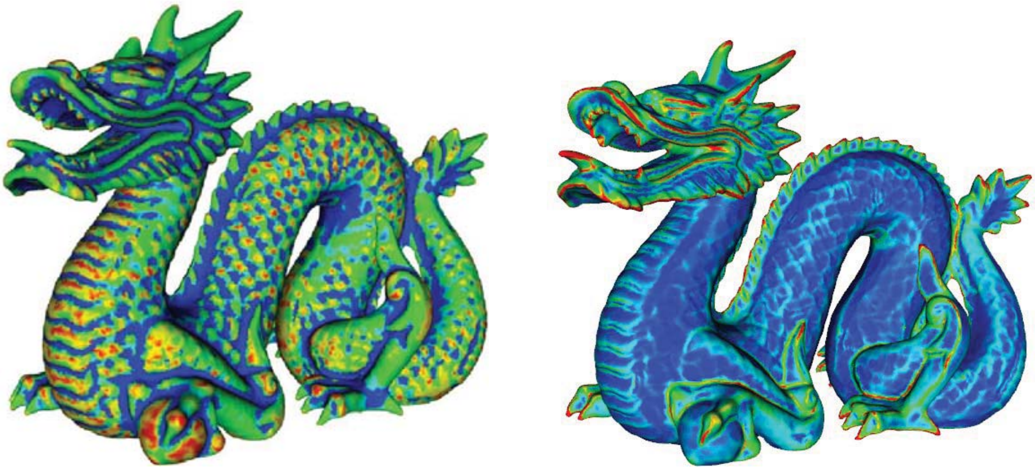
$$f(\mathbf{x}) = J_{B,n}(\mathbf{x}) + O(\mathbf{x}^{n+1}) \tag{3.8}$$

where $\mathbf{x} = (x, y)$ and $z = f(\mathbf{x})$ for a point $\mathbf{p} = (x, y, z)$ in 3D space and $O(\mathbf{x}^{n+1})$ contains higher order terms of the expansion. The n-jet $J_{B,n}(\mathbf{x})$ is the n-th order Taylor expansion of the height function

$$J_{B,n}(x, y) = \sum_{k=1}^{n} \sum_{j=0}^{k} B_{k-j,j} x^{k-j} y^j \tag{3.9}$$

where

$$B_{k-j,j} = \frac{1}{(k-j)!j!} \frac{\partial^k f(0,0)}{\partial x^{k-j} \partial y^j} \tag{3.10}$$



(a) Shape index (b) Curvedness

Figure 3.3: Shape index and curvedness of the 'Dragon' model.

First order properties of the above expansion correspond to the normal or tangent plane. The principal curvatures and directions are provided by its second order properties. Furthermore, the directional derivatives of the principal curvatures along the curvature lines could be estimated from the third order properties. The jet has a canonical form for non-umbilical points where their principal curvatures are not identical [8]

$$J_{B,3}(x,y) = \frac{1}{2}(k_1x^2 + k_2y^2) + \frac{1}{6}(b_0x^3 + 3b_1x^2y + 3b_2xy^2 + b_3y^3) \tag{3.11}$$

where k_1 and k_2 are the principal curvatures as defined in Section 3.2.2. The shape index and curvedness can be easily calculated from (3.6) and (3.7). The two vectors, $u = (b_0, b_1)$ and $v = (b_3, b_2)$, are the directional derivatives of k_1, k_2 along their curvature lines, respectively. The advantage of this fitting method is that it needs only a set of 3D points as input. It does not require the mesh connectivity to estimate the local curvature information.

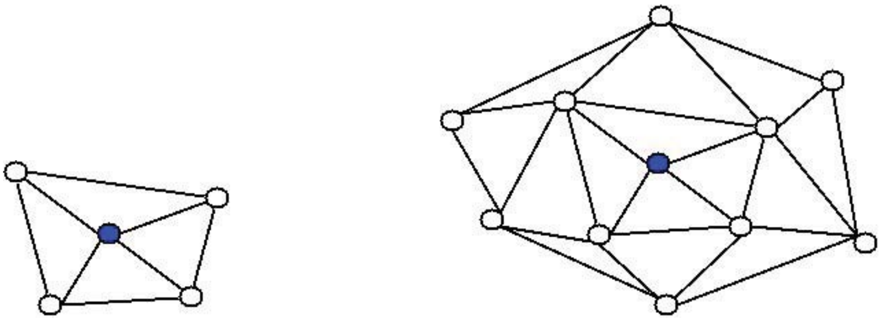
3.3 Feature Extraction using Curvedness

3.3.1 Algorithm Description

As we have already discussed in Section 2.3.1, there are 3D smoothing kernels that can be used to estimate the 3D scale space representation of a surface. However, these methods usually produce undesired artefacts such as changes in the geometric structures of the models. In the proposed framework, the scale of a point on the surface is defined as the size of the neighbourhood that we use to collect points as the input to the fitting. For unstructured point clouds, the scale r could be chosen as either Euclidean distance or geodesic distance [24]. All the surface points that are closer to the fitting point $\mathbf{p}$ than this distance will be picked. In this thesis, we use the Euclidean distance to specify the radius of the neighourhood. A drawback of using the Euclidean distance for collecting fitting points is that even the distance between two points is small, they may not be close neighbours on the underlying surface represented by the point cloud. However, it is inevitable as the connectivity information of surface points is not available. For 3D meshes, the scale r is the number of *rings* surrounding the fitting vertex $\mathbf{v}$. The scale r can also be called the radius of the neighbourhood. The first ring contains all vertices that are directly connected to the vertex $\mathbf{v}$. The second ring contains all the direct neighbours of the vertices in the first ring and so on[1]. Figure 3.4 shows an example of a vertex and its neighbourhoods of radius $r = 1$ and $r = 2$.

The proposed multi-scale curvedness-based feature selection algorithm is outlined in Algorithm 2. Features are surface positions $\mathbf{p}$ having extremum curvedness values $c_{\mathbf{p}}$ both in the neighbourhood of radius r_k as well as over the above and below scales (r_{k-1}, r_{k+1}). It can be seen that increasing the size of the local neighbourhood is similar to applying a smoothing filter but it avoids making direct changes to the 3D surfaces [64]. Thus, in addition to the ability to detect features at multiple scales, another benefit of this approach is that it reduces the effect of noise on the models.

[1] It is assumed that the mesh is relatively uniformly distributed across the surface.



(a) One ring ($r = 1$)

(b) Two rings ($r = 2$)

Figure 3.4: An example of rings. The fitting vertex is marked with blue color.

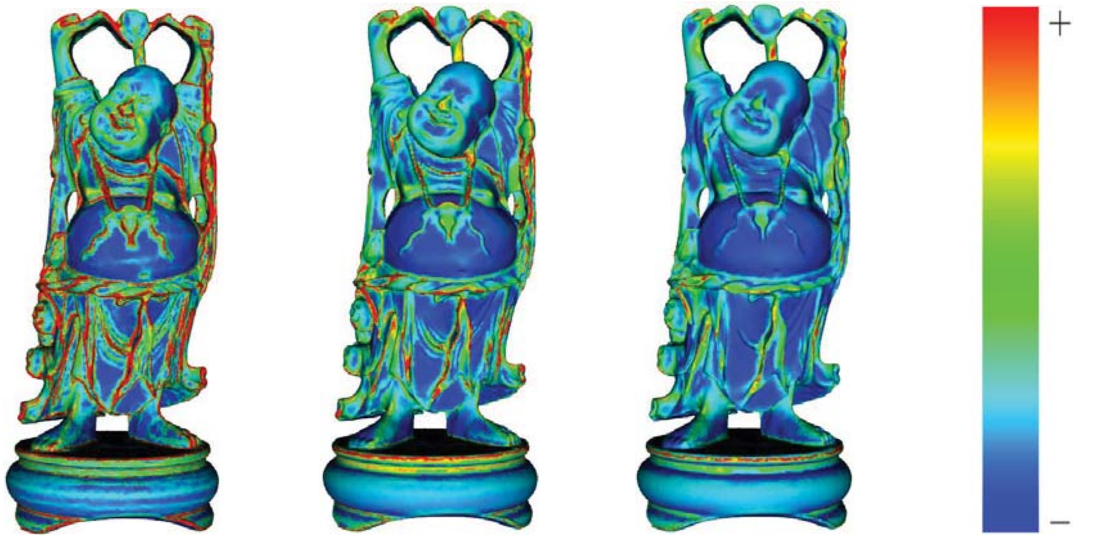
It is also worth noting that the scales r_k where features are selected can be used as support regions for many 3D object recognition algorithms such as spin-images [36] or the tensor-based approach [56]. Figure 3.5 shows the curvedness of the 'Buddha' models at different scale levels r_k.

Algorithm 2 Multi-scale Curvedness-based Feature Extraction Algorithm

```
Data:
$\mathbf{P} = \{\mathbf{p}_i \in \mathbf{R}^3\}$: set of 3D points sampled from the surface.
$\mathbf{R} = \{r_k\}$: a set of scales.
Algorithm:
for $r \in \{r_k\}$ do
    for $\mathbf{p} \in \{\mathbf{p}_i\}$ do
        Find the neighbourhood $N_r$ at scale $r$
        Fit a jet to $N_r$
        Compute principal curvatures $k_1$ and $k_2$
        Compute the curvedness $c_\mathbf{p}$
            $c_\mathbf{p} = \sqrt{(k_1^2 + k_2^2)/2}$
    end for
    Features are positions $\mathbf{p}$ having extremum values $c_\mathbf{p}$ both in the neighbourhood of radius $r_k$ as well as over the above and below scales $(r_{k-1}, r_{k+1})$.
end for
```

3.3.2 Feature Distinctiveness

In this work, the perceived reliability of a local feature is represented by its distinctiveness value. If the distinctiveness value is small, the feature may not be reliable



(a) $r_k = 2$ (b) $r_k = 5$ (c) $r_k = 7$ (d) Colormap

Figure 3.5: Curvedness of the 'Buddha' model estimated at different scale levels r_k. It can be seen that increasing the scale is similar to applying a smoothing filter to the 3D surface before calculating the surface curvature.

because its curvedness value does not deviate far enough from the other values in the immediate neighbourhood. The *distinctiveness* of a feature located at a surface point $\mathbf{p}$ at scale r_k is defined as

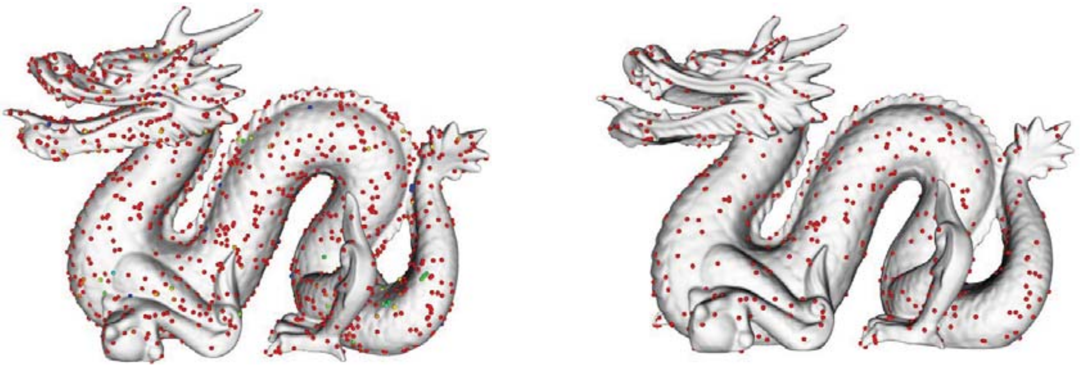
$$\gamma(\mathbf{p}, r_k) = \frac{|c_{\mathbf{p}} - \mu_{N_{\mathbf{p}}}|}{\sigma_{N_{\mathbf{p}}}} \tag{3.12}$$

where $N_{\mathbf{p}}$ is a set of all n 3D points in the neighbourhood of $\mathbf{p}$ at not only scale r_k but also at the two adjacent scales r_{k-1} and r_{k+1}. $c_{\mathbf{p}}$ is the curvedness of $\mathbf{p}$ as defined in (3.7). $\mu_{N_{\mathbf{p}}}$ and $\sigma_{N_{\mathbf{p}}}$ are the mean and standard deviation of the curvedness of all vertices in $N_{\mathbf{p}}$ respectively

$$\mu_{N_{\mathbf{p}}} = \frac{\sum_{\mathbf{p}_j \in N_{\mathbf{p}}} c_{\mathbf{p}_j}}{n} \tag{3.13}$$

$$\sigma_{N_{\mathbf{p}}} = \sqrt{\frac{\sum_{\mathbf{p}_j \in N_{\mathbf{p}}} (c_{\mathbf{p}_j} - \mu_{N_{\mathbf{p}}})^2}{n}} \tag{3.14}$$

It can be seen that if the distinctiveness value is small, the local maxima or minima may be due to measurement noise on the object's surface rather than a change in local object structure. Assuming the distribution of the curvedness of a surface



(a) Non-thresholded ($\gamma_t = 0$)

(b) Thresholded ($\gamma_t = 1.5$)

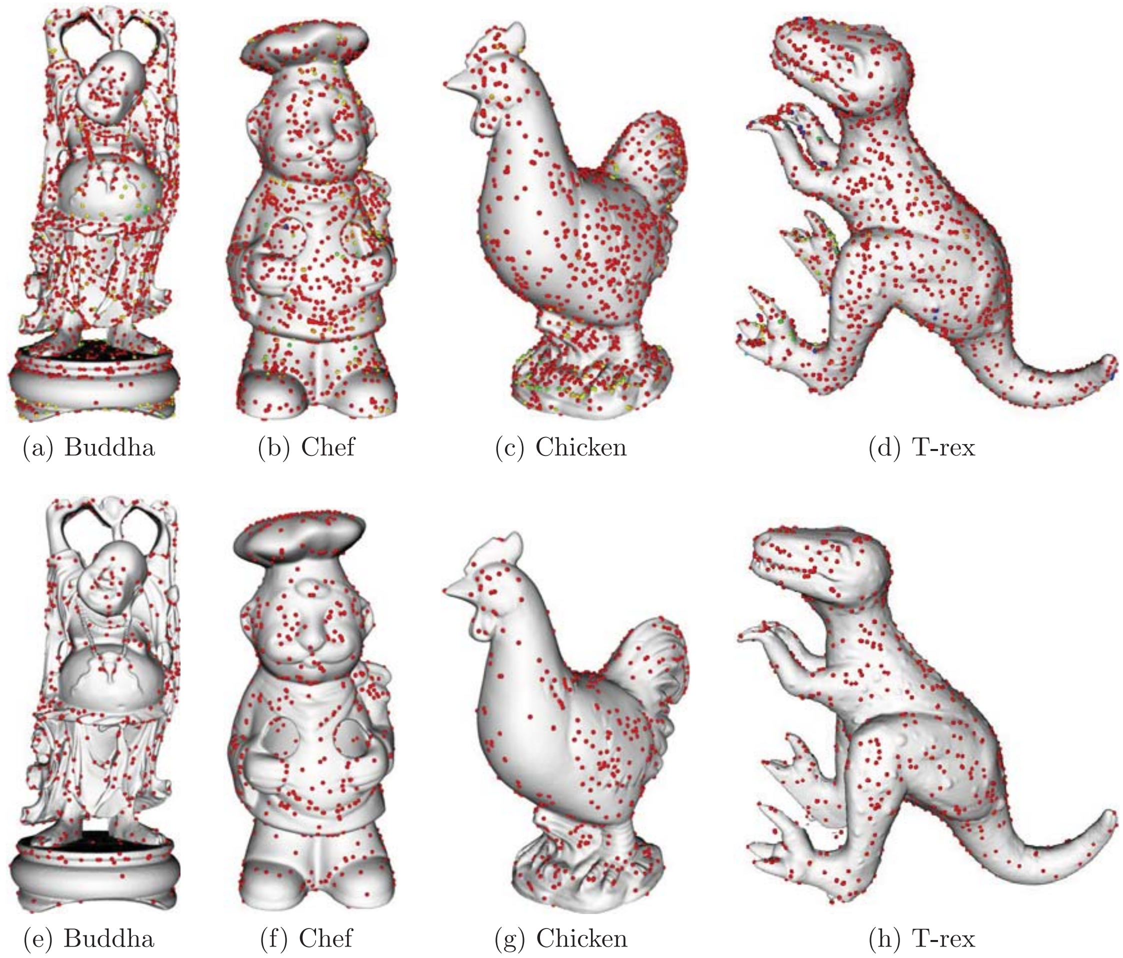
Figure 3.6: Local features detected on the ‘Dragon’ model. Features are selected from each model using 7 different scales ($r = 1$ to $r = 7$). The colormap of the features is from blue to red corresponding to low to high distinctiveness values. Figures are best seen in color.

region is approximated by a normal distribution, about 68% of the curvedness values would be within $[\mu_{N_\mathbf{p}} - \sigma_{N_\mathbf{p}}, \mu_{N_\mathbf{p}} + \sigma_{N_\mathbf{p}}]$ and a threshold γ_t could be used to remove less reliable features. Figure 3.6 shows the features extracted by the proposed algorithm from the Stanford ‘Dragon’ model [75] using two different values for the threshold.

3.3.3 Experiments

The proposed curvedness-based multi-scale feature extraction approach was tested on a variety of standard 3D models represented by triangular meshes. The ‘Dragon’ and ‘Buddha’ together with many other models are part of the Stanford 3D Scanning Repository [75]. The ‘Chef’, ‘Chicken’ and ‘T-rex’ models were found at Mian’s website [53]. Figures 3.6 and 3.7 visualise the features extracted from 5 different 3D models with two levels of threshold $\gamma_t = 0$ and $\gamma_t = 1.5$. It can be seen from the figures that most of the salient positions in the models such as positions near the noses, mouths or eyes of the Buddha and Chef or the tail of the Chicken were selected as feature points.

Table 3.1 shows the comparison between the number of features and the number of



(a) Buddha (b) Chef (c) Chicken (d) T-rex

(e) Buddha (f) Chef (g) Chicken (h) T-rex

Figure 3.7: Feature extraction results for four different 3D models with different distinctiveness thresholds $\gamma_t = 0$ and $\gamma_t = 1.5$ on the first row and second row, respectively. Features are selected from each model using 7 different scales ($r = 1$ to $r = 7$).

vertices in each model. It can be seen from the table that the number of features is significantly smaller than the number of vertices for all 5 surfaces. Without thresholding, the number of features is about 2% of the number of vertices in each model. When the threshold is set to $\gamma_t = 1.5$, the number of features reduces to just about 0.5% of the number of vertices. Although the set of features contains just a small percentage of the surface data, it is still a sparse but well-described representation of the geometric structures in the model as evident in Figures 3.6 and 3.7.

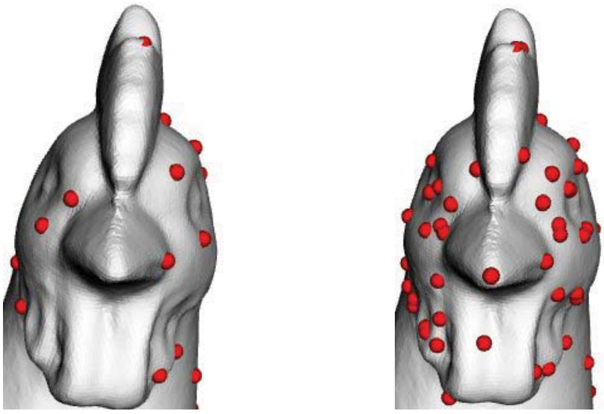
Table 3.1: Comparison between the number of vertices and number of features for all 5 models with different distinctiveness thresholds.

Model	No. of vertices	No. of features ($\gamma_t = 0$)	No. of features ($\gamma_t = 1.5$)
Dragon	134559	2910 (2.16%)	927 (0.69%)
Buddha	133127	2760 (2.07%)	745 (0.56%)
Chef	176920	2439 (1.38%)	804 (0.45%)
Chicken	135142	2273 (1.68%)	740 (0.55%)
T-rex	176508	2910 (1.65%)	983 (0.56%)

Figure 3.8 shows the features detected on the head of the 'Chicken' model using two different numbers of scales in the scale-space representations. Two different numbers of scales, 4 and 7, are used in estimating the local features in Figure 3.8a and Figure 3.8b, respectively. It can also be seen from Figure 3.9 that the more scales used in the scale-space representation, the more geometric structures of the surface are extracted. One advantage of the multi-scale approach is that it can detect coarse-scale features even though the curvature might be low. We can see that important feature points such as the one on the tip of the nose can only be detected using a high number of scales. There is, of course, a trade-off between the number of scales and the time taken to extract local features from 3D surfaces. The more scales used, the more number of salient features can be detected. However, it is also more computationally expensive to process the scale-space representation of a 3D surface.

In order to evaluate the repeatability of the feature points, white Gaussian Noise with standard deviation σ_G ranging from 0.001 to 0.1 was added to the 3D surfaces. When noise is introduced to the meshes, the variation of the local surface patches will increase. As a result, there would be more feature points detected in noisy images compared to the original one. However, it is important that the majority of the local features detected in the original surface will still be present in the noisy data.

Figure 3.10 shows the features extracted from the face of the 'Chef' model for



(a) 4 scales (b) 7 scales

Figure 3.8: Local features detected on the head of the ‘Chicken’ model using different numbers of scales in the scale-space representations.

different levels of noise. It can be seen from the figure that a large portion of local features from the original, smooth face are presented in the noisy versions. For example, there are still many feature points lying around salient structures such as the nose, chin, eyes, mouth and ears even in the noisiest surface in Figure 3.10d. With the noise level of $\sigma_G = 0.005$, most of the features in the original image appear in the noisy version. A quantitative evaluation of the repeatability of the features for five different 3D models is shown in Figure 3.11. At the noise level of $\sigma_G = 0.001$, nearly all of the features in the original model can be detected in the noisy surface. Even when the standard deviation of the noise goes to $\sigma_G = 0.1$, about 40% of the original features repeat in the noisy data.

The proposed approach was also employed to extract local features from different unstructured point clouds without connectivity information between the vertices. When applying to 3D point clouds, the Euclidean distance is used as the radius of the neighbourhood for collecting surrounding points. A kd-tree data structure is implemented to perform the local neighbour search efficiently. By fitting local manifolds directly to the surface points, the scale-space representation of the model can be created without the need to reconstruct the surface from the point cloud in advance which is a very error-prone and non-trivial process. Figure 3.12 shows the

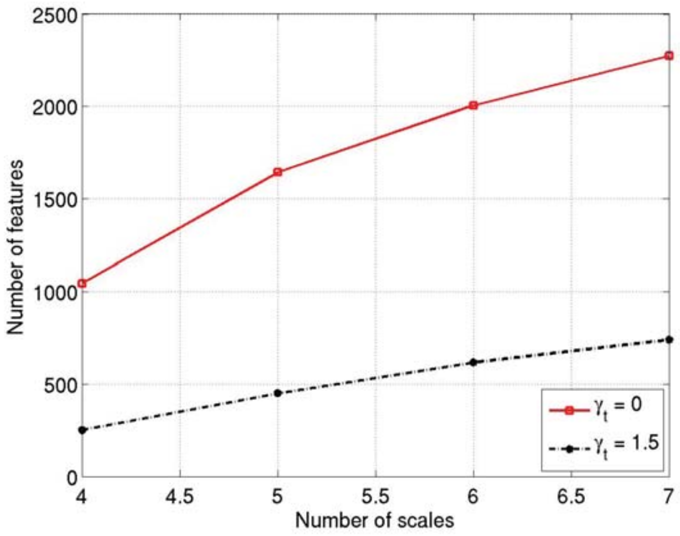
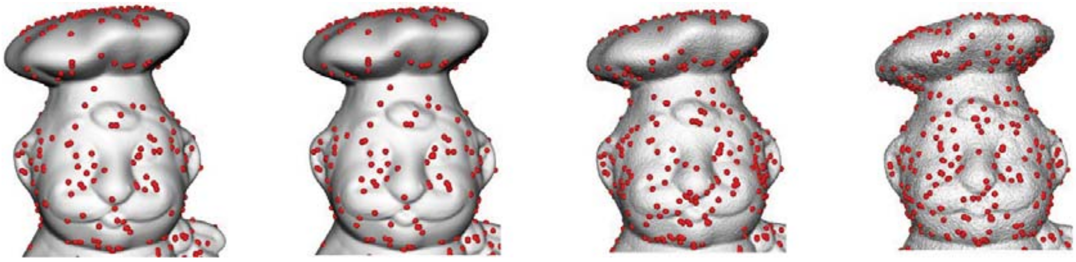


Figure 3.9: Number of features detected at different numbers of scales at two threshold levels $\gamma_t = 0$ and $\gamma_t = 1.5$ for the 'Chicken' model.



(a) Original (b) $\sigma_G = 0.005$ (c) $\sigma_G = 0.05$ (d) $\sigma_G = 0.1$

Figure 3.10: Features detected from the face of the 'Chef' model with different noise levels.

features extracted from two different point clouds of a truck and a tank constructed from simulated LADAR range images. The sizes of the spheres represent the scales where the features are extracted.

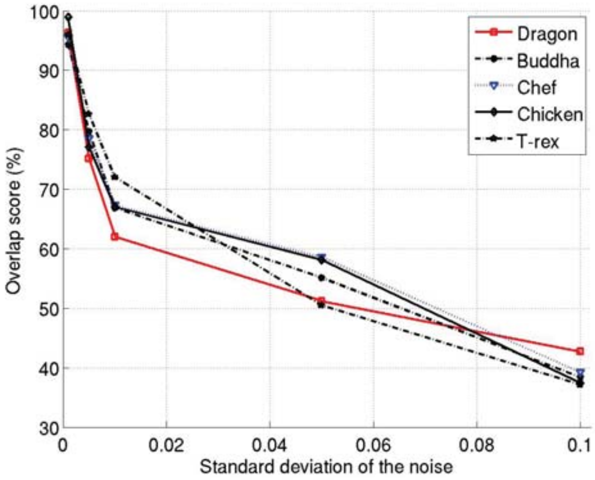


Figure 3.11: Repeatability of the features for 5 different models in different noise conditions.

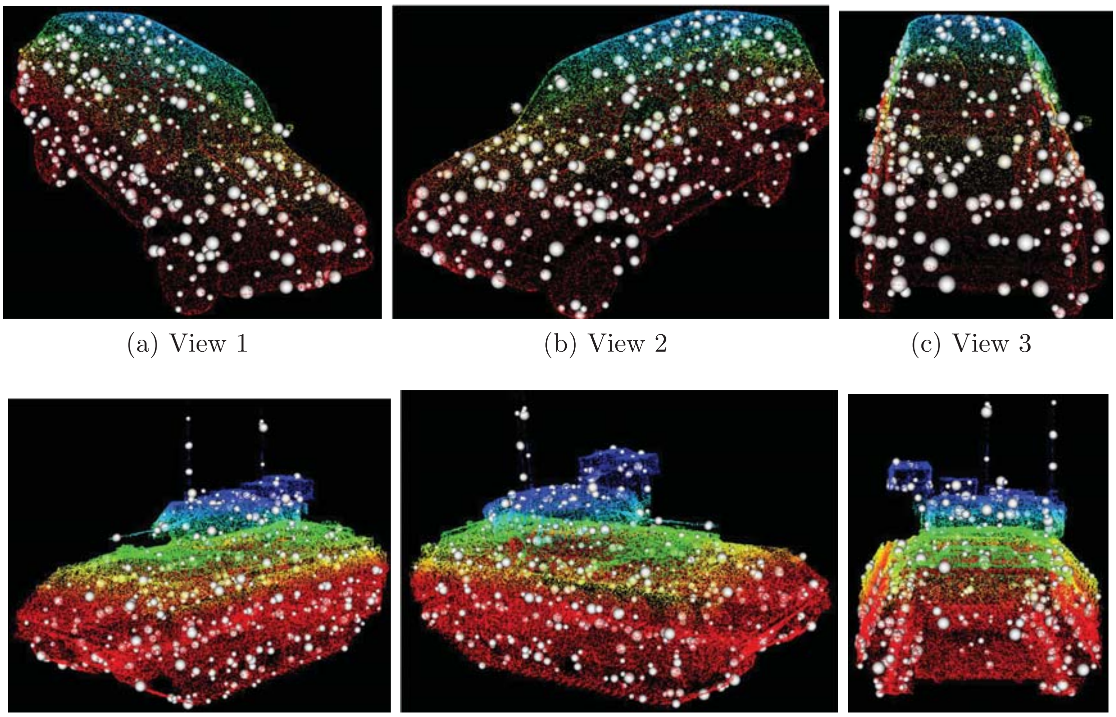
3.4 Feature Extraction using Local Shape Index Variation

3.4.1 Algorithm Description

While in Section 3.3, surface points with high local curvature were selected as salient features, another metric for choosing features is the variation in the local shape index. It is intuitively clear that regions with uniform shape index variation would not provide as much information about the surface as regions with high shape index variation. As the local surface topology at a point $\mathbf{p}$ is represented by its shape index, the local shape variation around $\mathbf{p}$ can be estimated by the standard deviation of the shape index values of $\mathbf{p}$'s neighbours based on the assumption that the distribution is Gaussian

$$\sigma_{N_{\mathbf{p}}} = \sqrt{\frac{\sum_{\mathbf{p}_j \in N_{\mathbf{p}}} (S_I(\mathbf{p}_j) - \mu_{N_{\mathbf{p}}})^2}{n}} \tag{3.15}$$

where $N_{\mathbf{p}}$ is a set of all n vertices in the neighbourhood of $\mathbf{p}$, $\mu_{N_{\mathbf{p}}}$ is the mean of the shape index in $N_{\mathbf{p}}$ and $S_I(\mathbf{p}_j)$ is the shape index at the point $\mathbf{p}_j$ as defined in



(a) View 1 (b) View 2 (c) View 3

(d) View 1 (e) View 2 (f) View 3

Figure 3.12: Feature extraction results for two different unstructured point clouds, a truck and a tank, respectively. The size of each sphere denotes the scale at which the feature is selected. The point clouds are colour-coded using the z coordinates of the 3D points.

Equation (3.6).

$$\mu_{N_{\mathbf{p}}} = \frac{\sum_{\mathbf{p}_j \in N_{\mathbf{p}}} S_I(\mathbf{p}_j)}{n} \tag{3.16}$$

The multi-scale feature selection algorithm based on local shape variation is outlined in Algorithm 3. Features are selected at surface positions $\mathbf{p}$ having maximum values σ_{N_r} both in the neighbourhood of radius r_k as well as over the above and below scales (r_{k-1}, r_{k+1}). In order to reduce the effect of noise on the algorithm and improve the reliability of the extracted features, the value of σ_{N_r} must also be greater than a threshold σ_t.

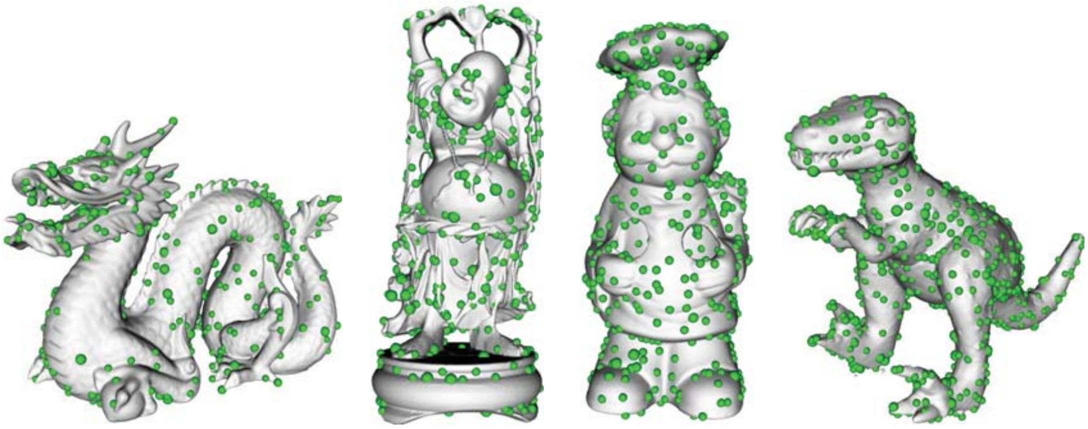
Algorithm 3 Multi-scale Feature Extraction using Local Shape Variation

```
Data:
$\mathbf{P} = \{\mathbf{p}_i \in \mathbf{R}^3\}$: set of 3D points sampled from the surface.
$\mathbf{R} = \{r_k\}$: a set of scales.
Algorithm:
for $r \in \{r_k\}$ do
    for $\mathbf{p} \in \{\mathbf{p}_i\}$ do
        Find the neighbourhood $N_r$ at scale $r$
        Fit a jet to $N_r$
        Compute principal curvatures $k_1$ and $k_2$
        Compute the shape index $S_I(\mathbf{p})$
            $S_I(\mathbf{p}) = \frac{2}{\pi} \arctan \frac{k_1+k_2}{k_2-k_1}$
        Compute the mean $\mu_{N_r}$ of the shape index
            $\mu_{N_r} = \frac{\sum_{\mathbf{p}_j \in N_r} S_I(\mathbf{p}_j)}{n_r}$
        where $n_r$ is the number of vertices in $N_r$
        Compute the local shape variation $\sigma_{N_\mathbf{p}}$
            $\sigma_{N_r} = \sqrt{\frac{\sum_{\mathbf{p}_j \in N_r} (S_I(\mathbf{p}_j) - \mu_{N_r})^2}{n_r}}$
    end for
    Features are positions $\mathbf{p}$ having maximum values $\sigma_{N_r}$ both in the neighbourhood of radius $r_k$ as well as over the above and below scales $(r_{k-1}, r_{k+1})$ and $\sigma_{N_r}$ must also be greater than a threshold $\sigma_t$.
end for
```

3.4.2 Experiments

The proposed framework for multi-scale feature extraction using local shape variation was also tested using the same standard 3D models as used in Section 3.3. Figure 3.13 shows the features extracted from four of these 3D models. The scales at which the features are selected are described by the size of the glyphs (blue spheres used to mark the positions of the features in each model). It can be seen that most surface positions with high local shape variation were chosen as features such as points on the head of the dragon model from Figure 3.13a.

The comparisons between the number of features and the number of vertices for each model are also shown in Table 3.2. In all the cases, the number of features was less than 0.5% of the number of vertices. This arose naturally from the selection process and was not a hard limit applied on the number of feature points.



(a) Dragon (b) Buddha (c) Chef (d) T-rex

Figure 3.13: Feature extraction results for four different 3D models using local shape variation. Different sizes of the glyphs correspond to different scales of the extracted features.

Table 3.2: Comparison between the number of vertices and number of features for all 4 models.

Model	No. of vertices	No. of features	Percentage
Dragon	134559	592	0.44%
Buddha	176920	640	0.36%
Chef	176508	698	0.39%
T-rex	99793	457	0.46%

A quantitative evaluation of the repeatability of features extracted from the proposed technique in different noisy conditions is shown in Figure 3.14. It can be seen that the repeatability of features extracted using local shape variation is better than the curvedness-based approach presented in Section 3.3 (Figure 3.11).

3.5 Surface Registration using Multi-scale Features and Spin-Images

In this section, the results of combining the proposed feature extraction framework and the spin-image algorithm to register 3D surfaces are presented. Without loss of

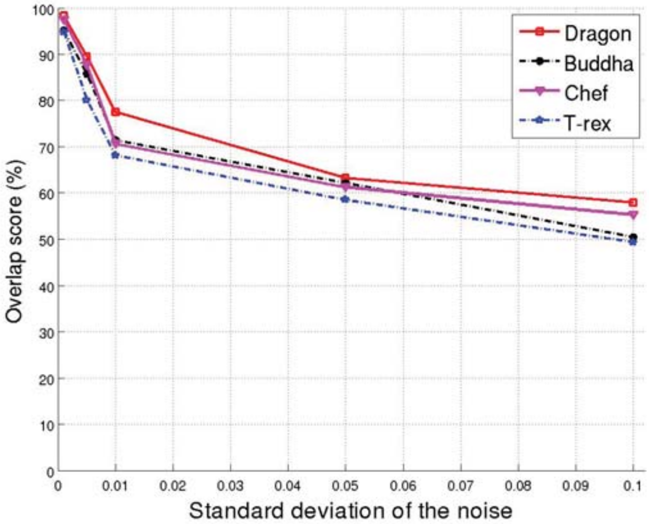
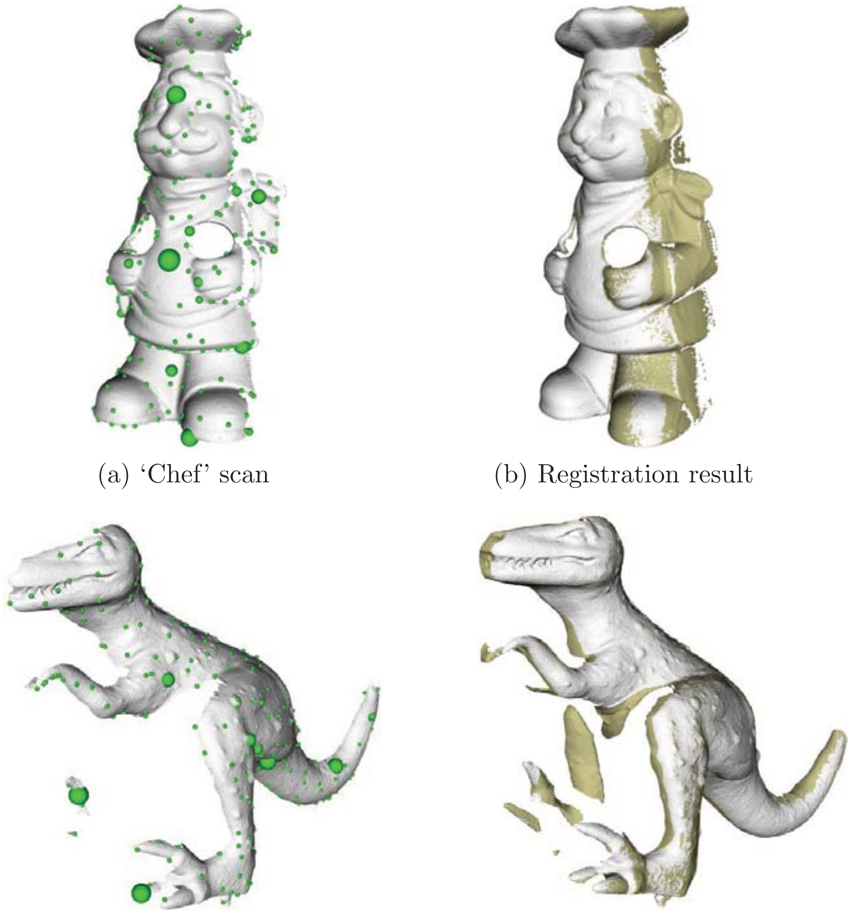


Figure 3.14: Repeatability of extracted features in different noise conditions.

generality, the feature extraction technique using local shape variation presented in Section 3.4 is selected as a preprocessing step in order to improve the accuracy and efficiency of the spin-image registration algorithm. Due to the limitation of the spin image implementation [36] in dealing with large datasets, each scan was re-sampled to contain about 10000 vertices using the cost driven approach proposed in [48]. Figures 3.15a and 3.15c show the features selected from the re-sampled scans of the 'Chef' and 'T-rex' models, respectively. To register two different views, we only performed the feature extraction on one view. The correspondences of these features were searched through all vertices of the other view to improve the probability of finding matches. Furthermore, the feature extraction step was performed off-line on the model scan so that it would not affect the actual matching time. The registration results of the two models are shown in Figure 3.15b and 3.15d.

A quantitative comparison of the registration results for two scans of the 'Chef' and 'T-rex' models by using randomly selected points and the proposed feature extraction approach is shown in Table 3.3. It can be seen from the table that the number of selected features is much smaller compared to the number of points if randomly selected. As a result, the registration time reduced significantly in both



(a) ‘Chef’ scan

(b) Registration result

(c) ‘T-rex’ scan

(d) Registration result

Figure 3.15: Registration results for 2 different views of the ‘Chef’ and ‘T-rex’ models. Positions marked in darker color are the scene scan aligned w.r.t the model scan using the estimated transformation.

cases by employing the proposed feature extraction technique. The experiments were done on a Intel Core2Duo 2.4 GHz laptop with 2GB of memory running Linux. Beside the improvement in the speed of the matching process, the proposed method also produced more accurate results as the average registration errors for the ‘Chef’ and ’T-rex’ were 0.801 and 0.615 compared to 0.887 and 0.615 if using random points, respectively. This error represents the average distance between all correspondences in a match after model points have been transformed by the match transformation [36]. The resolutions of the ‘Chef’ and ‘T-rex’ meshes were 2.06 and 1.48, respectively.

Table 3.3: Quantitative comparison of the registration using randomly selected points and local features.

Model	Random Points			Local Features		
	No. of selected points	Reg. time	Avg. error	No. of selected features	Reg. time	Avg. error
‘Chef’	5371	2m54s	0.887	187	3s	0.801
‘T-rex’	5183	2m34s	0.663	106	7s	0.615

3.6 Summary

In this chapter, we have presented two different schemes for locating salient features from 3D models as regions with high local curvature or shape variation. By fitting a truncated Taylor expansion called the jet to the neighbourhood of a surface point at different sizes, the two principal curvatures at that point are approximated at multiple scales. These principal curvature values are then used to calculate the curvedness or the shape index of the local patches in order to construct a scale-space representation of the surface. Surface positions corresponding to local extrema of the scale-space representation are chosen as feature points. The proposed algorithms are applicable to both 3D meshes with connectivity information or unstructured point clouds.

In order to demonstrate the approach, spin-images were selected as surface descriptors in order to perform surface registration with the proposed feature extraction framework. Experimental results on a number of different standard 3D models showed the effectiveness and robustness of the approach. In the next chapter, a novel registration framework for 3D datasets that searches for three pairs of corresponding features on the scene and model surfaces in order to recover the transformation between them will be presented.

Chapter 4

Surface Registration using Local Features

4.1 Introduction

In Chapter 3, the registration results of different 3D scans by combining the proposed feature extraction framework and the spin-image algorithm have been shown. It can be seen that the feature extraction technique has improved the alignment process both in terms of accuracy and efficiency. However, the registration approach using spin-images is only applicable to 3D meshes with connectivity information between the vertices. In addition, spin-images are also very sensitive to noise and variations in surface sampling [55].

In this chapter, the above issues are addressed by proposing an automatic and unified framework for the registration of both 3D meshes and unstructured point clouds using multi-scale local features, assuming that they differ by a rigid body transformation. By applying the feature extraction technique introduced in Chapter 3, salient features are automatically selected from each range image at multiple scales. A Delaunay tetrahedrization is then performed on the features to obtain a set of triangles from each range image. Possible correspondences are found by matching all possible pairs of triangles between the two views. Various geometric and rigidity constraints can be applied to prune unlikely correspondences. This pruning step is also the basis of other registration algorithms [9, 13, 70]. By finding the match that aligns the largest number of features between the two range images, the best transformation is able to be estimated. This transformation can be further refined using a variant of the Iterative Closest Point (ICP) algorithm [72]. Finally, the proposed approach is tested on a variety of 3D surfaces and point clouds with varying geometric complexity to demonstrate its effectiveness. The robustness of the proposed framework is also evaluated by applying it to register range images with different noise levels, different amounts of overlapping as well as variations in surface sampling. The block diagram of the 3D surface registration algorithm is shown in Figure 4.1.

4.2 Delaunay Tetrahedrization

Before performing the registration, scale-space representations of the input range images are constructed in order to extract local features using the curvedness-based technique discussed in Chapter 3. Without loss of generality, the method introduced in Section 3.3 is used for the feature extraction step in all experiments. Instead of trying to register all features directly, an estimate of the rigid transformation between two views of the same object can be recovered efficiently by just a pair of 3D triangles formed from the local features on each view [13]. Inspired by the approach in [70], rather than using all possible triangle combinations, the Delaunay tetrahedrization of the interest points at each scale in the scale-space can be computed. It is worth noting that the Delaunay tetrahedrization is unique and invariant to both rotation and translation [70]. The search for correspondences is only carried out between triangles at the same scale. This helps reduce the search space significantly as only a small subset of all possible triangles needs be chosen. It is not expected that all features extracted from one view are present on the other

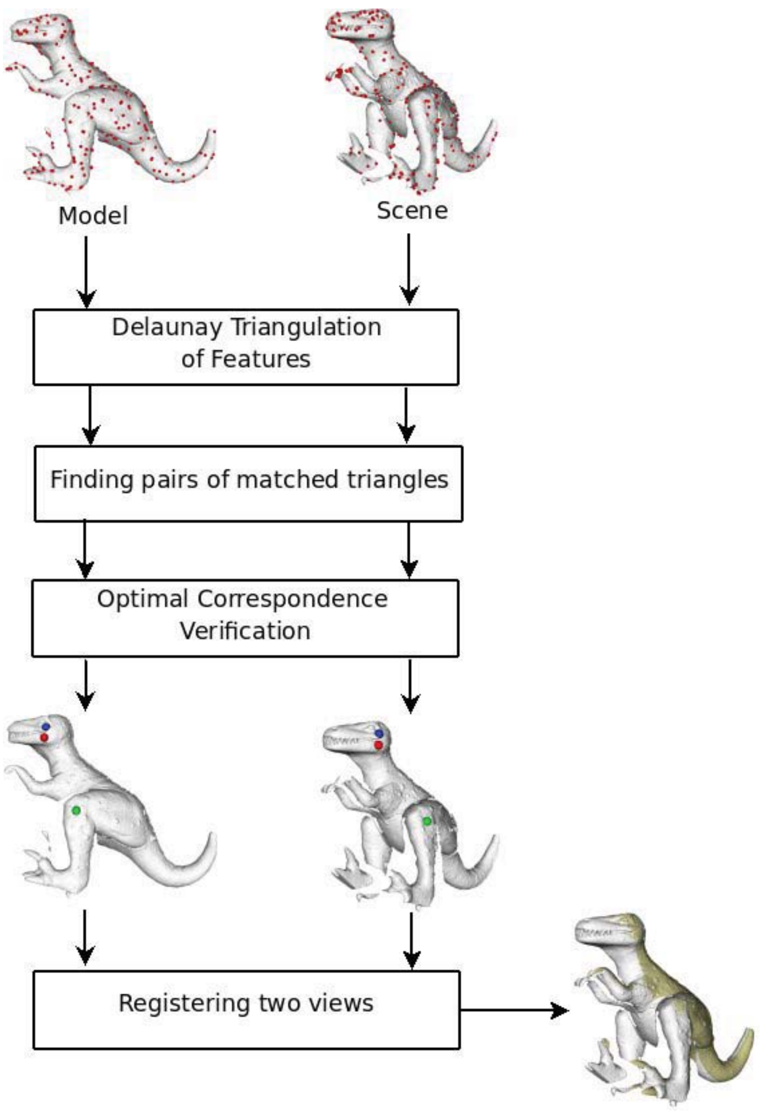

Model
Scene
Delaunay Triangulation of Features
Finding pairs of matched triangles
Optimal Correspondence Verification
Registering two views

Figure 4.1: Block diagram of the proposed 3D surface registration algorithm.

view. However, it is very likely that a majority of features chosen from the overlapping regions of the two views are similar. Thus, there is a high probability of finding a pair of matched triangles from the range images.

4.3 Rigidity Constraints

As it is still not practical to consider all possible matches from triangles at the same scale, further geometric and rigidity constraints are employed to improve both the efficiency and robustness of the registration process. If $\mathbf{s}_0$, $\mathbf{s}_1$ and $\mathbf{s}_2$ are the three vertices of a given scene triangle and $\mathbf{m}_0$, $\mathbf{m}_1$ and $\mathbf{m}_2$ are the known corresponding vertices of the matched triangle on the model view, there is always a slight mismatch error between the transformed scene vertices $\mathbf{s}'_0$, $\mathbf{s}'_1$ and $\mathbf{s}'_2$ and the model vertices. This is a result of measurement errors and noise introduced by the sampling process as well as the fact that only discrete surface points are available from each view. It can be seen that the greater the areas of the matched triangles are, the less their alignment error will propagate to the whole surface [13]. In other words, any alignment error between the transformed scene points $\mathbf{s}'_0$, $\mathbf{s}'_1$ and $\mathbf{s}'_2$ and the model points $\mathbf{m}_0$, $\mathbf{m}_1$ and $\mathbf{m}_2$ would cause greater errors for other points away from the triangles. Alternatively, choosing triangles with larger areas will result in smaller errors and better registration.

In the proposed framework, only triangles whose areas are greater than a threshold, A_t, will be considered in the subsequent steps. This will help both reduce the number of possible correspondences and minimize the possible error introduced when considering small triangles as the best matches. The value of A_t for each view was chosen as the average area of the scan's triangles for all experiments. Figure 4.2 plots the number of triangles whose areas are greater than the threshold for four different 3D models at different mesh resolutions. It can be seen that the number of triangles increases linearly with the number of vertices of each view. The use of the area threshold will prevent the algorithm from suffering combinatorial explosion when dealing with large 3D datasets. It is worth noting that the threshold value A_t can also be chosen so that selected triangles are in the upper quantile of the distribution of triangle areas.

Because the distance between two particular points does not change under a rigid transformation, it is expected that the corresponding edges of the matched triangles

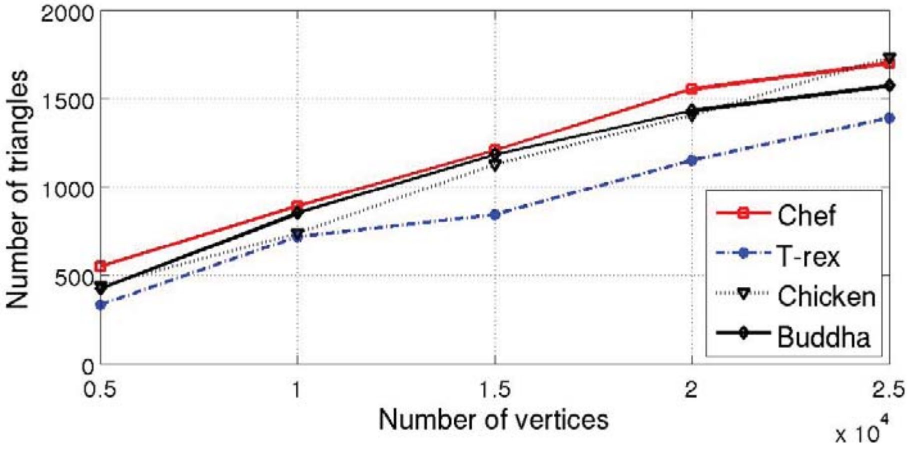


Figure 4.2: Number of triangles at different mesh resolutions.

on the scene and model views are approximately equal.

$$|d(\mathbf{s}_0, \mathbf{s}_1) - d(\mathbf{m}_0, \mathbf{m}_1)| < \tau_d(d(\mathbf{s}_0, \mathbf{s}_1), d(\mathbf{m}_0, \mathbf{m}_1)) \tag{4.1}$$

$$|d(\mathbf{s}_0, \mathbf{s}_2) - d(\mathbf{m}_0, \mathbf{m}_2)| < \tau_d(d(\mathbf{s}_0, \mathbf{s}_2), d(\mathbf{m}_0, \mathbf{m}_2)) \tag{4.2}$$

$$|d(\mathbf{s}_1, \mathbf{s}_2) - d(\mathbf{m}_1, \mathbf{m}_2)| < \tau_d(d(\mathbf{s}_1, \mathbf{s}_2), d(\mathbf{m}_1, \mathbf{m}_2)) \tag{4.3}$$

where $d(\mathbf{p}, \mathbf{q})$ is the Euclidean distance between two 3D points $\mathbf{p}$ and $\mathbf{q}$. The distance tolerance $\tau_d(d_1, d_2)$ can be estimated as a percentage of the maximum edge length $d_{max} = max(d_1, d_2)$. The reason for varying the value of τ_d with the edge length is to give more tolerance to correspondences based on larger triangles than those based on smaller ones.

The curvedness of a surface point is invariant to translation and rotation and so, a pair of triangles is only considered a possible match if the curvedness values of their corresponding vertices are similar

$$|c_{\mathbf{s}_i} - c_{\mathbf{m}_i}| < \varepsilon_c \quad i = 0, 1, 2 \tag{4.4}$$

where $c_{\mathbf{p}}$ is the normalised curvedness of the 3D point $\mathbf{p}$ and ε_c is the error tolerance. This compatibility test can also be considered as a geometric constraint on surface patches as their shapes do not change under the rigid transformation.

4.4 Transformation Estimation

If a pair of two triangles satisfies all the above rigidity and curvedness constraints, the transformation that aligns them is computed. In order to estimate the rigid transformation between them, two additional vertices, $\mathbf{s}_3$ and $\mathbf{m}_3$, are introduced to the scene and model triangles, respectively (Fig. 4.3)

$$\mathbf{s}_3 = \mathbf{s}_0 + \frac{(\mathbf{s}_1 - \mathbf{s}_0) \times (\mathbf{s}_2 - \mathbf{s}_0)}{|(\mathbf{s}_1 - \mathbf{s}_0) \times (\mathbf{s}_2 - \mathbf{s}_0)|} \tag{4.5}$$

$$\mathbf{m}_3 = \mathbf{m}_0 + \frac{(\mathbf{m}_1 - \mathbf{m}_0) \times (\mathbf{m}_2 - \mathbf{m}_0)}{|(\mathbf{m}_1 - \mathbf{m}_0) \times (\mathbf{m}_2 - \mathbf{m}_0)|} \tag{4.6}$$

where the × operator denotes the cross product of two vectors in the 3D Euclidean space.

The rigid transformation $\mathbf{T}$ between the two triangles contains a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$ satisfying

$$\mathbf{R}\mathbf{s}_i + \mathbf{t} = \mathbf{m}_i \quad i = 0, 1, 2, 3 \tag{4.7}$$

Assume that $\mathbf{S}^{-1}$ exists, the rotation matrix $\mathbf{R}$ can be estimated in closed form as [77]

$$\mathbf{R} = \mathbf{M}\mathbf{S}^{-1} \tag{4.8}$$

where

$$\mathbf{S} = \begin{bmatrix}\mathbf{s}_1 - \mathbf{s}_0 & \mathbf{s}_2 - \mathbf{s}_0 & \mathbf{s}_3 - \mathbf{s}_0\end{bmatrix} \tag{4.9}$$

$$\mathbf{M} = \begin{bmatrix}\mathbf{m}_1 - \mathbf{m}_0 & \mathbf{m}_2 - \mathbf{m}_0 & \mathbf{m}_3 - \mathbf{m}_0\end{bmatrix} \tag{4.10}$$

Once the rotation matrix has been caculated, the translation vector $\mathbf{t}$ can be obtained from (4.7)

$$\mathbf{t} = \overline{\mathbf{m}} - R\overline{\mathbf{s}} \tag{4.11}$$

where $\overline{\mathbf{s}}$ and $\overline{\mathbf{m}}$ are the average of $\mathbf{s}_i$ and $\mathbf{m}_i$

$$\overline{\mathbf{s}} = \frac{\sum_{i=0}^{3} \mathbf{s}_i}{4} \qquad \overline{\mathbf{m}} = \frac{\sum_{i=0}^{3} \mathbf{m}_i}{4} \tag{4.12}$$

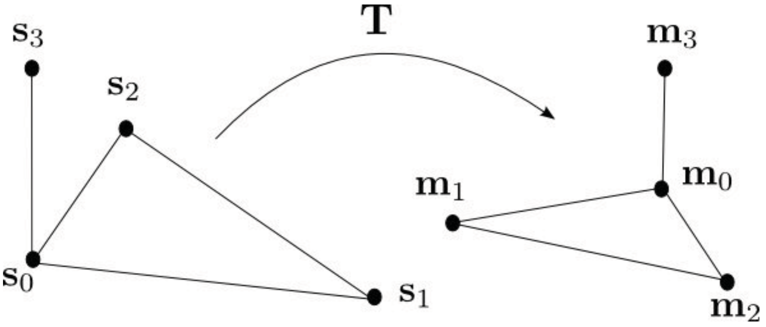


Figure 4.3: Estimating transformation from two 3D triangles.

4.5 Optimal Transformation Validation

As there is a possibility that more than one pair of triangles satisfy the above compatibility tests, the pair of triangles whose transformation aligns the most number of features is chosen as the best correspondence. This is carried out in the following way. After finding two matched triangles, the transformation is computed and applied to the rest of the scene features to align them with respect to the model features. Next, a search for the closest model feature for each of the above transformed scene features is performed. A k-d tree is implemented to do the nearest neighbor search efficiently. The alignment error between the transformed scene feature $\mathbf{s}'_i$ and its closest model feature $\mathbf{m}_i$ is estimated using the Gaussian weighted distance metric [73]

$$G_i = e^{-d_i^2/2\sigma^2} \tag{4.13}$$

where $d_i^2 = ||\mathbf{s}'_i - \mathbf{m}_i||^2$ is the squared Euclidean distance between a transformed feature and its closest model feature. The degree of interaction between the two sets of features is controlled by σ. Global interactions between the features are achieved with large σ while local interactions are enforced with a smaller value. The value of G_i ranges from 0 (the features are far away ($d_i = \infty$)) to 1 (the features are in exactly the same position ($d_i = 0$)). If the number of scene features is N, the global quality of the alignment can be calculated as

$$E = \sum_{i=0}^{N-1} G_i \tag{4.14}$$

The larger the value of E, the more features in the two scans are aligned. This metric works well in the case of little overlap between the views as the Gaussian distance limits the effect of outliers to the total alignment error.

The process is repeated for all pairs of triangles that satisfy the compatibility tests. The best correspondence is chosen as the pair of triangles whose transformation gives the largest value of the alignment quality metric E. The proposed pairwise surface registration algorithm using multi-scale local features is outlined in Algorithm 4.

Algorithm 4 Pairwise Surface Registration using Multi-scale Local Features

Data:
$\mathbf{M} = \{\mathbf{m}_i \in \mathbf{R}^3\}$: set of features extracted from the model surface.
$\mathbf{S} = \{\mathbf{s}_i \in \mathbf{R}^3\}$: set of features extracted from the scene surface.
$\mathbf{R} = \{r_k\}$: a set of scales.
A_t: area threshold.

Algorithm:

```
for $r \in \{r_k\}$ do
    Delaunay tetrahedrization of the features at scale $r$ in $\mathbf{M}$
        $TM_r = \{tm_{r_i}\}$: set of triangles obtained on the model surface at scale $r$
    Delaunay tetrahedrization of the features at scale $r$ in $\mathbf{S}$
        $TS_r = \{ts_{r_i}\}$: set of triangles obtained on the scene surface at scale $r$
    for $ts_r \in \{ts_{r_i}\}$ do
        Calculate the area $A_{ts_r}$of the scene triangle $ts_r$
        if $A_{ts_r} > A_t$ then
            for $tm_r \in \{tm_{r_i}\}$ do
                Calculate the area $A_{tm_r}$of the model triangle $tm_r$
                if $A_{tm_r} > A_t$ then
                    if $ts_r$ and $tm_r$ satisfy rigidity constraints then
                        Estimate the transformation $\mathbf{T}$ between the triangles
                        Transform all features in $\mathbf{S}$ using $\mathbf{T}$
                            $\mathbf{S}' = \{\mathbf{s}'_i\}$: set of transformed scene features
                        Calculate the alignment quality metric $E$ between $\mathbf{S}'$ and $\mathbf{M}$
                    end if
                end if
            end for
        end if
    end for
    The best correspondence is the pair of triangles whose transformation gives
    the largest value of the alignment quality metric $E$.
end for
```

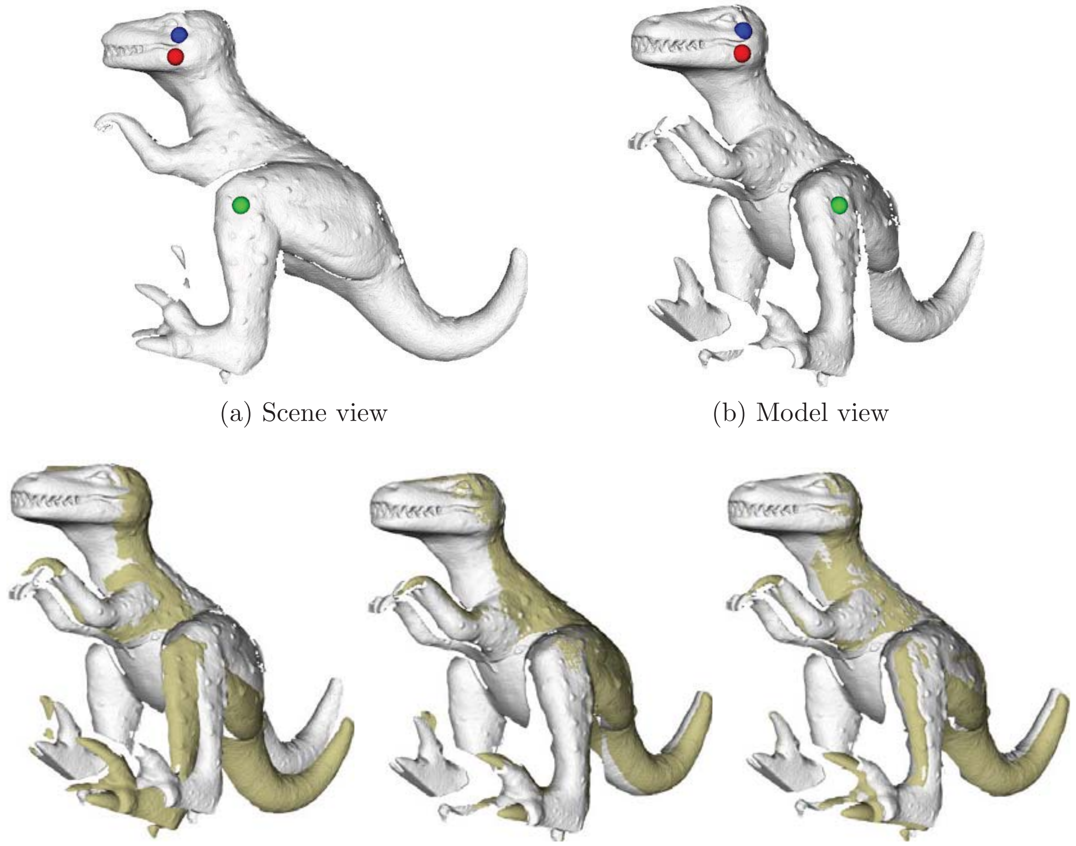
4.6 Experiments

The proposed registration framework was tested on a broad range of 3D datasets including both 3D meshes and unstructured point clouds. All the range images used in the experiments were taken from publicly available databases [53, 75, 79]. Furthermore, the robustness of the proposed algorithm is evaluated by testing it using range images with different noise conditions and surface resolutions. In all of experiments conducted, feature points are extracted from the input range image using five different scales. In the case of 3D meshes, the scales correspond to $r = 1$ to $r = 5$ (from the first to the fifth ring). For unstructured point clouds, the radius of the neighborhood for searching fitting points ranges from 1% to 10% of the length of the diagonal of the point cloud's bounding box.

4.6.1 3D Surfaces

In this section, the results of applying the above approach on different 3D meshes are presented. In order to better demonstrate the proposed method, the coarse registration results obtained from the above technique are compared with the ones using the spin-image algorithm [36]. The C++ implementation of the spin-image algorithm was downloaded from the authors' website [83].

Figure 4.4 shows the results of coarsely registering two views of the 'T-rex' model using the above approach and the spin-image method without any prior knowledge about the initial alignment. Each view has about 10k vertices. The corresponding vertices of the triangles used for matching are shown with the same color (Figures 4.4a and 4.4b). Positions marked in darker color are the scene scan aligned with respect to the model scan. The fine registration result obtained by applying the ICP [3] is shown in Figure 4.4e. From Figures 4.4a and 4.4b, it can be seen that the proposed method can accurately locate a pair of matched triangles from the two different views of the 'T-rex' model and calculate the transformation between the views. It is also obvious from Figures 4.4c and 4.4d that the two range images are



(a) Scene view

(b) Model view

(c) Spin-image result

(d) The proposed approach

(e) Result of the proposed approach after ICP

Figure 4.4: Pairwise registration results for the 'T-rex' model. Positions marked in darker color are the scene scan aligned w.r.t. the model scan using the estimated transformation.

aligned better using the proposed method than the spin-image technique. Further comparisons of the approach and the spin-image technique for three other models, namely, the 'Chef', 'Chicken' and 'Buddha', are shown in Figure 4.5. The proposed method clearly provides better alignments for the 'Buddha' and 'Chef' models. However, the proposed method has higher translation error in one direction in the case of the 'Chicken' model compared to the result by using spin-images.

These qualitative results are confirmed by the quantitative comparisons of the translation and rotation errors between the spin-image and the proposed approach in Tables 4.1 and 4.2, respectively. All models used in the experiments are at the same scale. For the Buddha model, the errors were estimated by comparing the coarse transformations obtained from the proposed approach and the spin-image method with the ground truth transformation. However, as the ground truth transformations of the scans of the 'Chef', 'Chicken' and 'T-rex' models are not available, the errors were calculated by comparing the coarse transformations with the refined transformations using ICP. It can be seen that the proposed method outperforms the spin-image approach in both the translation and rotation error metrics. For the 'Chicken' model, the proposed method has slightly higher translation error in the x axis that is consistent with the observations in Figure 4.5. Most of the rotation errors in the registration results using the proposed technique are smaller than 1°. This is a very desirable result for any coarse registration algorithm.

Table 4.1: Comparison of the translation errors.

	Spin-image			The proposed method		
Model	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$	$t_x(mm)$	$t_y(mm)$	$t_z(mm)$
Chef	2.29	23.78	111.29	1.64	5.94	12.67
T-rex	14.84	38.92	26.43	1.95	1.49	0.54
Chicken	0.41	22.05	24.21	3.25	9.42	10.21
Buddha	6.09	0.11	0.08	5.94	0.57	0.08

Table 4.2: Comparison of the rotation errors.

	Spin-image			The proposed method		
Model	$\theta_x(°)$	$\theta_y(°)$	$\theta_z(°)$	$\theta_x(°)$	$\theta_y(°)$	$\theta_z(°)$
Chef	22.95	1.39	1.92	0.18	0.39	0.19
T-rex	5.84	2.25	6.65	0.36	1.15	2.01
Chicken	4.08	0.45	0.68	0.79	0.36	0.82
Buddha	0.57	18.63	2.66	0.31	3.39	1.02

Table 4.3 compares the registration time between the proposed approach and the spin-image method for two views of the 'Chef' model. Each original scan of the Chef model which contains about 70k vertices was simplified to smaller meshes

using the approach in [48]. The registration time was measured at five different mesh resolutions for both the proposed approach and the spin-image method. The executing time for the proposed method includes both the time taken for extracting local features at different scales as well as finding the best pair of matched triangles. It can be seen from the table that the proposed approach is much more efficient than the spin-image method. In all the cases, it took the proposed method less than one minute to register the scans. For the pair of scans with 25k vertices, the spin-image took more than 2.7 hours to calculate the best transformation. Even at the original mesh resolution of 70k, the proposed approach only spent about 2 minutes for each pair of scans. Both the spin-image method and the proposed technique were implemented in C++ and tested on an Intel 2.4GHz laptop with 2GB of memory.

Table 4.3: Comparison of the registration time for the Chef model.

Number of Vertices	Spin-images	The proposed approach
$5k$	$71s$	$12s$
$10k$	$2m54s$	$22s$
$15k$	$16m12s$	$34s$
$20k$	$60m56s$	$44s$
$25k$	$166m$	$52s$

4.6.2 Noise and varying sampling densities

In order to evaluate the robustness of the proposed approach in the presence of noise, white Gaussian noise with standard deviation σ ranging from 0.025 to 0.5 was injected into the range scans. Figures 4.7a and 4.7b plot the average translation and rotation errors at different values of σ for the registration of two views of the T-rex model using the approach. It can be seen that even at the noise level of $\sigma = 0.5$, the proposed framework can still accurately align the scans with the average translation error less than $7mm$ and average rotation error less than $3°$.

The registration result at $\sigma = 0.5$ is shown in Figure 4.6.

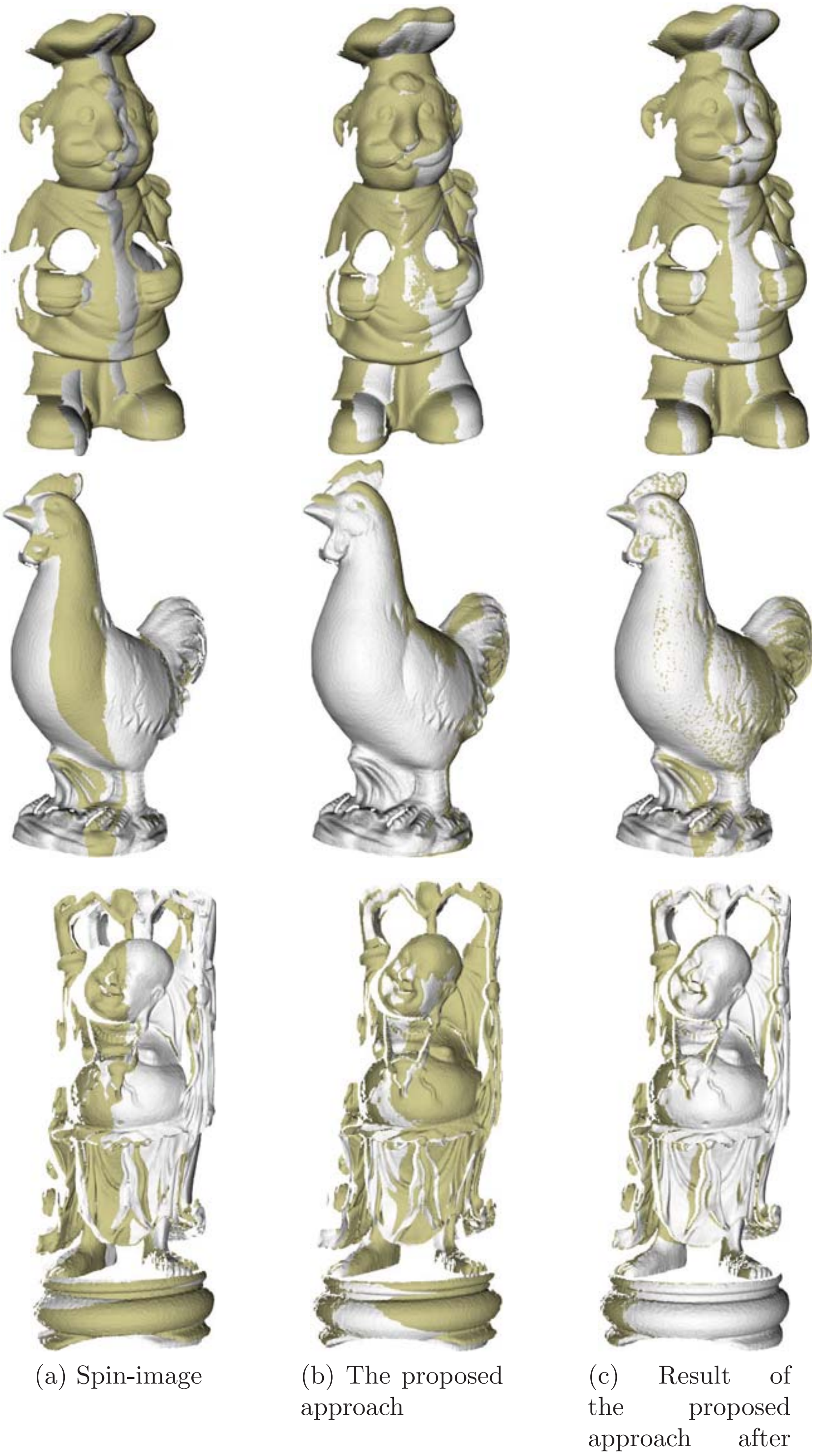
The performance of the proposed approach in registering range images with very different mesh resolutions was also tested. Figure 4.8a shows the scene scan of the T-rex model with 1k vertices. It is registered with the model scan containing 10k vertices. The results using the spin-image technique and the proposed approach are given in Figures 4.8c and 4.8d respectively. It is clear that the proposed framework still outperforms the spin-image method and provides relatively accurate results.

4.6.3 Unstructured Point Clouds

Results of using the proposed framework for registering unorganized point clouds are shown in Figure 4.9. Each view of the helicopter and carrier model contains about 10k and 24k data points, respectively. For the car model, there are approximately 50k points for each scan. All the views of the three models were obtained from the Stuttgart Range Image Database [79]. Despite a large rotation angle of about 90° between the two scans of the helicopter, the method was still able to align them accurately as seen in Figure 4.9c. Figures 4.9f and 4.9i present other successful registration results for the carrier and car model, respectively.

4.7 Summary

In this chapter, an automatic framework for registering two range scans that differ by a rigid body transformation has been presented. First, salient features are extracted from the input range images. A set of triangles is then created from each scan by performing the Delaunay tetrahedrization on its feature points. Rather than using an exhaustive searching strategy, possible pairs of matched triangles between the two views are found based on various surface geometric and rigidity constraints. The best correspondence is the pair of triangles whose transformation aligns the most number of features on the two range scans. The proposed technique provides a unified approach to the registration of both triangulated meshes and point clouds. Furthermore, it is robust to surface resolution as well as noise. Results of applying the framework to a number of different 3D models illustrate its effectiveness and robustness. In the next chapter, the algorithm will be extended to the problem of 3D object recognition from range images. Using a similar approach to this framework, it will be shown that the proposed method is able to recognize objects from 3D scenes with occlusion, clutter and noise.

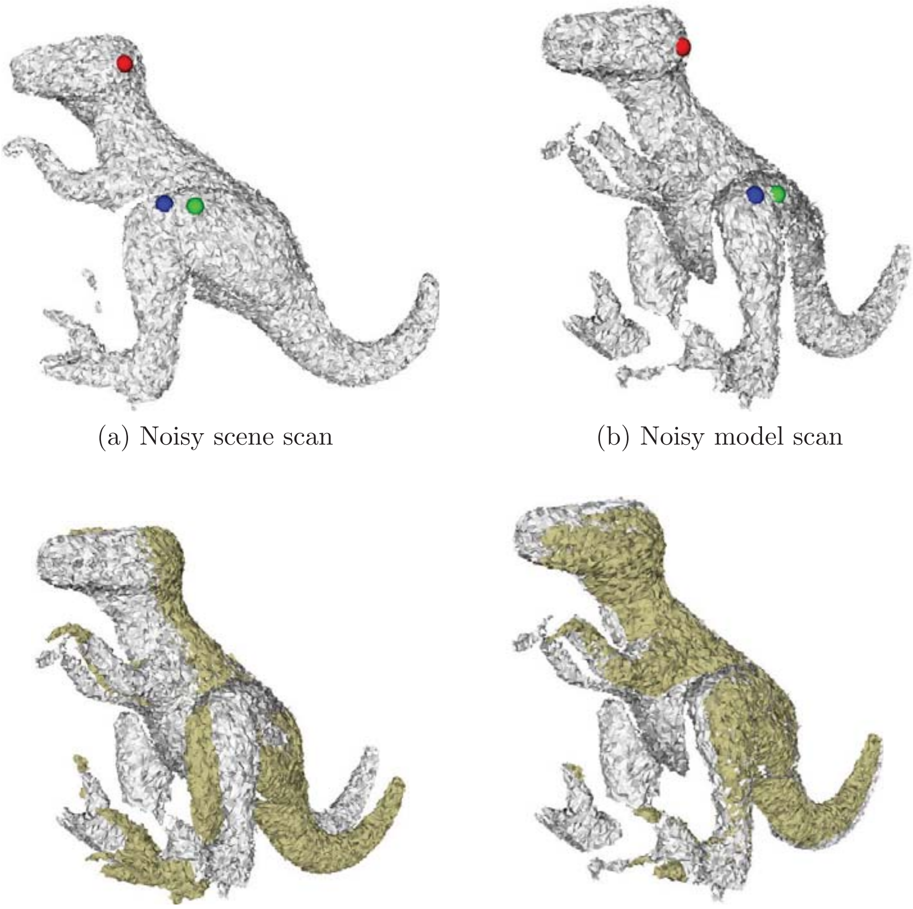


(a) Spin-image

(b) The proposed approach

(c) Result of the proposed approach after ICP

Figure 4.5: Pairwise registration results for other 3D models.



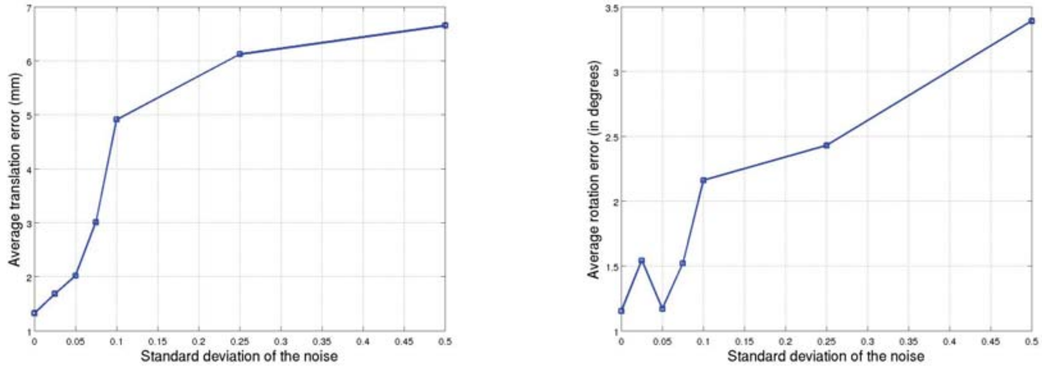
(a) Noisy scene scan

(b) Noisy model scan

(c) Spin-image

(d) The proposed approach

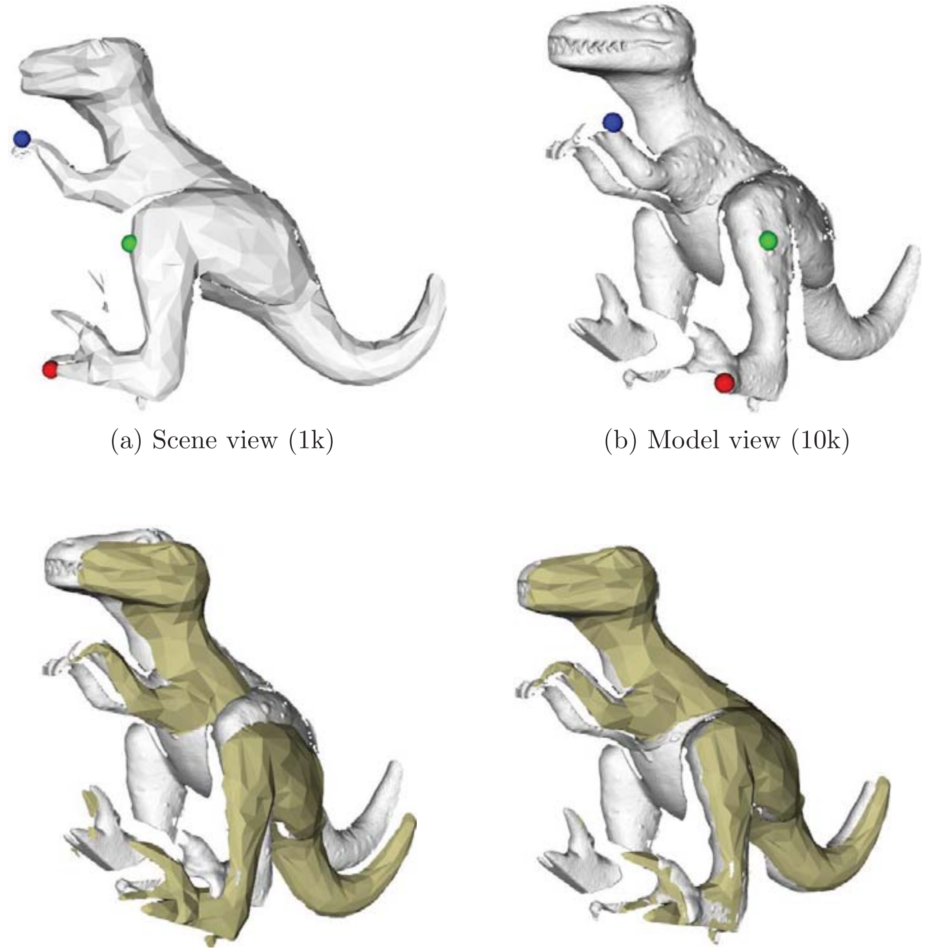
Figure 4.6: Pairwise registration results of noisy T-rex scans at $\sigma = 0.5$.



(a) Average translation error

(b) Average rotation error

Figure 4.7: Transformation errors for noisy scans of the T-rex model



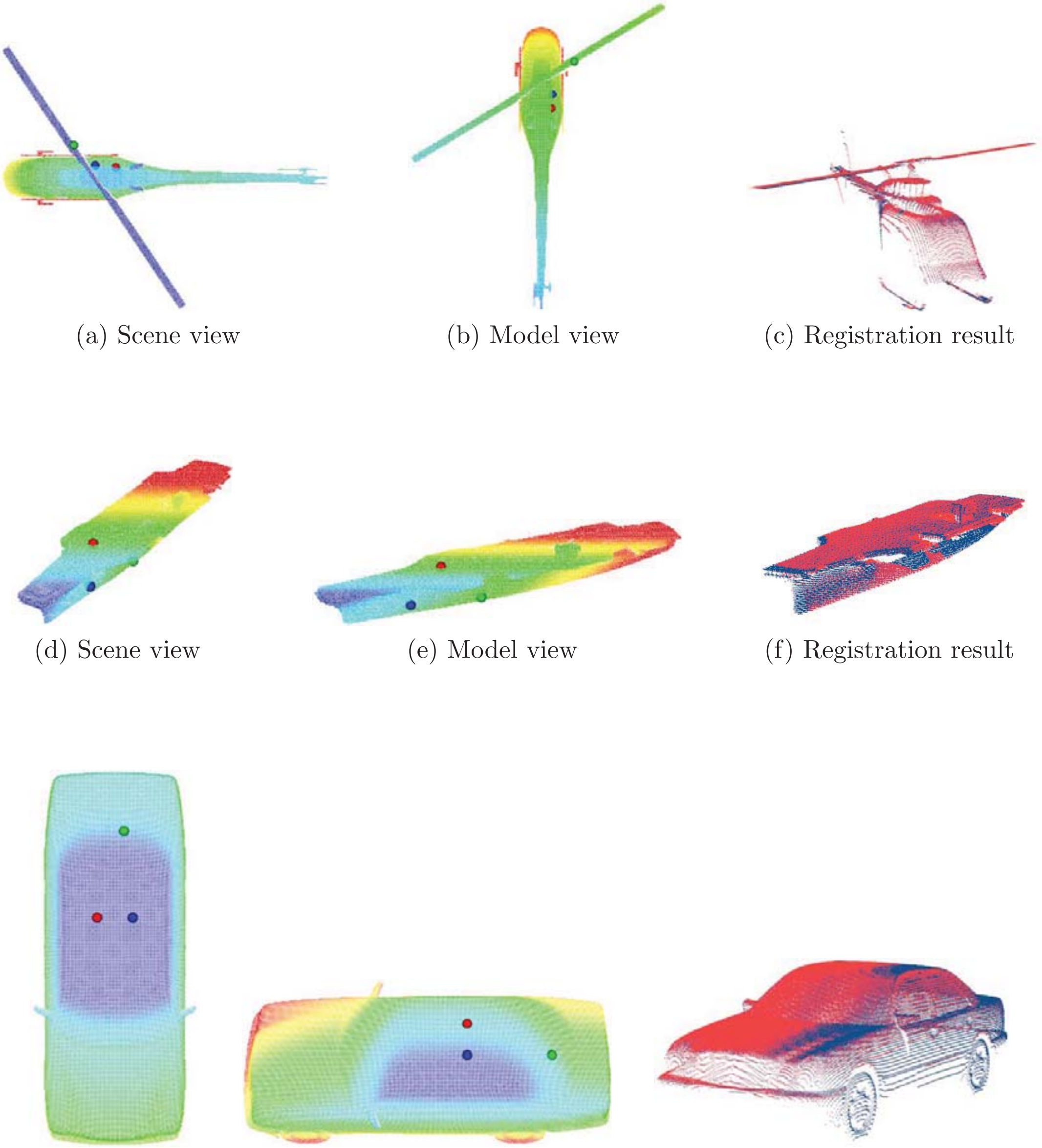
(a) Scene view (1k)

(b) Model view (10k)

(c) Spin-image

(d) Our approach

Figure 4.8: Pairwise coarse registration of T-rex scans with different mesh resolutions.



(a) Scene view

(b) Model view

(c) Registration result

(d) Scene view

(e) Model view

(f) Registration result

(g) Scene view

(h) Model view

(i) Registration result

Figure 4.9: Pairwise coarse registration results for unstructured point clouds. Points in each scan are color coded according to their z coordinates. The colormap is from blue to red corresponding to low to high values of z. In the registered views, the model points and scene points are shown in blue and red, respectively. Figures are best seen in color.

Chapter 5

3D Object Recognition from Range Images

5.1 3D Object Recognition Framework

Automatic recognition of objects in a 3D scene is an important task in computer vision and robotics. However, it is also a challenging problem due to occlusion and clutter generated by the presence of multiple objects and noise in the scene. As discussed in Chapter 2, existing 3D object recognition techniques still have many limitations in terms of accuracy, efficiency and especially the ability to handle scenes with a high degree of clutter and occlusion.

In this chapter, experimental results of applying the proposed multi-scale feature extraction framework in recognizing 3D objects from range images are presented. The approach is similar to the registration method presented in Chapter 4. First, local features are extracted from both the 3D model and the scene at multiple scales. Two sets of triangles are obtained by performing the Delaunay tetrahedrization both for the features on the scene and model surfaces. In order to estimate the location and orientation of the object in the scene, a pair of matched triangles between the scene and the model is found using the rigidity constraints as presented in Section 4.3. The verification step is performed by transforming the model's features to the scene coordinate system using the estimated transformation matrix and calculating the overlapping metric E as in Equation (4.14). The pair of triangles whose transformation matrix gives the largest value of E is chosen as the optimal correspondence. If the overlapping metric E is greater than a threshold E_t, we decide that the searched model exists in the scene and its pose is given by the transformation estimated from the optimal correspondence. The block diagram of the 3D object recognition algorithm is shown in Figure 5.1.

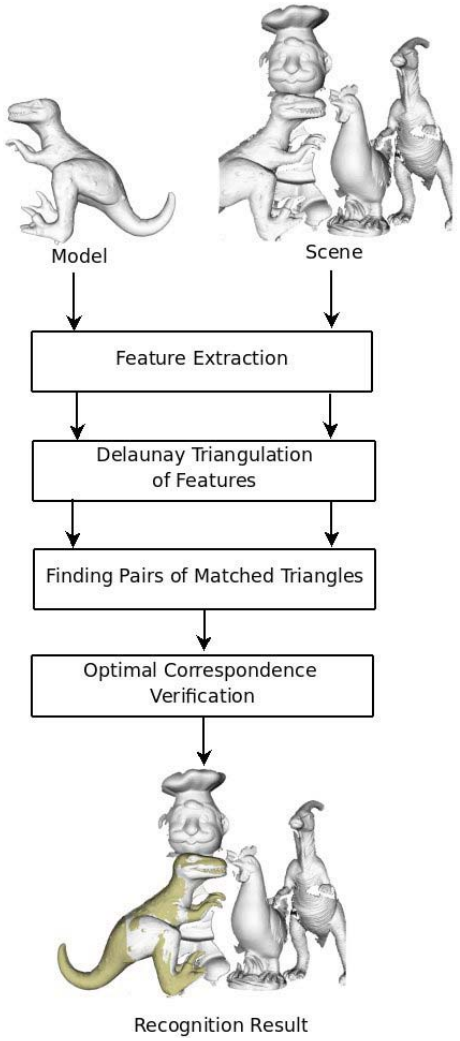
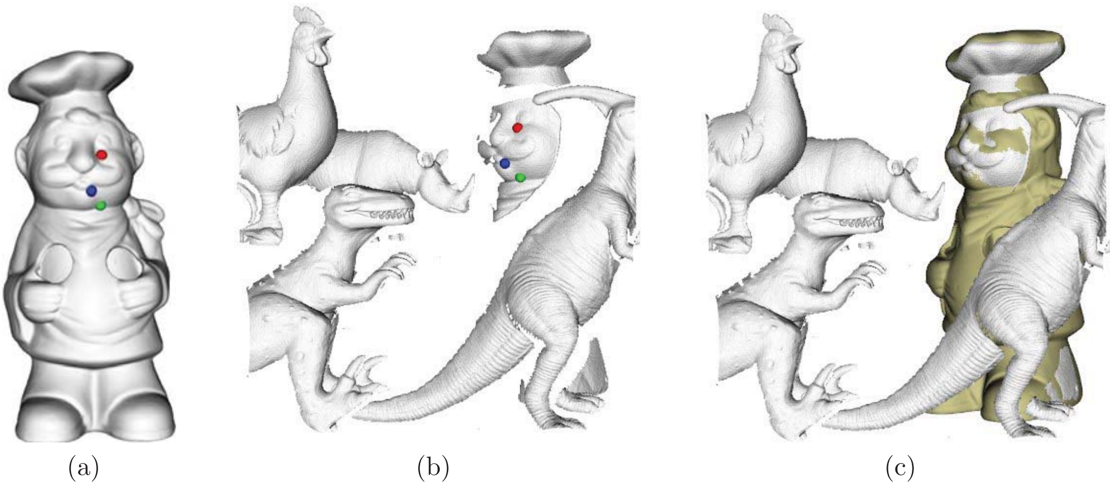

Model
Scene
Feature Extraction
Delaunay Triangulation of Features
Finding Pairs of Matched Triangles
Optimal Correspondence Verification
Recognition Result

Figure 5.1: Block diagram of the proposed 3D object recognition algorithm.

In order to improve the recognition results in scenes with a high degree of occlusion



(a) (b) (c)

Figure 5.2: Recognition result for a highly occluded scene. Only about 5% of the model's surface area appears in the scene. Location of the Chef is shown as model vertices superimposed on the scene.

and clutter, the features' distinctiveness values can be used in estimating the Gaussian error metric in the verification step. The Gaussian weighted distance metric between two features i and j in Equation (4.13) introduced in Section 4.5 can be rewritten as

$$G_{ij} = e^{-d_{ij}^2/(2\gamma_i\gamma_j\sigma^2)} \tag{5.1}$$

where γ_i and γ_j are the distinctiveness values of the two corresponding features i and j, respectively. These values are estimated using Equation (3.12). The purpose of adding the distinctiveness values in calculating the overlapping error metric is that a pair of highly distinctive features would contribute more significantly to the metric than a pair of features with low distinctiveness values. Figure 5.2 shows a successful example of recognizing the Chef model in a highly occluded scene using the above modified error metric. The amount of occlusion in the scene is about 95%. Without using the distinctiveness values in the overlapping metric, the technique has failed to locate the position of the Chef in this scene.

In experiments carried out, the level of occlusion and clutter are calculated using the formula defined in [36, 57]

$$occlusion = 1 - \frac{S_m}{S_M} \tag{5.2}$$

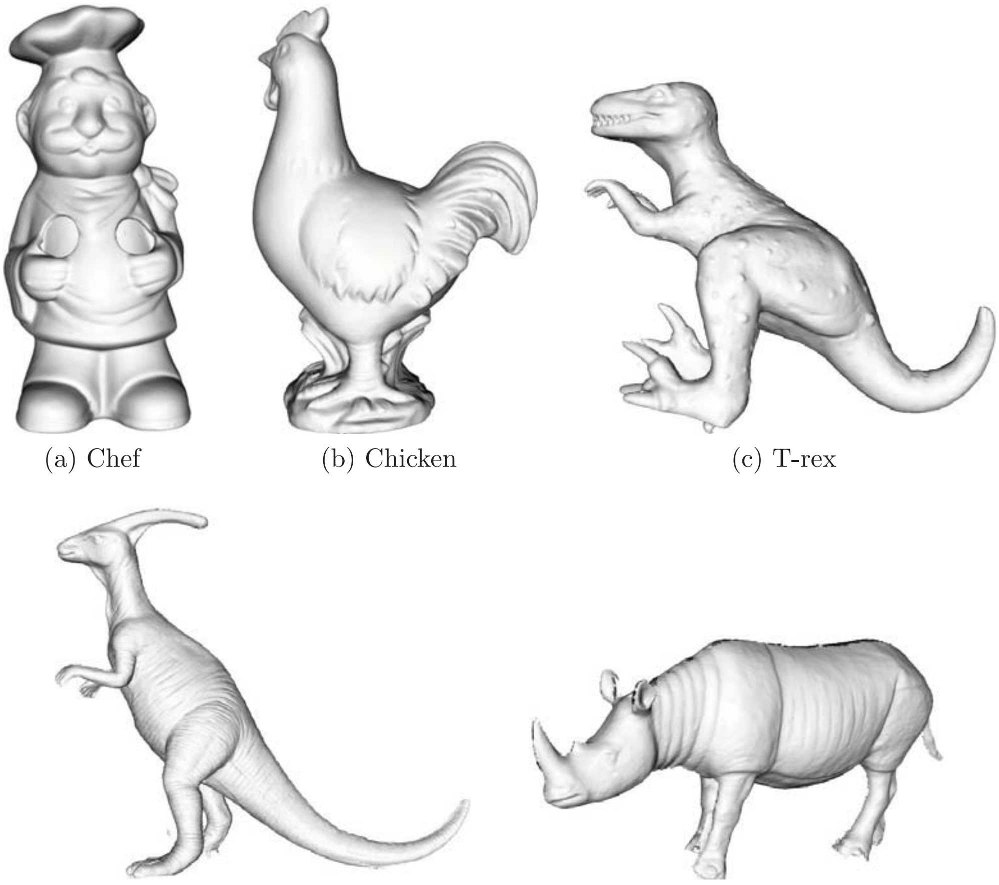
$$clutter = 1 - \frac{S_m}{S_S} \tag{5.3}$$

where S_m is the surface area of the model presented in the scene, S_M is the total surface area of the complete model and S_S is the total surface area of the scene. In order to calculate the amount of occlusion and clutter, it is necessary to estimate the overlap area of the model and the scene. This can be achieved by transforming the complete model to the scene coordinate system using the location and orientation parameters obtained after the recognition process. For each vertex on the model surface, the closest vertex on the scene surface is considered as overlapping if their normals are similar (the angle difference is within a threshold) [31]. The overlapping area is estimated by summing the area of all triangle facets in the overlapping region. For models that could not be automatically recognized from the scene, they were manually segmented in order to calculate the overlapping area.

5.2 Experimental Results

Figure 5.3 shows five different 3D models used in recognition experiments conducted, namely the Chef, Chicken, T-rex, Parasaurolophus and Rhino. These models were downloaded from A. Mian's homepage [53]. It is worth noting that the Rhino model was intentionally reconstructed with large holes in order to test the robustness of the matching algorithm with incomplete data [57]. The models were matched against 50 real scenes generated by randomly placing four or five of the objects together and scanning with the Minolta Vivid 910 scanner. The scenes are also available from the same website. Some sample scenes and the recognition results using the proposed approach are shown in Figure 5.4. It can be seen that the method was able to correctly recognize all five objects in these scenes.

Before performing the matching, salient local features are selected from the 3D



(a) Chef (b) Chicken (c) T-rex

(d) Parasaurolophus (e) Rhino

Figure 5.3: Five different 3D models used in the recognition.

models and scenes at 5 different scales corresponding to $r = 1$ to $r = 5$. The feature extraction step for all the models can be done off-line, thus it does not affect the actual matching time. Figure 5.5 shows the recognition results for the 'Chef' model in three real scenes with different amounts of occlusion and clutter. It can be seen that the method can accurately find the pair of matched triangles between the model and scene surfaces and thus, estimate the locations and orientations of the 'Chef' model in the scenes. Even with a highly occluded scene shown in Figure 5.5h, the recognition process was still successful when only about 15% of the model surface present in the scene. Further experimental results for other models including the T-rex, Rhino, Parasaurolophus and Chicken are shown in Figure 5.6.

To further gauge the performance of the matching approach, it is possible to break the success or failure of the recognition results into four states: *true positive*, *false positive*, *false negative* and *true negative* [36]. A recognition result is classified as true positive if the model exists and the algorithm can find it in the scene. The false positive is used if the algorithm decides that the model exists when it does not or the algorithm locates the model in an entirely wrong position in the scene. If the model exists in the scene and the algorithm concludes that it does not, this is termed as false negative. As the searched model was always in the scene, the true negative state was not used in the experiments. It is worth noting that if the threshold value E_t is too small, e.g. $E_t \approx 0$, false negatives can be eliminated. However, this will increase the false positive rate significantly.

Each of the five 3D models was recognized using the proposed algorithm on all 50 real scenes resulting in 250 recognition trials. Figure 5.7 shows the recognition rate of the approach versus occlusion and clutter. It can be seen from Figure 5.7a that when the amount of occlusion is smaller than 80%, the recognition rate (true positives) of the algorithm is always greater than 95%. This is an indication that the technique will almost always work if there is sufficient surface area of the model in the scene. However, the recognition rate drops significantly and the number of false negatives increases rapidly when the occlusion is higher than 85%. This is expected since, with high occlusion, the value of the overlapping metric E estimated from the verification step is very small as a result of little overlapping surface area between the model and the scene. The average recognition rate of the approach is 97.7% for the amount of occlusion up to 85% compared to the reported results of 87.8% of the spin-images [36] and 96.6% of the tensor-based approach [57] on these 50 real scenes. The recognition rate of the algorithm versus clutter is shown in Figure 5.7b. It is clear from the figure that the approach performs well in cluttered scenes. The effect of clutter on the recognition results only becomes significant when there is more than 90% clutter in the scene. However, it can be seen that even at 95% clutter, the number of true positives is still 60% of the total number of recognition trials. As the level of clutter reaches 90% and beyond, the false positive rate increases quickly as the method starts making more incorrect matches. This is an expected behaviour for any matching algorithm using local surface geometric features in highly confusing scenes.

The average recognition time per scene for each model using the proposed method is shown in Table 5.1. It can be seen from the table that the proposed approach takes on average about 3 minutes to recognize a model from a real scene. It is worth knowing that any preprocessing or simplification of the input range images before the recognition is not required. In the case of the spin-image algorithm [36] or the approach using 3D tensors [57], the input meshes have to be simplified or smoothed before performing the matching process. It took the spin-images an average recognition time of about 480 minutes per scene. The recognition time for the 3D tensor approach was reported by the authors at approximately 6 minutes per scene [57] or about 2 minutes per single object per scene. It can be seen that the algorithm is significantly faster than the spin-images and only slightly slower than the tensor-based method despite that it operates on full resolution meshes and the other techniques work on simplified versions.

Table 5.1: Recognition time for five 3D models.

Model	Number of vertices	Average recognition time
Chef	176920	175s
T-rex	176508	203s
Rhino	79902	151s
Parasaurolophus	184879	213s
Chicken	135142	148s

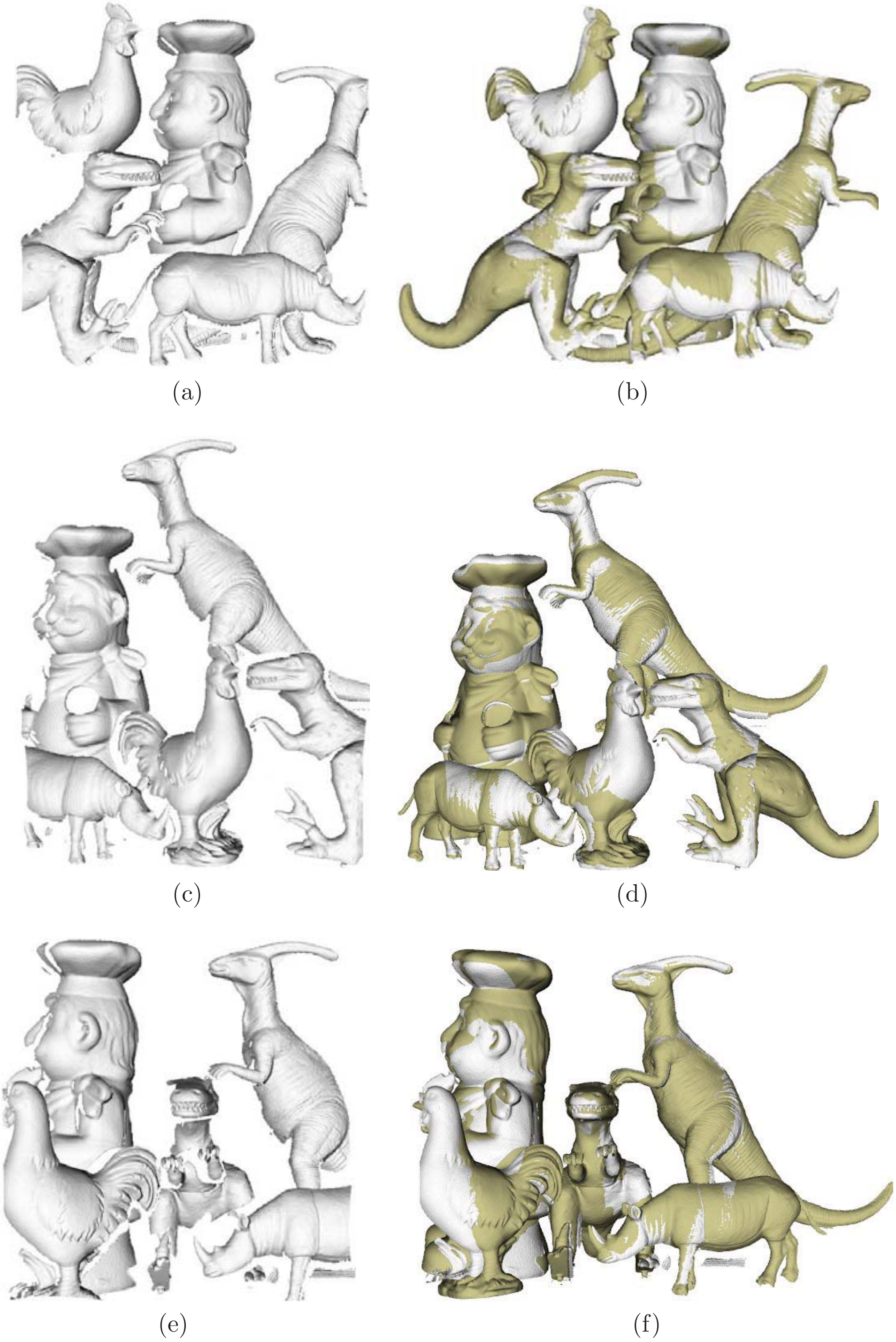
5.3 Robustness to Noise and Variations in Surface Sampling

In order to evaluate the robustness of the recognition framework to noise, Gaussian noise with standard deviation σ ranging from 0.05 to 0.5 was added to the 3D scenes. The recognition results for the Chef and Parasaurolophus models in a noisy scene with $\sigma = 0.5$ are shown in Figure 5.8. The first two columns of the figure show the pairs of the matched triangles used in locating the models in the scene. Even in this highly contaminated scene, the algorithm can still find the optimal correspondences and thus, accurately identify the models and determine their poses. The robustness of the method with respect to noise is a result of matching features extracted from multiple scales. It can be seen that features extracted at higher scales are not so affected by noise as using a large neighbourhood for collecting fitting points is similar to applying a smoothing filter to the noisy surface.

Figure 5.9 shows the recognition results for the Chef and Parasaurolophus models in the same 3D scene but with lower resolution. The original scene containing 114k vertices was down-sampled to about 52k vertices using the mesh simplification algorithm in [48]. It can be seen from the results that the framework was able to correctly locate both objects in despite of the large variations in the surface sampling between the models and the scene.

5.4 Summary

In this chapter, an extension of the surface matching framework using multi-scale features to the problem of 3D object recognition has been presented. Experiments were performed on 5 models and 50 real scenes. Recognition results show that the method is able to accurately locate a 3D model from highly occluded scenes with the presence of other objects. The proposed method also outperforms the spin-images [36] and tensor-based approach [57] in terms of recognition rate. It is comparable to the tensor-based approach regarding the recognition time. It has also been demonstrated that the framework is robust to both noise and variations in surface resolution.

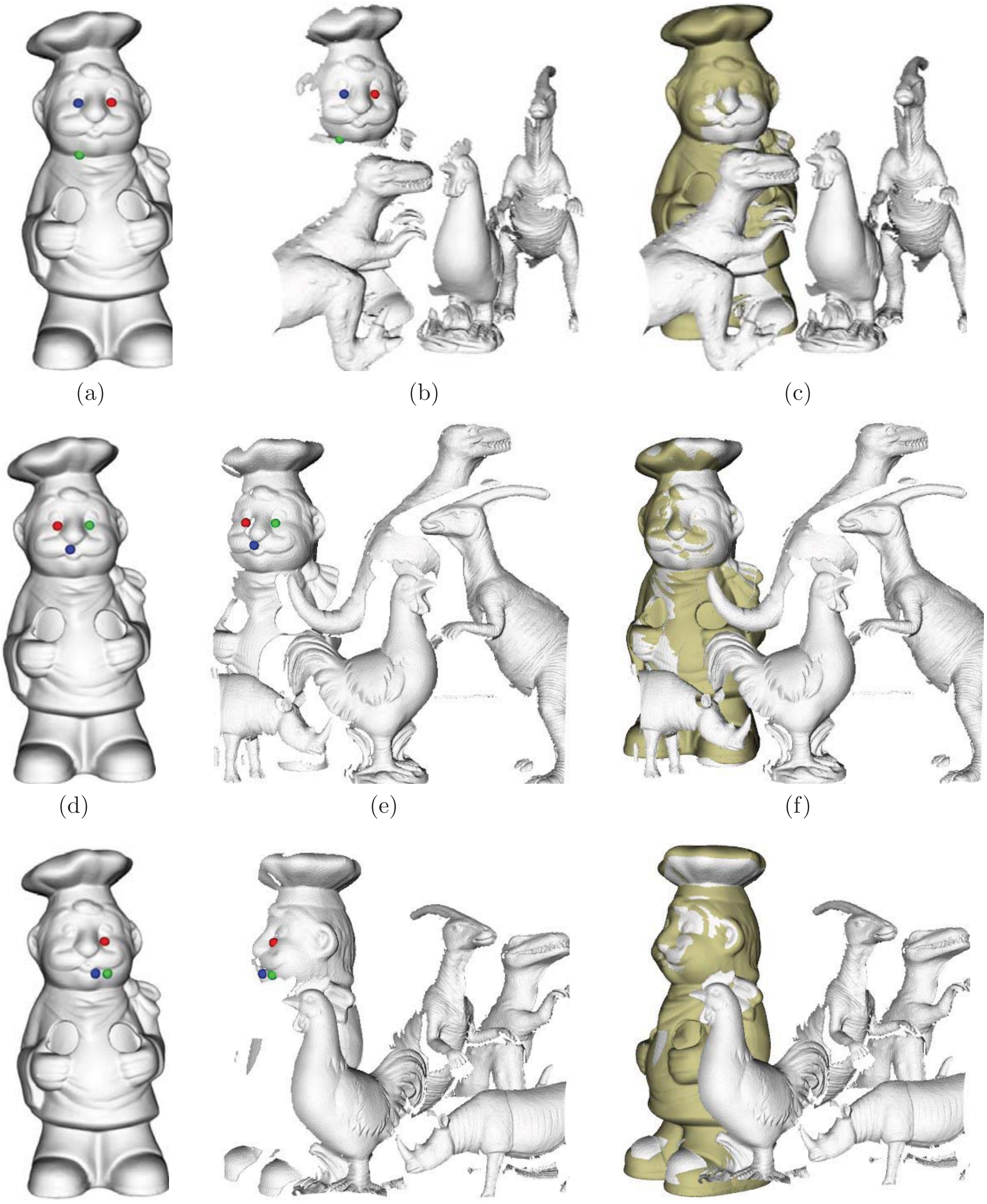


(a) (b)

(c) (d)

(e) (f)

Figure 5.4: Some real scenes used in the recognition and the simultaneous recognition results for five models. All models are correctly recognized and their positions are shown by superimposing model vertices on the scenes.

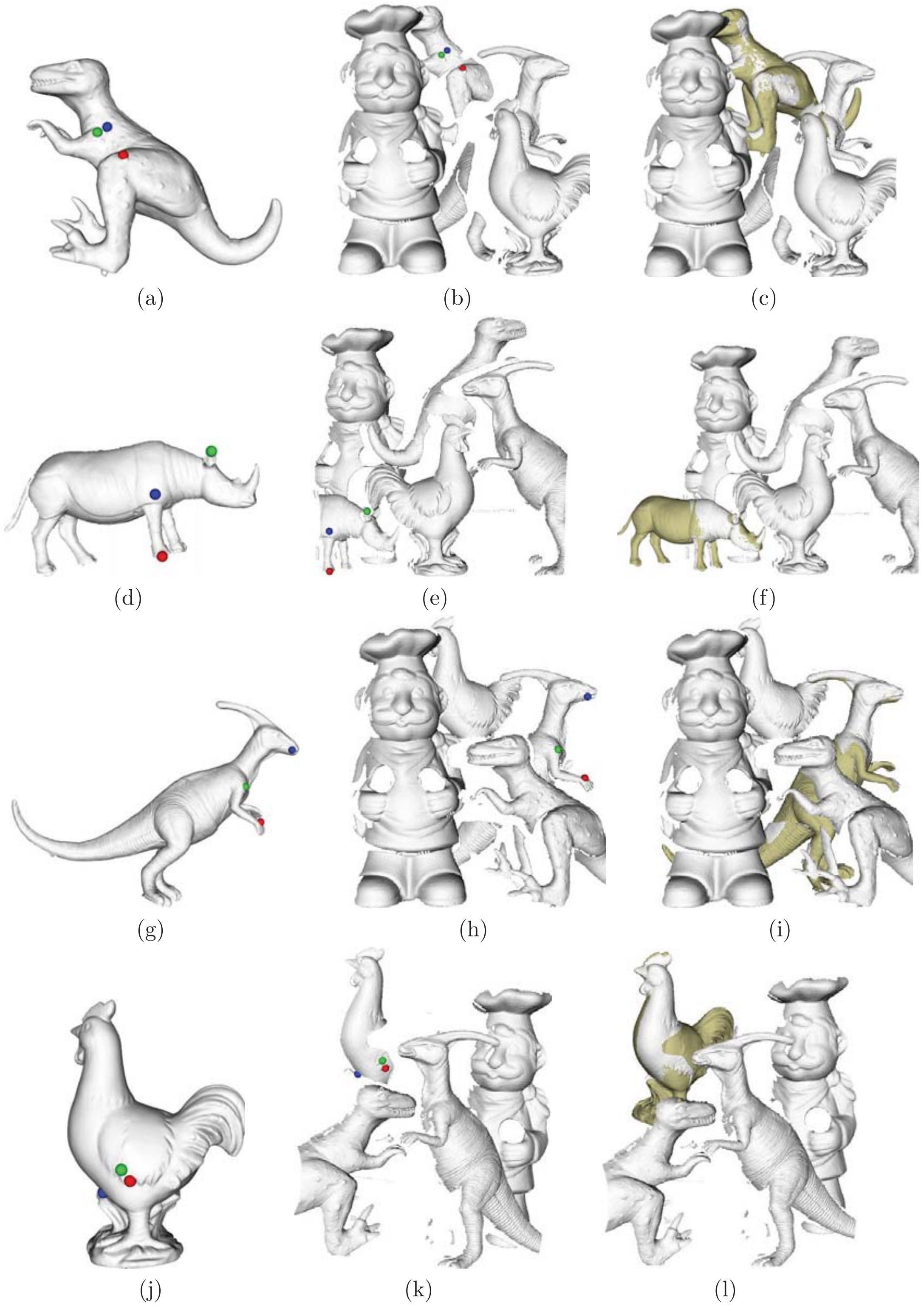


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.5: Recognition results for the Chef model. (a), (d), (g) The matched triangle found on the model surface. (b), (e), (h) The matched triangle found on the scene surface. (c), (f), (i) Location of the Chef is shown as model vertices superimposed on the scene.



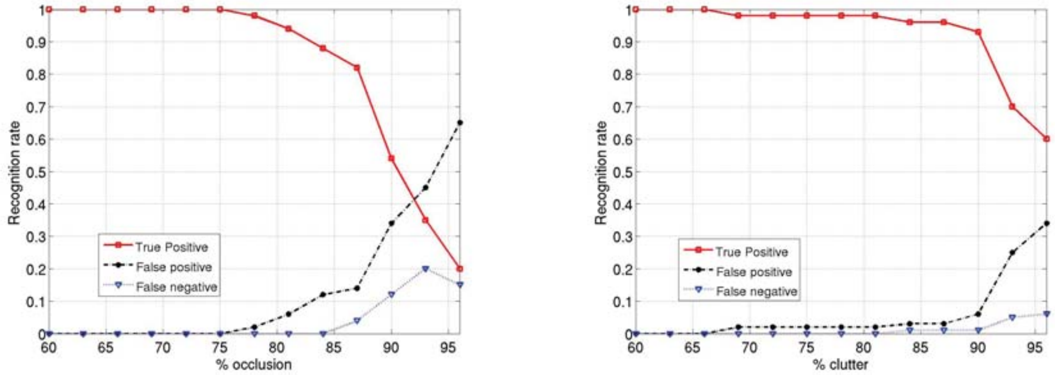
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

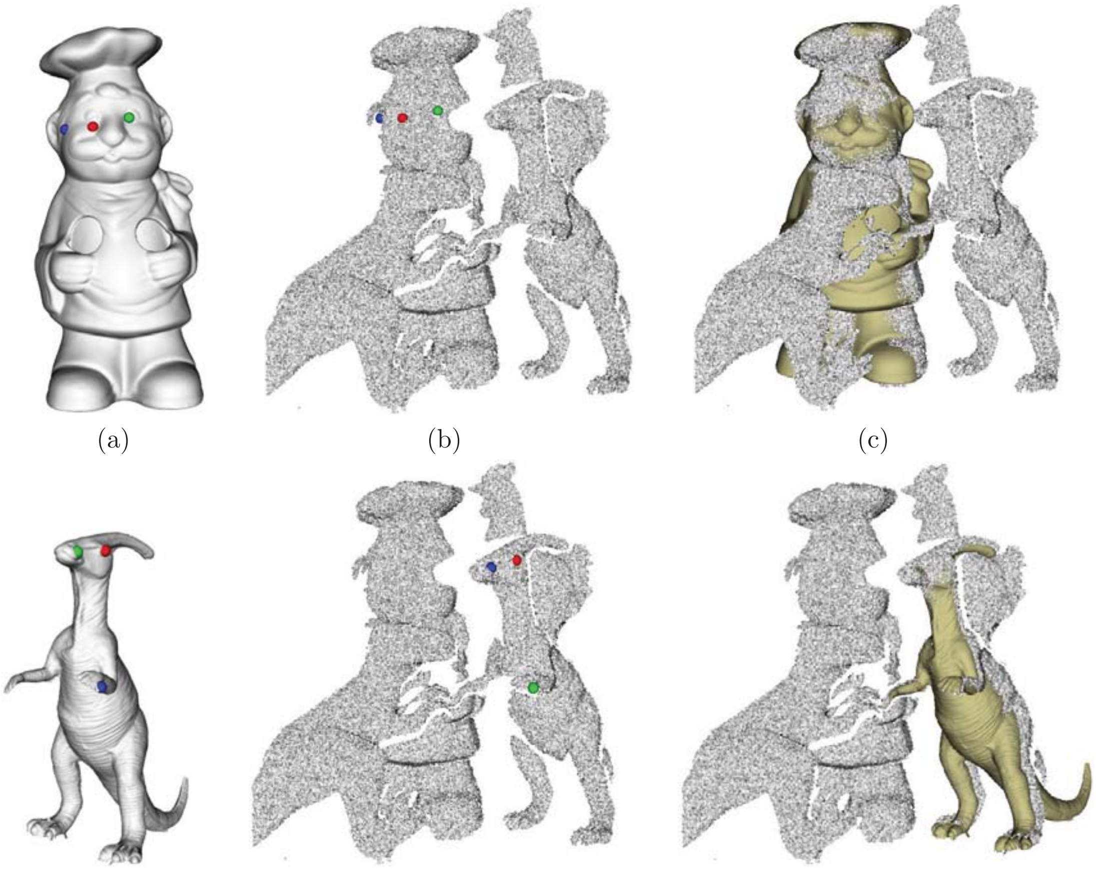
(j) (k) (l)

Figure 5.6: Additional recognition results for other models.



(a) (b)

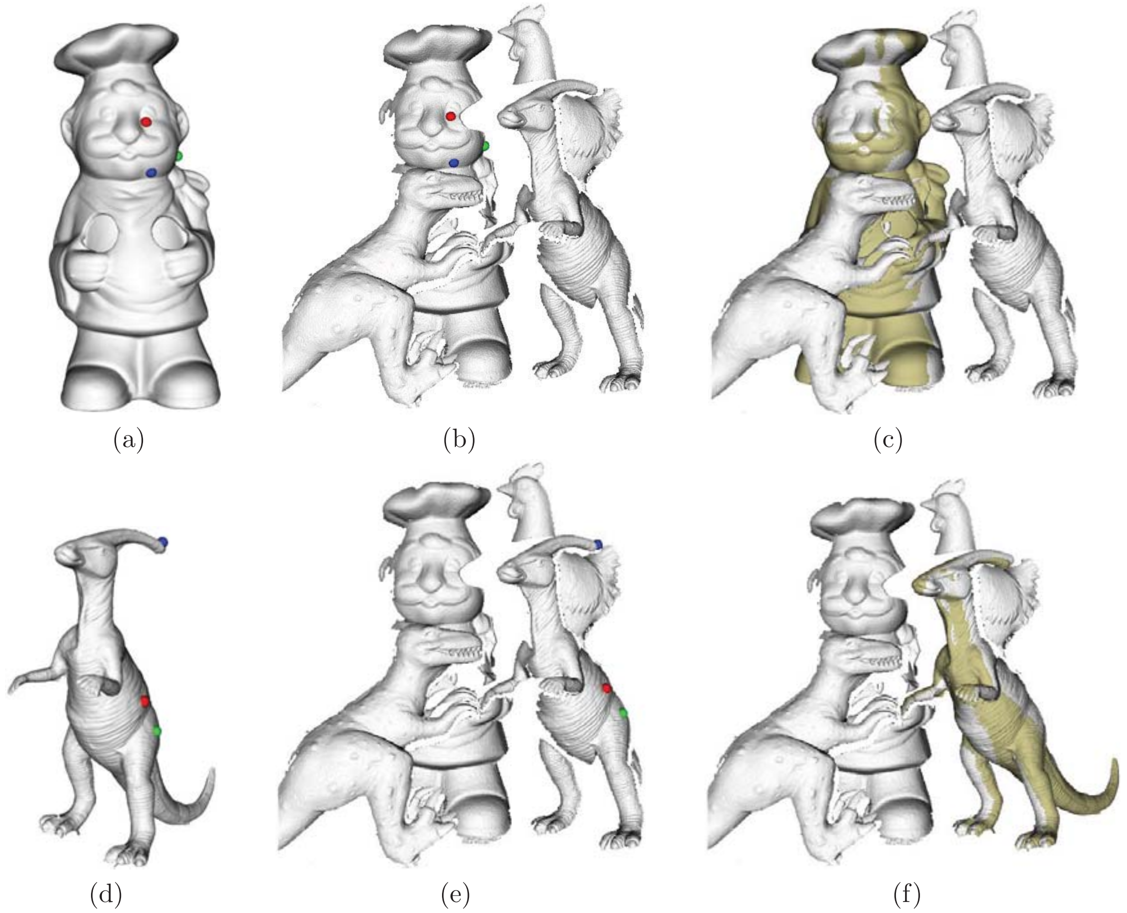
Figure 5.7: (a) Recognition rate of the proposed algorithm versus occlusion. (b) Recognition rate of the proposed algorithm versus clutter



(a) (b) (c)

(d) (e) (f)

Figure 5.8: Recognition results for the Chef and Parasaurolophus models with Gaussian noise in the scene ($\sigma = 0.5$). The algorithm could accurately find the pairs of matched triangles and thus, correctly locate the their poses in this very noisy scene.



(a) (b) (c)

(d) (e) (f)

Figure 5.9: Recognition results for the Chef and Parasaurolophus models with different mesh resolution in the scene. The original mesh of the scene containing 114k vertices was down-sampled to 52k vertices. The poses of the Chef and Parasaurolophus were accurately estimated in despite of the large variations in the surface sampling between the models and the scene.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

In this thesis, two different schemes for extracting salient local features from 3D datasets have been proposed. By fitting a truncated Taylor expansion called the jet to the local surface patch at different sizes, the two principal curvatures are estimated at multiple scales. These principal curvature values are then used to calculate the curvedness or the shape index of the local patches in order to construct a scale-space representation of the input surface. Feature points are selected at surface positions corresponding to the local extrema of the scale-space representation. Furthermore, the distintiveness of each feature is evaluated by using a value that measures how far the local surface curvature at the point is from the mean value of its neighbourhood. The proposed framework is applicable to 3D meshes as well as unstructured point clouds and thus, creates a generic approach to both types of datasets. It is also demonstrated to work well in noisy conditions given the high repeatability of the features between the original and noisy range images. In order to demonstrate the benefits of the technique, it was applied as a preprocessing step to a surface registration algorithm using spin-images as local descriptors. The proposed feature extraction method helped to improve the registration results in both terms of accuracy and efficiency.

This thesis also presents an automatic framework for registering two range scans that differ by a rigid body transformation. The transformation between two views of the same object can be estimated by finding a pair of matched triangles created from the triangulations of the features on the scene and model views. In order to increase the accuracy and efficiency of the algorithm, various surface geometric and rigidity constraints are applied to prune unlikely correspondences. The method can provide good alignment results for noisy surfaces and those with different resolutions. When compared on the same datasets, the proposed technique was shown to be faster and more accurate than the spin-image algorithm.

As a natural extension to the 3D surface registration algorithm, the framework was extended to perform the recognition of 3D models on scenes with occlusion and clutter. The proposed 3D recognition technique was used to recognize 5 objects on 50 real scenes. Experimental results shows that the method could achieve an average recognition rate of 97.7% compared to 87.8% of the spin-images [36] and 96.6% of the tensor-based approach [57]. The method was also able to recognize 3D objects from scenes with noise and variations in surface resolutions.

6.2 Future Research

There are several directions that can be further investigated to improve this work. When extracting salient features for 3D point clouds, the Euclidean distance was used to search for fitting points in the local neighbourhood at different scales. If portions of the surface are close in 3D space but their geodesic distance is large, the feature extraction method may include these portions simultaneously. As a result, erroneous features may be created if the Euclidean distance is used as the distance measure [62]. It would be of further research interest to investigate the use of the geodesic distance in creating the scale-space representation of unstructured point clouds in order to improve the reliability of extracted features.

In this thesis, only the problem of pairwise registration of 3D surfaces was investigated. An approach similar to the graph method proposed by Huber [31, 32] can be used to perform the automatic registration of unordered views. In this method, each view is considered as a vertex in the graph and the weighted edges are represented by the pairwise transformations and areas of overlap between the views. Edges with areas of overlap less than a threshold are pruned. A global optimization is applied to search for the maximum spanning tree of the graph in order to compute the final poses of all views.

In the proposed approach, the transformation between two surfaces was limited to rigid body transformation, i.e. translation and rotation. However, as the information about the scale is already contained in each feature, the method can be further developed to register range scans with inconsistent global scales. The geometric constraints used in finding correspondences need to be adjusted to incorporate the scale difference between range scans.

Another one of future work would be to develop a new type of surface descriptor that uses not only the curvature information but also the spatial distribution of vertices on a local surface patch to improve the recognition results. Unlike existing descriptors like spin-images, 3D shape-context or 3D tensors only encoding the spatial information of surface vertices, it is believed that incorporating the curvature information will increase the discrimination as well as the reliability of the descriptors.

Bibliography

[1] A. Barr. Superquadrics and Angle-Preserving Transformations. *IEEE Trans. Computer Graphics and Applications*, 1(3):11–23, 1981.

[2] P. Besl and R. Jain. Three-Dimensional Object Recognition. *ACM Computing Survey*, 17:75–145, 1985.

[3] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[4] R. Campbell and P. Flynn. Eigenshapes for 3D Object Recognition in Range Data. In *IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 2, June 1999.

[5] R. Campbell and P. Flynn. A Survey of Free-Form Object Representation and Recognition Techniques. *Computer Vision and Image Understanding*, 81(2):166–210, Feb. 2001.

[6] O. Carmichael, D. Huber, and M. Hebert. 3D Cueing: A Data Filter For Object Recognition. In *Proc. IEEE Int'l Conf. Robotics and Automation*, volume 2, pages 944–950, 1999.

[7] O. Carmichael, D. Huber, and M. Hebert. Large Datasets and Confusing Scenes in 3D Matching and Recognition. In *Proc. IEEE Int'l Conf. 3D Digital Imaging and Modeling*, pages 358–367, 1999.

[8] F. Cazals and M. Pouget. Estimating Differential Quantities using Polynomial Fitting of Osculating Jets. *Computer Aided Geometric Design*, 22(2), 2005.

[9] C. Chen, Y. Hung, and J. Cheng. RANSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(11):1229–1234, 1991.

[10] H. Chen and B. Bhanu. 3D Free-Form Object Recognition in Range Images using Local Surface Patches. In *Proc. IEEE Int'l Conf. Pattern Recognition*, volume 3, pages 136–139, 2004.

[11] J. Cheng and H. Don. A Graph Matching Approach to 3-D Point Correspondences. *Int'l J. Pattern Recognition and Artificial Intelligence*, 5(3):399–412, 1991.

[12] C. Chua and R. Jarvis. Point Signatures: A New Representation for 3D Object Recognition. *International Journal of Computer Vision*, 17:63–85, 1996.

[13] C. Chua and R. Jarvis. 3D Free-form Surface Registration and Object Recognition. *International Journal of Computer Vision*, 25(1):77–99, 1999.

[14] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal. Registration of 3D Multimodality Medical Images Using Surfaces and Point Landmarks. *Pattern Recognition Letters*, 15(5):461–467, May 1994.

[15] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computation Geometry: Algorithms and Applications.* Springer, 3 edition, 2008.

[16] H. Dellingette, M. Hebert, and K. Ikeuchi. A Spherical Representation for the Recognition of Curved Objects. *Image and Vision Computing*, 10(3):132–144, 1992.

[17] M. Desbrun, M. Meyer, P. Schöder, and P. Barr. Implicit Fairing of Arbitrary Meshes using Diffusion and Curvature Flow. In *Proc. SIGGRAPH*, 1999.

[18] H. Dinh and S. Kropac. Multi-Resolution Spin-Images. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 863–870, June 2006.

[19] C. Dorai and A. Jain. COSMOS: A Representation Scheme for 3D Free-Form Objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, Oct. 1997.

[20] A. Flint, A. Dick, and A. van den Hengel. Thrift: Local 3D Structure Recognition. In *Proc. DICTA 2007*, pages 182–188, Dec. 2007.

[21] A. Flint, A. Dick, and A. van den Hengel. Local 3D Structure Recognition in Range Images. *IET Computer Vision*, 2(4):208–217, Dec. 2008.

[22] G. Foresti. Object Recognition and Tracking for Remote Video Surveillance. *IEEE Trans. Cir. Sys. Video Tech.*, 9(7):1045–1062, Oct. 1999.

[23] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing Objects in Range Data using Regional Point Descriptors. In *Proc. European Conf. Computer Vision*, May 2004.

[24] A. Gray. *Modern Differential Geometry of Curves and Surfaces.* CRC Press, 1993.

[25] J. Guerrero-Bermdez, J. Meneses, and O. Gauldrn. Object Recognition using Three-Dimensional Correlation of Range Images. *Optical Engineering*, 39(10):2828–2831, Oct. 2000.

[26] M. Hebert, J. Ponce, T. Boult, and A. Gross, editors. *Object Representation in Computer Vision.* Springer-Verlag, Berlin, 1995.

[27] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3D Object Recognition from Range Images using Local Feature Histograms. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 2, pages 394–399, 2001.

[28] Y. Hirano, C. Garcia, R. Sukthankar, and A. Hoogs. Industry and Object Recognition: Applications, Applied Research and Challenges. *Toward Category-Level Object Recognition*, pages 49–64, 2006.

[29] R. Hoffman and A. Jain. Segmentation and Classification of Range Images. *IEEE Trans Pattern Analysis and Machine Intelligence*, 9(5):608–620, 1987.

[30] C. Huang, O. Camps, and T. Kanungo. Object Recognition using Appearance-Based Parts and Relations. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 877–883, June 1997.

[31] D. Huber. Automatic 3-D Modeling using Range Images Obtained from Unknown Viewpoints. In *Proc. 3DIM*, pages 153–160, May 2001.

[32] D. Huber and M. Hebert. Fully Automatic Registration of Multiple 3D Data Sets. *Image and Vision Computing*, 21(7):637–650, 2003.

[33] A. Jagannathan and E. Miller. Three-dimensional Surface Mesh Segmentation using Curvedness-based Region Growing Approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(12):2195–2204, 2007.

[34] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching.* PhD thesis, Carnegie Mellon University, Aug. 1997.

[35] A. Johnson and M. Hebert. Surface Registration by Matching Oriented Points. In *Proc. 3DIM*, pages 121–128, May 1997.

[36] A. Johnson and M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(5):674–686, May 1999.

[37] T. Joshi, J. Ponce, B. Vijayakumar, and D. Kriegman. HOT Curves for Modeling and Recognition of Smooth Curved 3D Objects. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 876–880, June 1994.

[38] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:103–108, 1990.

[39] J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50:363–370, 1984.

[40] J. Koenderink and A. van Doorn. Surface Shape and Curvature Scales. *Image and Vision Computing*, 10:557–565, 1992.

[41] V. Koivunen and R. Bajcsy. Spline Representations in 3-D Vision. In M. Hebert, J. Ponce, T. Boult, and A. Gross, editors, *Object Representation in Computer Vision*, Berlin, 1995. Springer-Verlag.

[42] A. Kushal, C. Schmid, and J. Ponce. Flexible Object Models for Category-Level 3D Object Recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2007. DOI: 10.1109/CVPR.2007.383149.

[43] R. Lakemond, C. Fookes, and S. Sridharan. Affine Adaptation of Local Image Features Using the Hessian Matrix. In *Proc. IEEE Int'l Conf. Advanced Video and Signal Based Surveillance*, pages 496–501, 2009.

[44] Y. Lamdan and H. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 238–249, 1988.

[45] X. Li and I. Guskov. Multi-scale Features for Approximate Alignment of Point-based Surfaces. In *Proc. Eurographics Symp. on Geometry Proc.*, 2005.

[46] X. Li and I. Guskov. 3D Object Recognition from Range Images using Pyramid Matching. In *Proc. IEEE Int'l Conf. Computer Vision*, 2007. DOI: 10.1109/ICCV.2007.4408829.

[47] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer, Dordrecht, 1994.

[48] P. Lindstrom and G. Turk. Evaluation of Memoryless Simplification. *IEEE Trans. Visual. and Comp. Graph.*, 5:98–115, 1992.

[49] D. Lowe. Distinctive Image Features From Scale-Invariant Keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004.

[50] A. Mahalanobis. Automatic Target Recognition using Correlation Filters. In *Proc. SPIE*, volume 2484, pages 546–557, 1995.

[51] A. Mahalanobis, B. VijayaKumar, S. Song, S. Sims, and J. Epperson. Unconstrained Correlation Filters. *Applied Optics*, 33:3751–3759, 1994.

[52] B. Matei, Y. Shan, H. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert. Rapid Object Indexing using Locality Sensitive Hashing and Joint 3D-Signature Space Estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(7):674–686, July 2006.

[53] A. Mian. 3D Models. http://people.csse.uwa.edu.au/ajmal/3Dmodeling.html.

[54] A. Mian. *Representations and Matching Techniques for 3D Free-form Object and Face Recognition.* PhD thesis, The University of Western Australia, Mar. 2006.

[55] A. Mian, M. Bennamoun, and R. Owens. Automatic Correspondence for 3D Modeling: An Extensive Review. *International Journal of Shape Modelling*, 24(2):253–291, 2005.

[56] A. Mian, M. Bennamoun, and R. Owens. A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images. *International Journal of Computer Vision*, 66(1):19–40, 2006.

[57] A. Mian, M. Bennamoun, and R. Owens. Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1584–1601, Oct. 2006.

[58] F. Mokhatarian, N. Khalili, and P. Yuen. Multi-scale Free-form 3D Object Recognition using 3D Models. *Image and Vision Computing*, 19:271–281, 2001.

[59] F. Mokhtarian. Silhouette-Based Isolated Object Recognition through Curvature Scale Space. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17:539–544, 1995.

[60] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[61] R. Nevatia and T. Binford. Description and Recognition of Curved Objects. *Artificial Intelligence*, 8:69–76, 1977.

[62] J. Novatnack and K. Nishino. Scale-Dependent 3D Geometric Features. In *Proc. IEEE Int'l Conf. Computer Vision*, Oct. 2007.

[63] J. Novatnack and K. Nishino. Scale-Dependent/Invariant Local 3D Shape Descriptors for Fully Automatic Registration of Multiple Sets of Range Images. In *Proc. European Conference on Computer Vision*, pages 440–453, Oct. 2008.

[64] M. Pauly, R. Keiser, and M. Gross. Multi-scale Feature Extraction on Point-sampled Surface. In *Proc. Eurographics 2003*, 2003. DOI: 10.1111/1467-8659.00675.

[65] S. Petitijean. A Survey of Methods for Recovering Quadrics in Triangle Meshes. *ACM Computing Surveys*, 34(2):1584–1601, 2001.

[66] J. Ponce and D. Kriegman. On Recognizing and Positioning Curved 3D Objects from Image Contours. In *Proc. IEEE Workshop Interpretation 3D Scenes*, pages 61–67, 1989.

[67] J. Ponce, S. Lazebnik, F. Rothganger, and C. Schmid. Toward True 3D Object Recognition. In *Congres de Reconnaissance des Formes et Intelligence Artificielle*, Jan. 2004.

[68] N. Raja and A. Jain. Recognizing Geons from Superquadrics Fitted to Range Data. *Image and Vision Computing*, 10:179–190, 1992.

[69] J. Rosen. Three-Dimensional Joint Transform Correlator. *Applied Optics*, 37(32):7538–7544, 1998.

[70] G. Roth. Registering Two Overlapping Range Images. In *Proc. 3DIM*, pages 191–200, 1999.

[71] S. Ruiz-Correa, L. Shapiro, and M. Melia. A New Signature-Based Method for Efficient 3-D Object Recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 1, pages 769–776, 2001.

[72] S. Rusinkiewicz and M. Levoy. Efficient Variants of the ICP Algorithm. In *Proc. 3DIM*, pages 145–152, 2001.

[73] L. Shapiro and J. Brady. Feature-based Correspondence: an Eigenvector Approach. *Image and Vision Computing*, 10:283–288, 1992.

[74] M. Spivak. *A Comprehensive Introduction to Differential Geometry.* Publish or Perish, 1999.

[75] Standford University. Standford 3D Scanning Repository. http://www-graphics.stanford.edu/data/3Dscanrep/.

[76] F. Stein and G. Medioni. Structural Indexing: Efficient 3D Object Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:125–145, 1992.

[77] R. Sumner and J. Popovic. Deformation Transfer for Triangle Meshes. In *Proc. SIGGRAPH*, 2004.

[78] G. Taubin. A Signal Processing Approach to Fair Surface Design. In *Proc. SIGGRAPH*, 1995.

[79] The University of Stuttgart. Stuttgart Range Image Database. http://range.informatik.uni-stuttgart.de/htdocs/html.

[80] J. Thirion. The Extremal Mesh and the Understanding of 3D Surfaces. *International Journal of Computer Vision*, 19:115–128, 1996.

[81] M. Turk and A. Pentland. Face Recognition using Eigenfaces. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[82] R. Unnikrishnan and M. Hebert. Multi-scale Interest Regions from Unorganized Point Clouds. In *Workshop on Search in 3D*, June 2008.

[83] Vision and Mobile Robotics Laboratory. Mesh Toolbox. http://www.cs.cmu.edu/˜vmr/software/meshtoolbox.

[84] A. Witkin. Scale-Space Filtering. In *Proc. IJCAI*, pages 1019–1022, 1983.

[85] S. Yamany and A. Farag. Surface Signatures: An Orientation Independent Free-form Surface Representation Scheme for the Purpose of Objects Registration and Matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(8):1105–1120, 2002.

[86] J. Yi and D. Chelberg. Model-Based 3D Object Recognition using Bayesian Indexing. *Computer Vision and Image Understanding*, 69(1):87–105, 1998.

[87] D. Zhang. *Harmonic Shape Images: A 3D Free-form Surface Representation and Its Applications in Surface Matching.* PhD thesis, Carnegie Mellon University, Nov. 1999.