



HAL
open science

MULTISENSOR INTEGRATION AND DISCRETE GEOMETRY PROCESSING FOR COORDINATE METROLOGY

Haibin Zhao

► **To cite this version:**

Haibin Zhao. MULTISENSOR INTEGRATION AND DISCRETE GEOMETRY PROCESSING FOR COORDINATE METROLOGY. Mechanics [physics.med-ph]. École normale supérieure de Cachan - ENS Cachan, 2010. English. NNT: . tel-00451244

HAL Id: tel-00451244

<https://theses.hal.science/tel-00451244>

Submitted on 28 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° ENSC-2010/???

**THESE DE DOCTORAT
DE L'ECOLE NORMALE SUPERIEURE DE CACHAN**

Présentée par

Monsieur ZHAO Haibin

pour obtenir le grade de

DOCTEUR DE L'ECOLE NORMALE SUPERIEURE DE CACHAN

Domaine:

MECANIQUE - GENIE MECANIQUE - GENIE CIVIL

Sujet de la thèse:

**MULTISENSOR INTEGRATION AND DISCRETE GEOMETRY
PROCESSING FOR COORDINATE METROLOGY**

Thèse présentée et soutenue à Cachan le 18 Janvier devant le jury composé de:

M. J.-F. Fontaine	Professeur - Université de Bourgogne	Rapporteur
M. J.-C. Léon	Professeur - Grenoble INP	Rapporteur
M. P. Bourdet	Professeur émérite -ENS Cachan	Directeur de thèse
M. N. Anwer	Maître de conférences- Université Paris Nord	Examineur
M. J. M. Linarès	Professeur- Université de la Méditerranée	Examineur
M. B. Hautbergue	Directeur R&D – SPRING Technologies	Invité



Laboratoire Universitaire de Recherche en Production Automatisée
(ENS CACHAN / EA 1385)

61, avenue du Président Wilson - 94 235 Cachan cedex

Acknowledgements:

The works proposed in this thesis are the fruits of the research activities in LURPA (Laboratoire Universitaire de Recherche en Production Automatisée) of ENS Cachan (Ecole Normale Supérieure de Cachan).

First of all, I want to give my great acknowledgements to my thesis director, Professor Pierre Bourdet. His profound knowledge and experiences guide my works under the correct direction. And his respectable personality also makes me feel pleasure during the three years when I stay in LURPA.

I address my unstinting appreciations to Nabil Anwer for his high quality supervising. Without him, I can't reach the fruitful results of my research activities. He searched the opportunities for me to come in France to continue my Ph.D studies. During the whole three years, he gives me his great helps both in my academic research and in my living conditions. He also tried to find work opportunities for me after my graduation. For all of what he has already done for me, Thanks.

A great thankfulness is given to Prof. Jean-Jacque Lessage, who provides the opportunity and great conveniences for me to stay and study in the laboratory. The same thankfulness is also given to Prof. Luc Mathieu.

Thanks to Prof. Bernard Anselmetti and Prof. Claire Lartigue, who provide the opportunity and conveniences for me to stay and study in the team of Geo3D.

I want to thank Prof. Jean-Claude Paul and Prof. Wang Junying for their agreeable receptions during the ten months when I stayed in Tsinghua University, China. The same thankfulness is also given to Mr. Li and Mrs. Yang who work in CAD Teaching Center in Tsinghua University.

Thanks to the office of SRI (Service des Relations Internationales) of the ENS Cachan who provides the financial support to my first year staying in France. Thanks to the Sino-French Laboratory PLMIC (PLM Innovation Center) who finances me the second year of my research works in China. And thanks to LURPA, who supports me the finances needed for my third year in France.

I give my acknowledgements to the members of the jury for their contributions, their

elegant constructive suggestions to the thesis and their presences on the oral defense.

Thanks to all the members in LURPA. They give me their sincere assistances which make feel comfortable during the periods I spend in LURPA. The periods I stayed in France are amazing and very important for me. Great thanks to Nicolas Audfray who assists me to do the experiments. Great thanks also to Laurant Tappie, Charyar Mehdi-Souzani, Renaud Costadoat, et al. who provide the testing models and contributions to the thesis. My great thankfulness is also given to Labib Daoud, who provides his kind helps to solve my academic and living difficulties during my staying in France.

Thanks to Dr Yang Jianxin who is now working in LURPA. As one of my senior fellow apprentice in Tsinghua University, He gives me his sincere helps both in research and in living. Thanks. I also want to thank my Chinese friends living in ENS Cachan and living in Paris, who provide their helps for me. It is pleasure to share my time with them.

Finally, I prefer to give my heartfelt appreciations to ZHANG Min, and also to my parents, my sisters and my other kindred. Thanks for their selfless supports and encouragements in my life.

Contents

Introduction.....	1
Chapter 1: Multisensor Integration in Coordinate Metrology	7
1.1. Introduction	8
1.2. Sensing techniques for coordinate metrology	9
1.2.1. Contact probing.....	9
1.2.2. Laser range triangulation.....	13
1.2.3. Chromatic confocal imaging	17
1.3. Multisensor integration in coordinate metrology	22
1.3.1. Multisensor configuration	22
1.3.2. Multisensor data fusion	23
1.3.3. Related works in multisensor integration	24
1.4. Development of an integrated multisensor system for coordinate metrology.....	32
1.5. Conclusion	33
Chapter 2: Function and Data Modeling for Multisensor Integration	35
2.1. Introduction	36
2.2. Functional analysis of the system.....	36
2.2.1. Function modeling using IDEF0	36
2.2.2. Function models	38
2.2.3. Conclusion.....	51
2.3. Ontology-based data modeling.....	52
2.3.1. Introduction to ontology modeling.....	52
2.3.2. Data modeling	55
2.3.3. Ontology development	61
2.3.4. Conclusion.....	63
2.4. Conclusion	64
Chapter 3: Discrete Differential Geometry	65
3.1. Introduction	66
3.2. Polyhedral surface approximation.....	67
3.2.1. Approximation method and mesh data generation	67
3.2.2. File formats for mesh data representation and exchange	70
3.3. Differential geometry properties Estimation	72
3.3.1. Differential geometry of smooth surface.....	72
3.3.2. Normal vector estimation	76
3.3.3. Discrete curvature estimation methods	79
3.3.4. Our estimation method	85
3.3.5. Shape index and curvedness.....	86
3.4. Testing results.....	88
3.4.1. Normal estimation results.....	88

3.4.2.	Curvature estimation results and comparison	88
3.4.3.	Time performance	92
3.5.	Conclusion	94

Chapter 4: Registration of Discrete Shapes.....95

4.1.	Introduction	96
4.2.	Literature review	96
4.3.	Method overview	98
4.4.	Coarse registration	100
4.4.1.	Principal pose estimation	100
4.4.2.	Transformation calculation.....	102
4.4.3.	Overlapping alignment.....	104
4.5.	Fine registration	108
4.5.1.	Flow chart of the CFR method.....	108
4.5.2.	Geometric distance definition	110
4.5.3.	Corresponding point pairs searching.....	111
4.5.4.	Corresponding point pairs registration.....	112
4.5.5.	Convergence condition.....	115
4.6.	Testing results.....	116
4.6.1.	Full-full overlapping	116
4.6.2.	Full-partial overlapping.....	117
4.6.3.	Partial-partial overlapping.....	118
4.6.4.	Performance analysis.....	120
4.7.	Conclusion	127

Chapter 5: Discrete shape recognition and segmentation129

5.1.	Introduction	130
5.2.	Literature review	130
5.3.	Method overview	134
5.4.	Local surface type recognition	137
5.4.1.	Surface types based on Gaussian and mean curvatures.....	137
5.4.2.	Surface type definition based on shape index	137
5.5.	Vertex clustering.....	141
5.5.1.	Sharp edges and high curvature regions detection	141
5.5.2.	Vertex clustering and cluster refining.....	143
5.6.	Connected region generation.....	148
5.6.1.	Connected region labeling.....	149
5.6.2.	Region visualization.....	151
5.6.3.	Region merging and refining.....	153
5.7.	Experiments and Results	156
5.7.1.	Testing cases.....	157
5.7.2.	Noise effect	161
5.7.3.	Time performance	163
5.7.4.	Limitations	166

5.8.	Conclusion	169
Chapter 6: A Case Study		171
6.1	Introduction	172
6.2.	Multidata acquisition.....	172
6.2.1.	Multisensor system configuration	172
6.2.2.	Data acquisition.....	179
6.3.	DSP-COMS platform overview	180
6.3.1.	Main interface	181
6.3.2.	Menu specification.....	182
6.4.	Discrete geometry processing	184
6.4.1.	Registration	184
6.3.2.	Segmentation.....	188
6.5.	Conclusion	191
Conclusion		193
References.....		199

Figures

Figure 1-1: Classification of the sensing techniques for coordinate metrology	9
Figure 1-2: Limitations of the touch probing system with considering the tip size	10
Figure 1-3: Coordinate systems in contact probing systems	11
Figure 1-4: Triangulation principles of laser scanning	13
Figure 1-5: Modeling of the laser scanning system on CMM platform	15
Figure 1-6: Working principle of chromatic confocal imaging system	19
Figure 1-7: Modeling of the STIL sensor mounted on CMM platform	20
Figure 1-8: Calibration of the STIL sensor	22
Figure 1-9: Three sensors configurations in multisensor systems	23
Figure 1-10: Interface standards in dimensional metrology systems [Hor05]	30
Figure 1-11: The constructed system layout of the multisensor system	32
Figure 2-1: Graphic format of IDEF0	37
Figure 2-2: Hierarchical decomposition of IDEF0 diagrams	38
Figure 2-3: Top-layer IDEF0 diagram of the system – A-0 diagram	39
Figure 2-4: A0 diagram – the first layer of the hierarchical structure	41
Figure 2-5: A2 diagram for measurement strategies planning	44
Figure 2-6: Two trajectory generation methods	47
Figure 2-7: A4 diagram for discrete geometry processing	48
Figure 2-8: Product model	57
Figure 2-9: Measurement strategies planning model (MSP model)	58
Figure 2-10: Execution model	59
Figure 2-11: Result analysis model	61
Figure 2-12: Top hierarchical structure of ontologies built on Protégé-OWL	62
Figure 2-13: Ontology editing	63
Figure 3-1: Examples of the triangle mesh generation	68
Figure 3-2: Schematic diagram of the half-edge data structure	69
Figure 3-3: STL file format and a typical STL model	70
Figure 3-4: OFF file format and an example of OFF model	72
Figure 3-5: Normal and principal curvature directions of a smooth surface at point p	75
Figure 3-6: The notations mentioned in normal estimation	78
Figure 3-7: Two main types of local regions	81
Figure 3-8: The notations in the method of Cohen-Steiner and Morvan	85
Figure 3-9: The notations in our proposed method	86
Figure 3-10: Three examples of the Normal estimation	88
Figure 3-11: The maximum and minimum principal curvature fields	93
Figure 3-12: Time performance of our curvature estimation method	94
Figure 4-1: Procedures of the proposed registration method	99
Figure 4-2: Principal coordinate systems of two discrete shapes	102
Figure 4-3: Notations for coordinate system transformation calculation	103
Figure 4-4: Full overlapping alignment based on principal poses	105
Figure 4-5: Partial overlapping alignment based on principal poses	105

Figure 4-6: Overlapping alignment results with different	107
Figure 4-7: The low accurate alignment in coarse registration	108
Figure 4-8: General registration flow of the CFR method	109
Figure 4-9: Full-full overlapping registration of the armadillo data	117
Figure 4-10: Full-partial overlapping registration of the Chinese dragon data (Dragon 1)	118
Figure 4-11: Partial-partial overlapping registration of the Chinese dragon data (Dragon 2).....	119
Figure 4-12: Partial-partial overlapping registration of the Stanford bunny data	120
Figure 4-13: Partial-partial overlapping registration of the future Buddha data	120
Figure 4-14: The two data used to test the influence of λ on registration performance.....	124
Figure 4-15: Registration of unitary shapes	125
Figure 4-16: Registration of the data without sufficient correspondences	125
Figure 5-1: Procedures overview of shape recognition and segmentation.....	135
Figure 5-2: The locations of the surface types on shape index scales and their color scales	139
Figure 5-3: Representative shapes of the 10 defined surface types based on shape index.....	139
Figure 5-4: Discrete shapes and their local surface types' visualization.....	140
Figure 5-5: Example of mislabeling of sharp edge and high curvature region	141
Figure 5-6: Curvedness maps of two discrete shapes	142
Figure 5-7: Detection of the points in sharp edges and high curvature regions	143
Figure 5-8: General flow of the proposed clustering algorithm.....	145
Figure 5-9: The defined surface types on the $\kappa_1 - \kappa_2$ plane.....	146
Figure 5-10: Examples of vertex clustering and cluster refining	148
Figure 5-11: Studied cases for the connected region labeling.....	150
Figure 5-12: Four conditions a triangle possibly encountered for visualization.....	152
Figure 5-13: Region label adjusting for visualization	153
Figure 5-14: Connected region merging and refining	156
Figure 5-15: Boundary types in continuous and discrete shapes	158
Figure 5-16: Segmentation of regions surrounded by sharp edges	159
Figure 5-17: Segmentation of regions surrounded by tangent edges	160
Figure 5-18: Segmentation of regions surrounded by curvature edges.....	161
Figure 5-19: Segmentation of complex shapes	162
Figure 5-20: Segmentation of a discrete shape with noise.....	163
Figure 5-21: The tested discrete shapes in table 5-3	164
Figure 5-22: Shape used for testing the time performance	165
Figure 5-23: Time performance of the testing case with different tessellations.....	166
Figure 5-24: Characteristic points identification from complex shapes.....	167
Figure 5-25: Limitations to the measured noisy point data.....	167
Figure 5-26: Segmentation with different sizes of vertices.....	168
Figure 6-1: The multisensor measurement system in LURPA.....	173
Figure 6-2: The component to install the STIL pen on CMM arm	177
Figure 6-3: Two configurations of the sensor physical integration.....	177
Figure 6-4: Automotive water pump cover	179
Figure 6-5: Measurement system configuration and examples of the acquired data	180
Figure 6-6: The main interface of DSP-COMS	182

Figure 6-7: Detailed menu specification of the software	183
Figure 6-8: The scene data and the model data in case 1	184
Figure 6-9: Coarse registration results in case 1	185
Figure 6-10: Fine registration results in case 1	185
Figure 6-11: The scene data and the model data in case 2	186
Figure 6-12: Coarse registration results in case 2	186
Figure 6-13: Fine registration results in case 2	186
Figure 6-14: The scene data and the model data in case 3	187
Figure 6-15: Coarse registration results in case 3	187
Figure 6-16: Fine registration results in case 2	188
Figure 6-17: Final discrete shape of the workpiece	188
Figure 6-18: The maximum principal curvature map of the workpiece.....	189
Figure 6-19: The minimum principal curvature map of the workpiece	189
Figure 6-20: The shape index map of the workpiece	190
Figure 6-21: The curvedness map of the workpiece	190
Figure 6-22: Recognition results of the high curvature points.....	191
Figure 6-23: The final segmentation result	191

Tables

Table 3-1: Surface types specified by shape index intervals and their type labels.....	87
Table 3-2: Estimation of curvature values by the three methods on different surface types.....	90
Table 3-3: Estimation errors of the three methods on different surface types.....	91
Table 3-4: Sorting the methods based on the relative average error	91
Table 3-5: Sorting the methods based on the relative maximum error.....	91
Table 3-6: Sorting the methods based on the relative minimum error	92
Table 3-7: Time performances for the mentioned models.....	93
Table 4-1: Computational costs in coarse registration	121
Table 4-2: Time performances of the tested cases in coarse registration	122
Table 4-3: Time complexities of the two compared methods	123
Table 4-4: Performances of the two compared algorithms.....	123
Table 4-5: The performances of the proposed method with different values of λ	126
Table 5-1: Surface types specified by signs of Gaussian and mean curvatures	137
Table 5-2: The denotations used in connected region labeling	149
Table 5-3: Computing times for the tested cases.....	164
Table 5-4: Computing times for the same shape with different tessellations.....	166
Table 6-1: Specifications of the Kreon Zephyr KZ25 laser scanner	174
Table 6-2: Specifications of the Renishaw TP2 touch trigger probe.....	174
Table 6-3: Specifications of the STIL CL2 optical pen with CHR150 controller	175
Table 6-4: Comparison of the main characteristics of the three sensors	178

Introduction

Acquiring 3D point data from physical objects is increasingly being adopted in a variety of product development processes, such as quality control and inspection [WRS00], reverse engineering [VMC97], etc. In coordinate metrology, many sensing technologies are available for data acquisition, such as tactile probing, laser scanning, confocal imaging, etc. In general, different sensors capture the information with different details. The complex design specifications are increasingly required to embed them in the mechanical parts such as shapes and surfaces description, dimensions, macro and micro geometrical tolerances, material characteristics for either functional or aesthetical reasons. It becomes difficult to satisfy all the measurement requirements with only a single sensor in coordinate metrology. An individual sensor can get neither holistic information of a workpiece accurately, nor spatial and temporal coverage with small measurement uncertainties in such measurement. Multiple sensors are hence employed to achieve both holistic geometrical measurement information and improved reliability or reduced uncertainty of measurement data [WJS09].

A multisensor integration system provides efficient solutions and better performances than the single sensor based system. Significant efforts are being devoted to the development of multisensor integration system in coordinate metrology. Different information sources (sensors) should be integrated in one common system. This makes complex system development to achieve the multisensor integration. Another fundamental question is how to merge the data provided by multiple sensors together and process them to extract meaningful geometric information.

In this dissertation, we investigate the multisensor integration and discrete geometry processing for coordinate metrology from two main parts: function and data modeling, and discrete geometry processing.

In order to achieve the multisensor integration, a comprehensive function and data modeling of the system are necessary to obtain clear and unambiguous understanding of the whole system. The system can be developed with the guidelines of the consistent function models and unambiguous data representation.

In multisensor integration system, the multiple sensors should be configured properly to fulfill the measurement. The specifications of the integrated sensors are usually different. The products measured by multisensor integration system usually

comprise complex specifications. The discrete geometry processing methods are also required for the achievement of multisensor integration.

A multisensor integration system contains various information and data. They should be described and managed unambiguously. The system should be also integrated with a PLM (Product Lifecycle Management) context, which also requires a unified and consistent data representation. Ontology development provides an effective way to represent and manage the data and the knowledge of multisensor integration system in coordinate metrology.

Multisensor integration requires suitable methods and algorithms of discrete geometry processing to process the multiple data acquired from different sensors. The methods in the three important phases of discrete geometry processing are investigated in this dissertation: normal and discrete curvature estimation; registration; shape recognition and segmentation.

Reliable estimation of normal and discrete curvatures is often required in many applications when the surface is defined by a set of discrete points rather than by mathematical formulae [MW00]. It is an essential task for discrete shape modeling and processing. Reliable estimation of discrete curvatures is also the basis of discrete shape registration and segmentation.

Registration is one of the most important and decisive steps of multisensor integration. The point data acquired by multiple views/sensors are usually represented in their own coordinate systems. During the registration process, the measurement data captured in the respective sensor's coordinate system are aligned and transformed to one common coordinate system. The complete shape of the measured workpiece is generated after the registration process.

Recovering and extracting the useful information from discrete shapes play critical roles in discrete shape modeling domain. Generally, the shape recognition and understanding processes are based on the decomposition of the data into smaller parts [She08]. The shape recognition and segmentation are the most critical parts of discrete shape processing [VMC97]. Many applications including classification [JM07], object recognition [DGG04, SP08], reverse engineering [AMF00, BV04, DVV07, WKW02, YL99], and dimensional metrology [LX08, SHM01] need to solve the shape recognition and segmentation problem.

Contributions

The goal of this thesis is to investigate the multisensor integration and geometry processing in the context of coordinate metrology. The main contributions of the thesis are:

- A set of function models to analyze the functional requirements of multisensor integration system in coordinate metrology
- A Structured data modeling method based on ontologies for data representation and sharing.
- A set of methods to estimate the normal and the discrete curvatures of polygon mesh
- A set of methods for registration of multidata represented by respective sensors' coordinate systems.
- A set of methods for discrete shape recognition and segmentation.
- A developed software framework for discrete geometry processing.

Outline

The thesis is organized as follows:

Chapter 1 introduces three sensing techniques commonly used in coordinate metrology. The multisensor configurations and multidata fusion procedures, as well as the related works in multisensor integration for coordinate metrology are discussed. The layout of the developed multisensor measurement system is also presented.

Chapter 2 presents the detailed function model and data model of the developed multisensor integration system in coordinate metrology. The hierarchical decomposition of the function model is specified in details. An ontology based data model is developed to manage the various data in the developed system according to the different measurement stages defined by function model.

Chapter 3 proposes the methods to estimate normal vectors and curvatures of discrete shapes. The normal vector at each vertex is estimated from the normal vectors of the triangle facets in its neighbor region. A tensor based method is used to estimate the principal curvatures of discrete shapes. Two surface descriptors, shape index and

curvedness, are also introduced.

Chapter 4 presents a method to register the discrete shapes with unknown correspondences. The method uses the combination of the curvatures' information and Euclidean distance to help the corresponding point searching. The method can provide accurate registration results with fast convergence. A comparative analysis of the proposed method and the initial ICP (Iterative Closest Point) algorithm is discussed in this chapter.

Chapter 5 presents new methods for discrete shape recognition and segmentation. Finally, several contributions are made in local surface type recognition based on shape index; vertex clustering and connected region partition. The methods provide satisfying results for segmentation of discrete shapes. The time performances of the proposed methods are also analyzed.

Chapter 6 presents a detailed case study of an industrial workpiece -automotive water pump cover. The multisensor integration platform and the measurement strategies of the workpiece are described. The developed software DSP-COMS, which serves as the test platform for the algorithms developed in this thesis. The detailed processing results of the studied workpiece are presented.

In conclusion, we summarize the thesis and propose some promising directions for the future research.

Chapter 1

Multisensor Integration in Coordinate Metrology

1.1. Introduction

Acquiring 3D point data from physical objects is increasingly being adopted in a variety of product development processes, such as quality control and inspection [WRS00], reverse engineering [VMC97], etc. Complex shaped parts and tighter tolerances are increasingly required in modern engineering applications, either for functional or aesthetical reasons. The geometric specifications embedded in these parts such as shapes and surfaces, dimensions, macro and micro geometrical tolerances, material characteristics make it difficult to satisfy all the measurement requirements with only a single sensor in coordinate metrology. An individual sensor can get neither holistic information of a workpiece accurately, nor spatial and temporal coverage with small measurement uncertainties. Multiple sensors are hence employed to achieve both holistic geometrical measurement information and improved reliability or reduced uncertainty of measurement data [WJS09].

A multisensor integration system in coordinate metrology is a measurement system which combines several different sensors so that the measurement result can benefit from all available sensor information and data. With multisensor integration system, particular features of a workpiece can be measured with the most suitable sensor, and the measurement with small uncertainty can be used to correct data from other sensors which exhibit relevant systematic errors but have a wider field of view or application range.

This chapter introduces different sensing techniques commonly used in coordinate metrology and surveys the related work in multisensor integration. The integrated multisensor measurement system developed in our laboratory is also presented.

This chapter is organized as follows:

Section 1.2 introduces the different sensing techniques for surface digitizing in coordinate metrology. Section 1.3 discusses the multisensor configuration and data fusion process in multisensor methods. A comprehensive survey of related work in multisensor integration is presented in section 1.3. Section 1.4 describes the layout of the developed multisensor system for coordinate metrology.

1.2. Sensing techniques for coordinate metrology

Different sensor technologies are available for surface digitizing in coordinate metrology. According to their working principles, the sensing techniques for data acquisition in coordinate metrology can be classified into two categories: contact and non-contact [VMC99, CN04]. Figure 1-1 gives an overview of this classification..

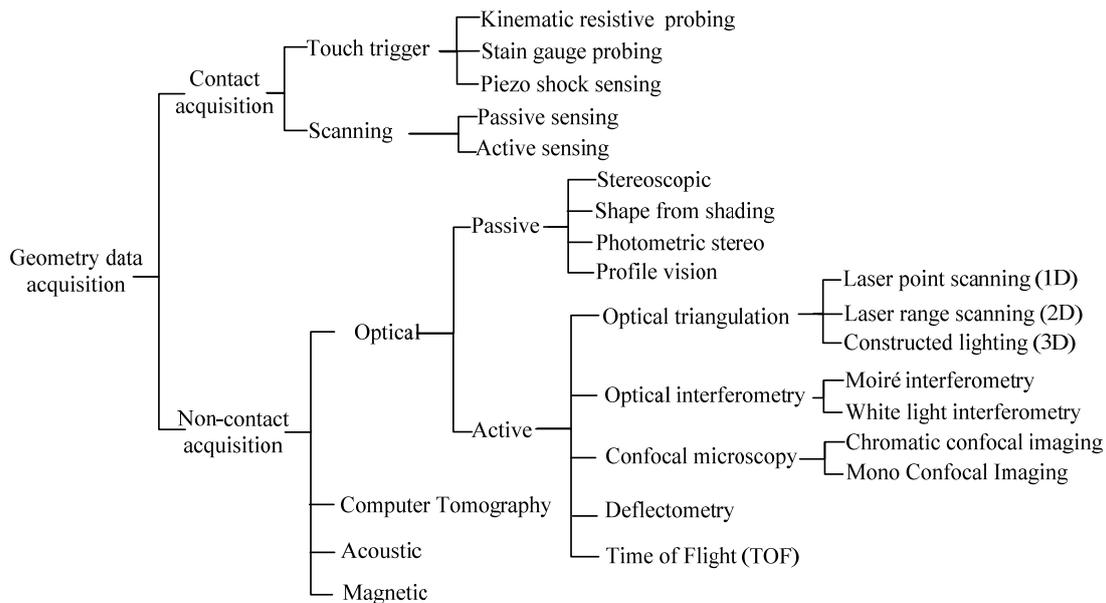


Figure 1-1: Classification of the sensing techniques for coordinate metrology

In the following sections, three data acquisition techniques that are widely used and cover different measurement scales are presented. Their working principles coordinate frames and calibrations are discussed below.

1.2.1. Contact probing

(1). *Probing principle*

Contact probing systems are usually applied in cases where surface measurements allow or require lower point data density, such as the inspection of prismatic objects, known surfaces. The measuring ranges span from about a micrometer to several millimeters in one, two or three dimensions [WJS09]. The contact probing sensors are usually slow (1~2 point per second). Other limitations of the probing system are that the regions will be inaccessible if the sizes of these regions are smaller than the

diameter of the tip ball or that the peaks might lead to smoothed approximation of the surface (see the blue regions in figure 1-2 (1)). Moreover, the different sizes of the stylus tips also influence the measurement results (see figure 1-2(2)) [WJS09].

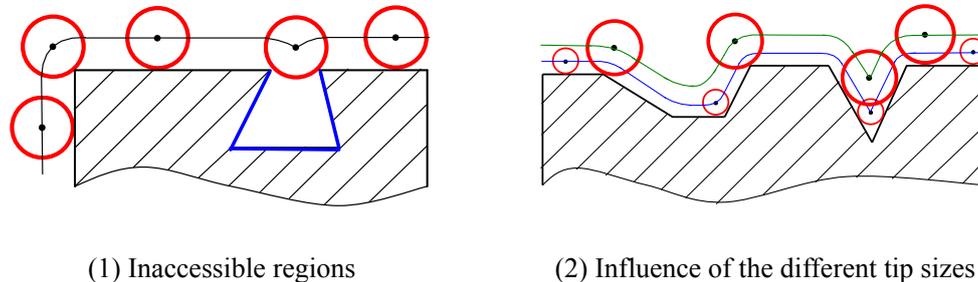


Figure 1-2: Limitations of the touch probing system with considering the tip size

However, the accuracy of contact probing sensor is higher. The contact probing sensor is more adaptive to the environment and simple.

The working principle of contact probe is based on a mechanical interaction. There are two models: touch trigger and scanning.

The touch trigger mechanism will generate a trigger signal when the stylus is deflected away from its fixed seated position. The trigger signal generated by the probe in real time will be processed to record the position of the contact point in the three axes. Hence, the touch trigger probing contains two basic steps. The first one is approaching the measuring point on the surface to generate the trigger signal. And then followed by a withdrawing procedure in which the probe is back off the surface, the stylus returns to its initial position and is ready for next point probing. According to different sensor structures, this mechanism can be decomposed into three types: Kinematic resistive probing (i.e. Renishaw TP20 stylus), Stain-gauge probing (i.e. Renishaw TP7M stylus), and Piezo shock sensing (i.e. Renishaw TP800 stylus) [Ren].

On contrast, in scanning mechanism, the probe tip is always in contact with the surface during the measurement process. The touching element is guided on a curve along the surface while a set of coordinates are sampled in a time sequence [WSJ09]. The scanning mechanism works based on passive sensing (i.e. Renishaw SP600) or active sensing (i.e. Zeiss VAST system) technologies [Ren, WEP04].

In general, the scanning sensors are more complex in structure, data analysis and monitor control than the touch trigger sensors. The points acquired by scanning sensors (up to 500 points per second) are much more than the trigger sensors, but with

more uncertainty [WJS09]. Accordingly, the scanning sensors are suitable to perform the measurement of size, position and profile of precise geometric features, while the touch trigger sensors can be employed for prismatic parts without significant variations.

(2). *Modeling of the contact probing system*

A contact probing system is a 3D data acquisition system in general, which means that the initial acquired data are 3D with (x, y, z) coordinates along the axes. The geometrical information can be derived from these 3D data. Most often, the functional characteristics of the contact probing systems can be derived from a Cartesian cylindrical or a spherical coordinate system. The probing process requires the definition of the coordinate systems for data acquisition. There are three coordinate systems in a 3D contact probing system [LG04, Ren] as shown in figure 1-3.

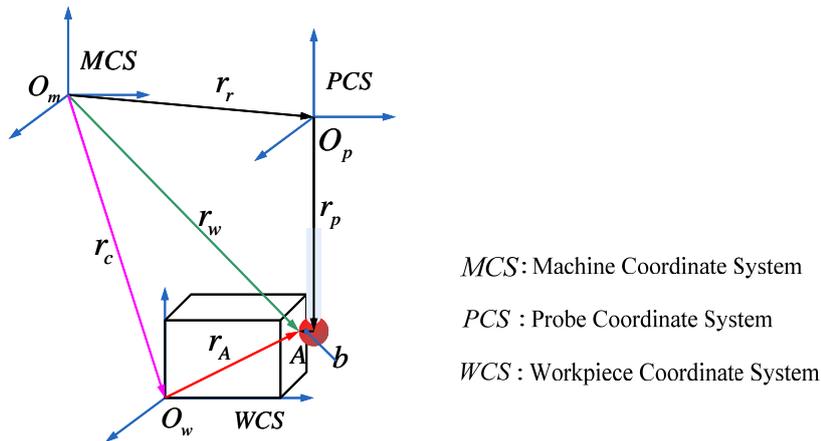


Figure 1-3: Coordinate systems in contact probing systems

When a workpiece coordinate system (WCS) is defined on a workpiece, all the measured points can be output as \vec{r}_A after transformation from the machine coordinate system (MCS) into WCS. From the relationships described in figure 1-3, the final result of \vec{r}_A can be derived as follows.

The coordinates of the point A in MCS is represented as:

$$\vec{r}_A = \vec{r}_r + \vec{r}_p + \vec{b} \quad (1-1)$$

\vec{r}_w can be represented as:

$$\vec{r}_w = \vec{r}_c + \vec{r}_A \quad (1-2)$$

Finally, we can obtain the final result as:

$$\vec{r}_A = \vec{r}_r + \vec{r}_p + \vec{b} - \vec{r}_c \quad (1-3)$$

Where, \vec{r}_A is the position vector of the probed point A in workpiece coordinate system. \vec{r}_r is the position vector of the origin of the probe coordinate system in machine coordinate system. \vec{r}_c denotes the position vector of the origin of the workpiece coordinates system in machine coordinate system. \vec{r}_w stands for the point vector of the probed point in machine coordinate system. \vec{r}_p is the position vector of the center of the probe tip in the probe coordinate system. and \vec{b} denotes the radius vector which starts from the center of the tip and ends at the contact point.

(3). *Calibration of the probing system*

Before inspection, the position of the center point of the probe tip related to the reference point (\vec{r}_p in figure 1-3) and also the radius of the tip (\vec{b} in figure 1-3) should be known first for a correct measurement [EW96]. This is the main purpose of a calibration process. Many factors influence these parameters, such as probing force, pre-traveling of the probe, elastic attribute of the probing system, temperature and other parameters [Ren].

The calibration can be done by experiments with a calibrated artifact to determine the compensations for each influencing factor in the probing system. The most common used calibrated artifact is a sphere [WEP04]. Because the great varieties of the different probing systems, the calibration process is also done in different strategies. However, for each stylus needed to be calibrated in the system, the general calibration strategy is composed of the following steps:

- (a) Selecting the calibrated artifact (with the same condition).
- (b) Choosing the location and orientation of the artifact.
- (c) Determining the number, location and sequence of the probing points on the artifact.

With the probed points, the experimental position and the tip ball size can be derived by surface fitting and the parameters can be compensated for the final data acquisition from the workpiece.

1.2.2. Laser range triangulation

(1). Principles of laser scanning

The laser scanner can acquire a high density of point data from the surface with high speed. Its non-contact nature makes it suitable to measure the surfaces with flexible or soft materials [SNK02]. Even the laser scanner has its own demerits, such as limited viewpoint, occlusion, and sensitiveness to the surface optical conditions (specular for example), noise and redundancy in the acquired data, the laser scanners are frequently employed in coordinate metrology [NYH97].

The laser scanner works based on optical triangulation. With the triangulation principle, a point on the measured object can be determined by the trigonometric relations between the projector, the camera, and the object itself [WRS00]. The basic spot triangulation principle in 1D is shown in figure 1-4 (1). The depth distance d can be calculated as the formula (1-4):

$$d = b \cdot \frac{\sin \alpha \cdot \sin \beta}{\sin(\alpha + \beta)} \quad (1-4)$$

Where, b is the basic distance between the projector and the camera, which is a known physical parameter with a given laser scanner. α and β are the angles of the projector and the camera respectively.

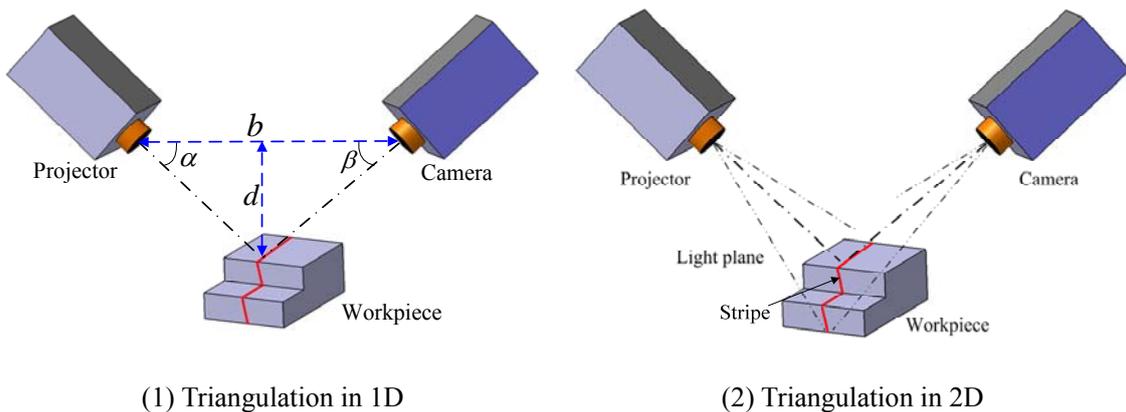


Figure 1-4: Triangulation principles of laser scanning

The spot scanner only captures the depth information of a single point each time. it needs to move in two directions for whole surface measurement. The point scanning in 1D is limited in accuracy and efficiency. Hence, the laser range scanner in 2D is more employed in industries. The range triangulation scanner generates a laser plane

to intersect the object and so a profile stripe can be acquired in each time, as shown in figure 1-4 (2). The calculation of each point on the scan line is similar as the spot triangulation. However, a whole line or profile is measured in each time rather than only a single point. The accuracy and efficiency are both improved greatly.

The accuracy of a laser scanner is usually related to many factors, such as the relative position of the scanner and the object, the view angle, the condition of the measured surfaces, etc.. Some authors have already done some research works on that domain [GCB07, ML08].

(2). *Modeling of the laser scanning system on CMM platform*

The laser range scanner is a 2D sensor. In each scanning, the scanner acquires a line image (represented by two dimensional parameters, such as (R, C)). Each point on the scanner line may correspond to a series of points on the object. An extrinsic calibration is therefore necessary to transform the 2D image data into 3D spatial data to represent the object naturally.

The laser scanner is usually mounted on a CMM platform or on a robotic arm when doing the data acquisition. In our research, we consider the Laser scanning system integrated on CMM platform. There are five basic coordinate systems in the scanning system [LBM04] as shown in the figure 1-5.

(a) Machine coordinate system (MCS) is associated with the CMM.

(b) Workpiece coordinate system (WCS) is the reference coordinate system of the workpiece when the workpiece is in the design stage.

(c) Laser coordinate system (LCS) is constructed on the laser light plane virtually during the scanning. The origin of LCS is the intersection point between the optical axis of the camera lens and the laser light plane. The z-direction is along the normal of the laser light plane. And the x-, y-directions are the projections of the two perpendicular directions of the imaging plane of the camera, see the figure 1-5.

(d) Camera coordinate system (CCS) is the coordinate system associate with the camera lens. The origin of CCS is the center of the lens and z-direction is along the optical axis of the lens. The x-, y-directions are the projections of image plane matrix.

(e) Image plane coordinate system (PCS) is the coordinate system of the camera on the image plane. The z-direction is along the optical axis. and the x-, y- directions

are represented as (R, C) . R and C denote the lines and the columns of the CCD matrix respectively. Hence, the coordinates of a point on image plane can be referred as (R_i, C_i) .

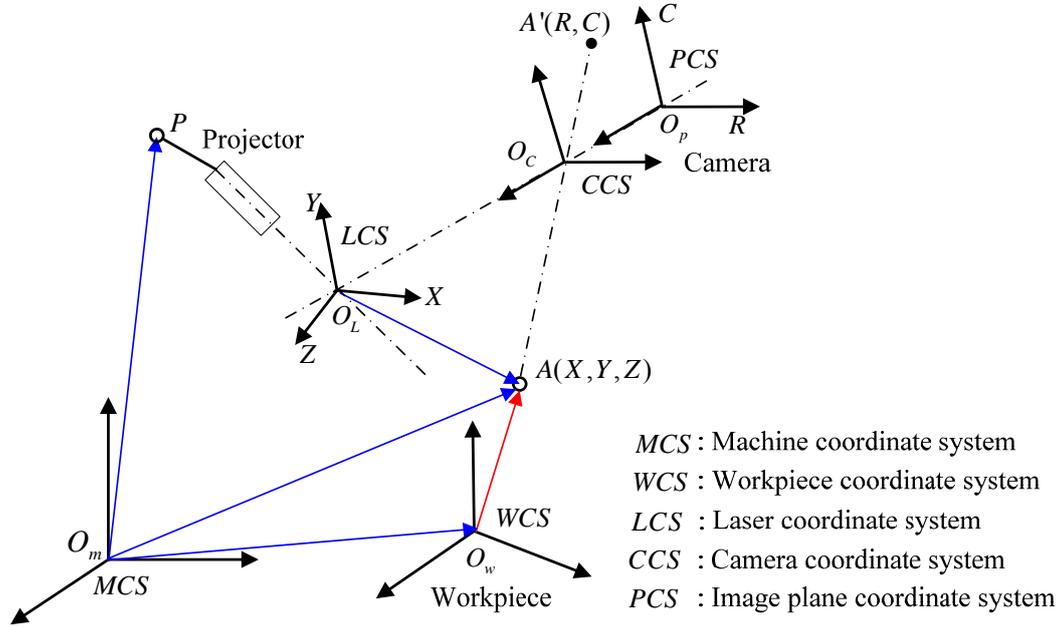


Figure 1-5: Modeling of the laser scanning system on CMM platform

Considering a point A on the surface, the three coordinates of A can be represented with WCS. The point A can also be represented with LCS. Therefore, we can get:

$$\overline{O_w A} = \overline{O_w O_m} + \overline{O_m P} + \overline{P O_L} + \overline{O_L A} \quad (1-5)$$

Where, the point P in formula (1-5) and in figure 1-5 is the reference point of the laser scanner fixed on the CMM arm. The coordinates of P are represented in MCS as $\overline{O_m P}$. $\overline{O_w O_m}$ and $\overline{P O_L}$ generate the translation vectors. $\overline{O_L A}$ represents the position vector of A in the coordinate system LCS.

Considering the coordinates transformation between the two systems, WCS and LCS, we can obtain the following equations:

$$\overline{O_w A} = R_{lw} \overline{O_L A} + T_{lw} \quad (1-6)$$

Where, R_{lw} is the 3×3 rotation matrix from LCS to WCS while T_{lw} is the 3×1 translation matrix.

In order to acquire the vector $\overline{O_L A}$, The relationships among the three coordinate

systems LCS, CCS and PCS should be formulized. According to [LBM04], the final relationship between LCS and PCS can be expressed as:

$$\overrightarrow{O_p A'} = G(\overrightarrow{O_L A}) \quad (1-7)$$

Where, A' denotes the imaging point corresponding to A . And so $\overrightarrow{O_p A'}$ represents the position vector of A' in PCS, which can be specified as (R_A, C_A) .

Considering the coordinates on the three axes of A in LCS, The equation (1-2-7) can be represented as:

$$\begin{cases} X_A^L = G_X^{-1}(R_A, C_A) \\ Y_A^L = G_Y^{-1}(R_A, C_A) \\ Z_A^L = 0 \end{cases} \quad (1-8)$$

Where, $G^{-1}()$ is the inverse function of $G()$.

Finally, the coordinates of the point A represented in WCS which can be viewed the measured point data can be obtained from the following formula:

$$\begin{bmatrix} X_A^W \\ Y_A^W \\ Z_A^W \\ 1 \end{bmatrix} = \begin{bmatrix} R_{lw} & T_{lw} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} G_X^{-1}(R_A, C_A) \\ G_Y^{-1}(R_A, C_A) \\ 0 \\ 1 \end{bmatrix} \quad (1-9)$$

The method to calculate the parameters mentioned in formula (1-9) can be found in a calibration process.

(3). *Calibration of the laser range scanning system*

The calibration considered here is the extrinsic calibration which is mainly used to determine the transformation relationships between the 2D image data in CCD space and the 3D spatial coordinates.

Considering the laser range sensor implemented in our experiments, which is a Kreon Zephyr KZ25 laser scanner [Kre], the laser beam is assumed as a plane. According to [BCD93], the formula (1-8) can be constructed as:

$$\begin{cases} X_A^L = \frac{b_1 \cdot R_A + b_2 \cdot C_A + b_3}{d_1 \cdot R_A + d_2 \cdot C_A + 1} \\ Y_A^L = \frac{b_4 \cdot R_A + b_5 \cdot C_A + b_6}{d_1 \cdot R_A + d_2 \cdot C_A + 1} \\ Z_A^L = 0 \end{cases} \quad (1-10)$$

Formula (1-10) brings out 8 additional parameters. These intrinsic parameters are linked to the sensor architecture and correspond to the model of the scanning system. According to formula (1-10), the global transformation of formula (1-9) can be represented as:

$$\begin{bmatrix} X_A^W \\ Y_A^W \\ Z_A^W \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & X_P^M + t_1 \\ r_{21} & r_{22} & r_{23} & Y_P^M + t_2 \\ r_{31} & r_{32} & r_{33} & Z_P^M + t_3 \\ & 0 & & 1 \end{bmatrix} \begin{bmatrix} \frac{b_1 \cdot R_A + b_2 \cdot C_A + b_3}{d_1 \cdot R_A + d_2 \cdot C_A + 1} \\ \frac{b_4 \cdot R_A + b_5 \cdot C_A + b_6}{d_1 \cdot R_A + d_2 \cdot C_A + 1} \\ 0 \\ 1 \end{bmatrix} \quad (1-11)$$

The final transformation relationship can be derived as:

$$\begin{bmatrix} X_A^W \\ Y_A^W \\ Z_A^W \end{bmatrix} = \begin{bmatrix} X_P^M \\ Y_P^M \\ Z_P^M \end{bmatrix} + \begin{bmatrix} \frac{a_1 \cdot R_A + a_2 \cdot C_A + a_3}{a_{10} \cdot R_A + a_{11} \cdot C_A + 1} \\ \frac{a_4 \cdot R_A + a_5 \cdot C_A + a_6}{a_{10} \cdot R_A + a_{11} \cdot C_A + 1} \\ \frac{a_7 \cdot R_A + a_8 \cdot C_A + a_9}{a_{10} \cdot R_A + a_{11} \cdot C_A + 1} \end{bmatrix} \quad (1-12)$$

Where, $[X_P^M \ Y_P^M \ Z_P^M]^T$ is the coordinates of P in MCS. a_i ($i=1,2,\dots,11$) are the 11 global parameters need to be determined in the calibration process.

The strategy of the calibration is quite similar as the calibration for contact probing system. We have to note that at least 11 points on the selected artifact need to be captured in order to identify the 11 unknown parameters in formula (1-12).

1.2.3. Chromatic confocal imaging

(1). *Principle of chromatic confocal imaging*

The chromatic confocal imaging based sensors focus on the applications in

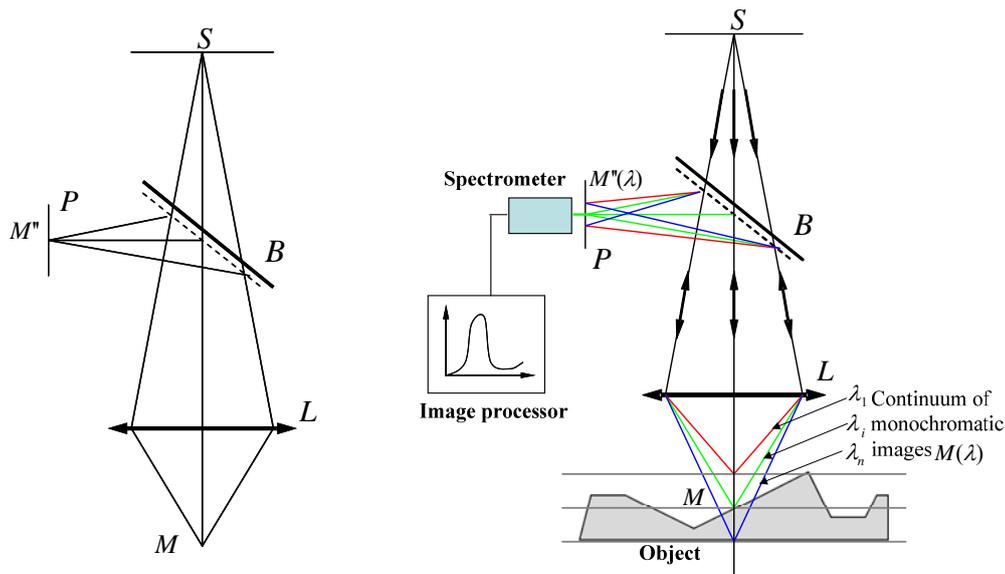
micro-domains, such as micro-metrology, micro-topography, etc. The measurement range of sensors belonging to this category is from several nanometers to several millimeters with high resolution.

The work principle of chromatic confocal imaging [Sti] is based on a quasi confocal configuration with extended z-axis field. The field extension is obtained by spectral coding of the z-axis. The confocal microscopy images only one object point at a time. The field of view must be reconstructed by (x, y) scanning.

Figure 1-6 (1) describes a basic confocal imaging configuration as a single point viewing system. The point source S is imaged by the objective lens L on the object point M . The retro-diffused light passes back through L and is then directed towards the detector by a beam splitter B . The diaphragm P located at the image of M given by L stops light coming from all points except M .

The altitude (z coordinate) of each point of the surface can be acquired by dynamic focusing, in other terms, by moving some mechanical part along z axis. This operation is very inconvenient in practices. An alternative approach is to extend the z axis by stretching the axial chromatism and working with a polychromatic point source (the most used, for example a white light source). With a univocal color coding to index the different wavelengths, the confocal system turns to a single segment view system, which is the basic idea of the chromatic confocal imaging system. The setup of a classical chromatic confocal imaging system is shown in figure 1-6 (2).

Practically, a white source is imaged by an objective lens with extended axial chromatism on a series of monochromatic point images in the measurement space. When the measured sample intercepts the measurement space at point M , a single of the monochromatic point images is focalized at M . Due to the confocal configuration, only the wavelength λ_M will pass through the spatial filter with high efficiency, all other wavelength will be out of focus.



(1) Single point confocal imaging (2) Chromatic confocal imaging for 3D surface

Figure 1-6: Working principle of chromatic confocal imaging system

In general, the confocal imaging principle yields an excellent spatial resolution regardless of ambient illumination. The chromatic coding ensures that measurement is insensitive to reflectivity variations in the sample and allows working with all types of materials. The chromatic confocal imaging sensor can provide high performance in micro-measurement domains.

(2). *Modeling of the chromatic confocal imaging system*

The sensor based on chromatic confocal imaging system implemented in our research works are STIL sensor [Sti]. The STIL sensor is composed of a CHR controller with an optical pen. The STIL sensor is a 1D sensor. It can only capture the altitude information (z coordinate) of each point. It hence needs (x, y) scanning to reconstruct its view field. Generally, this operation is achieved by mounting the STIL sensor onto the CMM platform. The x -, y -coordinates are acquired by CMM system. Figure 1-7 specifies the model of the STIL sensing system based on CMM platform.

There are four coordinate systems in the sensing system, as shown in figure 1-7. Machine coordinate system (MCS) is associated with the CMM machine. PH10 coordinate system (PCS) is the coordinate system attached with the PH10 header by which the STIL sensor is mounted on the CMM arm. The STIL coordinate system (SCS) is the coordinate system of the STIL sensor and the workpiece coordinate system (WCS) is the part design coordinate system. The SCS and PCS conform the

same three axes directions.

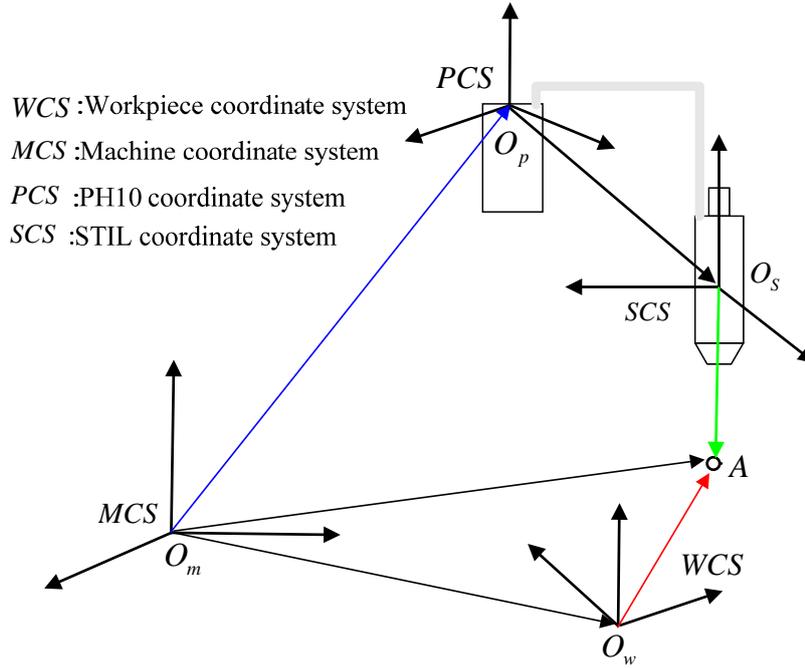


Figure 1-7: Modeling of the STIL sensor mounted on CMM platform

Considering a point A on object surface, the position vector of A can be represented in WCS, and also can be represented in SCS, Therefore, we can get the following equations:

$$\overrightarrow{O_w A} = \overrightarrow{O_m A} - \overrightarrow{O_m O_w} \quad (1-13)$$

$$\overrightarrow{O_m A} = \overrightarrow{O_m O_p} + \overrightarrow{O_p O_s} + \overrightarrow{O_s A} \quad (1-14)$$

Where, $\overrightarrow{O_m O_p}$ is a translation vector between the PCS and MCS. $\overrightarrow{O_p O_s}$ is a vector to record the position of the sensor in PCS. When the sensor is fixed on the PH10, $\overrightarrow{O_p O_s}$ should be known. $\overrightarrow{O_s A}$ is the coordinates of A in SCS. The z coordinate of $\overrightarrow{O_s A}$ can be acquired by the sensor and the coordinates along x and y axes can be acquired from the CMM. $\overrightarrow{O_m O_w}$ is the translation vector between MCS and WCS.

Combining the equations (1-10) and (1-11), we can finally obtain:

$$\overrightarrow{O_w A} = \overrightarrow{O_m O_p} + \overrightarrow{O_p O_s} + \overrightarrow{O_s A} - \overrightarrow{O_m O_w} \quad (1-15)$$

From the view of coordinate transformation, the equation (1-15) can be represented as:

$$\overrightarrow{O_w A} = R \cdot \overrightarrow{O_s A} + T \quad (1-16)$$

Where, R and T are the rotation matrix and translation matrix from the SCS to the WCS. When the system is setup, all the parameters can be acquired by a calibration process.

(3). *Calibration of the chromatic confocal imaging system*

The chromatic confocal imaging sensor used in our experiments is STIL sensor, which is a 1D sensor used in micro-domain with very high resolution (about 0.06 μm). However, the resolution of the CMM on z-axis is only 0.5 μm , which is much higher than the STIL resolution. Therefore, during each measurement process, the x-y plane of the CMM has to be fixed to an unchangeable z-value. Otherwise, the resolution of the results will lose greatly.

The calibration process is used to calculate the rotation matrix and translation matrix in equation (1-16). Unlike the contact probing or laser scanning system, the STIL sensor does its position and orientation calibration in two separate ways [Cos07].

The position calibration has the similar strategy with the probing sensors. an artifact should be selected properly considering the micro application nature of the STIL sensor. If the classical sphere is used, the measurement of a small region of the sphere may be not interesting enough to evaluate the spherical center. Instead, we use a facet sphere (shown as figure 1-8 (1)) by measuring the three planar facets to determine the center position of the sphere for the position calibration.

The orientation calibration is used to determine the orientation of the STIL sensor when doing the measurement tasks. For this purpose, two thin films with pre-known thickness values are used as shown in figure 1-8 (2). Assuming the STIL measurement orientation is along the direction z_1 (also z_2), the orientation angle between the measurement orientation and the plummet line can be computed as:

$$\theta = \arccos\left(\frac{b-a}{z_1-z_2}\right) \quad (1-17)$$

Where, a and b are the thickness of the two films, which are known before calibration.

Here, we must note that when the orientation of the STIL sensor changes, the orientation calibration has to be done again for correct measurement.

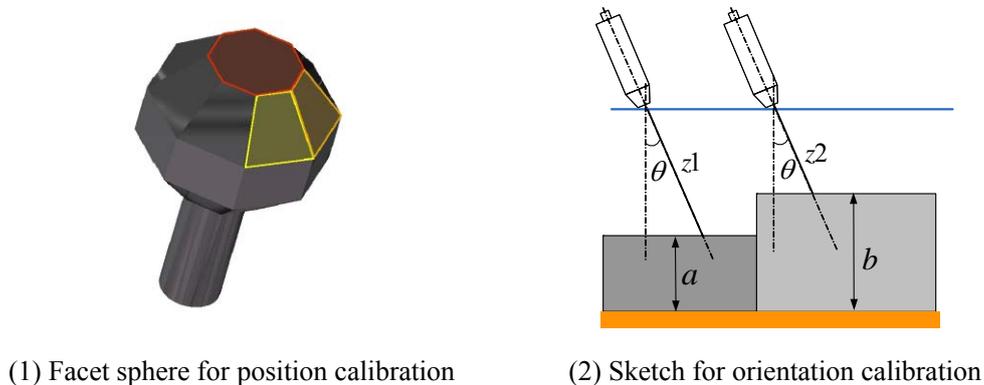


Figure 1-8: Calibration of the STIL sensor

1.3. Multisensor integration in coordinate metrology

Besides the mentioned three data acquisition techniques, there are also many methods widely employed for surface digitizing, some of them are represented in figure 1-1. These methods have distinct merits and demerits [NYH97] and hence have their special application ranges. With only one sensor, it could be very difficult to satisfy all the requirements of the ever-increasing complexities of modern data acquisition. The multisensor data fusion methods provide effective solutions to that problem. With multisensor methods, the particular features of a workpiece can be digitized by the most suitable sensor and the small uncertainty can be used to correct data from other sensors which exhibit relevant systematic error but have a wider field of view or application range [WJS09]. Thereby, the merits of each integrated sensor can be fully utilized and their demerits can also be avoided to improve the acquisition performance of the whole system.

1.3.1. Multisensor configuration

An important issue for multisensor measurement is how to integrate the multiple sensors onto a common measurement platform (e.g. CMM). The physical sensor configurations can be roughly classified into three categories [Dur88, WJS09] as shown in figure 1-9.

Complementary configuration is that the sensors work independently but the

acquired data can be combined to give more complete information of the measured object. An example of this configuration is the fusion of images captured with different illumination series to achieve images with higher contrast [HL03].

Competitive system is that when the sensors are configured as independent to measure the same feature in order to reduce the measurement uncertainty.

Cooperative configuration uses the information provided by two or more independent inhomogeneous sensors to derive data that would not be available from the sensor individually. The particular examples mentioned this kind of sensors configuration are related to multisensor integrated on the CMM platform and use the data acquired by an optical sensor to guide the tactile probing measurement [CBV01, NYH97].

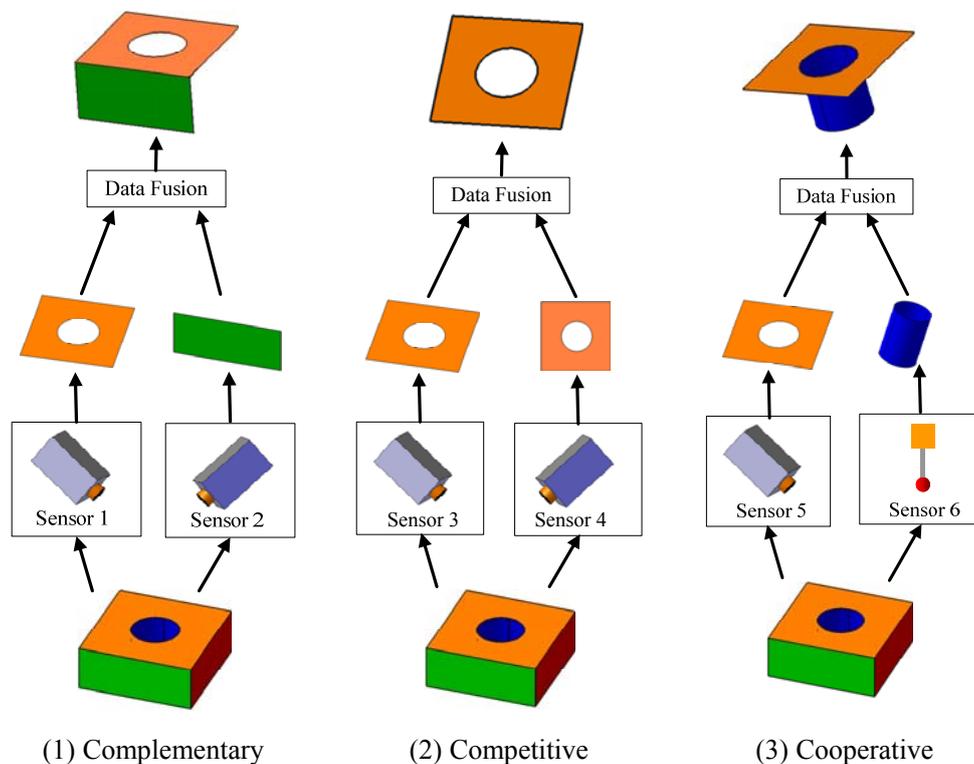


Figure 1-9: Three sensors configurations in multisensor systems

1.3.2. Multisensor data fusion

In applications, a particular multisensor measurement system may be configured as the combination of the two or more models to fulfill the complete measurement tasks. The data acquired by the multiple sensors, dependently or independently, are embedded in their own coordinate systems which are distinctly different from each

other. Thereby, how to merge the multiple data models and process to extract the geometrical features should be one of the most important stages for multisensor measurement. There is a great deal of issues that should be considered properly for data processing in order to achieve the multisensor integration. Generally, the process of the multisensor data fusion should comprise the following procedures:

(a) Pre-processing: the raw data acquired from multiple sensors are typically preprocessed to improve their qualities, such as denoising, filtering, etc. After pre processing, the data are more reliable for further processing. The pre-processing in multisensor data fusion should also include the data format conversion when it is necessary.

(b) Registration: every employed sensor has its own coordinate system which is usually different from each other. The measured data captured by respective sensor represented in its own coordinate system should be aligned and merged into a common coordinate system so as to represent the model completely. Moreover, in the recognition and position stages prior to the shape inspection, the digitized data from unfixed rigid objects also need to be registered with an idealized geometric model [BM92]. Therefore, registration is one of the most critical issues and decisive phases for data fusion and data processing and it hence should be taken into account.

(c) Fusion: data fusion procedure is performed to decide which data should be merged into the final data set, how to extract the useful information and how to handle the redundant data. The methods for data fusion should belong to one the following three clusters: estimation, inference, fuzzy or neural methods [WJS09]. Estimation methods, which include least square analysis [BRM92], weighted average [TGC04], Kalman filtering based methods [HQC09], etc., are suitable to analyze the measurement systems in which various results are acquired for the same measurand or for a regression plot are combined. Inference methods, like Bayesian probability theory [Pun99] are usually used to evaluate the measurement uncertainty.

1.3.3. Related works in multisensor integration

Comprehensive research activities have been addressed in multisensor based data acquisition. Multisensor fusion technique is an active issue in many applications, e.g. Robotics [ATS04], pattern recognition [GAG09], medicine [DVD02], non-destructive testing [KZD05], geo-sciences [MWW09], military reconnaissance and surveillance

[GPL04], etc. In the following sections, comprehensive research activities related to dimensional metrology and reverse engineering are surveyed.

(1). *Homogeneous optical sensors integration*

Measurement tasks in many applications require high quality data. The necessary quality is often achieved by applying data fusion methods across sensors, which is also called multi-model analysis. The integrated homogeneous sensors in such systems usually are cameras, laser scanners, or other optical sensors. Moreover, a series of images captured by the same sensor (named virtual sensors [WJS09]) also can be classified into this category.

One classical example of this integration setup is when applying the shape from shading technique. The setup consists of different illumination sources and a fixed camera. The camera captures a series of grayscale images with different illuminations. With the gradients analysis in these images, the height map of the object can be derived [Fra08, SRM06].

Another typical example is the multi-station photogrammetry network which integrated with several homogenous cameras. After calibration process [BA00] of each camera, each observation can be captured with several images simultaneously. These images finally can be registered to obtain a final global point cloud of the objects. Similar systems can be referred in [ALT05, SNK02, LPZ06].

For the shape measurement with complex structures, because the high accuracy of fringe projection systems [LLZ07], some researchers used the fringe projection sensor to replace the camera and constructed the measurement setup integrated with multi fringe projection sensors. Examples can be found in [WWH08, PZT02].

Some researchers, however, don't combine multiple sensors to capture multiple data. Instead, they use a single optical sensor to digitize the object several times to obtain a series of point data with different positions, focus depths or view orientations. More detail information of the object can then be extracted from these point data. Due to its economic cost cheap (only one sensor is required) and flexibility, these systems are quite widely researched and implemented.

C. Souzani, et al. [MTL06] developed a laser scanner self-guidance measurement system. In their system, the object is first scanned to generate the initial point data from where the characteristic edges can be extracted to guide the second scanning

using the same scanner.

For the objects which are beyond the measurement area of the sensor or too complex to be captured in one single measurement, Using one sensor to capture a series of partial views from different sensor positions and fusing them together is an effective solution. Similar systems are also mentioned in [Wil01, PZT02, XZJ06]. The methods in which the same feature is captured by a single camera from different positions to enhance the image quality are also discussed in [PK06, Pie03].

(2). *Contact and optical multisensor integration*

In modern dimensional metrology or reverse engineering, the increasing requirements in terms of flexibility and automation of the whole digitization process result in a great deal of research efforts on cooperative integration of inhomogeneous sensors. The sensors implemented in such systems mostly are mechanical probes and optical sensors [WJS09].

The optical sensors can be a simple video-camera, or a laser scanner, which acquires the global shape information and provides the guidance information to drive the CMM execute the local exploration with a more precise tactile probe [ZAB09]. In such systems, the strengths of the two kinds of sensors, i.e. the ability of an optical sensor to quickly generate the global information and the ability of a contact probe to obtain higher accurate measurement data, can be ensured at the same time. There have been considerable researches on the cooperative sensor integration.

Chan, et al. [CBV01] developed a multisensor system integrating a CCD camera and a tactile probe on CMM platform for reverse engineering. The two sensors are fixed on the CMM arm together. The images captured by the CCD camera are processed by neural network based method to provide the geometric data which can be used for planning the probing path of the tactile sensor. The CCD images play the role of the CAD model like in CAD model based inspection planning systems [ZWW06].

Similarly, Carbone, et al. [CCS01] proposed a method to combine a stereo vision system and a scanning touch probe. In their method, the 3D vision system is performed to acquire a number of clouds of points which are fused to approximate the initial CAD model and to guide the CMM programming of the touch probe. The touch points data are then import to the CAD environment to produce the final, accurate

CAD model. The approach mentioned in [CL97] uses the similar way for reverse engineering.

Nashman et al [NYH97] integrated a vision system and a touch probe for dimensional metrology purpose. In their method, The Vision camera is fixed on the CMM table and the touch probe is mounted on the CMM arm with three translation degree-of-freedom. The workpiece is located in the view field of the camera. The images captured by vision camera are used for the workpiece positioning. With comparison of the image data and the data generated by the machine scales and the probe, the position of the probe to the interested features are computed to guide the path generation of the tactile probe. The tactile measurement can provide the final inspection data.

Menq, et al. [SHM00, SHM01] presented a cooperative sensor integration system that fused a vision system and a touch probe for coordinate metrology. The objects are first scanned by the vision system to recognize the global information. Due to the occlusion of the vision system, some regions may be not able to be digitized. In their method, they estimated the unknown regions from the vision data to guide the touch probe to supplement the missing data. The regions that require higher accuracy are also probed again by the touch sensor. The multisensor based inspection planning algorithms were researched based on information integration.

More recently, Huang and Qian [HQ07] developed an approach to combine a laser scanner and a touch probe dynamically. A workpiece is first scanned by the laser scanner to capture the overall shape. It is the probed by a touch sensor where the probing positions are determined dynamically to reduce the measurement uncertainty according to the scanning data. They use the Kalman filtering to fuse the data together and to incrementally update the surface model based on the dynamic probed points.

In the methods demonstrated in [ASG04, EC01, JOM06], a laser scanner is used to detect the large point cloud data files required to define freeform surfaces, whereas a CMM touch probe is used to precisely define the boundary of bounding contours. Both sensors are mounted on the CMM arm. Generally, the objects need several scans with different views by the laser scanner to acquire complete point data. Multisensor systems in which, other sensors, like conoprobe [CWL03] or construct lighting sensor [XWZ05] are also mentioned in literatures. The digitizing errors are also analyzed in [EC01]. In some papers, the multisensor calibration problem has been researched to

reduce the measurement uncertainties [HQ07, HQC09 and SHM00].

(3). *Inhomogeneous optical sensors integration*

Like the optical/contact sensors integration, the multiple optical sensors with different principles can be well combined by cooperative integration. Hence the different resolutions and multi-scales measurement can be ensured in a single system with the inhomogeneous sensors integration.

In such systems, the lower resolution sensors (e.g. conventional camera) are usually used to capture the global information. Then, a data analysis phase follows to evaluate the integrity of the measurement data. If there is not enough information, further local measurements with higher resolution sensors are required in which the position of the local measurement regions, the configuration of the sensors are often derived from the global information. The final result data are updated after combining each additional measurement datum until the measurement tasks are fulfilled [WJS09]. The system developed by SoKolov, et al. [SKT05] combined a confocal sensor and a scanning probe sensor for nano coordinate metrology. The feasibility of combining an SPM (Scanning Probe Microscope) with an optical interference microscope in a single measurement is demonstrated in [TDK04]. Weckenmann, et al [WN03] combined a white light interferometer sensor and a fringe projection sensor to measure the wear of cutting tools.

Some systems have also mentioned the integration methods to combine several optical sensors for the complementary configuration [RRT01, SPM03].

Other multisensor systems applied to different purposes in dimensional metrology and reverse engineering are surveyed in [WJS09] and dimensional metrology for freeform shape measurements are well surveyed in [SDS07].

(4). *Commercial Systems*

Multisensor CMMs use a combination of several sensors to provide higher precise or larger ranges of the measurements. Many CMM manufacturers, like, Mahr, Hexagon, Werth Messtechnik, Zeiss, etc. [CN04], can provide the multisensor solutions. Most of these solutions combine the optical sensors with a tactile probe in a cooperative configuration, which yield the following advantages.

(a) The optical sensors are applied to acquire the global data of the object which

can guide the more precise sensors automatically.

(b) The inhomogeneous sensors capture the information that would be not available from a single sensor, which provides more details about the object. The holistic measurements can be achieved.

Some commercial systems integrate the measurement vision system, different optical and contact sensors, but also can include the sensors based on computed tomography, fiber probe [CN04, CZ08, Myc08]. However, the methods combining the different sensors in these systems are usually not published due to commercial purposes. Additionally, multisensor systems based on tracker sensors [Met08], interferometry or photogrammetry [GFM08] etc. are also available.

(5). *Integration within a PLM context*

The measurement system is not isolated and should be integrated with other activities in the context of PLM (Product Lifecycle Management). It is also an important issue to embed the measurement activity into the integrated manufacturing process. The reverse engineering should also consider the integration problem because the point data should reconstruct the CAD model which needs to satisfy the design intention and specifications. There are also considerable research works trying to achieve the system integration of measurement systems with CAD/CAM systems or measurement execution systems. We will survey the related research literature in this section.

The most common solution for the system integration is based on interface standards in which the data in each respective system which would be exchanged with other systems are specified with standard file formats. The systems parse and derive the useful information from these standard data. Many standards and neutral files are published for these purposes, like STEP (Standard for the Exchange of Product Model Data) [Step], DMIS (Dimensional Measuring Interface Standard) [Dmis], DML (Dimensional Markup Language) [Dml], I++DME (Inspection plus plus Dimensional Measurement Equipments) [Nist] etc. The figure 1-10 shows the general interfaces in the dimensional metrology systems proposed by NIST [Hor05].

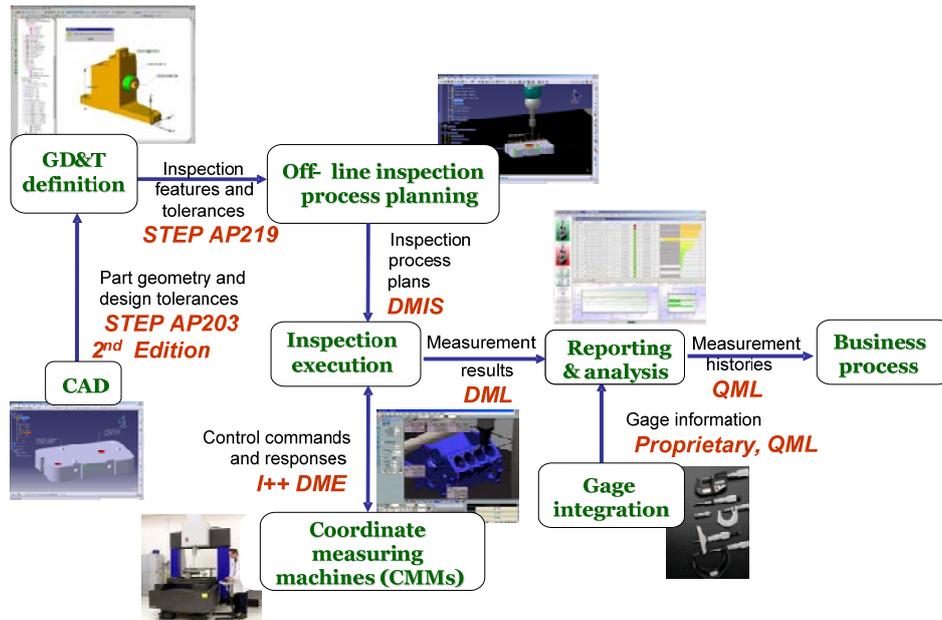


Figure 1-10: Interface standards in dimensional metrology systems [Hor05]

Due to the different purposes of the measurement results, the interface standards for results reporting and analysis are still immature. Here, we only consider the interfaces of the inspection planning system with the CAD/GD&T and also with the CMM execution systems.

(a) Interface between CAD/GD&T and inspection process planning. Inspection process planning strategy is critical for the whole measurement because it influence the global performance of the measurement system. If the inspection process is properly planned, the measurement could generate reliable result data within an object time. Otherwise, the system may fail to measure the objects with unexpected conditions. In inspection process planning strategy, the following tasks should be solved [WCV06]:

- Recognizing and extracting the inspection features and their associated constraints
- Planning the appropriate inspection method for each respective feature
- Integrating the inspection operations and generating an effective and efficient collision-free inspection path for the total features

Inspection features are usually derived from the design information. Traditionally, the inspection process is planned manually based on 2D drafts. Efforts have been

performed in recent years to address automated features recognition and extraction from 3D CAD models.

Considerable researches are oriented using API (Application Programming Interface) tools provided by the CAD systems to extract the design features and then interpret them as inspection features [HPM07, LMM01, ZAW00, SKL03]. The main limitation of these methods is that they greatly rely on the special CAD platforms.

More commonly, some institutes attempted to solve the problem based on standard neutral interfaces, like STEP APs, IGES, etc [Sri08-1]. According to the system developed in WEPROM project, the CAD based inspection process planning system for touch probing is designed based on STEP AP214. In the subsequent project ProSens is combined with STEP and an industrial standard Q-DAS [Qdas] is implemented to extract tolerance information [ZWW06]. The system was extended to support inspection process planning for optical sensors (stereo vision and conoprobe) [CWL05]. Other feature extraction methods based on production standards are also mentioned in the literature [Fen94, GC96, KHM01, MZL03, SCK06, ZWM01].

(b) Interface between inspection planning and execution systems. The off-line planned inspection processes need to be transformed into executable code files and drive the CMM to measure the objects conforming to the planned processes. DMIS is the only international standard language (ISO 22093) used for the control of measurement equipments, especially for CMM [KH09]. A majority of the CMM software vendors implement DMIS. Some commercial software packages, like PC-DMIS [Pcd], INSPEX [Ins], can generate the DMIS files automatically after the inspection process planning. Some inspection systems developed in academia also use the DMIS files to store and transfer the inspection processes [Fen94, KHM01, SCK06].

DMIS is an interface standard for the CMM software. On contrast, I++DME is used to support the interface between the CMM software and the machine, to send the machine commands and receive the feedback or response. I++DME is first proposed by five automotive manufacturers (AUDI, BMW, Daimler-Chrysler, VW, Volvo). Zeiss, Renishaw as well as others announced I++DME compliant products [Hor05]. Some improvements have already been made for the harmonization between DMIS and I++DME specifications [Hor05]. However, I++DME focuses on the hardware control which is of interest for the measurement equipment vendors. Until now, there

is still few researches in this area.

1.4. Development of an integrated multisensor system for coordinate metrology

In the previous sections, a comprehensive review of multisensor integration techniques and related research works showed the necessity to develop suitable measurement systems in order to get holistic, more accurate and reliable information about measured workpieces.

To achieve an important level of multisensor integration in coordinate metrology, we developed an integrated multisensor system based on three sensors (laser range scanner, trigger touch probe and chromatic confocal sensor) as shown in figure 1-11.

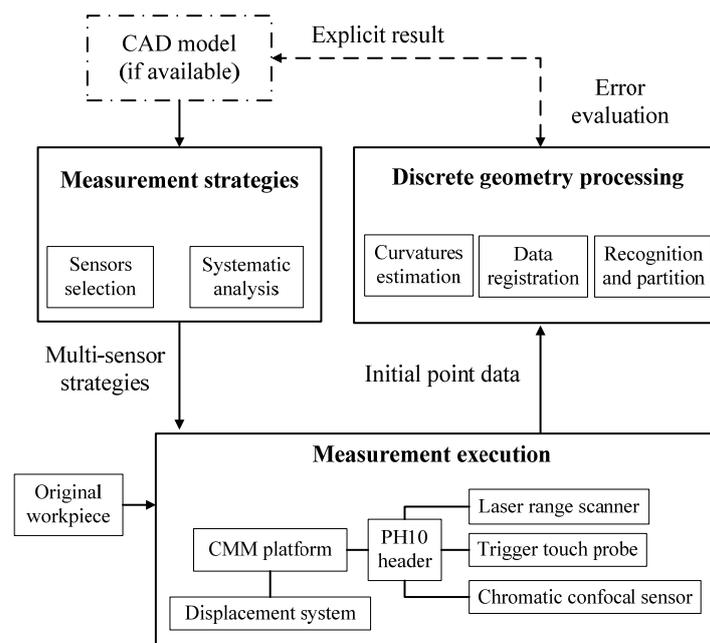


Figure 1-11: The constructed system layout of the multisensor system

The system is composed of two main components: data acquisition system and data processing system.

Data acquisition system includes two parts. One is measurement strategies and the other is measurement execution, in which the techniques for multisensor data acquisition are achieved. The three used sensors are inhomogeneous. They can be configured as complementary or competitive according to different measurement tasks. The proposed system can also achieve multiscales measurement.

Data processing system is used to recover and process discrete geometric information from the acquired point data in order to retrieve necessary information for shape registration and segmentation.

The development of such multisensor system requires both physical and logical integration for which system function analysis and information modeling techniques should be used efficiently (chapter 2).

1.5. Conclusion

Multisensor integration is proven to generate measurements with holistic, more accurate and reliable information. In this chapter, we focus on the multisensor integration techniques for coordinate metrology.

Firstly, three sensors which are widely used and cover different measurement scales in coordinate metrology are discussed considering their working principles, coordinate systems and calibrations.

Secondly, the multisensor integration for surface digitizing is introduced. The different multisensor configurations and general procedures for multidata fusion are presented and we discussed the related work in multisensor integration. Homogeneous optical sensors integration, inhomogeneous optical and touch probe integration, inhomogeneous optical sensors integration and system integration within a PLM context are surveyed. Commercial systems for multisensor integration in dimensional metrology are also described.

Finally, we introduced the layout of the developed multisensor system within our research work. The system can cover the multiscales measurements with complementary or competitive sensor configuration.

Chapter 2

Function and Data Modeling for Multisensor Integration

2.1. Introduction

Due to the complex function requirements and various kinds of existing data (specifications of workpiece and sensors, measurement process planning, measurement results, etc.), the development of multisensor integration system in coordinate metrology is still difficult and immature.

In order to achieve the multisensor integration, a comprehensive function analysis and data modeling of the system are necessary to obtain clear and unambiguous understanding and modeling of the whole system. Function modeling is used to specify the function requirements of the system while data modeling is used to manage the information of the system with structured data and unified representation which is convenient for data reusing and sharing.

In this chapter, we build the function models of the multisensor integration system using IDEF0 diagram. The activities, information flows and resources for multisensor integration are specified clearly in the function models. We also propose an ontology-based approach to build the structured data model using Protégé-OWL. The classes, instances and their properties are defined unambiguously for data management in multisensor integration.

This chapter is organized as follows:

Section 2.2 presents the functional analysis of the system. Section 2.3 gives the detailed data models which support the multisensor integration and section 2.4 is the conclusion.

2.2. Functional analysis of the system

2.2.1. Function modeling using IDEF0

The achievement of the measurement system for surface digitizing based on multisensor integration relies on the achievements of a series of functions or activities. The standard method to analyze the decisions, actions and activities of a system is IDEF0 (Integration Definition for Function Modeling) method [IDE93]. IDEF0 is

derived from the well-established graphic modeling language Structured Analysis and Design Technique (SADT) to build the system function models. A function model is a structured representation of functions, activities or processes within the modeled system or subject area. The purposes of function modeling are to provide a clear description of the functions and processes, to help specifying the information needs, and establish a basis for determining product and service costs.

The primary strength of IDEF0 is that the method has been proven effective in detailing the system activities for function modeling and structuring the analysis of the information communication within the activities. Activities in IDEF0 diagram can be described by their Inputs, Outputs, Controls, and Mechanisms (IOCMs). The hierarchical nature of IDEF0 facilitates the ability to construct models that have a top-down representation. In each hierarchical level, an activity illustrates the transformation from input to output with the support of control and mechanism. Using a simple box represents the activity and the arrows represent the data. The graphic format of IDEF0 diagram can be shown as figure 2-1.

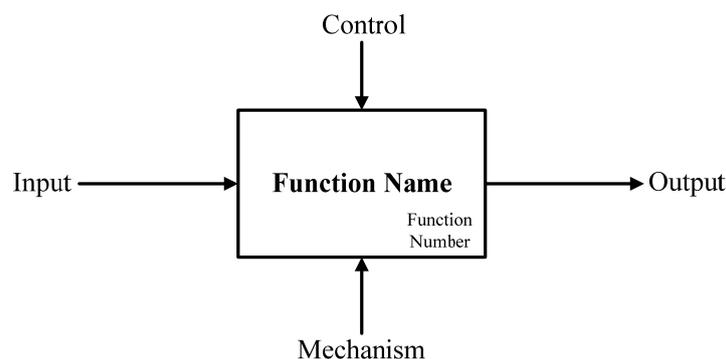


Figure 2-1: Graphic format of IDEF0

The IDEF0 models describe a system for a specific purpose and from a selected viewpoint. For different purposes or from different viewpoints, the IDEF0 models might be very different even for the same modeled system. The functions can be decomposed into more detailed diagrams until the subject is described at a level necessary to support the goals of a particular project. Generally, the top level diagram provides the most general or abstract descriptions of the system. The top level diagram is followed by a series of child diagrams providing more details about the modeled subject. The decomposition of IDEF0 diagram can be represented as figure 2-2.

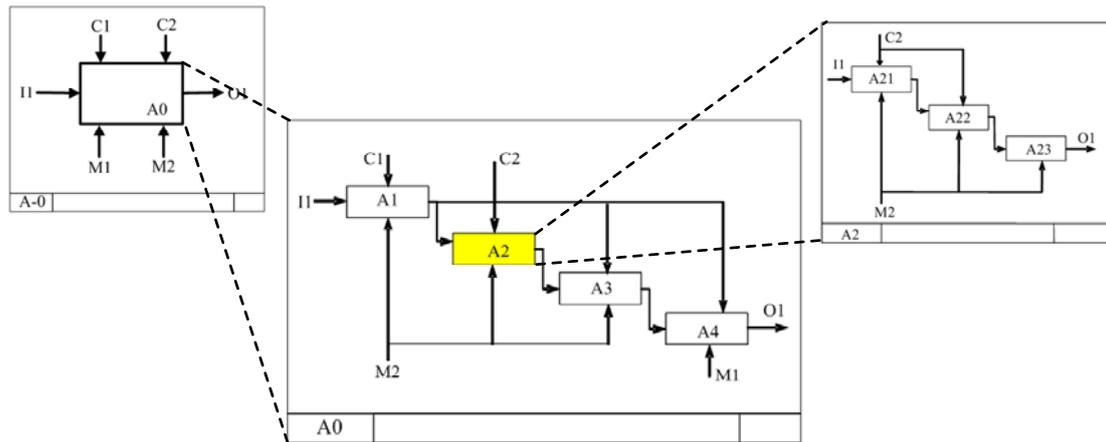


Figure 2-2: Hierarchical decomposition of IDEF0 diagrams

Several attempts have been made to analyze manufacturing systems using IDEF0 method. Ang et al. [ALG95] proposed a knowledge-based IDEF0 method to model manufacturing process planning systems. The function modeling of an inspection integrated framework was developed by Barreiro, et al [BLV03]. From the standpoint of the inspection as an integrated part of a manufacturing system, Lin et al [LL00] built an IDEF0 model of the measurement process planning system. Lin and Chow [LC01] defined a system model based on IDEF0 and STEP to define the process and information flows for the CMM based measurement systems. There is few research work in function modeling for multisensor integration systems in coordinate metrology.

2.2.2. Function models

(1). *A-0 diagram*

One of the most important features of IDEF0 methodology is that it gradually introduces greater and greater levels of detail through the diagram structure describing the model. In this way, an IDEF0 model starts by representing the whole system as a single unit – a box with arrow interfaces to functions outside the system. This is called the A-0 diagram. The A-0 diagram provides the most general or the maximum abstract description of the system function and its complete external interfaces. The A-0 diagram of the subject – Multisensor integration and geometry processing for coordinate metrology is shown in figure 2-3.

The A-0 diagram specifies the general ICOM data of the modeled system. In our

model, the input data mainly contains the physical workpiece to be digitized, its existing specifications and the inspection sequence. If the CAD model of the workpiece is available, it also can be used to perform the measurement strategy planning and resulting data analysis for quality control. The output information of the system includes the digitized point data and the analyzed results for different purposes, etc. The system is operated under the guidance of Measurement equipments specifications, related standards and policies, and the related knowledge. The supporting elements or mechanisms are human operator interactions, digitizing software and hardware tools.

Two important statements specifying the model’s viewpoint and purpose shall be briefly presented in A-0 diagram, which helps guiding and constraining the creation of the model. The viewpoint of the IDEF0 model is set for the system developer focusing on the multisensor integration achievements. The purpose is to analyze the function requirements for the data acquisition and processing based on a multisensor measurement system.

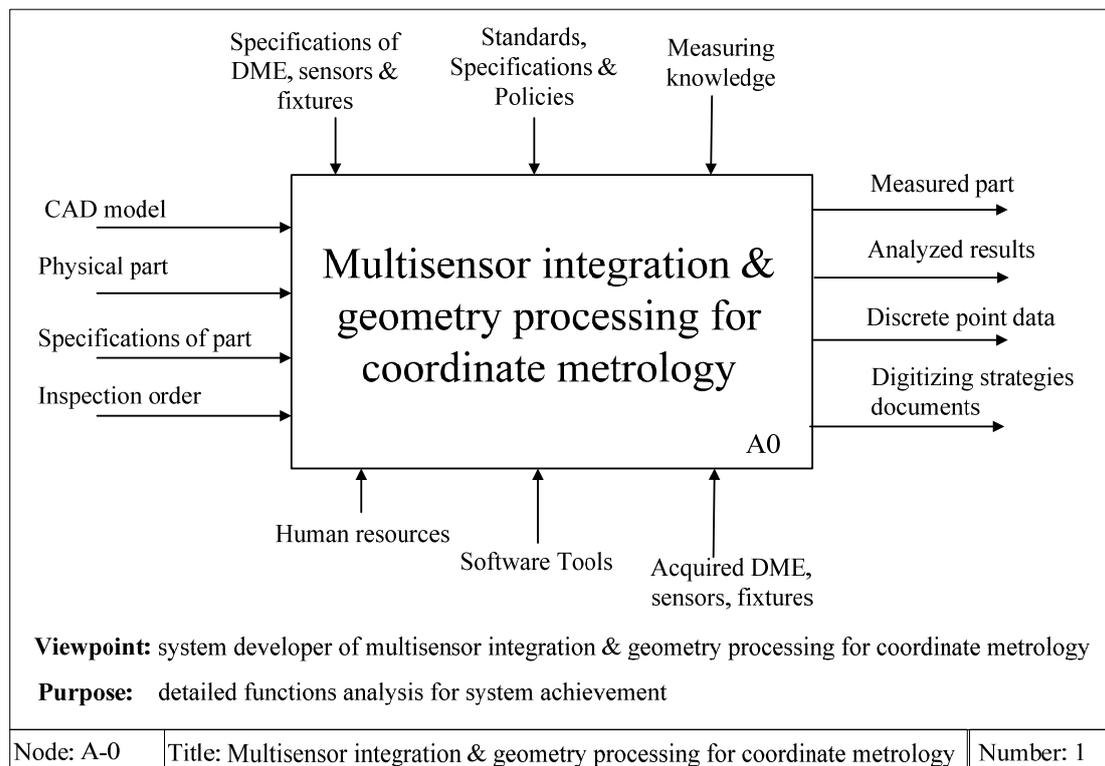


Figure 2-3: Top-layer IDEF0 diagram of the system – A-0 diagram

The A-0 diagram is the start-point and the root of the analyzed IDEF0 model from which a more detailed model – A0 diagram – can be derived by decomposing the A-0

context diagram in top level.

(2). A0 diagram for multisensor system construction

A simple activity box in IDEF0 diagram, if detailed, is always detailed on a child diagram into no fewer than three boxes and no more than six boxes. In order to specify the system with more detailed information, the top level A-0 diagram is always required to be decomposed to construct the A0 diagram.

The A0 diagram of the modeled system is illustrated in figure 2-4, from where we can find that the system contains four basic activities: data extraction and recognition, measurement strategies planning, Measurement execution and discrete geometry processing.

(a) A1: Data extraction & recognition.

The activity A1 – Data extraction & recognition- is used to interpret the design specifications of the workpiece for measurement purpose. If the CAD model is available, the activity A1 will extract the features, dimensions & tolerances data, and other specifications from the CAD model. The operation to associate the dimensions & tolerances data with their corresponding features is usually required because these two kinds of specifications are stored and expressed separately in a CAD database. The measurement elements extraction and recognition based on a CAD model can be performed in two ways: either using further API development tools associated with the CAD software package; or using standard neutral file formats. The first way relies on the development environment. The second method which is more common and adaptable is based on the neutral files used for inspection elements extraction such as STEP 10303, IGES, DXF, etc.

If the CAD model is unavailable, which is the most frequent in reverse engineering configurations, the data extraction and recognition can be done based on the scanning data. In this way, the workpiece is first scanned by an optical sensor to acquire the global information. The scanning data is then analyzed to recognize and extract the shape information and characteristic points, based on which new features and their specifications can be acquired through more detailed measurements.

The output of activity A1 contains the labeled inspection features, dimensions & tolerances, measurement requirements and the associations within the geometry features.

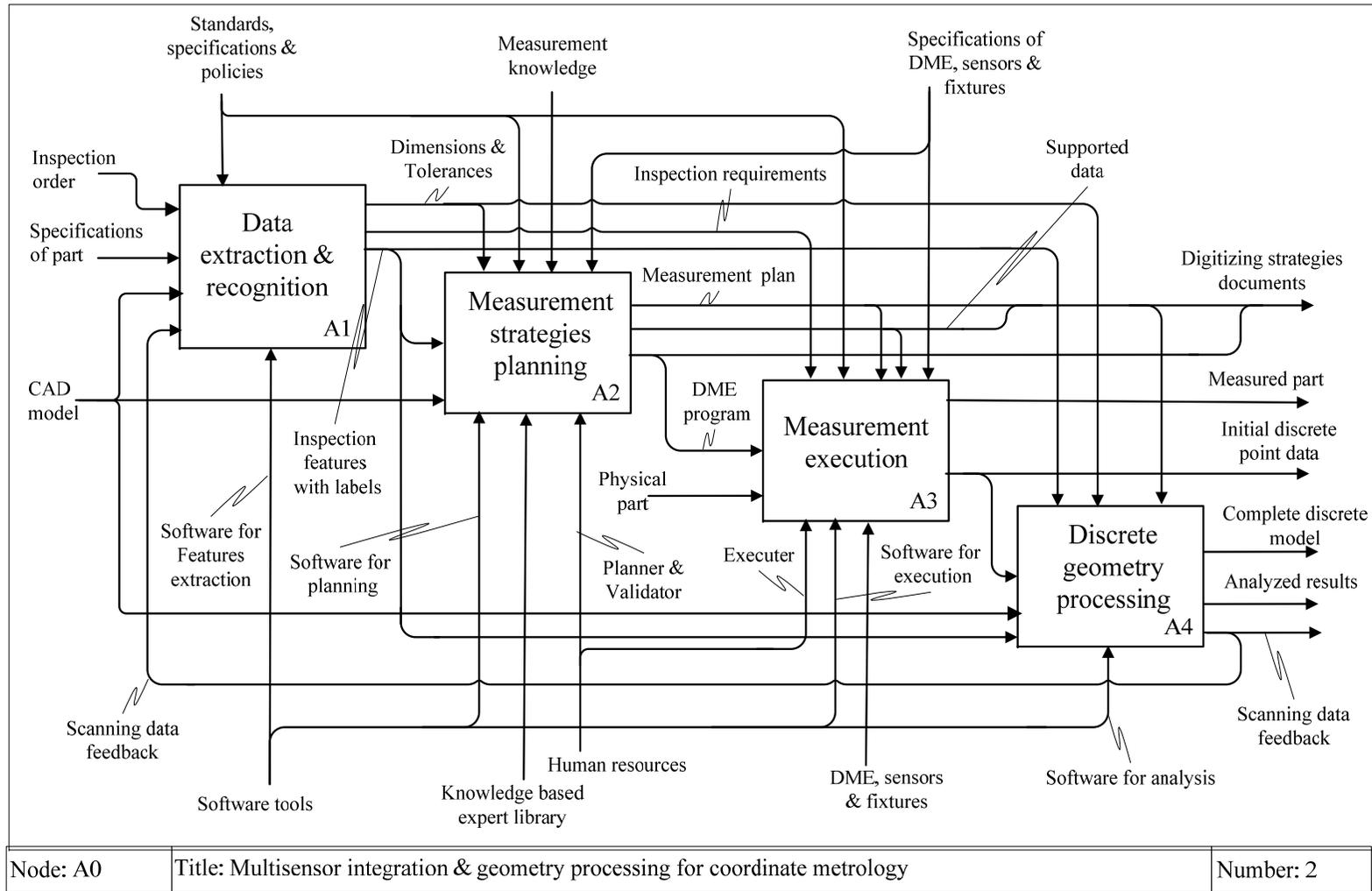


Figure 2-4: A0 diagram – the first layer of the hierarchical structure

(b) A2: Measurement strategies planning.

The activity A2 is dedicated to specify the detailed measurement planning of the whole digitizing task. Each sensor has its own characteristics which may be different from other sensors. Therefore, each measurement feature should be matched with the most suitable sensors and the proper measurement process should be planned accordingly. Finally, the executable DME (Dimensional Metrology Equipment) program which is used to drive the DME to perform the measurement is generated that conform to the DMIS standard for instance. In this activity, besides the sensor complementary configuration, the functions to achieve the sensor cooperative configuration based measurement are specified with more considerations. The more detailed information for Activity A2 can be found in its child diagram which is specified in section 2.2.2 (3).

(c) A3: Measurement execution.

The activity A3 is used to model the function requirements to achieve the measurement execution. The measurement execution is controlled by the DME program generated from the activity A2. In this activity, the setup for the measurement including the fixtures and calibrations of multiple sensors, fixture of the workpiece, etc should be first considered. Considering the laser scanner guide cooperative measurement, in which the initial scanning is executed firstly. The scanned data analysis and the guidance to other sensors need the interaction with the activity A4 (result analysis) and activity A2 (measurement strategies planning) respectively. Thereby, during the measurement execution process, some commutations might be needed with the activity A2 and the activity A4. The output of activity A3 should be the measured workpiece and the raw point data from different sensors which are the initial input to the data analysis activity A4.

(d) A4: Discrete geometry processing.

The activity A4, discrete geometry processing, focuses on the function modeling for the result analysis to recover geometric information underlying the digitized point data. Generally, different sensors capture the geometric information with different file formats. The activity A4 should have the capability to process all these kinds of data obtained from the integrated sensors in the measurement system. Basically, the data processing procedures are different according to different purposes. For example, in dimensional metrology, the digitized data are processed to extract the actual

dimensions and tolerances of the workpiece. With the comparison to the corresponding design nominal values, the quality of the measured workpiece can be controlled; whereas in reverse engineering, the shape information is more considered for the model reconstruction purpose. However, there are some common procedures which are often required by general engineering applications, such as multidata registration and fusion, shape recognition and segmentation, etc. In our method, the activity A4 is conformed on these common procedures. The detailed description of activity A4 is specified in its child diagram which is shown in section 2.2.2 (4).

(3). *A2 diagram for measurement strategies planning*

Measurement strategies planning activity A2 is one of the critical stages to achieve the functions of multisensor measurement system. There are three kinds of sensor configurations, complementary, competitive and cooperative [WJS09]. In activity A2, they should be considered properly. In general, the first two kinds of sensor configurations are simpler than the cooperative configuration since the multiple sensors in the first two kinds are configured independently of each other. The measurement process planning for each sensor can be made independently and then they can be combined together. However, in cooperative configured systems, the communications among the respective sensors should be the first requirement for the function achievement.

According to different sensors integrated in the system, the strategy plan may differ. In the developed system, three distinct different sensors are combined on the CMM platform, which are: a Kreon laser range scanner, a Renishaw trigger tactile probe and a STIL sensor based on chromatic confocal imaging principle. The function specifications of the measurement process planning based on the three sensors are shown in figure 2-5.

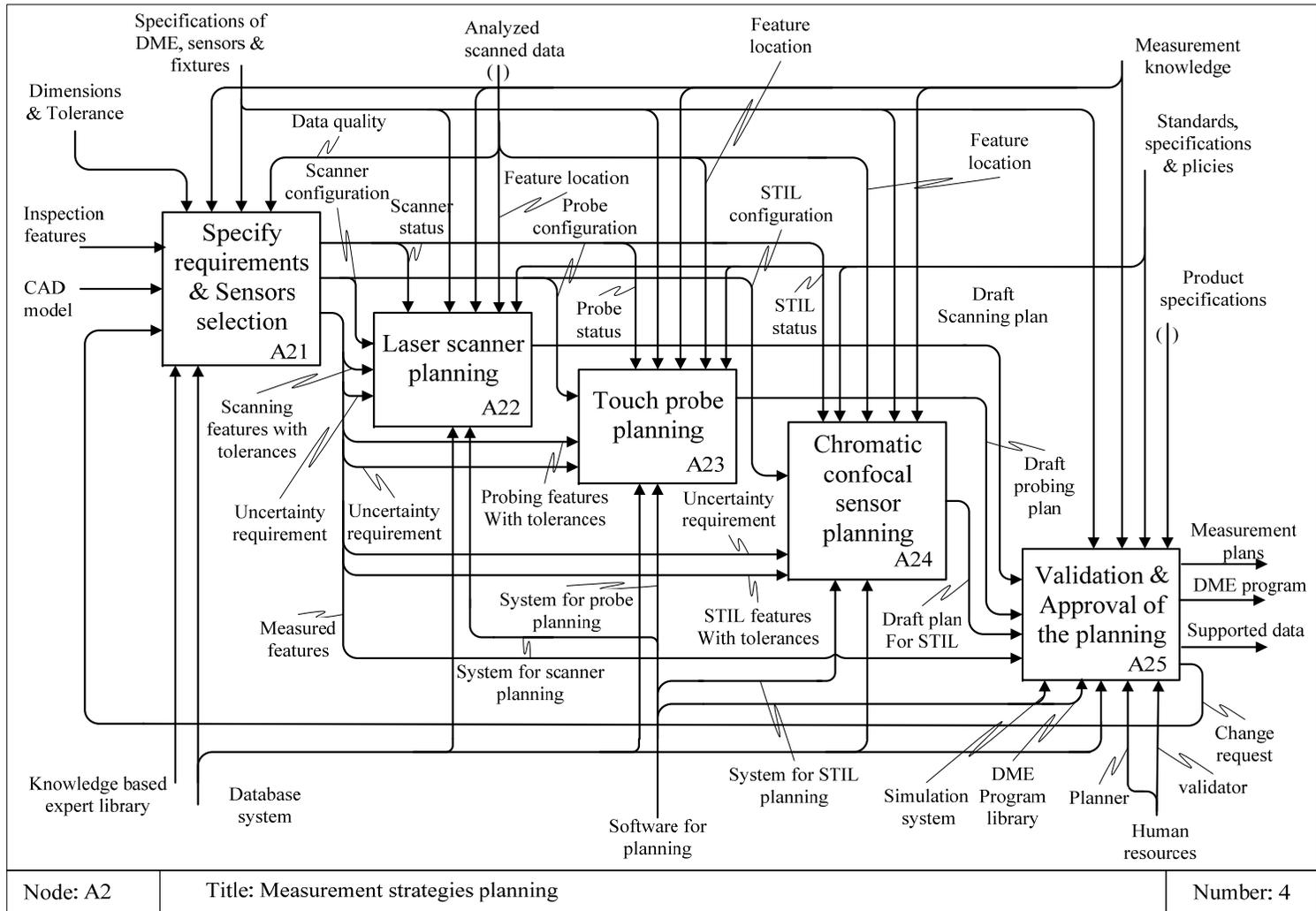


Figure 2-5: A2 diagram for measurement strategies planning

The activity A2 can be decomposed into 5 sub-activities.

(a) **A21:** Specify requirements and sensor selection.

The main purpose of the activity A21 is to specify the measurement requirements for each of the measurement features in the feature sequence generated in the activity A1. Based on the measurement requirements, the suitable sensors are selected to perform the digitization based on measurement knowledge. The measurement requirements are specified based on the feature types and the associated dimensions & tolerances. The measurement requirements could include the measurement uncertainties, fixtures, position and orientation of the workpiece and other measurement constraints. The sensor selection is determined based on both the attributes of the inspection features and the sensors. The sensors based on different work principles have their work ranges, merits and limitations all of which should be evaluated in the sensor selection phase. In competitive or cooperative configuration measurements, the same feature may require two or more sensors for more detailed acquisition.

If the CAD model is unavailable, an optical sensor is usually applied to acquire the initial data from the workpiece. The analyzed initial scanned data is then applied to help specifying the later measurement requirements and sensor selection. For example, if some regions of the initial scanned data appear with a low quality or incomplete and need more detailed measurements, the complementary measurement methods can be determined based on the analysis result of the initial scanned data.

(b) **A22:** Laser scanner planning.

The activities A22, A23 and A24 are used to plan the measurement processes for the three kinds of sensors respectively. In an actual measurement, perhaps, it is not necessary to apply all the three sensors. Hence, we use three parameters (scanner status, probe status and STIL status) to control whether a sensor is enabled or not. The activity A22 will be activated if the parameter “scanner status” is TRUE.

Because the laser scanner is limited by its own field of view, and occlusions, the pose (position and orientation) of the measurement features and the laser scanner should be properly determined. The continuous collision-free scanning trajectories should also be considered in this activity [MGC05, MTL06].

(c) **A23:** Touch probe planning.

Because of the simple implementation and great adaptability, the touch probes are widely used in general applications for surface digitization. In multisensor system, the touch probe can be applied for point collection from primitive surfaces or surfaces with known CAD model, or measurement with higher accuracy in special local regions to complement the optical sensing data. The activity A23 will be activated when the parameter “probe status” is set to TRUE.

To generate an appropriate touch probing measurement process, three main problems should be solved properly: determining the measurement points' number and placement; determining the probing orientation for each point; and generating a collision-free path for the whole measurement points [ALA99]. The locations of the probing features can be derived from the CAD model or from the scanned data. With the measurement uncertainty and accuracy requirements, the measurement points can be determined, and the probing process is planned accordingly.

(d) A24: Chromatic confocal sensor planning.

The STIL sensor based on chromatic confocal sensor planning is a 1D sensor used only in micro domain. In practice, the sensor is mounted on the CMM arm and the coordinates in x-, y- axes are recorded with the assistance of CMM. Basically, the measurement process for STIL sensor can be made based on the known specifications of the workpiece [SLY04]. It is nearly impossible to capture the data of a large area of a workpiece with only STIL sensor. In multisensor system, the multi-scales measurement can be achieved by integrating several sensors with different measurement scales. In the implemented system, the laser scanner is used to capture the complete characterization of the object. The STIL sensor can be guided by the scanned data to perform the local measurement.

We have to note that the gap between two connected sensing lines should be at least $4\mu\text{m}$ [Cos07]. However, because of the limited field view, the gap between the neighbor sensing lines are usually less than $4\mu\text{m}$. Therefore, the measurement path should be made according to figure 2-6 (2), which is different from the traditional path generating for macro measurement, as shown in figure 2-6 (1).

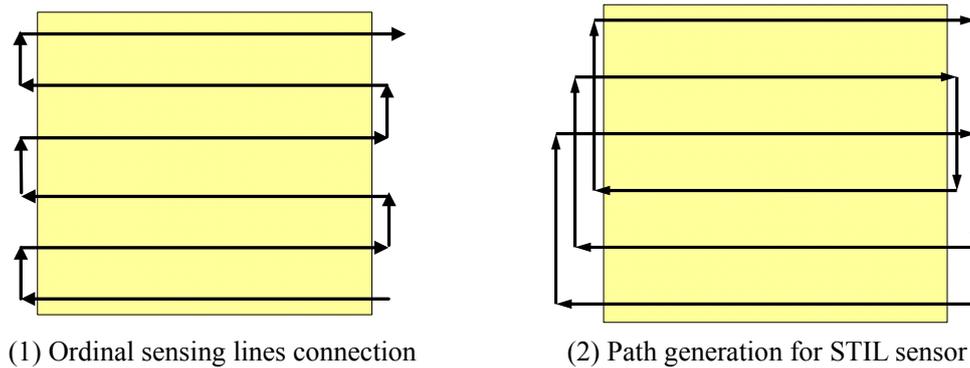


Figure 2-6: Two trajectory generation methods

(e) A25: Validation and approval of the planning.

The activity A25 is used to synthesize the process plans, validate and approve it for measurement execution. All the three sensors are driven by the CMM. The measurement sequence of the features is as: Laser scanning-> touch probe-> chromatic confocal sensor. If only one sensor is implemented, the plan can be directly driven for validation.

The validation of the measure process plans is based on simulation. The simulation software package is used or developed to simulate the trajectories of the sensors in measurement processes for collision checking and optimizing. The measurement process plans are finally approved by the process approver.

Another critical issue in this activity is generating the DME program. The DME program is compiled based on DMIS standard. The DMIS file can be identified by the execution unit and can be interpreted into DME codes based on I++DME standard to drive the CMM to perform the measurement.

(4). A4 diagram for discrete geometry processing

The result analysis activity A4 has to support the multisensor integration which requires the function to analyze the data from different sources and recover the geometric information. As discussed above, different applications focus on different information recovery, and this leads to very different data processing methods. In our system, we consider the common procedures for discrete geometry processing, which includes the discrete differential geometry attributes recovery, multidata registration and merging, shape recognition and segmentation. The corresponding decomposition of the A4 diagram is illustrated in figure 2-7.

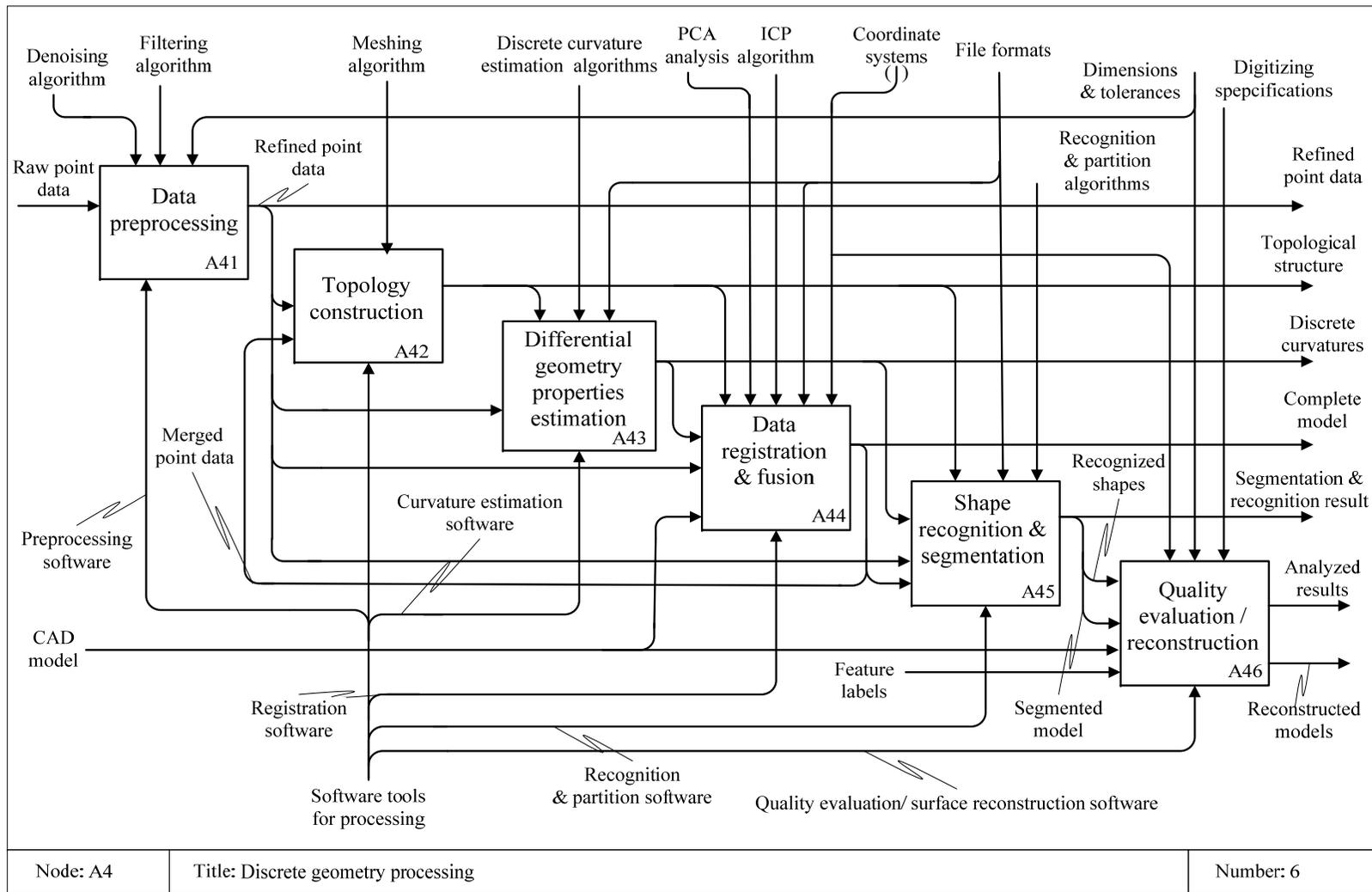


Figure 2-7: A4 diagram for discrete geometry processing

(a) A41: Data preprocessing.

The raw data acquired directly from the sensors are influenced by many factors, such as measurement environments, surface conditions, uncertainties and accuracies of the sensors, etc. The qualities of the raw data hence are not high and need to be improved for further processing by a preprocessing procedure.

The quality of point data [HDD92] can be evaluated by many indicators, such as the four indicators proposed by Lartigue, et al. [LCB02], which are: noise, density, completeness and accuracy. If the quality of the point data is not good enough, a preprocessing perhaps should be applied to refine the point data. The preprocessing procedures might enable denoising, filtering, complementing data, etc. However, after preprocessing, if the quality of the point set is still not good enough, the laser scanner guide measurement, rescanning, touch probing or chromatic confocal imaging, can be applied to acquire again the data with more details.

(b) A42: Topology construction.

The topological structure is always required by most computations on scattered point data, from where the neighborhood information for each point can be queried. Generally, there are three kinds of topological structures popular used in engineering applications and computer graphics, etc.: direct point based (e.g. k-nearest neighbors) approximated polyhedral surface based (e.g. polygon mesh), volume based (e.g. Voxel representation). Which kind of topological structure should be appropriate depends on the different applications. For general purposes, we prefer to use the polygon mesh to construct the topological structure of a point set.

Even after preprocessing the point data, the noise and other defects are usually unavoidable in measurement point data. The particular challenge for engineering data is how to construct the polygon mesh from these noisy, unorganized point sets. Several Voronoi/delaunay based methods have declared that they can provide robust meshing results for the noisy point data [ACT07, DG06, ST09, VHB08, XMQ04].

(c) A43: Differential geometry properties estimation.

Differential geometry attributes cover the tangent vector field (the first order), curvatures (the second order), and other attributes with higher orders. The differential geometry attributes record many shape information of the smooth or discrete geometries. Estimation of the differential geometry attributes is one of the most

common tools for shape modeling and understanding, especially for the analysis of point data without detailed knowledge, like in reverse engineering.

In engineering applications, frequently used differential geometry attributes are normal vectors and curvatures which are defined not at single isolated points, but cover local regions. The neighbor information around a given point is thus always necessary to evaluate differential geometry attributes. The differential geometry attributes suffer from various definitions for the discrete point data based on different neighbor regions and different methods are developed for the estimation purpose. Take the curvature estimation for example. The commonly used methods can be classified into three categories: surface fitting based methods, discretizing the formulae of continuous surface and tensor based methods. Further details work can be found in chapter 3.

(d) A44: Data registration and fusion.

Multidata registration and fusion is used to align or merge the point data acquired from different views of one sensor or from different sensors together in order to provide the complete information of the workpiece. It is thereby quite critical to achieve the multisensor based surface digitizing function.

If the correspondences among the multidata are known, the data can be aligned accordingly. The correspondence can be the relationships among the coordinate systems of the multidata, or the artificial markers. Otherwise, if the correspondences are unknown, the methods to recover the overlaps between each two point sets should be done and the point data can then be registered accordingly. The most widely used methods used here is the ICP (Iterative Closest Point) algorithm [BM92] and its variants. The ICP based methods do the registration in two steps: coarse registration and fine registration. However, the ICP algorithm fails to merge the point data with only partial overlaps. Therefore, for multisensor based systems, an improved method has to be developed to align the data with only partial overlaps. The detailed work can be found in chapter 4.

(e) A45: Shape recognition and segmentation.

The activity A45 addresses the discrete shape model recognition and segmentation. The point data is decomposed into a series of simplex, analytical regions with recognizable surface types for many engineering purposes. This information is quite useful for surface reconstruction in reverse engineering or for

actual features partition for tolerance evaluation in dimensional metrology. In laser scanner guide measurement, the initial scanned data should be processed to recognize the locations of the regions where need more detailed measurements.

The methods for shape type recognition are generally based on the discrete curvatures, such as the principal curvatures, Gaussian and mean curvatures, In these methods, a set of shape type categories are first predefined, and all the points of the discrete model are then classified into these shape categories according to their curvatures. According to different used surface indicators, different methods are discussed [BJ88, KD92].

Most of the existing segmentation methods can be classified into three categories: edge-based, region-based and hybrid-based. For any of these methods, the normal and curvature information are usually required. In our case, we first defined 10 basic surface types which can cover the most generally used surfaces in engineering applications, and then, the points are clustered accordingly. After the cluster refinement, a connected region generation method is applied to recover the connected regions in the discrete model. More details can be found in chapter 5.

(f) A46: Quality evaluation / reconstruction.

The activity A46 is a customized procedure to address different purposes. In our implementation, we consider the quality evaluation and model reconstruction purposes. The measured point data provide the representation of the physical workpiece. In model reconstruction applications, when the discrete model is properly decomposed, for each segmented region, the surface can be simply reconstructed since the surface type is also recognized. This is the same for quality control purpose. With the segmented results, the features associated to each tolerance can be queried easily. The evaluation of the actual tolerances and dimensions and comparison to the related nominal values are then performed.

2.2.3. Conclusion

The detailed functional analysis to achieve the multisensor based measurement system is described by IDEF0 diagrams in this section. The functional analysis is used to specify the requirements to achieve multisensor based surface digitizing and multidata processing from the viewpoint of a system developer.

The function model of the system defines the process, the information flow and the material flow by ICOM code. Basically, the system is decomposed into four functional activities: measurement data extraction and recognition; measurement strategies planning; measurement execution and discrete geometry processing. We focused on the measurement strategies planning and discrete geometry processing activities in more details. The function model provides us a clear view of the system, based on which the system can be developed to achieve the function requirements.

2.3. Ontology-based data modeling

2.3.1. Introduction to ontology modeling

(1). *Ontology definition and its benefits*

In the surface digitizing world, the quantity of generated knowledge and information is growing at an ever increasing pace. The growing information makes it important to facilitate ways of sharing and reusing the information among people, and computers automatically. Ontologies, introduced in Artificial Intelligence in the field of Knowledge Management, provide the means to structure and manage all this information and make it easily accessible. In philosophy, ontology is the study of the nature of being, existence or reality in general. However, the ontology concept used here is in information science and has the following definition: an ontology is a rigorous specification of a conceptualization for some knowledge domain of discourse – definitions of classes, relations, functions and other objects [Gru93]. In general, ontologies describe formally a domain of discourses clearly and unambiguously [Gru95].

An ontology embodies some sort of world view with respect to the given domain. The world view is often conceived as a set of terms (e.g. entities, attributes, and processes), their definitions and inter-relationships. Terms denote important concepts in the domain. This is referred to as a conceptualization. Recording such conceptualization with an ontology is referred to as ontology development [AV04].

Ontologies provide a shared understanding of a domain. Such a shared understanding is necessary to overcome inconsistencies in terminology. Such inconsistencies can be overcome by mapping the particular terminology to a shared

ontology or by direct mapping the particular ontologies. When the knowledge of a domain is represented in a declarative formalism, computers can semantically interoperate when it comes to processing, exchanging or presenting information. The motivation behind the development of ontology model for a particular domain falls into the following [AV04]:

- (a) Sharing a common understanding of the information in a knowledge domain;
- (b) Improving interoperability among applications that use the domain knowledge;
- (c) Making domain assumptions explicit so that applying changes becomes easier as these assumptions evolve;
- (d) Enabling reuse of the domain knowledge by multiple applications.

In short, Ontologies should have the ability to standardize the definition of real-world concepts and support semantic interoperability. A common ontology is a guarantee of consistency, but not completeness, with respect to queries and assertions using the vocabulary defined in the ontology [Gru92, Gru95].

(2). *Ontology development*

Ontology development doesn't stand for a new approach to knowledge representation and management. Ontologies are developed to facilitate knowledge sharing and understanding. In ontology development, an ontology is usually represented by classes, properties (or slots) and individuals (or instances). Classes describe all those features that encapsulate the concepts falling in a knowledge domain. Individuals or instances represent the objects in the target domain. A class can be viewed as a set of individuals whose types are characterized by the similar concept. Properties define the relations or associations between classes [OWL].

According to [AV04], three phases are identified in an ontology development in general:

- (a) Knowledge acquisition. This involves the extraction of knowledge from unstructured or semi-structured sources of information and also the acquisition of expert knowledge from human domain experts.
- (b) Knowledge representation. The acquired domain knowledge is recorded using an ontology language and a platform to build the ontology of the domain.

(c) Knowledge use and reuse. The structured knowledge stored in an ontology model should be able to be accessed and reasoned in order to draw some conclusions.

Nowadays, there are several languages and tools for defining ontologies in an arbitrary domain, such as RDF (Resource Description Framework) and the RDF schema [Mil00], UML (Unified Modeling Language) [Uml], OWL (Web Ontology Language) [OWL], IDEF5 [IDEF5], etc. In industrial production applications, the data models of STEP defined by its own modeling language EXPRESS (ISO10303-11) are already structured, which can be viewed as ontology-like for the presentation of product model information.

OWL is the prevailing standard today in ontology design and provides a much richer vocabulary than other languages. Furthermore, it provides support for reasoning and has a formal semantics. OWL ontologies can be categorized into OWL-Lite, OWL-DL and OWL-Full [HKR04].

One of the platforms used for ontology development is Protégé, which is an open source platform developed by Stanford University to construct ontologies. Protégé implements tools that support the ontology creation, visualization, manipulation and reasoning. It can be customized to provide user-friendly interface for specific domain [Stal].

(3). *Related works on ontology development*

Since ontologies were first introduced in AI (Artificial Intelligence) for knowledge management [Gru93, Gru95], many related works on ontology development have been made in different domains.

Ontologies can be classified into two types: domain ontology and upper ontology. A domain ontology (domain-specific ontology) specifies a domain knowledge. It represents the particular meanings of terms applied in a specific domain. Patil [PD05] proposed an ontology approach to formalize product semantics into a Product Semantic Representation Language (PSRL) for data exchange in a PLM context. The ontologies to support process integration in enterprise engineering are discussed in [GAF00]. An ontology-based method for information management in engineering design process is presented by Brendt et al. [BMT06]. In AimAtShape project [Aim], Andersen built an ontology of CAD model [AV04].

An upper ontology is a model of the common objects that are used across a wide

range of ontologies. There are several standard upper ontologies available for use, like Dublin Core (ISO 15836) [Dub], SUMO [Sumo], GFO [Gfo], etc.

There are little researches on ontology development in the specific domain of multisensor based surface digitizing. Considering the complexity of information in such a domain, it is necessary to manage the related information with consistent and unambiguous representation.

2.3.2. Data modeling

From the view point of data processing, the measurement system is a function activity to transform the specifications data of a product into the digitized point data with recovered geometric data that reflect the product specifications. According to the functional analysis in section 2.2, the process of multisensor based surface digitizing can be decomposed into four main procedures: data extraction and recognition, measurement strategies planning, measurement execution and discrete geometry processing. In the different procedures, the required information and resources are different, which provides an intuitive way to organize the knowledge. According to the different stages, the data can be categorized into four basic classes: Product model, measurement strategies planning model, execution model and result analysis model. Each class can be decomposed into more details to construct the hierarchical structure of the data model. This data analysis is based on standard AP219 (Dimensional Inspection Information Exchange) and ISO/TC 213 (Dimensional and Geometrical Product Specification and Verification) which are used to specify the product data in design and measurement.

However, it is difficult to construct the data model directly on Protégé-OWL platform since the information and their relationship net in the multisensor based domain are complex. The tool used here for the data modeling is an EXPRESS-G like formalism. EXPRESS-G is a standard graphical representation for data modeling based on EXPRESS modeling language. EXPRESS-G diagrams show relationships and data structures in clear and more understandable way.

(1). *Product model*

The product model describes the design specifications of the workpiece to be digitized. The design data are usually generated at a product design stage. They can be

embedded in the CAD model of the workpiece and other related documents. However, the CAD model or specification documents are not always available in some applications, like reverse engineering. The product data can be recognized from the physical workpiece by human brain and roughly recorded in documentations. The intuitive product information can be confirmed by analyzing the acquired data initially scanned over the workpiece.

The product information specified at a design stage needs to be interpreted to satisfy the requirements at the measurement stage. In other words, the product data should be decomposed and then the measurement elements can be recognized and extracted. According to the standard STEP part 42 (Geometric and Topological Representation), the product data consists in three basic categories: (a) geometry schema which defined the geometric entities to compose the product; (b) topology schema which specifies the topological relationships within geometric entities; and (c) property schema which describes the attributes the geometry and topology schemas have. The first two schemes provide the basic structure necessary for specifying the representation of a CAD model.

However, the design and manufacturing specifications, such as the dimensions and tolerances associated to the geometric entities and topological elements are not included in the STEP Part 42. These specifications are necessary for the measurement to specify the measurement requirements, such as uncertainty and accuracy should be satisfied during the measurement. It is important to add the property schema to specify them. Moreover, the support data, such as the software package, the workpiece administrative data (workpiece id, version, etc.), are also recorded as property data.

The schemes refer to each other. The complete product model can be constructed accordingly. Figure 2-8 illustrates the general view of the data structures in product model.

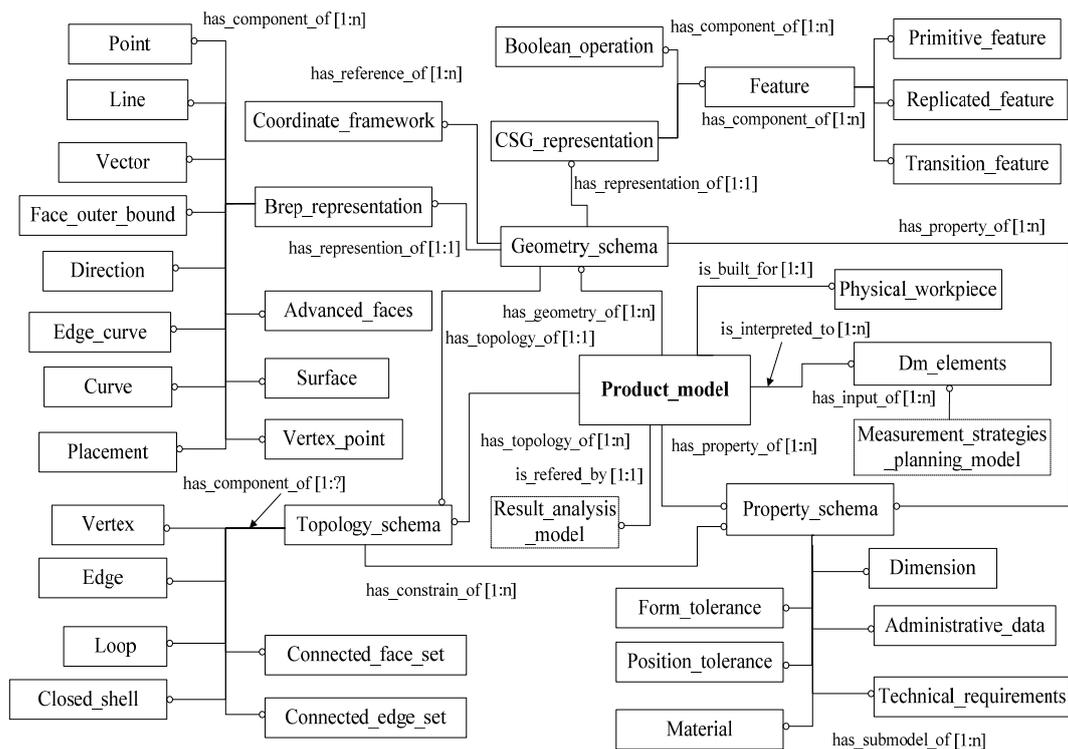


Figure 2-8: Product model

(2). *Measurement strategies planning model (MSP model)*

The MSP model specifies the related information or data required to generate the measurement process planning. In general, it contains mainly two sub groups. One group is related to the knowledge and algorithms to produce the measurement process plans. The other group describes the resources of the DME (Dimensional Measurement Equipments) and software tools. The planning results, which contain the measurement process specification documents, the DME programs and the support data, are related to the MSP model. The MSP model refers to the product model to obtain the decomposed features associated with dimensions and tolerances and the measurement requirements. Figure 2-9 gives a schematic view of the MSP model.

The information in the first group which specifies the knowledge and algorithm to generate the measurement process plans is represented as the control information in IDEF0 diagrams. The knowledge included here is redundant, which could contain the knowledge to select the suitable sensors for each measurement elements, the algorithms to generate the sensing positions and orientation, the algorithm to connect the adjacent sensing paths for the global optimized, collision-free trajectories generation, etc. In order to plan the measurement processes properly for a special

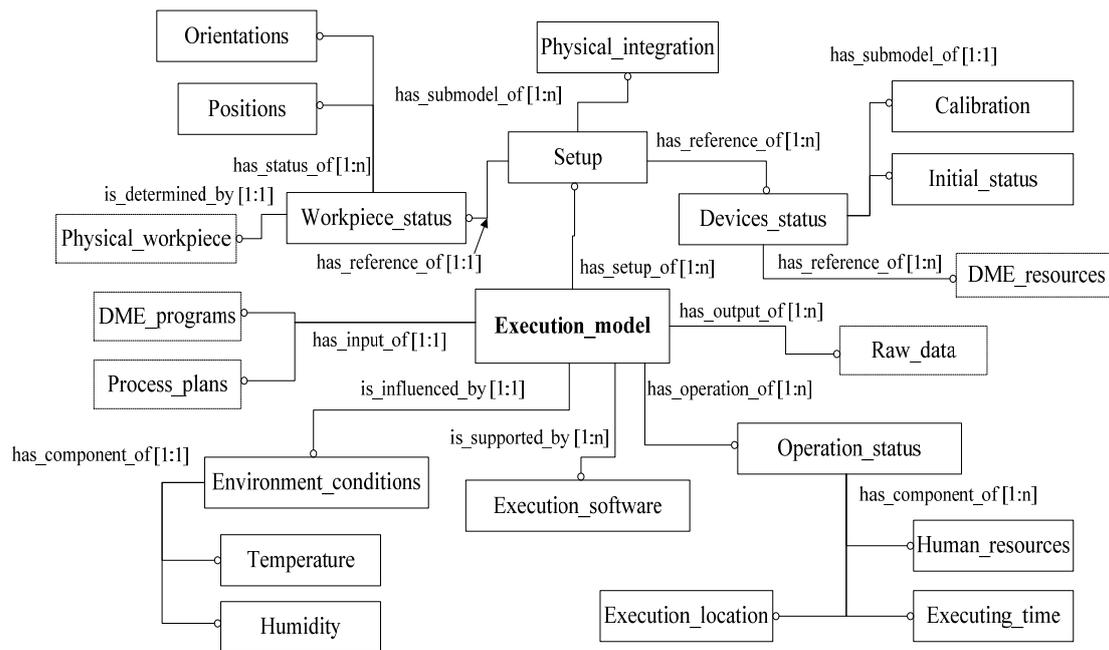


Figure 2-10: Execution model

(4). *Result analysis model*

The last group of data is used to record the information related to the result analysis. The result analysis model can be decomposed into four clusters: raw data, final result data, intermediate and supported data.

Raw data is the initial point data acquired from the sensors directly, which are viewed as the input data to be processed. Considering that multiple sensors are used in the surface digitizing procedures, the raw data also consist in various data files with different formats, containing different information, In order to recover the explicit geometric information underlying them, the data processing procedures are always required.

Final result data relates to the data which record the final processed results. For different applications, different geometric data should be recovered. In dimensional metrology, the actual information about the actual features, dimensions and tolerances should be recovered from the raw data for quality control. In reverse engineering, the surfaces' information is more considered for shape modeling.

In general, it is difficult to extract the geometric information from the raw data directly. Some intermediate procedures are required for assistances. The intermediate data group is then used to collect the results generated in these intermediate procedures. For different applications, different intermediate procedures may be

implemented. However, there are still some common procedures which are always required by different purposes. In our data modeling, we consider the intermediate data in the following procedures.

(a) Data preprocessing. The raw data can't be directly used for further processing in general because of their bad qualities. The data preprocessing aims at improving the qualities of the raw data, like reducing the noise, filtering the redundant points, removing the outliers, etc. The intermediate data in this phase are the refined data.

(b) Topological structure construction. This is performed to build the neighbor relationships within the point data. Different methods will generate different topological structures. For example, the polyhedral surface approximation will generate the polygon mesh as the intermediate data in this phase, while voxel based method will generate voxel-volume as the intermediate data

(c) Differential geometry properties estimation. The differential geometry properties include the normal vector, curvatures, and the higher order of geometry. They record the important properties of shapes and can be used for geometry processing and recovering. The intermediate data in this phase are the estimating results of the differential geometry properties.

(d) Data registration and fusion. The point data acquired from different views of single sensor and/or from different sensors capture the different information of the measured workpiece. In order to represent the workpiece completely, these data should be registered and merged together. The intermediate data in this phase is the generated complete shape.

(e) Shape recognition and segmentation. In this stage, the complete shape is decomposed into pieces of simple, analyzable regions for surface recognition. The segmented regions could be used to estimate the associated dimensions and tolerances, or used for surface reconstruction. The intermediate data in this phase are the segmented data and the recognized data.

The supported data are the resources which are required for result analysis. The supported data includes the algorithms used to obtain the intermediate data and the final result data, etc.

The data related to the result analysis model are illustrated in figure 2-11.

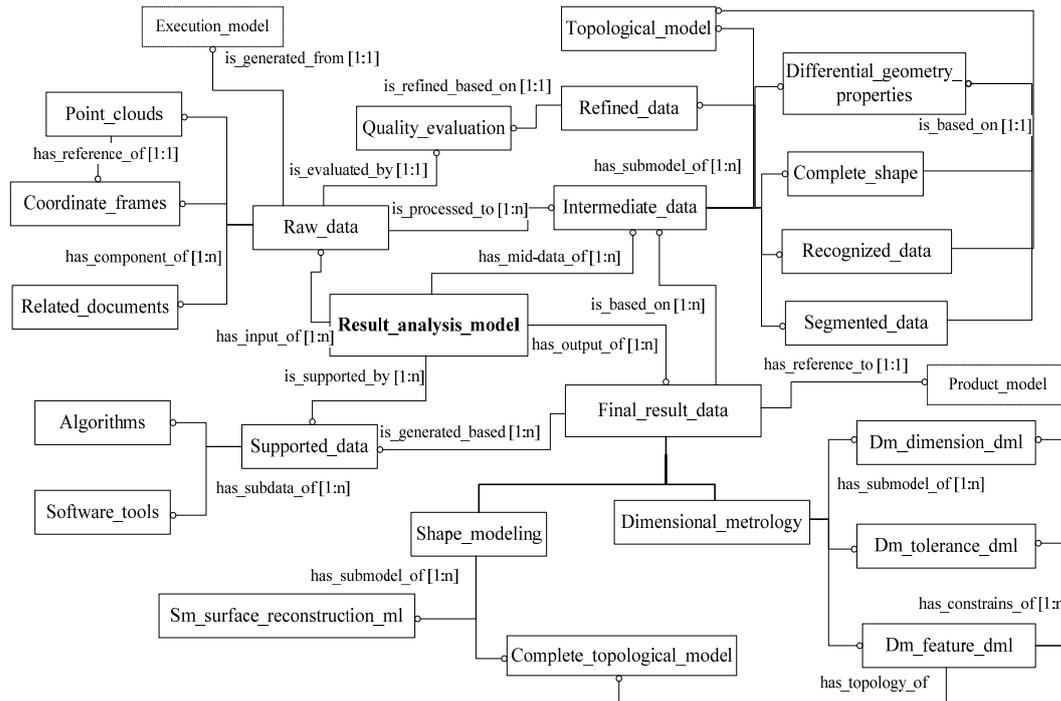


Figure 2-11: Result analysis model

The standard STEP AP 219 has already defined some data structures to specify information requirements to manage dimensional inspection, which include administrating, planning, and executing dimensional inspection as well as analyzing and archiving the results. However, the defined data structures only support the final requirements for dimensions & tolerances estimation in dimensional metrology. The definitions for the data in the intermediate procedures of the data processing are not considered. But we define the complete data model for the general purposes of surface digitizing.

2.3.3. Ontology development

The four groups of the data model specified for multisensor based surface digitizing domain contain knowledge and data in much more details. The data model would be very complex when trying to describe the detailed data. Moreover, the graphic based EXPRESS-G language can not support the information reasoning like querying, editing, inferring, etc. Fortunately, the ontology development language OWL provides an effective way to organize the data in a given domain. The OWL, built on XML (Extensible Markup Language), adheres to an object-oriented philosophy.

The commonly used platform to build the ontology model using OWL is Protégé-OWL [Stal]. The Protégé -OWL platform organizes the data using hierarchical structure. The classes, properties, individuals and the relationships or constraints within them can be clearly defined on the Protégé-OWL. Moreover, if the ontologies are properly defined, the platform provides tools for information reasoning. Based on the above data analysis, we can build the ontologies of multisensor based surface digitizing. The figure 2-12 shows the top hierarchies of the four data groups which are built on Protégé-OWL platform and the figure 2-13 highlights the ontology of Kreon Zephyr KZ 25 laser scanner.



Figure 2-12: Top hierarchical structure of ontologies built on Protégé-OWL

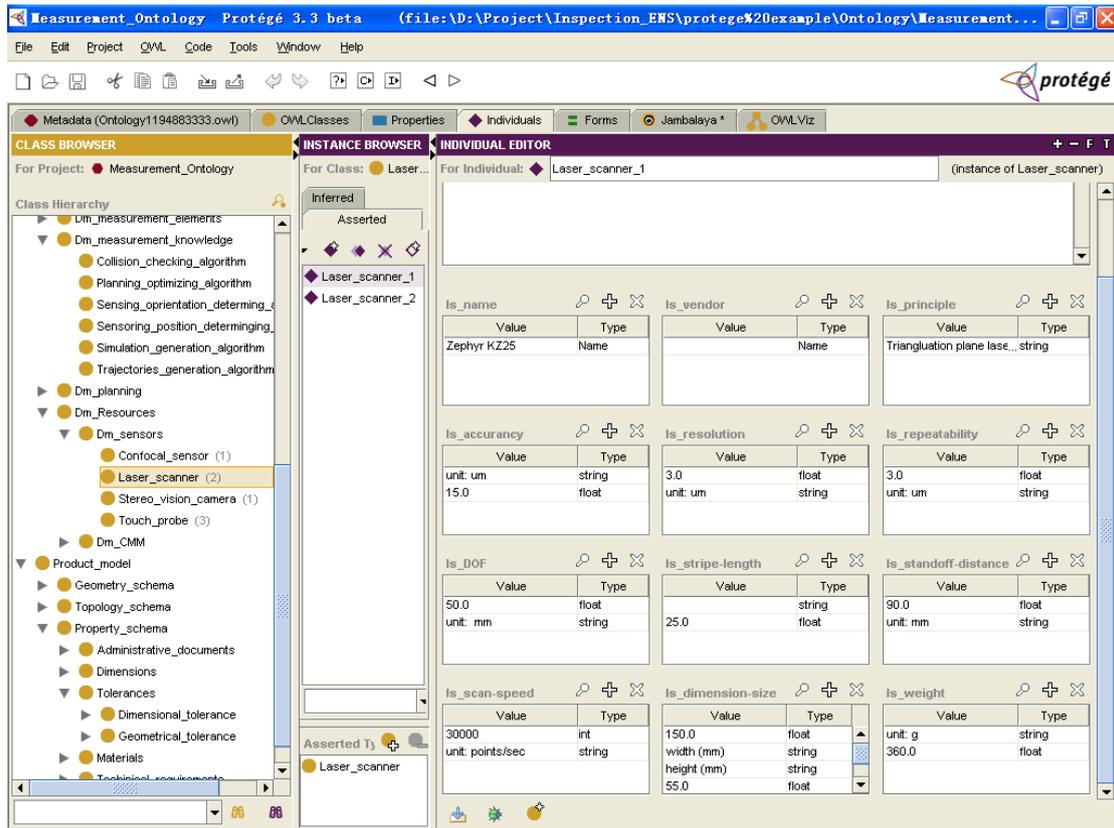


Figure 2-13: Ontology editing

2.3.4. Conclusion

This section specifies an ontology-based approach to manage the data for multisensor integration. Using a graphic method, the information entities, and their properties are organized clearly. The existing data in the multisensor integration domain can be decomposed into four basic groups according to the different measurement stages: product model recording the product knowledge, measurement strategies planning model is used to organize the data related to the measurement planning strategies, execution model constructing the data for measurement execution and result analysis model which is used to organize the data used in result analysis stage. The four data groups refer to each other and the relationships among them are specified. Finally, we develop the complete ontologies for multisensor integration using Protégé-OWL based on the above data analysis.

2.4. Conclusion

This chapter specifies the detailed function models and data models for multisensor integration.

The detailed hierarchical function models of the multisensor integration system developed in our laboratory are defined using IDEF0 methodology. The system is modeled with four basic activities: Data extraction and recognition that specifies the functions to interpret the workpiece specifications for measurement purpose; measurement strategies planning which specifies the activities for measurement process planning with multisensor integration; measurement execution that model the function requirements for measurement execution and discrete geometry processing which defines the functions to recover and process the discrete geometries underlying the acquired point data. The information flow and resources of each activity are defined in the function models.

The concepts of ontology in information science are briefly introduced and an ontology based approach is used to build the structured data model of the multisensor integration system in coordinate metrology using Protégé-OWL. The data and knowledge in the system are organized based on the different measurement stages which are defined in function modeling. Accordingly, the whole data model of the system comprises four models: product model which is used to define the data related to the specifications of the workpiece; measurement strategies planning model that specifies the data related to the measurement process generation; measurement execution model which is used to collect the data required for measurement execution and results analysis model that records the data for result analysis. The inter relationships among them are also specified.

The function models and data models provide clear ways to understand the multisensor integration for coordinate metrology. They can be guidelines for the system development and implementation.



Chapter 3

Discrete Differential Geometry

3.1. Introduction

In the previous chapter, function and data modeling highlighted the subsequent uses of 3D point data processing. Digitizing and geometry processing are increasingly being interrelated in a variety of engineering applications and have spurred extensive research.

Estimation of the discrete differential geometry properties, such as normal vector, curvatures, etc. are often required in many applications when the surface is defined by a set of discrete points rather than by mathematical formulae [MW00]. The estimation of the discrete differential geometry properties is an essential task for discrete shape modeling and processing. Reliable estimation of discrete curvatures is also the basis of discrete shape registration and segmentation.

In discrete differential geometry, the topological structure of 3D point data should be constructed first, from which the neighborhood of each point can be queried. The most used topological structure is the polygon mesh.

In this chapter, we propose methods to estimate normal and curvatures of discrete shapes. The normal vector at each vertex is estimated from the normal vectors of the triangle facets in its neighbor regions. A tensor based method is used to estimate the principal curvatures of discrete shapes. A comparative analysis of different methods for discrete curvature estimation is presented. We introduce also two surface descriptors, shape index and curvedness both of which are derived from the principal curvatures.

This chapter is organized as follows:

Section 3.2 introduces the methods to generate the polygon mesh from the discrete point data. Section 3.3 overviews different methods for normal and curvatures estimation and introduces a new estimation method. Section 3.4 presents a comparative analysis and testing results. Section 3.5 is the conclusion.

3.2. Polyhedral surface approximation

3.2.1. Approximation method and mesh data generation

(1). *Mesh generation*

The scattered point sets are usually approximated to proper watertight polyhedral surfaces in order to build the topology structures [HD06]. The polyhedral surfaces are composed of polygon meshes from which the local neighbor information of each vertex can be found. The local neighbor information is always required by most computations on the scattered point sets. There have been considerable techniques applied to this problem in computational geometry, medical imaging, cartography, etc. [AB98, BC02, CVY02, Dey07]. The particular challenge in engineering is that the scattered point sets generated by laser scanner or other measurement sensors are always noisy, unorganized, or incomplete, most of which are not considered by the mentioned methods. However, the algorithms for polygon mesh generation are beyond our research works. We only consider the performances of the mesh generation algorithms for noisy point sets.

Many authors or software packages developers declared that their algorithms could construct the satisfying mesh from unorganized point sets, even in presence of noise, redundant or incomplete data [ACT07, DG06, ST09, VHB08, XMQ04]. The method developed by Alliez, et al. [ACT07], denoted as Alliez method in the following, is one of the most popular methods for surface reconstruction of noisy defective point samples. Hence, we developed this method to approximate polyhedral mesh in our work. The Alliez method is a Delaunay/Voronoi diagram based and the general procedures can be described briefly in two main steps.

Firstly, if no normal information is provided in the input noisy point set, a Voronoi-PCA based approach is performed to estimate the unoriented normal information. For this purpose, the Voronoi diagram and the Delaunay balls [AB98] of the input point set are first computed. And then, a series of the covariance matrices over a union of the Voronoi cells of a given point are built, and the matrix with maximum anisotropy is selected. The eigenvalues and eigenvectors of the selected covariance matrix are estimated. The eigenvector corresponding to the largest

eigenvalue is a good approximation of the normal vector at the given point.

Secondly, an implicit function is computed via generalized eigenvalue problem so as to make the gradient of the function best fit the normal directions. [ACT07] uses the energy function called anisotropic dirichlet energy to measure the alignment of the gradient of the implicit function and the normal directions. The implicit function provides a best-fitting iso-surface reconstruction. With the implicit function optimization and the isotropic Laplace operator, the method provides resilience to the noise, adjustable fitting to the data and controllable smoothness of the reconstructed surface.

In practices, we implemented the Alliez method with the help of the CGAL libraries [CGAL]. Figure 3-1 shows some examples of the final results of the polygon mesh generation. In the example of car case, the point data is acquired by scanning over the car model using Kreon scanner, which is noisy, defective and redundant. The other point set are downloaded from the website of Aim-at-shape [Aim]. The Alliez method can also generate reasonable topological results.

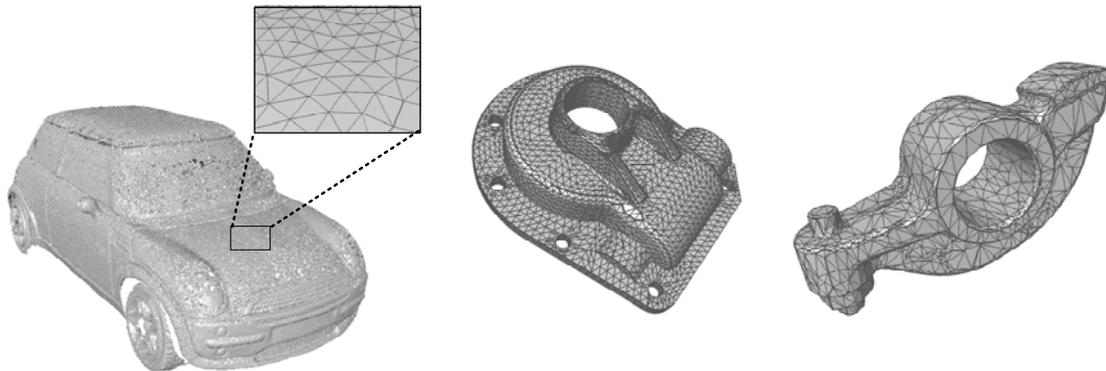


Figure 3-1: Examples of the triangle mesh generation

(2). *Data structure for mesh representation*

A proper data structure for mesh representation plays a critical role in geometry processing on the polyhedral surfaces. Because of the absence of the adjacent relationships, the face-based representation by which a polygon mesh is represented only as a shared list of vertices and a list of faces storing the pointers to the vertices demonstrates its inconvenience for geometry computing. Instead, as mentioned in [Ket99], the boundary representation is given increasing considerations. A boundary representation (B-rep) of a polyhedral surface consists of a set of vertices, a set of edges, a set of facets and incidence relations on them [Ket99]. There are various types

of boundary representations, such as Winged-edge, Quad-edge, Half-edge, etc. A good survey of data representations for polygon mesh can be found in [Ket99].

The Half-edge Data Structure (HDS) is a slightly more sophisticated B-rep, which allows all the queries of the vertices, edges, facets and incidence relations can be performed in a constant time and the adjacent relations don't change the size of the representation [McG00]. These properties make the HDS be an excellent choice for mesh representation as we do in our application. The HDS is only capable of manifold surface. For polygon mesh, this means that every edge is bordered exactly by two faces.

As the name implies, the HDS splits each edge into two oriented half-edges having two opposite directions. The mutual relation between the two half-edges is added for convenient query. A half-edge is thus considered belonging to exact one triangle. The list half-edges of one triangle can be oriented as clockwise or counter-clockwise around the facet as long as the convention is used throughout. In our works, the convention conforms to counter-clockwise. Figure 3-2 shows a schematic diagram of the HDS for triangle mesh representation.

The HDS provides an efficient way for mesh representation of an oriented manifold polygon surface and also provides a convenient way to query the elements (vertices, edges, facets). It is sufficient for modeling the topology structure of commonly-used surfaces in engineering even in the presence of loops and multi-edges [Ket99]. The data structure is built with the help of the libraries of CGAL [CGAL] in our research work.

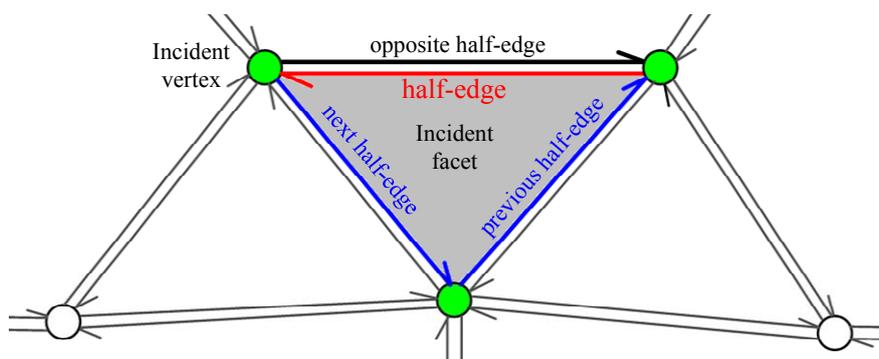


Figure 3-2: Schematic diagram of the half-edge data structure

3.2.2. File formats for mesh data representation and exchange

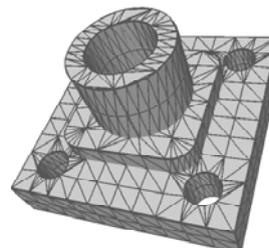
The generated polyhedral mesh data need to be exported with specified file formats which can be recognized by the downstream applications. There are many file formats widely used to store and exchange the mesh data in computer graphics and engineering applications, such as VRML (Virtual Reality Modeling Language) with an extension “.wrl”, PLY (Polygon File Format or Stanford Triangle Format) with an extension “.ply”, etc. Among them, STL and OFF files are two of the file formats which are more popular.

(1). *STL file format (.stl)*

STL (Stereo Lithography or Standard Triangulation Language) is a widely used file format to specify the triangle mesh model in rapid prototyping, computer-aided manufacturing, or other engineering applications. Typically, the STL file is saved with the extension “.stl”. The file format is supported by many CAD software packages, such as CATIA, etc. The STL format specifies both the ASCII and Binary representations [Bru93, Stla09]. The binary STL files are more compact; however, the ASCII STL files are more comprehensible by the human. Therefore, we only consider the ASCII STL formats in the following sections.

A STL file is a facet based representation of the raw unstructured triangulated surfaces or solid entities. A STL file consists of a list of triangle facet data. Each triangle facet is uniquely identified by a unit normal vector and by three vertices. The normal vector points to the exterior of the model and the three vertices are given by counter-clockwise direction. The normal and each vertex are specified by 3D coordinates. Figure 3-3 (1) indicates the representation of a complete triangle facet unit extracted from a typical STL file. The complete STL model is also shown in figure 3-3 (2).

```
facet normal 0.000000e+000 -9.957342e-001 9.226814e-002
  outer loop
    vertex 4.500000e+001 -2.250000e+001 2.755364e-015
    vertex 6.000000e+001 -2.250000e+001 2.755364e-015
    vertex 4.500000e+001 -2.211690e+001 4.134364e+000
  endloop
endfacet
```



(1) The complete facet definition in STL file

(2) STL model

Figure 3-3: STL file format and a typical STL model

The STL format is easy to be generated. However, there are also some drawbacks of the STL files:

(a) Lack of topological information. The STL file is only a “bucket of facets” [RW92]. The connectivity entities, such as lines, curves, etc. and the attributes, such as layers, colors, etc. are ignored in the STL files. Moreover, the triangle facets stored in STL files are independent and there is no relationship with each other. This leads to great amount of redundant data directly. The topological information is important for the further curvature estimation. Hence, the topological structure of the discrete model represented in STL format should be first constructed before any further processing.

(b) STL errors. As other triangle mesh carriers, STL data sometimes contain errors and ill-formatted values. According to [Wab99, SP08, SPC04], the common errors are manifold, vertex-to-vertex rule error, missing data, etc.

Therefore, when a STL file is encountered, the model healing and topology construction should be done first to build the neighbor information in the discrete model for the further use.

(2). *OFF file format (.off)*

Object File Format (OFF) is used to represent the geometry of a model by specifying the polygons of the model’s surface [Ros86]. The polygon can have any number of vertices, like triangle facet with three vertices, quadrangular facet with four vertices, etc. The OFF files are generally with the extension “.off”. Some other extensions are also possible based on the different purposes of the users. OFF data is an ASCII data (There is also a binary version. However, the binary version is not as popular as the ASCII).

The OFF file begins with the ASCII keywords “OFF”. The next line states the number of vertices, the number of facets and the number of edges. However, the number of edges can be safely ignored. And then the coordinates (x , y , z) of each vertex are listed one per line. After the list of vertices, the facets list is followed with one facet per line. For each facet line, the number of included vertices is the first listed followed by the line indices of the vertices. Figure 3-4 shows a simple example of the OFF file format of a simple object.

Besides the information of the vertices and facets, other attributes of these elements, such as color, textures, etc., can also be specified in the OFF files, which is

the main difference from the STL file format. Moreover, considering the same object represented in both of the STL file and the OFF file, the OFF file always has a smaller file size than the STL file. However, the normal information is not included in the OFF format.

In our method, the recognizable file formats of the input discrete model are the STL and OFF files. The generated mesh surfaces from the clouds of points should be saved as one of them.

```

OFF
1176 2356 0
-5.3171 -7.39159 78.9116
-8.10315 -13.0152 75.0302
-10.4236 -16.5081 76.055
-13.2511 -17.533 72.5621
... ..
3 236 0 244
3 0 218 244
3 244 218 1
3 218 221 1
... ..

```

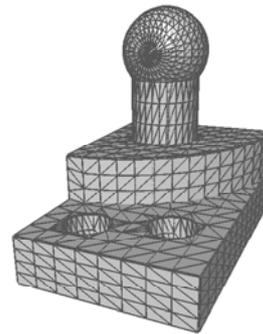


Figure 3-4: OFF file format and an example of OFF model

3.3. Differential geometry properties Estimation

Differential geometry properties, mainly contains normal vector, curvatures, etc., which specify the local attributes of the first order and second order derivatives of a curve or surface. Differential geometry properties calculation is an important goal in differential geometry both for continuous and discrete curves and surfaces. It is the most common tools for shape understanding and modeling especially for discrete models. In this section, firstly, we will introduce the classical differential geometry on continuous surface, followed by the surveys of the commonly used estimation methods for discrete models. Then, an improved method to estimate the principal curvatures on polygon mesh surfaces is proposed.

3.3.1. Differential geometry of smooth surface

We first recall some classical definitions and notations of differential geometry in the case that the surface is smooth. A good review can be found in [DoC76, Sri08, One97].

Let Σ denote an orientable surface embedded in the three dimensional Euclidean space \mathbb{R}^3 , such that the surface Σ is described by an arbitrary parameterization of two variables $\mathbf{X}(u, v)$. For an arbitrary point on the surface, $p \in \Sigma$, the unit normal vector at p pointing outside of some volume bounded by Σ , is referred as $\vec{n}(p)$, which is given by:

$$\vec{n}(p) = \frac{\mathbf{X}_u \times \mathbf{X}_v}{\|\mathbf{X}_u \times \mathbf{X}_v\|} \quad (3-1)$$

Where, the subscripts indicate partial derivatives and \times denote the cross product.

The first and second fundamental forms of Σ are defined as:

$$I(u, v, du, dv) = d\mathbf{X} \cdot d\mathbf{X} = \begin{bmatrix} du & dv \end{bmatrix} \Pi \begin{bmatrix} du \\ dv \end{bmatrix} \quad (3-2)$$

$$II(u, v, du, dv) = -d\mathbf{X} \cdot d\vec{n}(p) = \begin{bmatrix} du & dv \end{bmatrix} \Lambda \begin{bmatrix} du \\ dv \end{bmatrix} \quad (3-3)$$

Where,

$$\Pi = \begin{bmatrix} \mathbf{X}_u \cdot \mathbf{X}_u & \mathbf{X}_u \cdot \mathbf{X}_v \\ \mathbf{X}_u \cdot \mathbf{X}_v & \mathbf{X}_v \cdot \mathbf{X}_v \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix}, \quad (3-4)$$

$$\Lambda = \begin{bmatrix} \mathbf{X}_{uu} \cdot \vec{n}(p) & \mathbf{X}_{uv} \cdot \vec{n}(p) \\ \mathbf{X}_{uv} \cdot \vec{n}(p) & \mathbf{X}_{vv} \cdot \vec{n}(p) \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$$

E, F, G are the coefficients of the first fundamental form and L, M, N are the coefficients of the second fundamental form.

The first fundamental form measures the small amount of movement on the surface for a given small movement in the space of the parameters (u, v) . It is invariant to surface parameterization changes and to translations and rotations of the surface. It doesn't depend on the way the surface is embedded in 3D space and is therefore an intrinsic property of the surface. The second fundamental form, which measures the change in the unit normal for a movement in parameter space, depends on the embedding of Σ and is thus an extrinsic property of the surface.

Curvatures are defined to measure the local bending of the surface. Take the given point p for example, for each unit direction \vec{u} on its tangent plane $T_2(p)$, the normal curvature $\kappa_n(\vec{u})$ is defined as the curvature of the curve that belongs to both the surface itself and a perpendicular plane containing both $\vec{n}(p)$ and \vec{u} . Formally,

the normal curvature is defined by the following equation [One97]:

$$\kappa_n(\vec{u}) = S_p(\vec{u}) \cdot \vec{u} \quad (3-5)$$

Where, $S_p(\vec{u})$ denotes the shape operator at point p along the direction \vec{u} . The shape operator which is also known as Weingarten endomorphism is defined as the derivate of $\vec{n}(p)$ with the tangent direction \vec{u} . Mathematically, it's defined as:

$$S_p(\vec{u}) = -\nabla_{\vec{u}} \vec{n}(p) \quad (3-6)$$

Where, $\nabla_{\vec{u}} \vec{n}(p)$ indicates the gradients of $\vec{n}(p)$ along \vec{u} at point p .

The shape operator can be shown to be symmetric. The eigenvalues and eigenvectors of the shape operator are respectively called the principal curvatures and the principal directions. In other words, both the principal curvatures and the principal directions can be recovered from the shape operator matrix.

From the previous analysis, we will see that the principal curvatures are related with the normal curvature. The relationship can be represented as the famous Euler's theorem:

$$\kappa_n(\vec{u}) = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta \quad (3-7)$$

Where, κ_1 and κ_2 represent the maximum and the minimum principal curvatures respectively. θ is the angle between \vec{u} and the principal direction associated with the maximum principal curvature. If \vec{T}_1 and \vec{T}_2 denote the principal directions respectively, we can get the following equation:

$$\vec{u} = (\cos \theta) \cdot \vec{T}_1 + (\sin \theta) \cdot \vec{T}_2 \quad (3-8)$$

Figure 3-5 shows the normal direction and the directions of these curvatures at the point p .

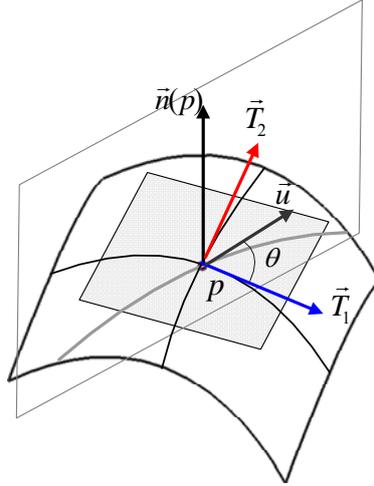


Figure 3-5: Normal and principal curvature directions of a smooth surface at point p

Considering the fact that the principal curvatures and the principal directions are the eigenvalues and the eigenvectors of the shape operator, the shape operator at p , S_p can be represented as follows:

$$S_p = \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \quad (3-9)$$

The two principal directions, $\{\vec{T}_1, \vec{T}_2\}$ define an orthonormal basis of the tangent space to Σ at p .

The shape operator defines a 2×2 symmetric bilinear form of the surface Σ at p associated with the second fundamental form. As we see, the bilinear form needs to be extended to 3×3 symmetric form [CM03, Tau95]. We do so by adding the normal vector $\vec{n}(p)$ to the basis $\{\vec{T}_1, \vec{T}_2\}$. Then, we use the orthonormal basis $\{\vec{n}(p), \vec{T}_1, \vec{T}_2\}$ to describe the three-dimensional space, which changes from point to point, similar as Taubin [Tau95] and Cohen-Steiner and Morvan [CM03] did. The following formula extends the definition of the normal curvature to an arbitrary non-tangent direction:

$$\kappa_n(\vec{u}) = \begin{bmatrix} n \\ t_1 \\ t_2 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa_1 & 0 \\ 0 & 0 & \kappa_2 \end{bmatrix} \begin{bmatrix} n \\ t_1 \\ t_2 \end{bmatrix} \quad (3-10)$$

Where, $\vec{u} = n \cdot \vec{n}(p) + t_1 \cdot \vec{T}_1 + t_2 \cdot \vec{T}_2$ is an arbitrary direction. Tangent directions to Σ at p are those for which $n = 0$.

With the extended definition of normal curvature, the shape operator at point p

can also be extended to 3×3 form. If we use its eigenvalues to represent the extended shape operator, it is defined as:

$$S_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa_1 & 0 \\ 0 & 0 & \kappa_2 \end{bmatrix} \quad (3-11)$$

In other representations, the two principal curvatures can be derived from the Gaussian and mean curvatures. Based on the first and second fundamental forms of the surfaces in R^3 , the Gaussian curvature (K) and mean curvature (H) are defined as [DoC76]:

$$K = \frac{\det II}{\det I} = \frac{LN - M^2}{EG - F^2}, \quad H = \frac{LG - 2MF + NE}{2(EG - F^2)} \quad (3-12)$$

The two principal curvatures are then given by:

$$\begin{cases} \kappa_1 = H + \sqrt{H^2 - K} \\ \kappa_2 = H - \sqrt{H^2 - K} \end{cases} \quad (3-13)$$

For discrete models, such as the point cloud, or the polygon mesh, the discrete differential geometric information is different from the smooth cases. In the following sections, the methods for discrete Differential geometry properties estimation are discussed in details.

3.3.2. Normal vector estimation

Normal vector is a local geometric property of a 3D surface, which is specific to a given point or a planar facet. Many attempts have already been done for reliable estimation of normal vector from discrete point data, by smooth parametric local surface association [OYF05, YL99], or by generating polyhedral surface [WKW02, HM01].

Reliable estimation of the normal vector at a point is much dependent on the identification of its valid neighbor points. Many researchers have used a fixed number of Euclidean nearest neighbor points to estimate the normal vector at the given point. These k -nearest neighbors ($kNNs$) are used to approximate the tangent plane [HDD92] or the local quadric surface patch [YL99] at the given point. However, the appropriate parameter k is difficult to be determined in these straightforward

methods. Too many neighbor points may degrade the local characteristics of the normal vector. And too few neighbor points may be not sufficient to evaluate the real local geometry. Moreover, possibly, the parameter k should be different for different points in implementation.

Hence, in the proposed method, we construct the polyhedral mesh for the point cloud to identify the neighbor information at each point. Since the incremental mesh construction scheme depends on initial seed facets selection, the neighbors may vary according to different initial seed facets selection. However, this problem can be solved by Delaunay triangulation to obtain the consistent local neighbors and also the consistent normal vector field of the discrete model.

Given a polyhedral mesh surface, the normal vector at each vertex can be estimated as the weighted average of the normal vectors of the adjacent triangle facets around it. Considering an arbitrary vertex p in a discrete mesh surface Σ , assuming its neighbor contains N triangles, then, the normal vector at p could be estimated as:

$$\vec{n}(p) = \frac{\sum_{i=1}^N \omega_i \cdot \vec{n}_i}{\left\| \sum_{i=1}^N \omega_i \cdot \vec{n}_i \right\|} \quad (3-14)$$

Where, \vec{n}_i ($i = 1, \dots, N$) indicate the unit normal vector of the i -th triangle facet respectively. Mathematically, they can be estimated as:

$$\vec{n}_i = \frac{\vec{e}_{i1} \times \vec{e}_{i2} + \vec{e}_{i2} \times \vec{e}_{i3} + \vec{e}_{i3} \times \vec{e}_{i1}}{\left\| \vec{e}_{i1} \times \vec{e}_{i2} + \vec{e}_{i2} \times \vec{e}_{i3} + \vec{e}_{i3} \times \vec{e}_{i1} \right\|} \quad (3-15)$$

\vec{e}_{ij} ($j = 1, 2, 3$) are the three unit edge vectors of the i -th triangle facet in counter-clockwise.

ω_i ($i = 1, \dots, N$) are the weight coefficients corresponding to the normal vectors of facets f_i . There are many ways to determine the weight coefficients, such as area based, angle based, or distance based, etc [OYF05]. However, in our method, we introduce another weigh coefficient determining method. The weight coefficients consider the influences of the area of each adjacent triangle facet and the distance between the given vertex and the barycenter of each adjacent facet. Obviously, the more the area of the triangle is, the more contribution the triangle has; the more

distance between the given vertex and the barycenter is, the less contribution of the triangle is. In order to study the influences of different weights, we tested a various of weights, such as $\frac{A}{d}$, $\frac{1}{d}$, A , etc. According to the test results, the weight $\frac{A}{d^2}$ performs the best performance for the normal vector estimation. Hence, we specify the weight as:

$$\omega_i^* = \frac{A_i}{d_i^2} \quad (3-16)$$

In our method, we also normalize the weights to make the sum of all the weights equal to 1: Accordingly, the applied weight coefficients in our method can be defined as:

$$\omega_i = \frac{A_i/d_i^2}{\sum_{i=1}^N (A_i/d_i^2)} \quad (3-17)$$

Where, A_i ($i=1, \dots, N$) represent the area of the i -th triangle facet. d_i ($i=1, \dots, N$) are the distances between the vertex p and the barycenter of the i -th triangle facet. N is the number of all the triangle facets adjacent to the given vertex.

The figure 3-6 gives the geometric meanings of all the notations mentioned in the above formulae. In this figure, considering an arbitrary vertex p on a general triangle mesh surface, its one-ring vicinity is shown in the left as the area with gray color, the geometry meanings of the mentioned notation for normal estimation is shown in the right figure.

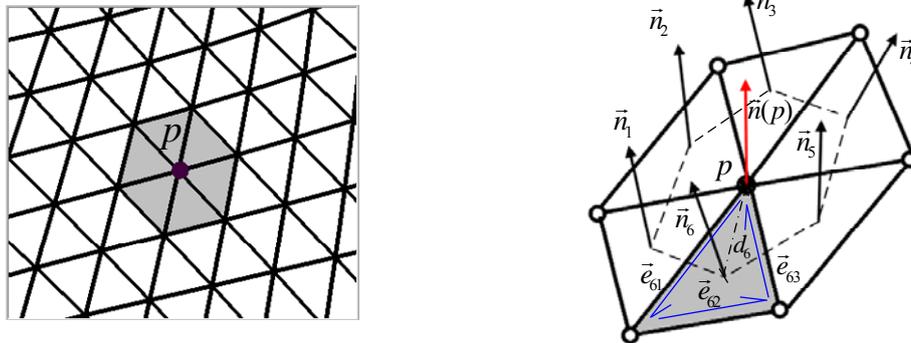


Figure 3-6: The notations mentioned in normal estimation

3.3.3. Discrete curvature estimation methods

Most of the classical discrete curvature estimation methods are based on the polygon mesh surfaces. However, due to the piecewise smoothness of the input mesh, the discrete curvature estimation is subject to various definitions [CM03, MDS02, Tau95]. According to the work of Meek and Walton [MW00], the classical methods of discrete estimation based on polygon mesh surfaces could be classified into three basic categories: ones may approximate a local quadric surface at a given vertex mesh and then apply the derivatives to obtain the curvatures [Sri08]. One may discretize the mathematic formulae that give the curvature information of continuous surface and extend the notations to discrete domains [MDS02]. One may use the tensor based techniques for discrete curvature estimation [CM03, Tau95].

(1). *Local surface approximation based method*

The first way to estimate the discrete curvatures on 3D meshes is by local surface fitting. Yuen, et al [MKY01] presented the general procedures of the methods in this category.

(a) Firstly, at a given vertex p , a local parameterization technique, such as local surface fitting is applied to get a local parametric representation of a surface in u and v . The representation is:

$$r(u, v) = (x(u, v), y(u, v), z(u, v)) \quad (3-18)$$

(b) When $r(u, v)$ correspond to semigeodesic coordinates [MKY01], Gaussian curvature and mean curvature could be derived from formula (3-18) by the following equations:

$$K = \frac{b_{uu}b_{vv} - b_{uv}^2}{x_v^2 + y_v^2 + z_v^2} \quad (3-19)$$

$$H = \frac{b_{vv} + (x_v^2 + y_v^2 + z_v^2)b_{uu}}{2(x_v^2 + y_v^2 + z_v^2)} \quad (3-20)$$

Where, K and H represent the Gaussian curvature and mean curvature respectively. And,

$$\begin{cases} b_{uu} = \frac{Ax_{uu} + By_{uu} + Cz_{uu}}{\sqrt{A^2 + B^2 + C^2}} \\ b_{vv} = \frac{Ax_{vv} + By_{vv} + Cz_{vv}}{\sqrt{A^2 + B^2 + C^2}} \\ b_{uv} = \frac{Ax_{uv} + By_{uv} + Cz_{uv}}{\sqrt{A^2 + B^2 + C^2}} \end{cases} \quad (3-21)$$

Where, $A = y_u z_v - z_u y_v$, $B = x_v z_u - z_v x_u$, $C = x_u y_v - y_u x_v$.

For each point of the surface, $P(x(u, v), y(u, v), z(u, v))$, the corresponding neighbor data is convolved with the partial derivatives of the Gaussian filter function, $G(u, v, \sigma)$ [KMY00]. The Gaussian filter function is a weighted average smoothing carried out at a vertex and its neighborhood, which is used to reduce the influences of the noisy points.

$$\begin{cases} x_u = x * \frac{\partial G}{\partial u}, \quad y_u = y * \frac{\partial G}{\partial u}, \quad z_u = z * \frac{\partial G}{\partial u} \\ x_v = x * \frac{\partial G}{\partial v}, \quad y_v = y * \frac{\partial G}{\partial v}, \quad z_v = z * \frac{\partial G}{\partial v} \\ x_{uu} = x * \frac{\partial^2 G}{\partial u^2}, \quad y_{uu} = y * \frac{\partial^2 G}{\partial u^2}, \quad z_{uu} = z * \frac{\partial^2 G}{\partial u^2} \\ x_{vv} = x * \frac{\partial^2 G}{\partial v^2}, \quad y_{vv} = y * \frac{\partial^2 G}{\partial v^2}, \quad z_{vv} = z * \frac{\partial^2 G}{\partial v^2} \\ x_{uv} = x * \frac{\partial^2 G}{\partial u \partial v}, \quad y_{uv} = y * \frac{\partial^2 G}{\partial u \partial v}, \quad z_{uv} = z * \frac{\partial^2 G}{\partial u \partial v} \end{cases} \quad (3-22)$$

Where, $*$ denotes convolution.

The most commonly used method in this category should be the quadric surface approximation. However, the fitted quadrics are inappropriate for approximating arbitrary data [SB95]. The segmentation based on quadric-based curvatures usually fails to produce acceptable results when the data originate from arbitrarily shaped curved objects [Pet02].

When the Gaussian and mean curvatures are obtained, the principal curvatures can be derived from them, using the equations given in formula (3-13). However, we must note that unlike the continuous case where $H^2 - K$ is always positive, we must make sure that $H^2 - K \geq 0$ is always true in discrete case to avoid any numerical problems.

(2). *Discretization of the curvature equation of continuous surface*

The method based on discrete Laplace-Beltrami operator [CLB09] proposed by Meyer, et al [MDS02] can be viewed as a classic method in this category. The basic idea of their approach is that a mesh is considered as either the limit of a family of smooth surfaces, or a linear approximation of an arbitrary surface; and the geometry properties on the surface at a vertex are defined as the spatial average around this vertex. Thus for each vertex, it is important to choose an appropriate associated surface patch over which the average will be computed. There are two main types of local regions commonly used in practice, shown in figure 3-7. However, Meyer et al [MDS02] proved that the Voronoi region plays the better performance than barycenter-based cell. Hence, the Voronoi-based region is used.

Similar as the methods based on local quadric surface association, the method of Meyer, et al. also begins with the calculations of mean curvature and Gaussian curvature. For this purpose, the mean curvature normal operator, which is also known as Laplace-Beltrami operator is introduced to the discrete case.

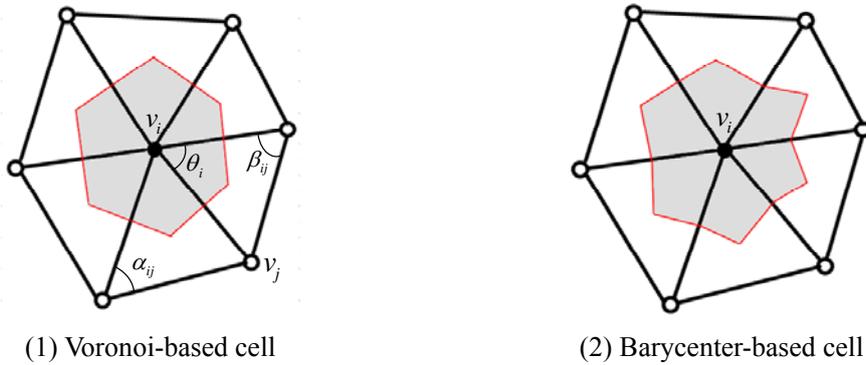


Figure 3-7: Two main types of local regions

According to differential geometry, the direction relationship between infinitesimal surface area and mean curvature flow can be represented as:

$$2H \cdot \vec{n}(p_i) = \lim_{A \rightarrow 0} \frac{\nabla A}{A} \quad (3-23)$$

Where, A is an infinitesimal area around a vertex p_i . ∇ is the gradient with respect to the coordinates of p_i in x, y, z directions.

If we denote by L the operator that map the vertex p_i to the vector

$$2H \cdot \vec{n}(p_i),$$

$$L(p_i) = 2H \cdot \vec{n}(p_i) \quad (3-24)$$

L is known as the Laplace-Beltrami operator for the surface. Laplace-Beltrami operator is a generalization of Laplacian operator from flat space to manifolds [DHK92]. We first discretize the surface in conformal parameter space with u and v , that is, for each linear triangle, the triangle itself defines the local surface metric. With the induced metric, the Laplace-Beltrami operator simply turns into Laplacian [DHK92], $\Delta_{u,v} = p_{iuu} + p_{ivv}$. So we can get:

$$\iint_A L(p_i) dA = -\iint_A \Delta_{u,v} p_i du dv \quad (3-25)$$

Using Gauss theorem, the integral of the Laplacian over a surface going through mid-point of each one-ring edge of a triangulated domain can be simply represented as follows [MDS02]:

$$\iint_A L(p_i) dA = \frac{1}{2} \sum_{j \in N_1(p_i)} (\cot \alpha_{ij} + \cot \beta_{ij})(p_i - p_j) \quad (3-26)$$

Where, $N_1(p_i)$ is the one-ring vicinity of the vertex p_i . The geometric meaning of the notations can be found in figure 3-1-7 (1). And if the area of the Voronoi based local regions is computed, the Laplace-Beltrami operator then can be represented as:

$$L(p_i) = \frac{1}{2A} \sum_{j \in N_1(p_i)} (\cot \alpha_{ij} + \cot \beta_{ij})(p_i - p_j) \quad (3-27)$$

From this expression, we can easily compute the mean curvature value at the vertex p_i with the combining of formula (3-1-24) by taking the half of the magnitude of $L(p_i)$:

$$H(p_i) = \frac{1}{4A} \left\| \sum_{j \in N_1(p_i)} (\cot \alpha_{ij} + \cot \beta_{ij})(p_i - p_j) \right\| \quad (3-28)$$

Similar to what is done for the mean curvature normal operator, the Gaussian curvature is also considered in local integral. However, thanks to the Gauss-Bonnet theorem which provides a simply, valid way for this purpose, the local integral of Gaussian curvature over a Voronoi based region can be stated as:

$$\iint_A K \, dA = 2\pi - \sum_{j \in N_1(p_i)} \theta_j \quad (3-29)$$

The discrete expression of the Gaussian curvature at the vertex p_i can then be represented as:

$$K(p_i) = \frac{1}{A} (2\pi - \sum_{j \in N_1(p_i)} \theta_j) \quad (3-30)$$

Since the Gaussian and mean curvatures are both solved, the principal curvatures can be easily computed based on the formulae (3-13). In order to determine the two principal directions, the method uses the tensor based technique and computes the eigenvectors of the curvature tensor. The tensor base techniques for curvature estimation are discussed in the next part.

(3). *Tensor based techniques*

Curvature tensor of a surface Σ is the map $p \mapsto \kappa_N$ that assigns each point p of Σ to the function that measures the normal curvature $\kappa_N(\bar{u})$ of Σ at p in the direction of the unit vector \bar{u} , tangent to Σ at p [Tau95]. The function of curvature measures of continuous surfaces always use the shape operator which we discussed in section 3.3.1. For the discrete cases, however the definitions are subject to various forms (see details in [GGR06, Tau95]). It seems a natural way to define a discrete shape operator as the map function to measure the curvatures on polyhedral mesh surfaces [CM03, HP04]. Cohen-Steiner and Morvan proposed a method to estimate the discrete curvature tensor on polygon mesh [CM03]. We will use this method to describe the tensor based techniques for curvature estimation.

From the section 3.3.1, we know that for smooth case, the symmetric shape operator matrix captures the complete normal curvature at a point. The eigenvalues and eigenvectors of shape operator matrix are the principal curvatures and principal directions respectively. Thus, for polyhedral case, if we build the shape operator matrix at each vertex, the discrete principal curvatures can be estimated by computing the eigenvalues of the shape operator matrix. The principal directions can also be obtained from eigenvectors. This is the basic idea of the discrete curvature estimation method proposed by Cohen-Steiner and Morvan. In order to build the continuous curvature tensor field over the whole surface, after estimating the curvature tensor at each vertex, the piecewise linear curvature tensor field can be built by interpolating

these values across triangles.

It is not a natural way to evaluate the discrete curvatures at an isolated vertex [ACD03]. One should consider the integrals of curvatures over a given local region around a vertex. The integral of curvatures associated with local region is called curvature measure in [CM03]. Given a polyhedral mesh surface Σ and an arbitrary vertex p on Σ , the curvature tensor at p is defined in a local region around p , denoted as B shown in figure 3-8. A natural curvature tensor can be defined at each vertex along a mesh edge in its local region B [ACD03, CM03]. This line density of curvature tensor can now be integrated over the local region B , based on which Cohen-Steiner and Morvan defined an anisotropic curvature measure in the following forms:

$$\begin{aligned} \bar{H}_v(B) = \sum_{e \in E} \frac{\text{length}(e \cap B)}{2} [& (\beta(e) - \sin \beta(e)) \cdot (\bar{e}^+ \times \bar{e}^{+T}) \\ & + (\beta(e) + \sin \beta(e)) \cdot (\bar{e}^- \times \bar{e}^{-T})] \end{aligned} \quad (3-31)$$

Where, $\bar{H}_v(B)$ is the curvature measure defined on mesh vertex, which can be regarded as the discrete shape operator matrix. E is the collection of all the mesh edges in B . \bar{e}^+ and \bar{e}^- denote the normalized sum and difference of unit normal vectors of the triangles incident at the edge e respectively. $\text{length}(e \cap B)$ is the length of the edge e in the local region B . $\beta(e)$ is the dihedral angle between the two normal vectors of the triangle incident with the edge e .

We must note that the shape operator $\bar{H}_v(B)$ is a 3×3 symmetric matrix which has three eigenvalues and eigenvectors respectively. The three eigenvectors correspond to the maximum principal direction, the minimum principal direction and the normal vector at the given vertex respectively. Therefore, The first two maxima eigenvalues of $\bar{H}_v(B)$ are the two principal curvatures at the given vertex.

In their research, Cohen-Steiner and Morvan defined another matrix which has the same eigenvectors but swapped eigenvalues with $\bar{H}_v(B)$, denoted as $\tilde{\bar{H}}_v(B)$. $\tilde{\bar{H}}_v(B)$ has the representation as shown in formula (3-31). Comparing to $\bar{H}_v(B)$, $\tilde{\bar{H}}_v(B)$ has a simpler form which is more convenience for computing. However, with the formula (3-32), we have to notice that the associated eigenvectors are switched: the eigenvector associated with the maximum principal curvature are the minimum principal direction and vice versa.

$$\bar{H}_v(B) = \frac{1}{A} \sum_{e \in E} \beta(e) \cdot \text{length}(e \cap B) \cdot (\bar{e} \times \bar{e}^T) \quad (3-32)$$

Where A is the area of the local region B . \bar{e} is the unit vector along the edge e . The other notations in formula (3-32) conform to the same meanings in the formula (3-31), which is described in figure 3-8. The method has solid foundations and convergence property which is defined in [CM03, Mor08].



Figure 3-8: The notations in the method of Cohen-Steiner and Morvan

3.3.4. Our estimation method

The Cohen-Steiner and Morvan method is simple and efficient to implementation for most kinds of polyhedral mesh models. However, the method fails to provide reliable results in some cases. For example, when we implement it for spherical polyhedral surfaces, the estimating errors always reach 33%, no matter how dense the vertices are. Until now, we haven't known the exact reason to this problem. However, the local region selection might play a role to cause the large estimation errors. Commonly, with the definition of local region B , the local region of a given vertex always overlaps with the local region of another vertex in its vicinity. In other words, the properties of other vertices may influence the curvature estimation at a given vertex. These influences might be the main reason to cause the larger errors.

Therefore, in our method, the main difference from the Cohen-Steiner and Morvan method is the selection of local regions. As discussed in the method developed by Meyer et al. [MDS02], the local region is defined based on the Voronoi diagram (the left picture in figure 3-9). This kind of local regions tiles the surface without overlapping and the total sum of all the local regions around every vertex composes the complete surface. In our method, avoiding using the local regions as Cohen-Steiner, we use the Voronoi based local regions as Meyer et al. did. Moreover, considering the contribution of line curvature tensor along each mesh edge to the

given vertex, we add weight coefficients to the formula (3-32). Mathematically, the weight equals to the value between the edge and its projection on the tangent plane. If the weight is standardized, the convergence of the estimation method will be same as the initial Cohen-Steiner method [CM03]. The formula for discrete curvatures estimation leads to the following expression:

$$\bar{H}_v(B) = \frac{1}{A} \sum_{e \in E} \lambda_e \cdot \beta(e) \cdot \text{length}(e \cap B) \cdot (\bar{e} \times \bar{e}^T) \quad (3-33)$$

Where, λ_e is the coefficient associated with the edge e , which has the representation as:

$$\lambda_e = \frac{\arccos \frac{\vec{n}(p) \cdot \vec{n}(e)}{\|\vec{n}(p)\| \cdot \|\vec{n}(e)\|}}{\sum_{e \in E} \left(\arccos \frac{\vec{n}(p) \cdot \vec{n}(e)}{\|\vec{n}(p)\| \cdot \|\vec{n}(e)\|} \right)} \quad (3-34)$$

$\vec{n}(e) = \frac{\vec{n}_1 + \vec{n}_2}{\|\vec{n}_1 + \vec{n}_2\|}$ is the edge normal of the edge e , which is equal to the average

normal vectors of the two triangles incident to the edge. The notation is described in the following figures.

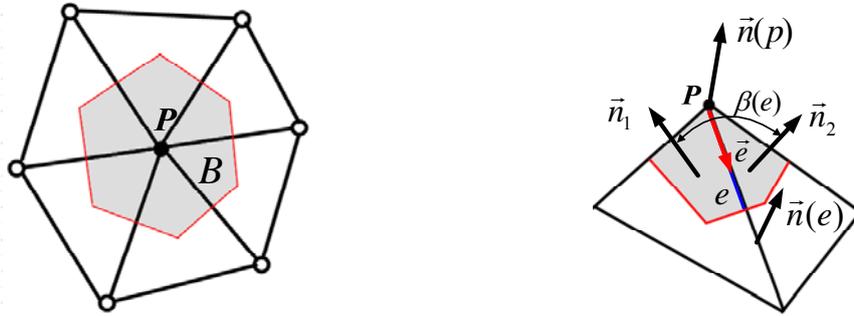


Figure 3-9: The notations in our proposed method

3.3.5. Shape index and curvedness

Similar to Gaussian curvature and mean curvature, we can derive another two surface descriptors from the two principal curvatures, called shape index and curvedness, which are very important indicators for the further processes of the discrete models in our work. We just give their definitions in this section. The applications of them can be found in chapter 4 and chapter 5.

Shape index first introduced by Koenderink and Doorn [KD92] is a single value within the range $[-1, 1]$ to measure the local surface types. For both of the discrete or continuous shapes, it has the definition as:

$$s = -\frac{2}{\pi} \arctan\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right) \quad (\kappa_1 \geq \kappa_2) \quad (3-35)$$

Where, κ_1 and κ_2 are the maximum and minimum principal curvatures of the local surface. [DJ97] and [LG09] gave some variant definitions of shape index respectively. The surface type of a vertex can be given by the following table according to the shape index intervals [KD92]. Detailed specification of the table is presented in section 5-4-2.

Table 3-1: Surface types specified by shape index intervals and their type labels

Surface type	Shape index interval	Type label	Surface type	Shape index interval	Type label
Spherical cup	$s \in [-1, -\frac{7}{8}]$	$T = -4$	Spherical cap	$s \in (\frac{7}{8}, 1]$	$T = 4$
Through	$s \in (-\frac{7}{8}, -\frac{5}{8}]$	$T = -3$	Dome	$s \in (\frac{5}{8}, \frac{7}{8}]$	$T = 3$
Rut	$s \in (-\frac{5}{8}, -\frac{3}{8}]$	$T = -2$	Ridge	$s \in (\frac{3}{8}, \frac{5}{8}]$	$T = 2$
Saddle rut	$s \in (-\frac{3}{8}, -\frac{1}{8}]$	$T = -1$	Saddle ridge	$s \in (\frac{1}{8}, \frac{3}{8}]$	$T = 1$
Saddle	$s \in (-\frac{1}{8}, \frac{1}{8}]$	$T = 0$	Plane	$s = 2$	$T = 5$

Curvedness, which is also known as bending energy [WBH07, Zor03], is always a positive number to specify the amount of the surface curvatures. There are several definitions of curvedness [LG09, KD03]. Here, Curvedness is defined as [KD92].

$$c = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \quad (3-36)$$

Unlike the dimension-independent shape index, curvedness is inversely proportional to the local size of the object and has the dimension of reciprocal length. Curvedness only vanishes at planar vertices. However, Gaussian curvature also vanishes on parabolic surface while mean curvature also vanishes on saddle surface.

Similar to couples of principal curvatures or Gaussian and mean curvatures, the

pair of shape index and curvedness is also geometric invariants and records the complete local second order geometric information of a shape [KD92].

3.4. Testing results

3.4.1. Normal estimation results

The figure 3-10 shows three examples of the normal estimation. We can see that the normal estimation results of our method are reasonable.

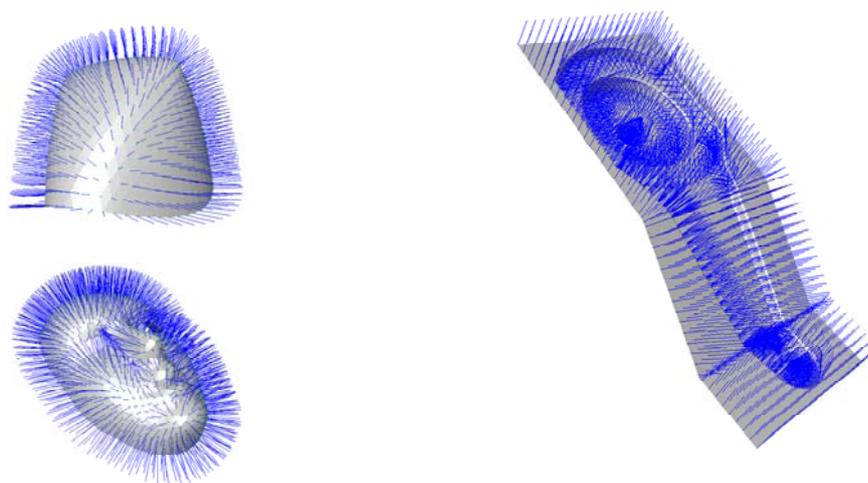


Figure 3-10: Three examples of the Normal estimation

3.4.2. Curvature estimation results and comparison

(1). *Comparison of the related methods*

We implemented the related three methods for discrete curvature estimation: our proposed method, Cohen-Steiner and Morvan method, and the method of Meyer et al. In order to compare the results obtained by the three methods, several discrete meshes approximating the simple surfaces where the curvatures are known analytically, such as sphere, cylinder, hyperboloid and flat plane, are considered. We use three errors to evaluate the performances of the three methods, respectively: relative average error (e_{ave}), relative maximum error (e_{max}) and relative minimum error (e_{min}), which are defined as:

$$e_{ave} = \left| \frac{\kappa_{ave} - \kappa_T}{\kappa_T} \right| \quad (3-37)$$

$$e_{max} = \left| \frac{\kappa_{max} - \kappa_T}{\kappa_T} \right| \quad (3-38)$$

$$e_{min} = \left| \frac{\kappa_{min} - \kappa_T}{\kappa_T} \right| \quad (3-39)$$

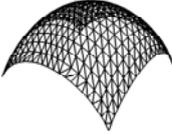
Where, κ_{ave} , κ_{max} , κ_{min} respectively represent the computed average curvatures, computed maximum curvatures and computed minimum curvatures. κ_T denotes the theoretical values of the corresponding curvatures. Notice that the curvatures can be the maximum or minimum principal curvatures.

The tables below list comparison results. In these tables, the method proposed by Cohen-Steiner and Morvan is denoted as Cohen; the method developed by Meyer et al. is denoted as Meyer, and our method is denoted as Zhao. Table 3-2 shows the theoretic and computed curvatures estimated by the three methods respectively. And table 3-3 shows the respective relative estimation errors of the three methods. The tables from 3-4 to 3-6 show the sorting results of the three methods based on their relative estimation errors respectively. Finally, we can find that our proposed method provides the most interesting performances from the global view.

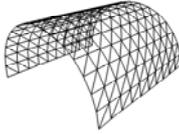
Here, we notice that the curvature values estimated by Cohen method are not the initial ones, but the values after adjusting by multiplying with 1.5, a multiplication factor when testing curvatures values by Cohen method we obtained large errors that suggested a missing 1.5 factor. Thanks to regression analysis, we obtained a 1.5 correction factor when testing Cohen method to estimate the curvatures on different simple shapes.

Table 3-2: Estimation of curvature values by the three methods on different surface types

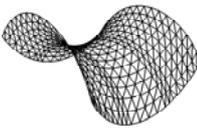
Types	Sphere		Cylinder		Symmetric saddle		Plane	
	κ_1	κ_2	κ_1	κ_2	κ_1	κ_2	κ_1	κ_2
Theoretic κ_T	0.1	0.1	0.1	0.0	0.1	-0.1	0.0	0.0
κ_{ave}	0.10021	0.10004	0.10016	0.0	0.09994	-0.09965	0.0	0.0
Cohen κ_{max}	0.10128	0.10074	0.10178	0.0	0.10315	-0.09933	0.0	0.0
κ_{min}	0.09936	0.09936	0.09933	0.0	0.09888	-0.10539	0.0	0.0
κ_{ave}	0.10057	0.09412	0.10029	0.0	0.09976	-0.10021	0.0	0.0
Meyer κ_{max}	0.10077	0.09716	0.10095	0.0	0.10245	-0.10009	0.0	0.0
κ_{min}	0.09987	0.09248	0.09954	0.0	0.09900	-0.10171	0.0	0.0
κ_{ave}	0.09996	0.09977	0.10005	0.0	0.10001	-0.09980	0.0	0.0
Zhao κ_{max}	0.10057	0.10005	0.10011	0.0	0.10298	-0.09946	0.0	0.0
κ_{min}	0.09987	0.09944	0.09954	0.0	0.09903	-0.10171	0.0	0.0



$R = 10$



$R = 10$



$R = 10$

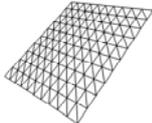


Table 3-3: Estimation errors of the three methods on different surface types

Types	Sphere		Cylinder		Symmetric saddle		Plane		
	κ_1 (%)	κ_2 (%)	κ_1 (%)	κ_2 (%)	κ_1 (%)	κ_2 (%)	κ_1 (%)	κ_2 (%)	
Cohen	e_{ave}	0.21	0.04	0.16	0.0	0.06	0.35	0.0	0.0
	e_{max}	1.28	0.26	1.78	0.0	3.15	0.76	0.0	0.0
	e_{min}	0.64	0.64	0.67	0.0	1.12	5.39	0.0	0.0
Meyer	e_{ave}	0.57	5.88	0.29	0.0	0.24	0.21	0.0	0.0
	e_{max}	0.77	3.84	0.95	0.0	2.45	0.09	0.0	0.0
	e_{min}	0.13	7.52	0.45	0.0	1.00	1.71	0.0	0.0
Zhao	e_{ave}	0.04	0.23	0.05	0.0	0.01	0.2	0.0	0.0
	e_{max}	0.43	0.05	0.11	0.0	2.98	0.54	0.0	0.0
	e_{min}	0.13	0.55	0.45	0.0	0.97	1.71	0.0	0.0

Table 3-4: Sorting the methods based on the relative average error

e_{ave}	Sphere		Cylinder	Symmetric saddle	
	κ_1	κ_2	κ_1	κ_1	κ_2
Cohen	2	1	2	2	3
Meyer	3	3	3	3	2
Zhao	1	2	1	1	1

Table 3-5: Sorting the methods based on the relative maximum error

e_{max}	Sphere		Cylinder	Symmetric saddle	
	κ_1	κ_2	κ_1	κ_1	κ_2
Cohen	3	2	3	3	3
Meyer	2	3	2	1	1
Zhao	1	1	1	2	2

Table 3-6: Sorting the methods based on the relative minimum error

e_{\min}	Sphere		Cylinder	Symmetric saddle	
	κ_1	κ_2	κ_1	κ_1	κ_2
Cohen	3	2	3	3	3
Meyer	1	3	1	2	1
Zhao	1	1	1	1	1

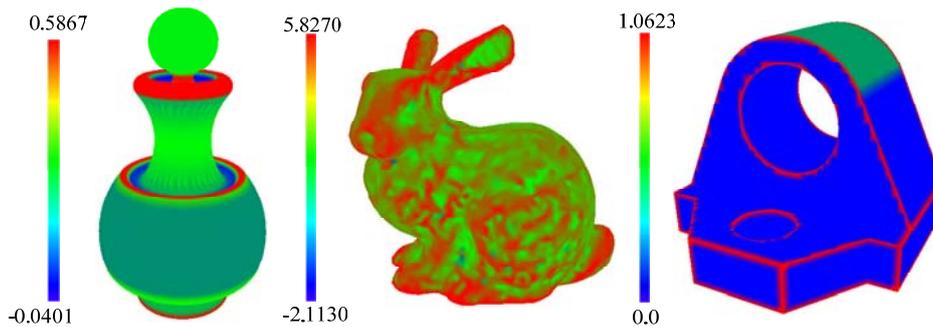
(2). *Estimation results of our method*

Figure 3-11 gives some principal curvatures fields of several polyhedral models estimated by our method. The curvatures fields are rendered with continuous B-R colors.

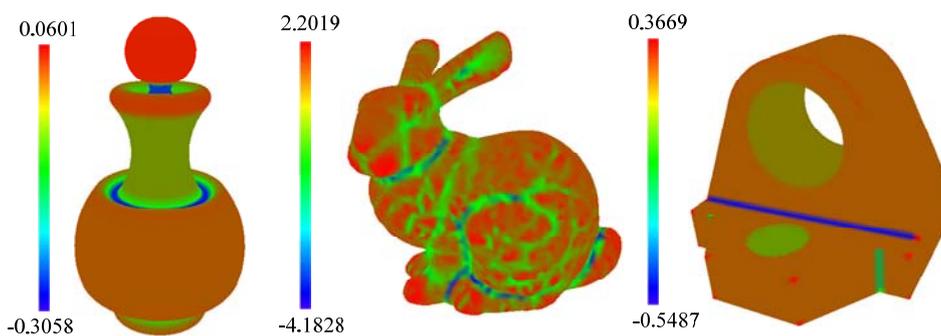
The proposed discrete curvature estimation method had been applied not only to the polyhedral models applied in engineering domains, but also to the models of natural freeform objects, like the Stanford bunny [Stan] shown in the figure 3-11.

3.4.3. Time performance

Time performance is obtained based on the following configuration. Platform: MS Visual C++ 2005 on Windows operation system; 1.83GHz Core II CPU with 1.0G RAM. The table 3-7 lists the curvatures' estimation time for some models with different vertices sizes. And the figure 3-12 demonstrates the trend of the time performance when the sizes of the discrete models become larger. The results show that the proposed method is efficient in time performance.



(1) The maximum principal curvature fields

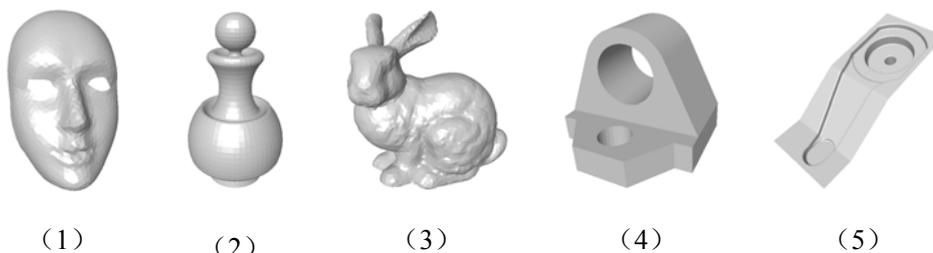


(2) The minimum principal curvature fields

Figure 3-11: The maximum and minimum principal curvature fields

Table 3-7: Time performances for the mentioned models

Index	Model	Size of vertices	Time (s)
(1)	Mask	1269	0.14
(2)	Rotated axis-symmetric part	3735	0.281
(3)	Bunny	4380	0.327
(4)	Composed workpiece	4970	0.359
(5)	Tool die	33862	2.453



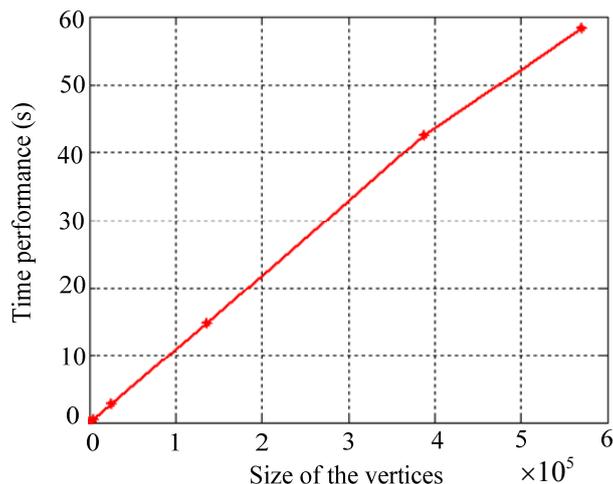


Figure 3-12: Time performance of our curvature estimation method

3.5. Conclusion

The methods to estimate the normal vectors and discrete curvatures of a given discrete shape are presented in this chapter.

The polygon mesh topological structure is associated to the scattered point data and the normal at each vertex is estimated as the weighted average of the normal vectors of the facets in its one-ring neighbor region.

We surveyed different discrete curvature estimation methods such as surface approximation based methods, discretization of the formula of continuous surface and tensor based techniques. Based on these different methods, we developed a new method to construct the discrete shape operator matrix at each vertex within its Voronoi cell. The principal curvatures and the principal directions can then be derived from the eigenvalues and the eigenvectors of the discrete shape operator matrix. We introduced the shape index and curvedness for local shape description.

A comparative analysis of different methods for discrete curvature estimation is presented. The results prove that the new proposed method provides better performances, robustness and efficiency.

Chapter 4

Registration of Discrete Shapes

4.1. Introduction

Registration is one of the most important and decisive steps of multisensor integration. The point data acquired by multiple views/sensors are usually represented in their own coordinate systems. During the registration process, the measurement data captured in the respective sensor's coordinate system are aligned and transformed to one common coordinate system. The transformation parameters including three rotations and three translations are usually determined by minimizing the distance between the shapes [BM92]. Another application of registration is to match digitized data from an unfixed rigid object with its idealized geometric CAD model for the automatic position prior to shape inspection.

This chapter presents a method to register the discrete shapes with unknown correspondences. The method uses the combination of the curvatures' information and Euclidean distance to help the corresponding point searching. The method can provide accurate registration results with fast convergence. In most of piecewise registration methods, one discrete shape is usually assumed as model data that is stationary and the other is viewed as scene data which is spatially transformed to match the model data. We also conform to that statement in this chapter.

The chapter 4 is organized as follows: Section 4.2 surveys the related works on registration. Section 4.3 gives the overview of the proposed method; Section 4.4 and section 4.5 discuss the detailed procedures in coarse and fine registration respectively. Section 4.6 provides several testing results. We conclude the chapter in section 4.7.

4.2. Literature review

The early approaches for registration of discrete shapes are based on matching discrete features [FH86]. Many of the difficulties inherent in feature based methods are overcome by the Iterative Closest Point (ICP) algorithm [BM92, CM91]. The ICP algorithm assumes that the two discrete shapes are initially aligned. Therefore, The ICP based registration process takes place in two stages: coarse registration and fine registration [JH02, KSL09].

The objective of coarse registration is to align the discrete shapes initially with lower resolution from global view. Approaches for coarse registration are usually

based on known correspondence recognition and matching. The correspondences can be the calibration relationships among the multiple sensors and views [ZSR08], the discrete features [ABR05, CHC98], or the markers attached to the target object [KSL09, LLS06]. Some papers attempt to solve the coarse registration problem automatically without known correspondences using Principle Component Analysis (PCA) [LNL08, ZPM04] or genetic algorithm [LCE04]. These automatic methods suffer from two main problems in practice.

The first problem is that these methods usually require large similarity between the two point sets, like methods based on PCA [LNL08, ZPM04] and least median of squares [LR09]. If the discrete shapes are only partial overlapping, which is more general, the initial registration may generate large matching errors and lead wrong trend for the later fine registration. The second problem is the computing complexity. Some methods try to solve the first problem using genetic algorithms [LCE04]. However, these methods lead directly to the complex computing problems.

The fine registration is performed to minimize the matching errors with full resolution. The Iterative Closest Point (ICP) algorithm [BM92, CM91] is one of the best known and widely used matching methods used for this purpose. The ICP algorithm finds the optimal match iteratively. It has a computing complexity of $O(N_s \cdot N_m)$, where, N_s and N_m represent the numbers of points respective in the scene data and in the model data. If the sizes of the points in discrete shapes are large, the traditional ICP algorithm requires heavy computations. Another difficulty in the traditional ICP algorithm is determining the overlapping sections between the two input data, which has determinant influence on the final registration performances.

Considerable variants have been developed to speed up the convergence and/or improve the accuracy of the traditional ICP algorithm. The common searching techniques, like k-D tree [GY03, NLH07], z-buffer [BS99], are introduced to accelerate the closest point finding. A good review of these variants can be found in [RL01]. Lomonosov, et al. proposed a trimmed ICP algorithm which is fast and robust to align two point clouds with overlaps under 50%. [CSS02, LCE04]. Almhdie, et al. built a comprehensive lookup matrix to improve the precise of correspondences computing [ALD07]. Jost and Hugli developed a multi-resolution ICP algorithm to reduce the computational cost of the traditional ICP algorithm by using different resolutions in different iterations [JH02].

Some additional information, like intensity of the range data [GRB94], intrinsic invariants [BDW04], color and texture information [KKW04, SJH98], invariant features [SLW02] or normal vectors [ZGX05] have also been used to improve the correspondence searching. Morency and Darrel used the closest point and inverse calibration techniques for corresponding point searching and defined a new error function to combine the two distances [MD02]. The registration problem has also been investigated from the geometry viewpoint by second order surface approximations [MGP04, PHY06].

Instead of the widely used Euclidean distances, such as point-to-point distance [BM92, ZSN03], point-to-plane distance [PS03,CM91] and point-to-projection distance [BL95], some new distances, like vector distance [MFF07], Hausdorff distance [VSH04], are implemented to evaluate the similarity between the two point cloud data too.

According to the author's knowledge, despite there have already been a large amount of work on registration, the determination of corresponding points hasn't been solved yet. It seems there is not a thorough investigation of registration from the view of curvature based local feature. Several feature based algorithms have been proposed in [BDW04, KKW04, JHS98, ZGX05]. However, these additional features information in these algorithms are indeed only used for coarse registration. Curvature was considered as the geometry invariants in the algorithm mentioned in [SLW02], but their method is only used for alignment of the scans data with the CAD model. [PHY06] considered the geometric features by local surface approximations which aggravated the computational complexity.

4.3. Method overview

The proposed approach addresses the piecewise registration problem of two arbitrary discrete shapes with partial overlapping regions. Our method is a new modified ICP algorithm in which we combined the curvature information for the correspondence searching. Without known correspondences, the proposed method can provide elegant registration results automatically for two input discrete shapes with arbitrary positions. Similar as other ICP variants, our method is executed in two main phases: coarse registration and fine registration. The overall registration procedures

are shown in figure 4-1.

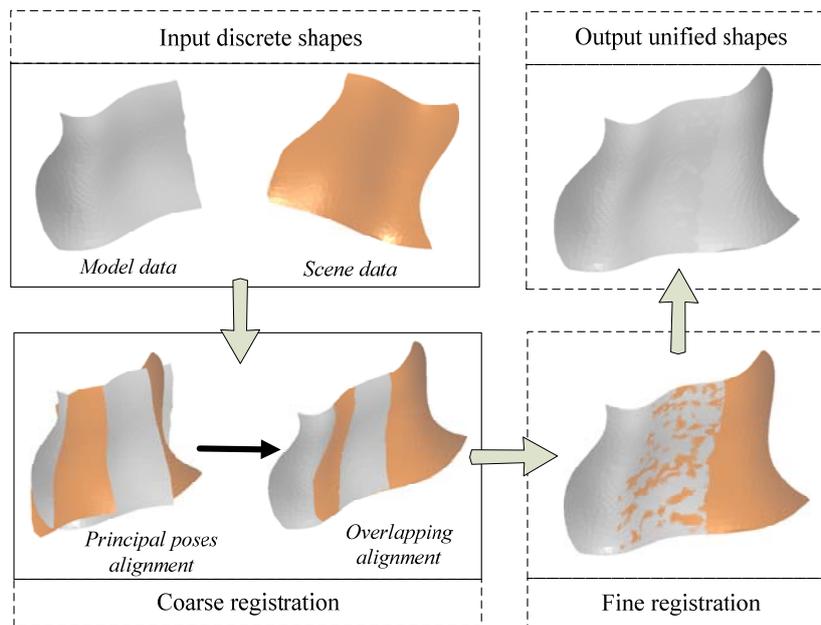


Figure 4-1: Procedures of the proposed registration method

The purpose of coarse registration is to roughly align the two shapes according to their overlaps and provide a correct trend for the fine registration. It includes two basic steps: principal poses alignment and overlapping alignment.

Principal poses alignment attempts to register the two discrete shapes according to their principal poses from the global view. The principal pose of a model contains a centroid and three principal axes, which can be evaluated by many statistic methods, like PCA (Principal Components Analysis), Least Median of Squares, etc. The method in this phase is quite like the existing methods do in literatures [LNL08, ZPM04].

Overlapping alignment is used to refine the result of principal poses alignment by registering the shapes based on their overlapping sections. In this phase, the corresponding regions between the two shapes are searched and the shapes are matched accordingly. Even the correspondences searching has limited resolution in general cases, it is helpful to provide correct trend for find registration.

Similar to other ICP variants, the fine registration in our method is also an iterative approach. The main difference from other ICP variants is that our algorithm provides a new way to combine the curvature information into the registration process to accelerate the corresponding points searching in the two shapes. The results prove that the proposed new method can provide more accurate registration results with less time-consuming than the initial ICP algorithm. In the following sections, we will

denote our Curvature-based Fine Registration method as CFR.

The main contributions of our works in this domain are:

(a) We proposed a new coarse registration method to align the two shapes according to their overlapping sections. It can provide more accurate results than the existing methods which only align the two shapes from the global view.

(b) We provided a new way to combine the curvature information into the fine registration process and defined a new distance to replace the Euclidean distance for the corresponding point pairs searching. The new distance can accelerate the corresponding point pairs searching with more accurate results. Thus, the proposed fine registration method can provide results with better performances in accuracy and time-consuming than the initial ICP algorithm.

(c) New rejection criterions are designed to trim the corresponding point pairs. The potential mismatched point pairs are rejected in the corresponding point pairs searching step, which ensures the robustness and efficiency of CFR method.

4.4. Coarse registration

Most of the previous works on coarse registration are done manually and with known correspondences. If the correspondences are unknown, as mentioned in section 4.2, the existing automatic methods suffer to the problems of larger matching errors [LNL08, LR09] or complex computations [LCE04].

Therefore, we propose a new method to align the two data based on their roughly recognized overlapping sections. Firstly, the two given shapes are initially aligned based on their principal poses. Secondly, we employ an additional procedure to reduce the matching errors by adjusting the alignment result according to their overlapping sections.

4.4.1. Principal pose estimation

The principal pose of an object comprises the location and orientation information in the sparse space, which can be represented as its centroid and three principal axes [LR09]. Its centroid records the location information while the three principal axes represent its orientation information. The principal pose can also be

described as the principal coordinate system of the object, where, the origin is its centroid and the three orthogonal coordinate axes are represented as the three principal axes organized by right-hand rule.

The centroid of an object can be easily estimated as the coordinate average of the elements composed to the object. The principal axes are the directions where the object is more extended. One of the most general methods for extracting the principal axes is PCA. PCA is a statistic method for principal components analysis by covariance analysis between factors [DSH01]. The goal of PCA is to compute the most meaningful basis to re-express the point sets. The new basis is hoped to filter the noise and reveal the hidden structure. Even the least median of squares [Rou84] seems more robust than PCA in computing the principal axes [LR09], PCA is much simpler than least median of squares and the result is satisfying for doing coarse registration. Therefore, the PCA method is implemented in our research works.

Considering a discrete shape represented by P_N , p_i is an arbitrary point in it, which is represented as a column vector $p_i = [x_i, y_i, z_i]^T$. In general, The PCA computes the principal axes as the following three steps:

(a) Calculating the origin of the principal coordinates system. The origin is determined as the centroid of P_N which is calculated as:

$$o_{pca} = \frac{1}{N} \sum_{i=1}^N p_i \quad (p_i \in P_N) \quad (4-1)$$

(b) Computing covariance matrix which is defined as [DSH01]:

$$M_{cov} = \sum_{i=1}^N (p_i - o_{pca})(p_i - o_{pca})^T \quad (p_i \in P_N) \quad (4-2)$$

(c) Eigenvalues and eigenvectors estimation of the covariance matrix. The first principal axis is the eigenvector corresponding to the largest eigenvalue. The two other principal axes are obtained from the remaining eigenvectors.

Figure 4-2 shows two principal coordinate systems constructed by the principal poses of the objects. The left one is an example of point cloud of a face mask. The right one shows the principal coordinate system of polygon mesh surface of a gear. In application, we define the X-axis and Y-axis of the principal coordinate system are along the first and the second principal axes respectively. The Z-axis of the principal

coordinate system can then be determined by the right-hand rule. The examples of the corresponding relationships between the three principal axes and the three orthogonal axes of the principal coordinate systems are illustrated in figure 4-2.

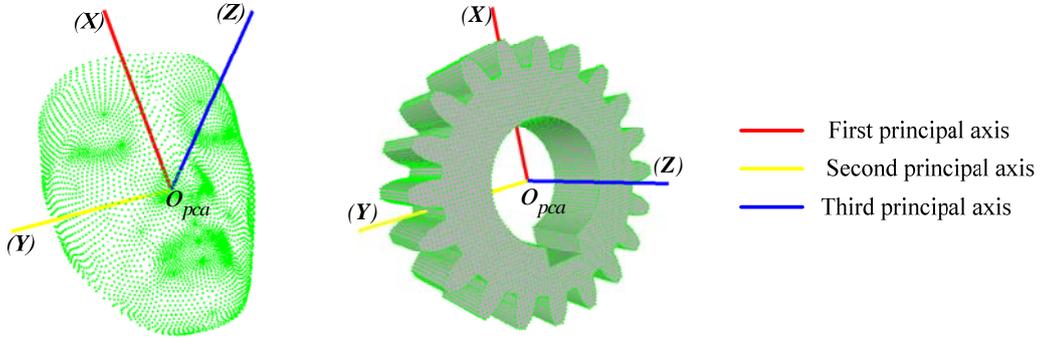


Figure 4-2: Principal coordinate systems of two discrete shapes

4.4.2. Transformation calculation

For both of the scene data and the model data, we can compute their principal coordinate systems by PCA respectively. In this section, we will calculate the rigid transformation to align the two principal coordinate systems together. The rigid transformation is parameterized in terms of rotations around and translations along each of the three coordinate axes.

Assuming that $C_1(o_1, \bar{x}_1, \bar{y}_1, \bar{z}_1)$ and $C_2(o_2, \bar{x}_2, \bar{y}_2, \bar{z}_2)$ denote the principal coordinate systems of scene data S_N and model data M_K respectively. Where, o_i are the centroids of the scene data ($i=1$) and model data ($i=2$) respectively; \bar{x}_i , \bar{y}_i and \bar{z}_i are the three normalized principal axes of each data. The subscript notes of S_N and M_K , N and K represent the point number of each data. All the data are embedded in the global coordinate system, which is represented as $C_0(o_0, \bar{x}_0, \bar{y}_0, \bar{z}_0)$

Let p_1 represent an arbitrary point in scene model, $\overrightarrow{o_0 p_1}$ and $\overrightarrow{o_1 p_1}$ represent the coordinates of p_1 in C_1 and C_2 respectively. Let p_2 represent the corresponding point in model data to p_1 . $\overrightarrow{o_0 p_2}$ and $\overrightarrow{o_2 p_2}$ have the similar meanings with $\overrightarrow{o_0 p_1}$ and $\overrightarrow{o_1 p_1}$ respectively. All the notations are described in figure 4-3.

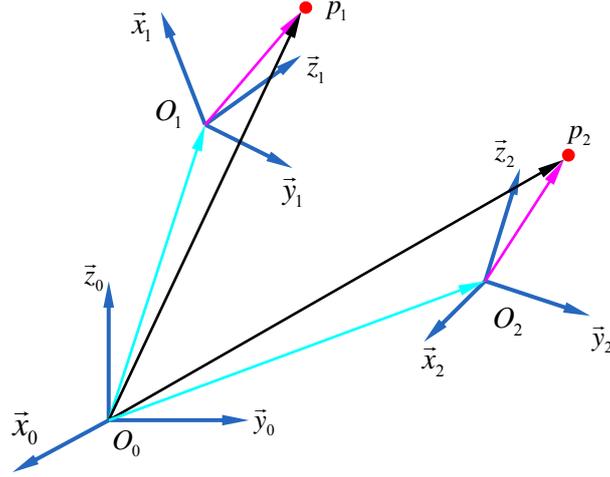


Figure 4-3: Notations for coordinate system transformation calculation

The objective of this section is to estimate an optimal rigid transformation to match the vector $\overrightarrow{o_0 p_1}$ to the vector $\overrightarrow{o_0 p_2}$.

The points both in the scene data and in the model data have the coordinates represented in the global coordinate system. Therefore, we first consider the transformations between the global coordinate system and the principal coordinate systems of the two data respectively. According to the relationships as shown in figure 4-3, for the scene data, the transformation from the global coordinate system to its principal coordinate system can be represented as:

$$\overrightarrow{o_1 p_1} = \overrightarrow{o_1 o_0} + R_{01} \cdot \overrightarrow{o_0 p_1} \quad (4-3)$$

Similarly, for the model data, we can obtain:

$$\overrightarrow{o_2 p_2} = \overrightarrow{o_2 o_0} + R_{02} \cdot \overrightarrow{o_0 p_2} \quad (4-4)$$

Then, we consider the transformation between the two principal coordinate systems. Because the p_1 and the p_2 are the corresponding points, the p_1 in $C_1(o_1, \bar{x}_1, \bar{y}_1, \bar{z}_1)$ should have the same coordinates as the p_2 in $C_2(o_2, \bar{x}_2, \bar{y}_2, \bar{z}_2)$. The transformation between $\overrightarrow{o_1 p_1}$ and $\overrightarrow{o_2 p_2}$ can be represented as:

$$\overrightarrow{o_2 p_2} = \overrightarrow{o_2 o_1} + R_{12} \cdot \overrightarrow{o_1 p_1} \quad (4-5)$$

Where, $\overrightarrow{o_i o_j}$ ($i, j = 0, 1, 2$) denote the translation vectors from the origin O_i to O_j . R_{ij} ($i, j = 0, 1, 2$) denote the rotation matrices from the coordinate system C_i to the coordinate system C_j . In our method, the rotation matrices R_{ij} are the direction cosine matrices, which are specified as:

$$R_{ij} = \begin{bmatrix} \cos(\bar{x}_j, \bar{x}_i) & \cos(\bar{x}_j, \bar{y}_i) & \cos(\bar{x}_j, \bar{z}_i) \\ \cos(\bar{y}_j, \bar{x}_i) & \cos(\bar{y}_j, \bar{y}_i) & \cos(\bar{y}_j, \bar{z}_i) \\ \cos(\bar{z}_j, \bar{x}_i) & \cos(\bar{z}_j, \bar{y}_i) & \cos(\bar{z}_j, \bar{z}_i) \end{bmatrix} \quad (4-6)$$

We combine the three formulae of (4-3), (4-4) and (4-5). Finally, we can obtain the final transformation formula between the vector $\overrightarrow{o_0 p_1}$ to the vector $\overrightarrow{o_0 p_2}$, which can be represented as:

$$\overrightarrow{o_0 p_2} = R_{02}^- (\overrightarrow{o_2 o_1} - \overrightarrow{o_2 o_0} + R_{12} \cdot \overrightarrow{o_1 o_0}) + R_{02}^- R_{12} R_{01} \cdot \overrightarrow{o_0 p_1} \quad (4-7)$$

From the knowledge of spatial kinematics, the invertible matrix of a rotation matrix is equal to its transpose. Therefore, the equation (4-7) can be represented as:

$$\overrightarrow{o_0 p_2} = R_{02}^T (\overrightarrow{o_2 o_1} - \overrightarrow{o_2 o_0} + R_{12} \cdot \overrightarrow{o_1 o_0}) + R_{02}^T R_{12} R_{01} \cdot \overrightarrow{o_0 p_1} \quad (4-8)$$

The formula (4-8) is the final transformation to match the scene data to the model data. For each point of the scene data, all the parameters in formula (4-8) are easily calculated. With this formula, the scene data and the model data can be aligned roughly. Figure 4-4 and figure 4-5 show two results of the principal poses alignment.

The alignment based on principal poses is just from the view of the whole sizes of the two discrete shapes, and not considering the overlapping sections within them. Therefore, the good registration quality always assumes that there is a large enough similarity between the two shapes (see figure 4-4). If the two shapes were partial overlapping, the registration result will not be good (see figure 4-5). We will propose an additional approach based on local feature matching to improve the registration accuracy from the overlapping view.

4.4.3. Overlapping alignment

As mentioned above, the previous coarse registration methods either depend on the explicit correspondences or fail to align the overlapping regions correctly. The discrete shapes should be registered based on their overlapping sections. We developed a new method for this purpose based on curvature-based local features. As one of the most familiar attributes associated with the geometric shape, curvature provides additional local shape information which is helpful for registration process. In our approach, the local feature is represented by the pair of shape index and curvedness (see section 3.3.5) which describes the shape information completely.

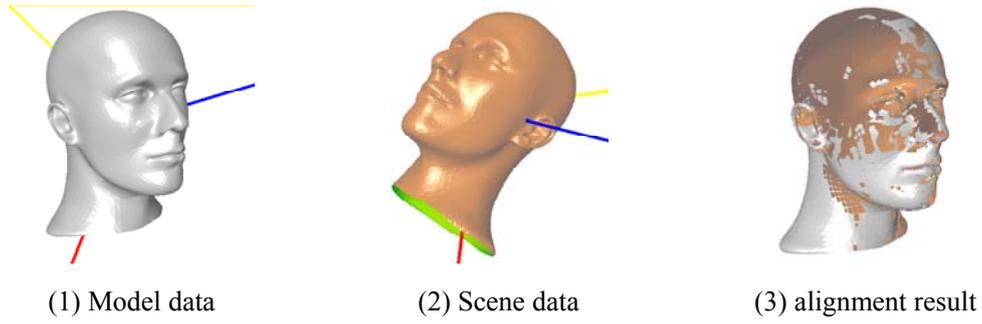


Figure 4-4: Full overlapping alignment based on principal poses

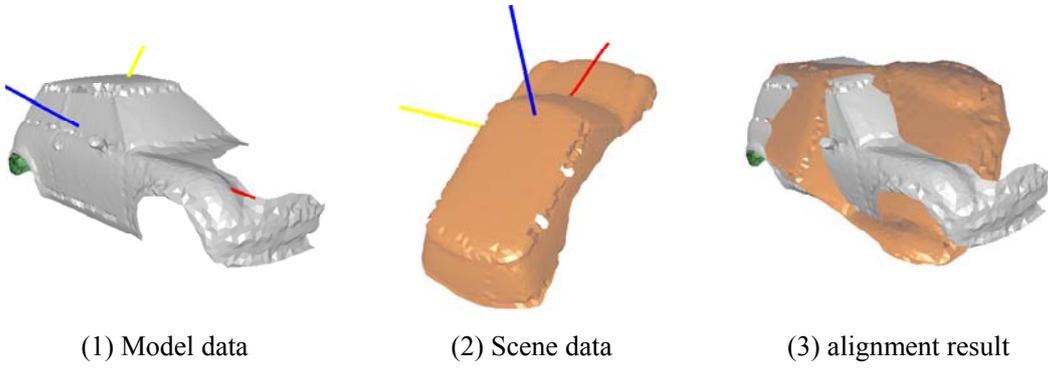


Figure 4-5: Partial overlapping alignment based on principal poses

With the hypothesis that the overlapping section is unknown, the first problem in overlapping alignment is to identify the correspondences between the two input shapes. Since the curvature information is geometry invariant, the points corresponding to the same point on the physical object should have the similar curvature attributes. The similarity between two points can be defined based on second order geometric attributes, like the pairs of shape index and curvedness, etc.

In the overlapping alignment, we use feature distance d_F to measure the similarity between two points, p in scene data and q in model data. The feature distance is defined as:

$$d_F = \begin{cases} c_p - c_q & T_p = T_q \\ \infty & T_p \neq T_q \end{cases} \quad (4-9)$$

Where, c_p , T_p denote the curvedness and surface type of the vertex p , while c_q , T_q have the similar meanings to denote the curvedness and surface type of the vertex q . All the parameters mentioned in formula (4-9) can be found their definitions in section 3.3.5. The feature distance is only considered when the two vertices have the same surface types.

Obviously, the less the feature distance is, the more similarity between two measured points is. In practice, given a point p in scene data and a threshold of d_F , denoted as th_{dF} , we can find all the points which has the less feature distance than th_{dF} in model data, denoted as $\Psi(p)$. If $\Psi(p) \neq \emptyset$, the point with the closest Euclidean distance will be defined as the corresponding point to point p . Else, $\Psi(p) = \emptyset$ means that there is no corresponding point in model data to point p .

After all vertices in scene data are traversed, a set of corresponding point pairs can be finally obtained, the algorithm proposed in section 4.5.4 is used to calculate the rotation and translation matrices to align the scene data and the model data together.

After several iterations (normally less than 10), the overlapping sections between the scene data and the model data can be aligned with good performance. In the alignment procedures, the threshold th_{dF} defines the matching resolution of two vertices, which affects the accuracy of the correspondences searching. If th_{dF} is set too large, which means the matching resolution is large, many point pairs which don't have good correspondences will be identified as corresponding point pairs. In this condition, the transformation calculation will be led to large error. By contrast, if th_{dF} is set too small, the point pairs with good correspondence will be ignored by the algorithm. The result of this condition might be that the algorithm can't find enough corresponding point pairs to calculate the transformation. Both of the two conditions will result in failure of the overlapping alignment. Experimentally, if the curvature estimation is reliable, th_{dF} set in $[10^{-3}, 10^{-7}]$ can usually generate satisfying alignments. Figure 4-6 shows an example of registration results based on different thresholds. From the figure, we can different thresholds will generate different results.

At the end of overlapping alignment, the minimum guaranteed overlapping proportion of the scene data could be estimated properly. We call it minimal overlapping proportion of scene data [CSS02], denoted as ξ_0 :

$$\xi_0 = \frac{N_C}{N_S} \quad (4-10)$$

Where, N_C is the number of the points in overlapping section of scene data which satisfy the constraint that $\Psi(p) \neq \emptyset$. N_S is the number of whole points in scene data. ξ_0 is an important parameter for the automatic fine registration process. Similarly, if N_M denotes the point number of the model data, the minimum

overlapping proportion of model data can be defined as:

$$\xi_1 = \frac{N_C}{N_M} \quad (4-11)$$

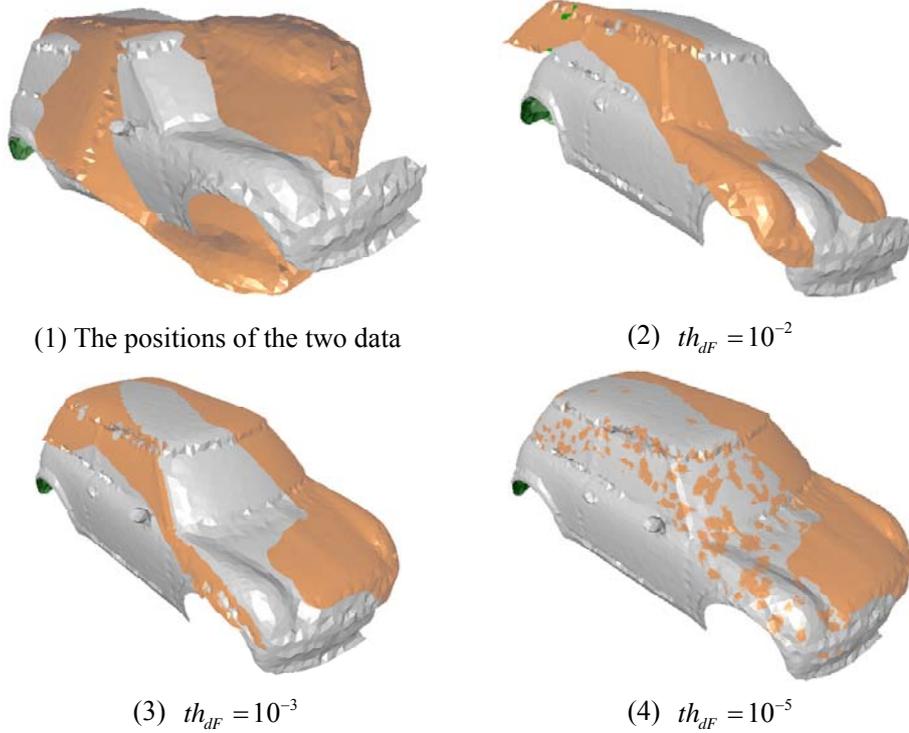


Figure 4-6: Overlapping alignment results with different

From the figure 4-6 (4), we can find that the proposed method can generate good registration results in some cases. Due to the estimation errors of the discrete curvature estimation for the noise data acquired from measurement systems, it is usually difficult to provide very accurate registration results only based on principal poses alignment and overlapping alignment. A common example is shown in figure 4-7, from where we can see the registration of the two data has large alignment errors. However, the proposed coarse registration ensures the right trend for the fine registration to generate accurate alignment.

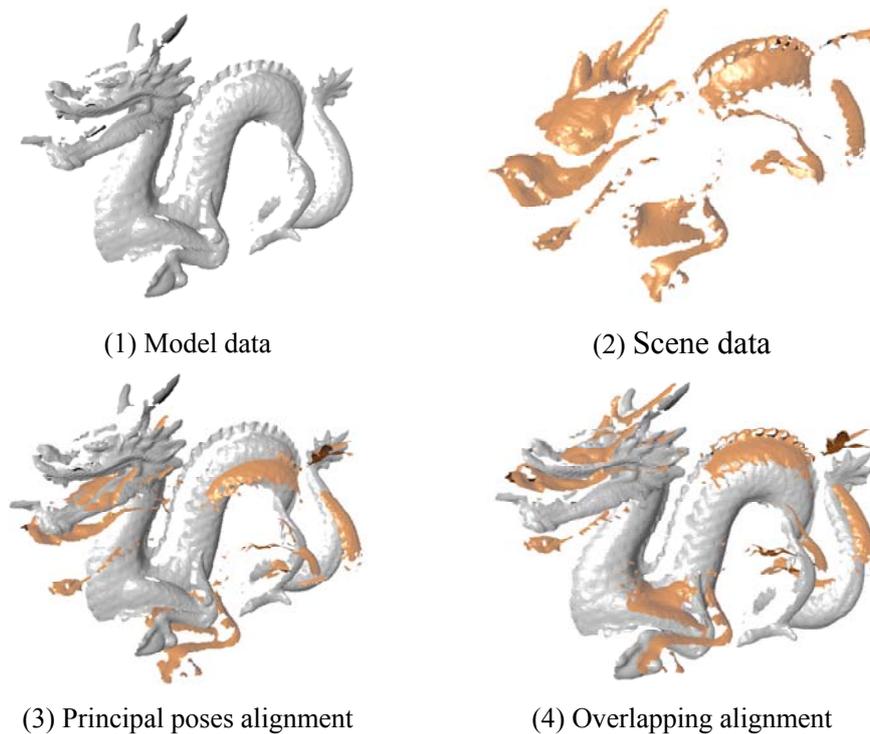


Figure 4-7: The low accurate alignment in coarse registration

4.5. Fine registration

The ICP algorithm has already been the most standard algorithm for the robust piecewise fine registration. Many ICP based variants are proposed to improve the performances of the initial ICP algorithm. But each variant has its own drawbacks [RL01]. In our study, a new extension of ICP algorithm is presented, called Curvature based Fine registration (CFR). The CFR works iteratively based on a new distance called geometric distance, which combines the Euclidean distance and the principal curvature ratio distance. The CFR can align two discrete shapes of unknown partial overlaps automatically with precise result.

4.5.1. Flow chart of the CFR method

The processing flow of the CFR method is illustrated in figure 4-8. Given a scene data S_N and a model data M_K , the CFR method can be decomposed into 5 basic steps as follows:

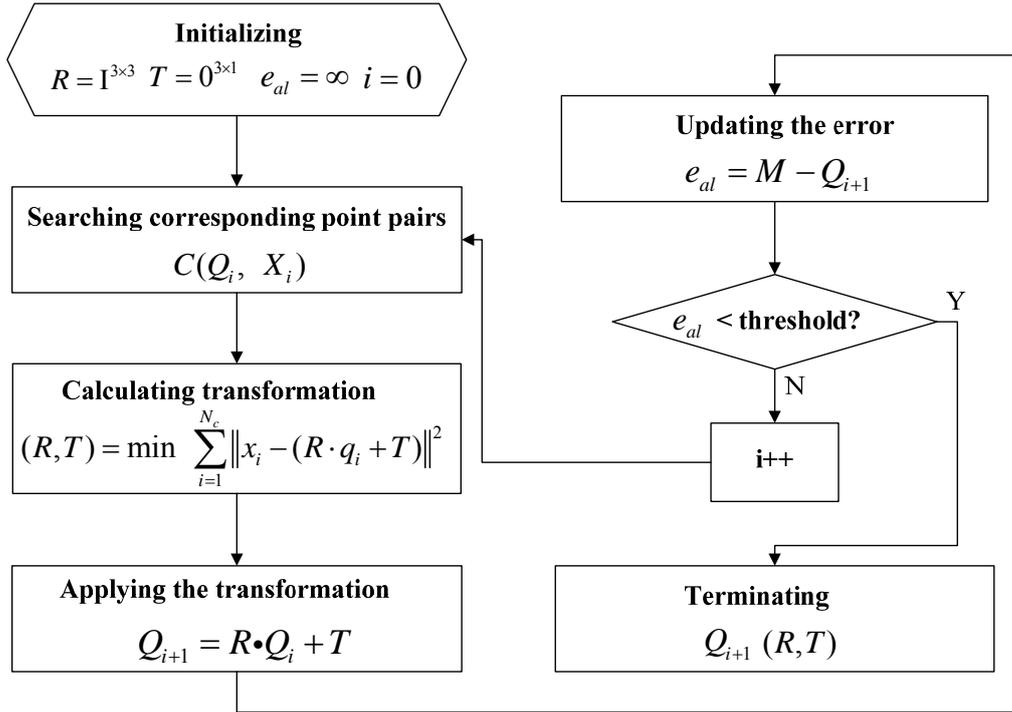


Figure 4-8: General registration flow of the CFR method

(a) **Initializing.** The rotation matrix is initialized as the 3×3 identity matrix and the translation matrix is initialized as the 3×1 zero matrix. The alignment error is initialized to an infinite value, $e_{al} = \infty$.

(b) **Searching.** The corresponding point pairs in the scene data and the model data are searched according to the nearest geometric distance defined in section 4.5.2. The searched corresponding point pairs are then weighted and the incorrect corresponding point pairs are rejected [CSS02]. At the i^{th} iteration, the final set of the corresponding point pairs is represented as $C(Q_i, X_i)$, where Q_i denotes the corresponding points in the scene data and X_i denotes the corresponding points in the model data.

(c) **Computing.** Assigning the error metric based on $C(Q_i, X_i)$ and computing the rotation and translation matrices to transform Q_i to X_i by minimizing the error metric.

(d) **Applying.** Applying the transformation matrices to the whole points of scene data and updating the scene data for next iteration.

(e) **Terminating.** The iteration is terminated when the change of error metric is below a given threshold.

4.5.2. Geometric distance definition

The most important step of ICP based algorithms consists in selecting the corresponding point pairs within the two discrete shapes [ALD07]. Since the accuracy of the search for corresponding points affects the estimation of the transformation parameters for registration, the output of this phase has a major impact over the downstream stages and influences the overall performance of the approach. In order to enhance the performance of this phase, we introduce the geometric distance to measure the similarity between two points.

Considering an arbitrary point in the scene data, let's denote it as p_i , and an arbitrary point in the model data, denoted as q_j , the geometric distance d_g is defined as:

$$d_g(p_i, q_j) = \lambda \cdot d_e + (1 - \lambda) \cdot k \cdot d_c \quad (4-12)$$

Where, d_e is the Euclidean distance between p_i and q_j :

$$d_e = \|p_i - q_j\| \quad (4-13)$$

d_c is defined as the curvature ratio distance between p_i and q_j . It has the same dimension scale with d_e .

$$d_c = \sqrt{(\rho_{1p} - \rho_{1q})^2 + (\rho_{2p} - \rho_{2q})^2} \quad (4-14)$$

ρ_1 and ρ_2 are the maximum and the minimum principal curvature radii respectively, which are defined as:

$$\rho_1 = \frac{1}{\kappa_1} \quad \rho_2 = \frac{1}{\kappa_2} \quad (4-15)$$

Where, κ_1 and κ_2 represent the maximum and minimum principal curvatures respectively. If ρ_1 and ρ_2 happen to be infinite values, ρ_1 and ρ_2 are initialized as very large values, such as 1×10^6 for example.

k is a constant to normalize d_c . In our method, it is defined as:

$$k = \frac{\max(l_S, l_M)}{\max((\max(\rho_1) + \max(\rho_2))_S, (\max(\rho_1) + \max(\rho_2))_M)} \quad (4-16)$$

Considering the minimum bounding box of the model data, it has three size parameters (length, width and height). l_M is the maximum value of the size parameters. Similarly, l_S is defined as the maximum value of the size parameters of the minimum bounding box of the scene data.

$\lambda \in [0, 1]$ is a user predefined coefficient to balance the contributions of d_c and d_e . The influence of λ with different values to the performance of registration is discussed in section 4.6.4.

4.5.3. Corresponding point pairs searching

In our method, a corresponding point pair is composed of a given point in scene data and the point in the model data which has the closest geometric distance to the given point. Given a point q_i in scene data, if the point x_k in model data satisfies the equation (4-17), the point pair $\{q_i, x_k\}$ can be viewed as a corresponding point pair.

$$d_g(q_i, M) = \min_{j \in \{1, \dots, N_M\}} d_g(q_i, x_j) = d_g(q_i, x_k) \quad (4-17)$$

The CFR searches the corresponding point in model data for each point in scene data. Some accelerating searching algorithms, like k-D tree, etc. can also be used here to speed up the CFR searching.

The hypothesis of the original ICP algorithm is that the overlapping section of scene data is always 100%. That is to say, each scene point will be searched a corresponding point in model data. In our method, we consider the more general condition that the scene data is not 100% overlapped, which means that the minimal overlapping proportion of scene data is less than one: $\xi_0 \leq 1$.

Because of partial overlapping, some scene points which have no corresponding point are also assigned a model point according to the closest geometric distance, which will lead to large errors in transformation calculation. The mismatching point pairs should be identified and rejected.

In our method, we consider two kinds of point pairs which should be rejected: the point pair that has large geometric distance and the border point pairs.

(a) Rejection of the point pairs with large geometric distance. Firstly, the

corresponding point pairs are sorted according to their geometric distances increasingly. The proportion ξ_r , computed from the bottom of the point pairs sequence will be rejected. The proportion ξ_r is determined based on the minimum overlap proportion ξ_0 , as $\xi_r = 1 - \xi_0$. It can be set manually when the overlapping section is known or automatically according to the formula 4-10 mentioned in section 4-4-3.

(b) Rejection of the border point pairs. A border point pair is the pair which contains the end points. The border points generally have larger uncertainty than the points in other regions, and often lead to wrong trend in a registration process. If any point in a point pair is identified as border point, the corresponding point pair will be rejected.

In each step of the iteration, we can finally obtain the corresponding point pairs after rejection process. The set of corresponding point pairs is denoted as $C(Q_i, X_i)$, where, Q_i and X_i respectively represent the corresponding points in the scene data and in the model data. Obviously, the number of points in Q_i and in X_i are the same, denoted as N_c . The point pairs in $C(Q_i, X_i)$ have high reliability of correct matching.

4.5.4. Corresponding point pairs registration

Given the point pairs $C(Q, X)$, the corresponding point pairs registration is a necessary procedure yielding to a least squares transformation to align Q and X . Assuming that each point q_i in Q corresponds to the point x_i in X with the same index. The objective function to be minimized in this phase is:

$$f(R, T) = \sum_{i=1}^{N_c} \|x_i - (R \cdot q_i + T)\|^2 \quad (4-18)$$

Where, the rotation matrix R satisfies the restriction [Ber07]:

$$\{R \in \mathbb{R}^{3 \times 3} \mid R^T R = I^3, \det(R) = 1\} \quad (4-19)$$

There are two most used methods to determine the rotation matrix: Singular Value Decomposition (SVD) method [AHB87, Cha95], and unit quaternion-based method [BM92, Hor87].

(1). *Singular Value Decomposition (SVD) method*

Firstly, we define a 3×3 matrix H :

$$H = \sum_{i=1}^{N_c} (q_i - \mu_Q)(x_i - \mu_X)^T \quad (4-20)$$

Where, μ_Q and μ_X are the centroids of the point sets Q and X .

Thereby, the SVD of H can be represented as follows:

$$H = U \Lambda V^T \quad (4-21)$$

Where, U and V are the unit matrices of singular vectors of H . Λ is defined as follows:

$$\Lambda = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \quad (4-22)$$

Where, σ_i ($i=1, 2, \dots, r$) are the singular values of H . $r = \text{rank}(H)$ is the rank of H , and $\sigma_i(H) = \sqrt{\lambda_i(H^T H)}$, $\lambda_i(H^T H)$ are the eigenvalues of the matrix $H^T H$.

Based on U and V , we can obtain a unit orthogonal matrix W which is defined as:

$$W = UV^T \quad (4-23)$$

W satisfies the constraint $|\det(W)|=1$. If $\det(W)=1$, the matrix W is the rotation matrix $R=W$. If $\det(W)=-1$, The matrix W is not the rotation matrix but a reflection matrix. In other words, one can obtain a reflection matrix of the desired rotation matrix. The formula (4-23) can be modified as the following formula to solve this problem [Cha95].

$$R = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{bmatrix} V^T \quad (4-24)$$

(2). *Unit quaternion-based method*

Besides the SVD method, another popular method used to calculate the rotation matrix is the unit quaternion based method. We prefer the unit quaternion method

because the reflection is not desired in our registration process. Therefore, we implemented the unit quaternion-based method and the procedures to generate the optimal rotation and translation matrices are as follows:

(a) Cross-covariance matrix calculation.

The cross-covariance matrix of the sets Q and X is given by:

$$M_{qx} = \frac{1}{N_C} \sum_{i=1}^{N_C} (q_i \cdot x_i^T) - \mu_Q \mu_X^T \quad (4-25)$$

Where, $\mu_Q = \frac{1}{N_C} \sum_{i=1}^{N_C} q_i$ and $\mu_X = \frac{1}{N_C} \sum_{i=1}^{N_C} x_i$ are the respective centroids of the sets Q and X .

(b) Quaternion matrix calculation.

To build the quaternion matrix, we define an anti-symmetric matrix A , which has the representation as:

$$A = M_{qx} - M_{qx}^T = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \quad (4-26)$$

Then, a row vector $\Delta = [a_{12}, a_{20}, a_{01}]$ is formed from the matrix A and the quaternion matrix can be constructed as follows:

$$W = \begin{bmatrix} tr(M_{qx}) & \Delta \\ \Delta^T & M_{qx} + M_{qx}^T - tr(M_{qx}) \cdot I \end{bmatrix} \quad (4-27)$$

Where, $tr(M_{qx})$ is the trace of M_{qx} , and I is the 3×3 identity matrix.

The unit eigenvector corresponding to the maximum eigenvalue of W , let it be denoted as $w_R = [w_0, w_1, w_2, w_3]$, constructs a unit quaternion. Where, $w_0 \geq 0$, and $w_0^2 + w_1^2 + w_2^2 + w_3^2 = 1$. w_R estimates the optimal rotation and can be used to generate the transformation matrices.

(c) Transformation matrices calculation.

With the unit eigenvector w_R , the rotation matrix can be formed as:

$$R = \begin{bmatrix} w_0^2 + w_1^2 - w_2^2 - w_3^2 & 2(w_1w_2 - w_0w_3) & 2(w_1w_3 + w_0w_2) \\ 2(w_1w_2 + w_0w_3) & w_0^2 - w_1^2 + w_2^2 - w_3^2 & 2(w_2w_3 - w_0w_1) \\ 2(w_1w_3 - w_0w_2) & 2(w_2w_3 + w_0w_1) & w_0^2 - w_1^2 - w_2^2 + w_3^2 \end{bmatrix} \quad (4-28)$$

The translation matrix is formed as:

$$T = \mu_x - R \cdot \mu_Q \quad (4-29)$$

(d) Applying the transformation.

The last step is to apply the rotation matrix and the translation matrix to each point in scene data to transform them into a new scene data. The original scene data is then updated to the new scene data. The CFR process is repeated until the convergence condition is satisfied.

4.5.5. Convergence condition

The CFR method registers the 3-D point data based on iterative operation. Considering the i^{th} iteration, the alignment error between the corresponding pairs, Q_i and X_i , is defined as:

$$e_{al} = \frac{1}{N_C} \sum_{k=1}^{N_C} \|x_{ki} - (R_i \cdot s_{ki} + T_i)\|^2 \quad (4-30)$$

If the alignment error e_{al} is less than a pre-defined threshold, the iteration will terminate.

Besl and McKay [BM92] proved that the ICP algorithm always converges monotonically to a local minimum with respect to the mean-square distance objective function defined in equation (4-18). Since the curvature information at a point is a geometric invariant, and it doesn't affect the convergence condition. There is no need to consider it into the alignment error minimization.

The CFR method has the similar convergence condition to the original ICP algorithm. However, because the curvature information helps improving the accuracy of corresponding point pair search, the convergence speed is much faster than the original ICP algorithm. The time performance of CFR is analyzed in section 4.6.4. According to [BM92], the general iterations of the original ICP algorithm is 40~50, but experimentally, after have been performed meanly 20 iterations, the CFR method

can usually generate satisfying results.

4.6. Testing results

We implemented the algorithms mentioned in the previous sections. All the programs were written in Visual C++ 8.0 and using the Open/GL libraries for visualization. We also implemented the initial ICP algorithm for comparison. In our experiments, According to the minimum overlapping proportions of the scene data and the model data, ξ_0 and ξ_1 (defined in formulae 4-10 and 4-11), we consider three kinds of registration: full-full overlapping and full-partial overlapping and partial-partial overlapping.

4.6.1. Full-full overlapping

If the overlapping sections cover the full size of the scene data ($\xi_0 = 1.0$) and also cover the full size of the model data ($\xi_1 = 1.0$), the registration can be classified into the full-full overlapping case.

Figure 4-9 shows an example of the full-full overlapping registration. The point data of armadillo are downloaded from Computer Graphics Laboratory of Stanford University [Stan]. We translate the scene data with 10mm and rotate it with 45° angles. The scene data contain 4078 points, while the model data compose of 7014 points as shown in figure 4-9 (1) and (2) respectively. The coarse registration result is given by figure 4-9 (3). The figures 4-9 (4) and (5) illustrate the final fine registration results of the initial ICP algorithm and the CFR method respectively.

In this case, the initial ICP algorithm is executed with 40 iterations and the final alignment error is 0.26298mm. The CFR method performs 20 iterations and the final alignment error is 0.10264mm. Obviously, the comparison shows that the CFR method provides the registration results with better performances.

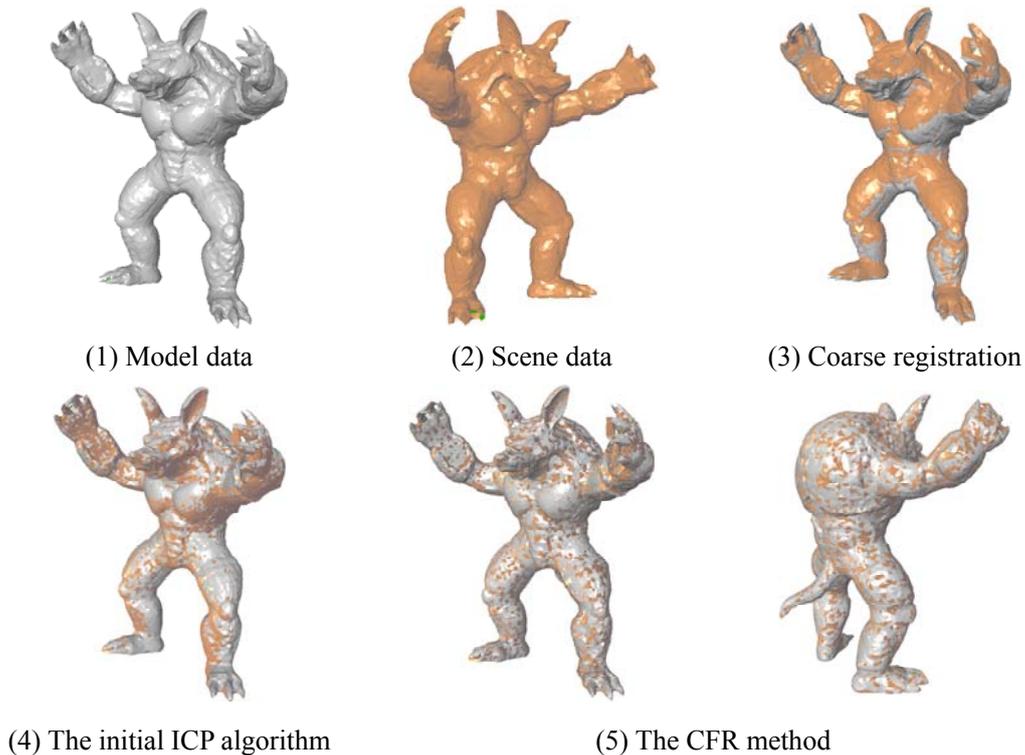


Figure 4-9: Full-full overlapping registration of the armadillo data

4.6.2. Full-partial overlapping

When the scene data is fully overlapped and the model data is partial overlapped, which means that $\xi_0 = 1.0$ and $\xi_1 < 1.0$, the registration belongs to the full-partial case.

Figure 4-10 presents an example of the full-partial overlapping registration. In this case, the model data which contains 11472 points captures the complete shape of the Chinese dragon. The scene data which contains 8897 points only captures the partial shape of the Chinese dragon. The two data are as shown in figure 4-10 (1) and (2).

Figure 4-10 (3) and (4) show the coarse registration results. Because of the different shapes of the two data, the principal poses alignment registers the two data with large error (see Figure 4-10 (3)). The alignment quality is improved by overlapping alignment (see figure 4-10 (4)), which can provide the right trend for fine registration.

Figure 4-10 (5) shows the fine registration result using the initial ICP algorithm

with 40 iterations. Figure 4-10 (6) presents the result of the CFR method with 20 iterations. In this case, the alignment error of the initial ICP algorithm is 0.00582mm and the CFR method is 0.00105mm. Therefore, the CFR method provides better performances for the full-partial overlapping registration than the initial ICP algorithm.

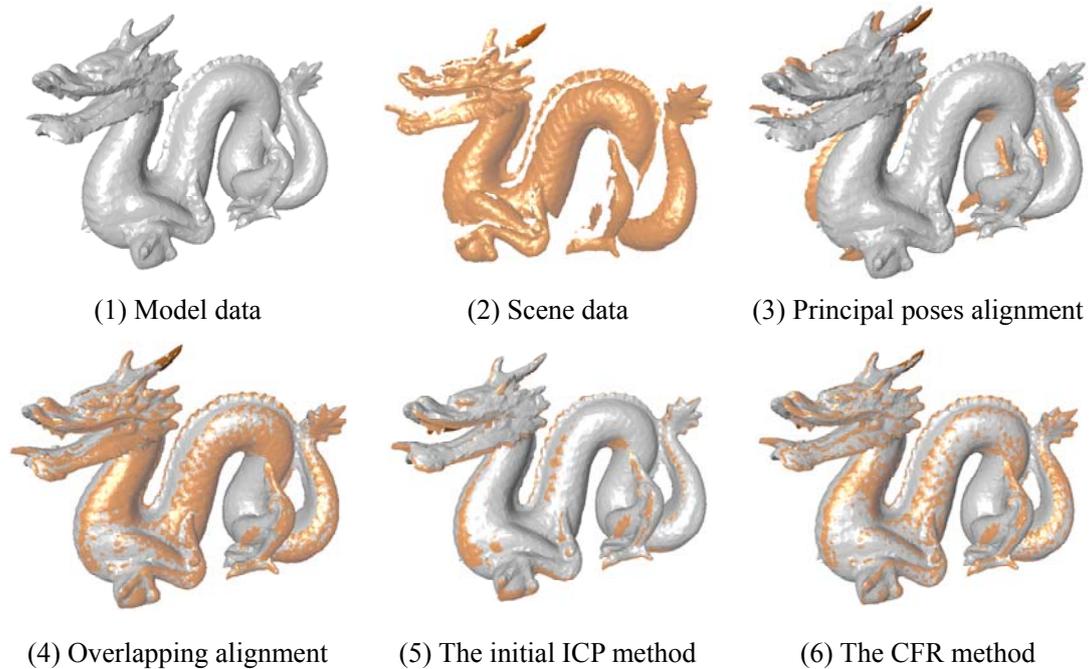


Figure 4-10: Full-partial overlapping registration of the Chinese dragon data (Dragon 1)

4.6.3. Partial-partial overlapping

Partial-partial overlapping registration considers both of the scene data and the model data are partially overlapped, which means $\xi_0 < 1.0$ and $\xi_1 < 1.0$. The point data acquired from different views of the sensors are usually only partially overlapped. It is hence very necessary to register the two data only with partial overlapping.

An example of partial-partial overlapping is shown in figure 4-11. The model data and the scene data of the Chinese dragon are both partially overlapped. In order to distinguish this case from the full-partial case (figure 4-10), we denote this partial-partial case as “Dragon 2” and denote the full-partial case as “Dragon 1”.

In Figure 4-11, the scene data and the model data are acquired by scanning over the object with different directions. The angle between the two scanning directions is 144° . The model data contains 8897 points and the scene data contains 7530 points.

Figure 4-11 (1) shows the relative positions of the two discrete shapes. The coarse registration results are shown in figure 4-7. Figure 4-11 (2) shows the final result of the initial ICP algorithm with 40 iterations. The registration result of the CFR method with 9 iterations is presented in figure 4-11 (3). The alignment errors of the two methods respectively are 0.006222mm and 0.000968mm.

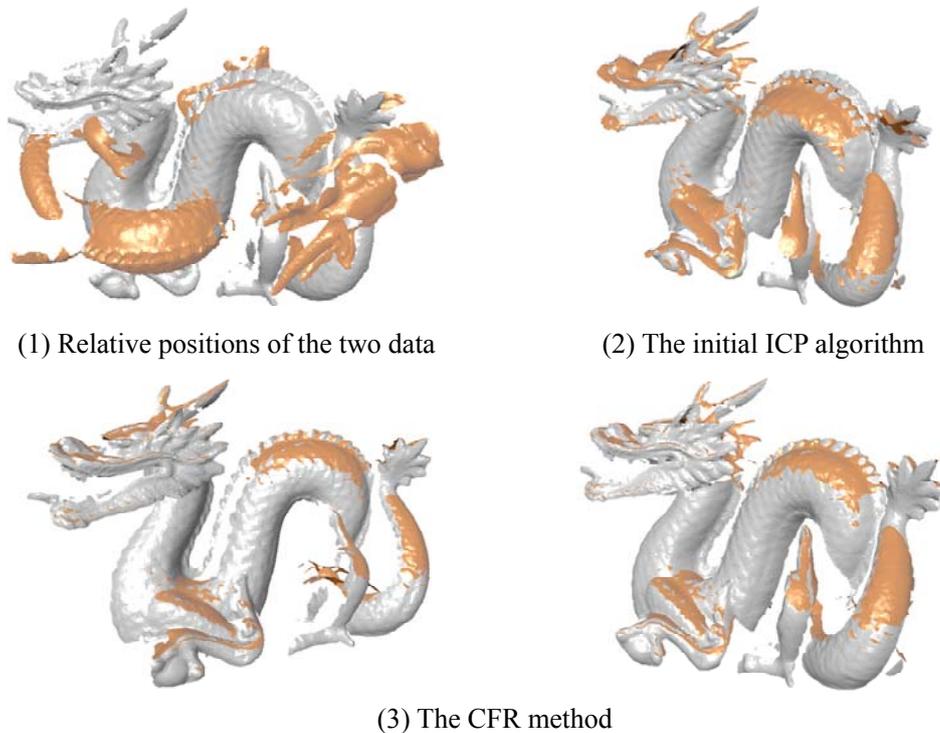


Figure 4-11: Partial-partial overlapping registration of the Chinese dragon data (Dragon 2)

Figure 4-12 and figure 4-13 show another two examples of partial-partial overlapping registration: the Stanford bunny and the future Buddha. In the case of the Stanford bunny, the initial ICP algorithm runs 40 iterations with an alignment error of 0.108mm. The CFR iterates 15 times with an alignment error of 0.00184mm. In the case of future Buddha, the initial ICP algorithm iterates 40 times while the CFR runs 13 iterations. The alignment error of the initial ICP algorithm is 0.015mm and of CFR is 0.000848mm.

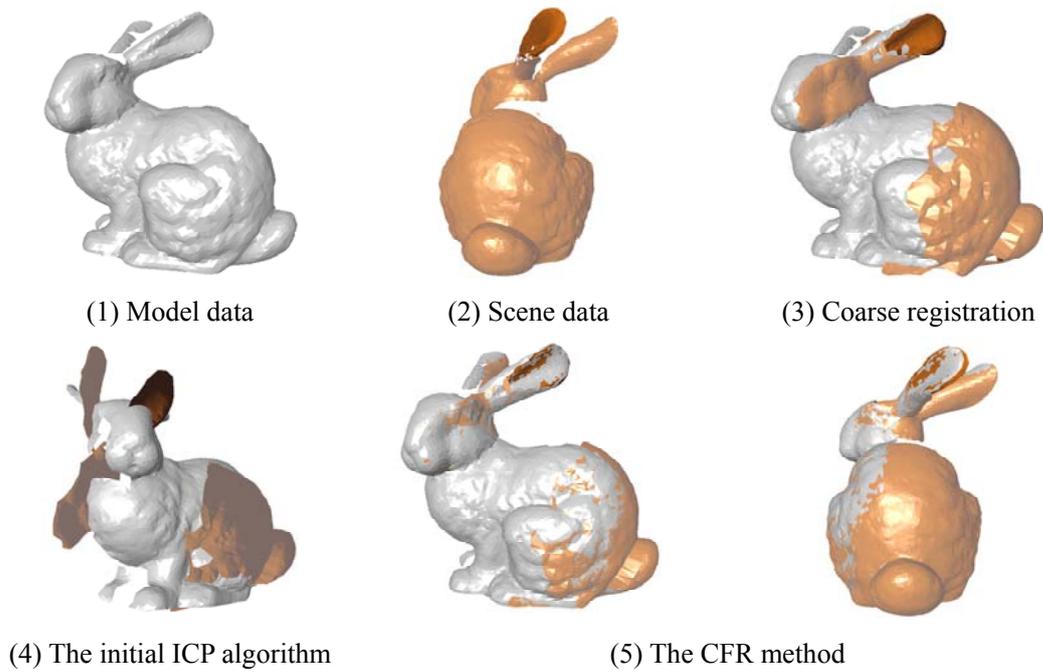


Figure 4-12: Partial-partial overlapping registration of the Stanford bunny data

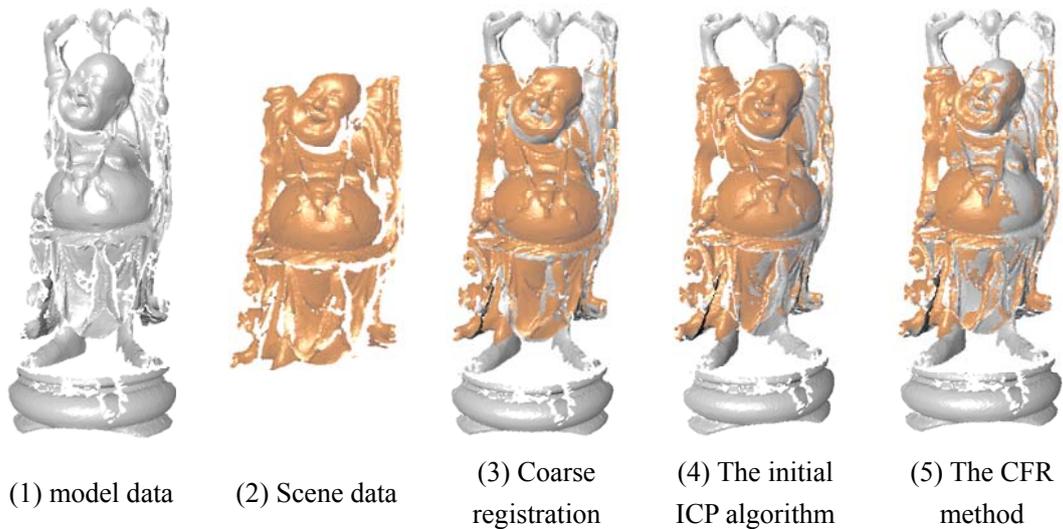


Figure 4-13: Partial-partial overlapping registration of the future Buddha data

From the testing results, we can see that the alignment errors of the original ICP algorithm usually are large and can fail to register the data (see figure 4-12). By contrast, the CFR method can provide more accurate results.

4.6.4. Performance analysis

In this section, we first present the time performance of our registration method. And then, the influence of the parameter λ mentioned in formula (4-12) on the

whole performance of the method is discussed. Finally, we discuss the limitations of the proposed method. All the results are obtained based on the MS windows operation system with the Intel Core 2 CPU (1.83GHz) and a 1.00G RAM.

(1). Time performance in coarse registration

The computational cost of coarse registration contains two parts: the cost for principal poses alignment and the cost for overlapping alignment.

The principal poses alignment is not an iteration based algorithm, the time complexity includes two portions. The first one is the principal poses estimation. The principal poses of the scene data and the model data can be estimated synchronously. Thereby, the computational cost of this phase should be the larger one within costs for the scene data and the model data, that is: $\max\{O(N_M), O(N_M)\}$. The second one is the transformation computation and its assignment to the scene data. The time complexity of transformation calculating is $O(1)$, and the applying the transformation is $O(N_S)$.

The overlapping alignment is based on iteration operation. In each iteration process, the computational cost contains three stages. The first one is searching the overlapping sections. The model data is pre-classified into 10 clusters based on the shape index, the corresponding point for each scene data only needs to be searched in the corresponding cluster which can reduce the computational cost, so the time cost in this step is $O(\frac{1}{10}N_S \cdot N_M)$. The computational costs in transformation calculation and the scene data updating are listed in the table 4-1.

Table 4-1: Computational costs in coarse registration

Registration stages	Step	Computational cost
Principal poses alignment	Principal poses estimation	$\max\{O(N_M), O(N_M)\}$
	Transformation estimation	$O(1) + O(N_S)$
Overlapping alignment	Overlapping searching	$O(N_S \cdot N_M / 10)$
	Transformation calculation	$O(N_C)$
	Transformation applying	$O(N_S)$

The alignment accuracy for coarse registration doesn't require very highly. Hence,

the overlapping alignment doesn't need a lot of iterations. Experimentally, 5 is enough for a general coarse registration purpose. Table 4-2 shows the time performances of the tested cases in coarse registration, in where, i_n denotes the iteration number.

Table 4-2: Time performances of the tested cases in coarse registration

Name	N_S	N_M	e_{al} (mm)	i_n	t (s)
Armadillo	4078	7014	0.8236332	5	133.8
Dragon 1	8897	11472	0.00197941	5	81.48
Dragon 2	7530	8897	0.00577915	5	152.6
Bunny	3533	3156	0.0266536	5	14.71
Buddha	10434	11831	0.001915	5	192.1

(2). *Time performance in fine registration*

Each iteration process includes 3 main steps in both the original ICP algorithm and the CFR method: searching the corresponding point pairs; computing the transformation and applying the transformation to the scene data. For convenience, we denote the 3 steps as searching, computing and updating respectively. The 3 main steps take up most of the time cost for ICP based methods. In our discussion, we only consider the time cost in the 3 main steps.

In the searching phase, the computational cost of corresponding point searching for per scene point is $O(N_M)$, so the total computational cost is $O(N_S \cdot N_M)$. The transformation should be applied to all the points in scene data, therefore, the computational cost in the updating phase is $O(N_S)$. The main difference within the two methods is in the calculation phase. In the initial ICP algorithm, the calculating is using all the scene points, the time cost is $O(N_S)$; by contrast, the transformation calculation is only based on the corresponding point pairs, so that the time cost should be $O(N_C)$. In the worst condition, $O(N_C) = O(N_S)$ occurs for the full registration.

The table 4-3 gives the computational costs in each step of the two methods. Finally, the total computational cost (C_{total}) should be the sum of the computation cost in the three phases multiplied by the number of iterations (i_n):

$$C_{total} = i_n \cdot (C_S + C_C + C_U) \quad (4-32)$$

As presented before, the CFR method needs less iterations than the initial ICP algorithm in general. The total computational cost of the CFR method should be much less than the initial ICP algorithm. The time performances of the tested cases are listed in table 4-4, from where, we can see that the CFR method can provide the results with better performances both in accuracy and convergence speed.

Table 4-3: Time complexities of the two compared methods

Method	Searching C_S	Calculating C_C	Updating C_U
The Initial ICP	$O(N_S \cdot N_M)$	$O(N_S)$	$O(N_S)$
The CFR	$O(N_S \cdot N_M)$	$O(N_C)$	$O(N_S)$

Table 4-4: Performances of the two compared algorithms

Data	N_S	N_M	Original ICP			CFR		
			e_{al_icp} (mm)	i_n	t_{icp} (s)	e_{al_cfr} (mm)	i_n	t_{cfr} (s)
Armadillo	4078	7014	0.26298	40	2607.61	0.10264	20	1543.22
Dragon 1	8897	11472	0.0058244	40	2978.55	0.00105624	20	1780.69
Dragon 2	7530	8897	0.0062218	40	1860.54	0.00096801	9	746.502
Bunny	3533	3156	0.108053	40	492.732	0.0018138	15	159.853
Buddha	10434	11831	0.015292	40	2911.78	0.0008479	13	1644.25

(3). *Discussions*

(a) The influence of the parameter λ on the registration performance.

The coefficient λ in the formula (4-12): $d_g(p_i, q_j) = \lambda \cdot d_e + (1 - \lambda) \cdot k \cdot d_c$ is used to balance the contributions of the Euclidean distance d_e and the curvature ratio distance d_c to the geometric distance d_g . When λ varies in $[0, 1]$, it affects the performances of the proposed registration method. We analyze the influence of λ on the final registration performance using two point data of Stanford bunny, the initial positions of the two data are shown in figure 4-14.

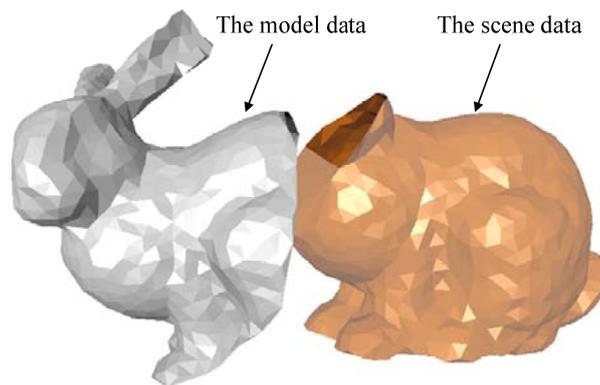


Figure 4-14: The two data used to test the influence of λ on registration performance

Table 4-5 shows the different performances of the proposed method with different values of λ . From the results, we can see that the proposed method could fail to register the point data when λ is set too large or too small. In the tested case, when λ varies within the region $[0.1, 0.75]$, the proposed algorithm can always provide reliable registration results with satisfying accuracies. However, in such interval, the smaller the λ is, the less iterations the method needs to obtain the satisfying accuracy. From table 4-5, we can see that it is necessary to combine the Euclidean distance and curvature ratio distance together to get better registration performance.

(b) The limitations of the proposed method.

As discussed above, the proposed method is a robust and accurate solution to the registration problem. However, some limitations should be discussed to restrict the application range of the proposed method.

The first limitation is the registration of unitary shapes. The unitary shape has theoretically the same curvatures everywhere. In these cases, the curvature is not helpful to determine the overlapping regions, and the proposed method regresses to the classical ICP algorithm. If the two shapes have not full-full overlapping, generally, the overlapping regions are indefinite, so it is difficult to identify the correct correspondences automatically. Therefore, the correct correspondences within the two data should be determined with a user-guidance. Otherwise, the method will give a solution, but, this solution might align the two data with unexpected results.

The figure 4-15 shows the examples of that failure model. The figure 4-15 (1) shows the satisfying registration result when the two input data are full overlapped. However, if the two data are partially overlapped, as shown in figure 4-15 (2), both the CFR method and the initial ICP algorithm are unable to provide acceptable results.

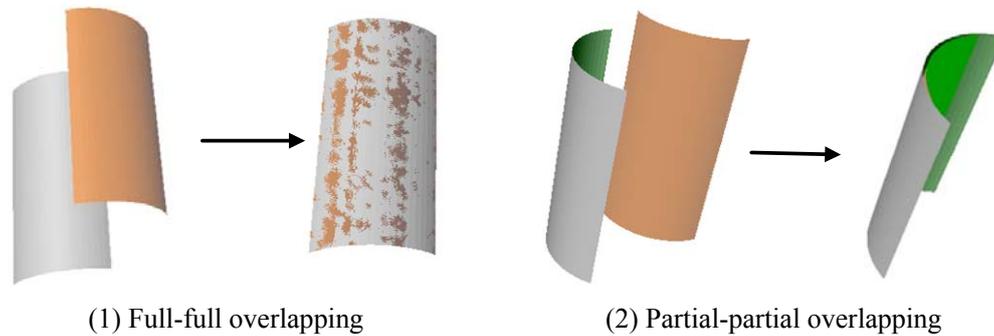


Figure 4-15: Registration of unitary shapes

Another limitation is that the CFR algorithm assumes that the correspondences between the two input data are sufficient enough to compute the transformation. If there is not enough corresponding point pairs in the two data, the method will fail to register them with satisfying results.

Several reasons may cause this problem. The first one is the unreliable curvature estimation. If the point density of one input data is not sufficient to estimate reliably curvatures, the curvature ratio distance calculation may also be unreliable and cause failure to register the data correctly. An example is shown in figure 4-16 (1). The second one is there is no sufficient overlapping sections between the two data. The minimum proportions of the overlapping sections are so small that they can't provide enough information to calculate the transformation. This condition may also lead to a wrong registration. An example is given by figure 4-16 (2).

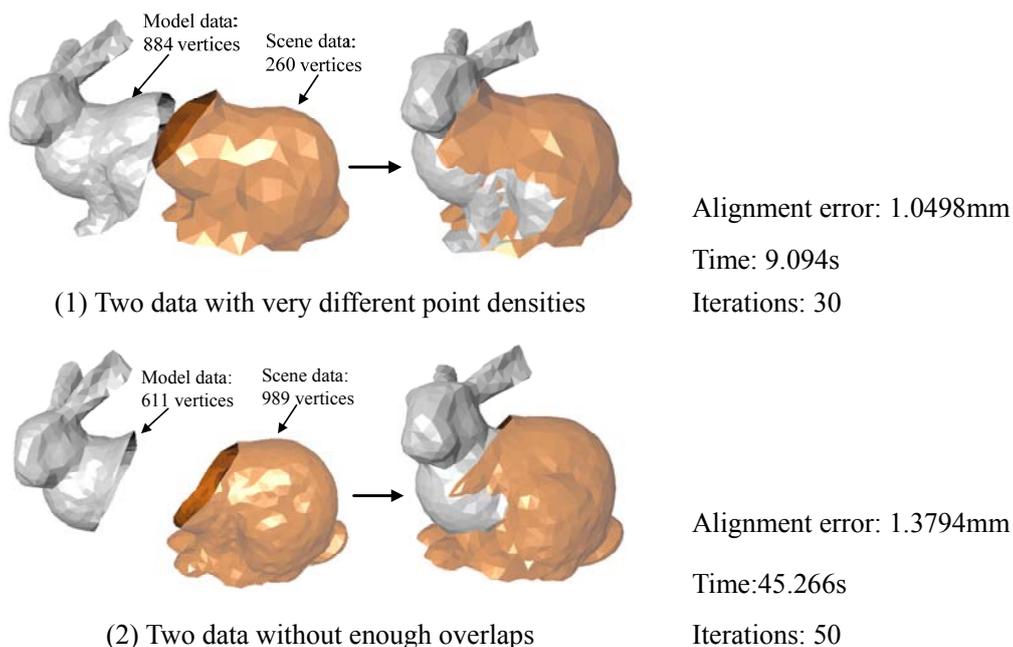
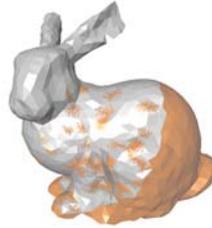
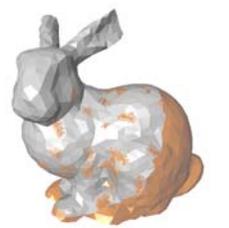
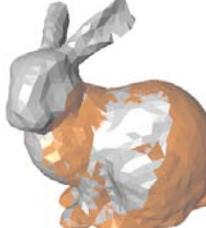


Figure 4-16: Registration of the data without sufficient correspondences

Table 4-5: The performances of the proposed method with different values of λ

λ	1.0	0.9	0.8	0.75	0.65
Time (s)	124.094s	124.265s	124.016s	111.859s	95.235s
Alignment error (mm)	0.399959mm	0.621262mm	0.462344	1.691×10^{-5} mm	1.672×10^{-5} mm
iterations	30	30	30	26	22
Result visualization					
λ	0.5	0.35	0.1	0.05	0.0
Time (s)	57.922s	45.235s	28.969s	24.769s	122.938s
Alignment error (mm)	1.714×10^{-5}	1.696×10^{-5}	1.686×10^{-5}	6.191×10^{-4} mm	2.82433mm
iterations	13	10	6	5	30
Result visualization					

4.7. Conclusion

In this chapter, we proposed a robust method for discrete shapes registration. The method performs the registration in two phases: coarse and fine registration. The curvature information is used to improve the registration performance in both phases.

In coarse registration, the principal poses of the two discrete shapes are first estimated based on PCA and they are aligned accordingly. A new distance based on shape index and curvedness is used to search the overlapping regions.

The fine registration is based on the initial ICP algorithm. However, instead of Euclidean distance, we defined a new distance, named geometric distance which combines the Euclidean distance and curvature ratio distance to measure the similarity between two points. The iterative fine registration based on geometric distance robustly provides accurate registration results with only partial overlapping regions. The comparative analysis with the initial ICP algorithm is presented and the results show that the proposed method has better performance both in registration accuracy and convergence speed.

The proposed registration method doesn't require any known correspondences. It is insensitive to the initial positions of the two input data and robust to provide registration results with satisfying performances.

The limitations of the proposed method are also discussed.

Chapter 5

Discrete shape recognition and segmentation

5.1. Introduction

Recovering and extracting the useful information from discrete shapes play critical roles in the discrete shape modeling domain. If a large and complex digitized model contains various features or freeform shapes, it would be very difficult to process it directly. Generally, the shape recognition and understanding processes are based on the decomposition of the data into smaller parts [Sha08]. The decomposition or segmentation process is used to group the initial model into a set of sub-components based on predefined criteria. Each of the segmented regions then has an appropriate, recognizable meaning [LA08, Sha08].

The shape recognition and segmentation are the most critical parts of discrete shape processing [VMC97]. Many applications including classification [JM07], object recognition [DGG04, SP08], reverse engineering [AMF00, BV04, DVV07, WKW02 YL99], and dimensional metrology [LX08, SHM01] need to solve the segmentation problem.

This chapter presents new methods for discrete shape recognition and segmentation. Finally, several contributions are proposed as a local surface type recognition based on shape index; vertex clustering and connected region partition. The methods presented in this chapter provide satisfying results for segmentation of discrete shapes.

This chapter is organized as follows:

Section 5.2 presents the related works. Section 5.3 gives an overview of our method. Section 5.4 presents the detailed procedures for local surface type recognition. Section 5.5 proposes the method for vertex clustering and cluster refining. Section 5.6 presents the procedures to generate the final segmentation results. Various testing cases are presented in section 5.7. Section 5.8 concludes this chapter.

5.2. Literature review

Considerable research activities in discrete shapes segmentation have been ensued in recent years, because of its great importance in shape modeling and understanding domain. Generally, according to different applications, the existing segmentation methods can be classified into two categories. The first one is devoted

to segmenting the discrete model of more natural object into pieces of meaningful regions from the viewpoint of human cognition. Most of the segmentation methods in computer graphics, biological and medical applications, are devoted to this purpose. The second category is aiming at partitioning the discrete model into patches and each patch can be fitted by a single, mathematically analyzable shape. The applications in engineering, especially in mechanical engineering, such as reverse engineering, dimensional metrology, belong to this category. For example, an animal model is segmented into head, limbs, trunk, etc. can be classified into the first category, while a mechanical part can be decomposed into planes, cylinders, spheres, etc. belongs to the second category.

For the applications classified in the first category, most of the existing methods are based on iterative clustering algorithms. Lien and Amato propose a partition method based on approximate convex decomposition [LA08] instead of the common used convex decomposition [CDS97]. The general strategy of their method is to iteratively identify the most concave features in the current decomposition, and then to partition the polyhedron so that the concavity of the identified features is reduced. This process continues until all components in the decomposition have acceptable concavity. Shlafman et al. [STK02] implemented the k -means clustering algorithm to iteratively improve the decomposition. Katz and Tal [KT03] improved the algorithm of Shlafman et al. by using fuzzy clustering and minimal boundary cuts to smooth the boundaries between the segmented patches. Lai, et al [LHM09] presented an efficient segmentation method based on random walks. Their method is rapid, but strongly relies on the initial seed points chosen and needs a coarse segmentation process. Jagannathan and Miller [JM07] developed a region growing approach based on curvedness to segment the 3D mesh. A curve skeleton-based algorithm developed by Reniers and Telea [RT07] is used to segment the 3D polyhedral shapes hierarchically. Dey, et al [DGG04] presented a Morse theoretic segmentation method for noisy point sets using Delaunay triangulation. Other algorithms, like the watershed algorithm [MW99], mean shift clustering [SSC04, YLL05] are also applied in segmentation algorithms. According to [ABM06], most of the segmentation methods can be generally classified into two types: geometry and structure based [JM07, RT07] or topology based [LA08, DGG04]. A good survey of mesh segmentation techniques can be found to [Sha08].

There are also considerable researches for the purpose of partitioning engineering

objects. The segmentation methods in this category are often used for reverse engineering [AMF00, BV04, DVV07, WKW02], dimensional metrology [LX08, SHM01], CAD model simplification [LDB05, She01] or feature recognition [CRB04, SP08], etc. The input data can generally be direct 3D point sets [DVV07, LX08, WKW02], range images [BJ88, MB05] or surface mesh [BV04, LDB05, RB03]. The continuous surfaces are also considered for feature recognition in [SP08]. However, in their method, the continuous models have to be transformed into a triangle mesh representation in STL format.

The methods for segmenting engineering objects can be generally classified into three types, namely, edge-based, face-based and hybrid-based [BV04, LX08, VMC97].

The edge-based approaches first detect the boundaries in the point data as representing edges between the surface patches and the points are partitioned accordingly. Some derivative attributes, like normal vectors and curvatures, are commonly used to find the boundary points. The points with discontinuities in these derivatives attributes are viewed as boundary points. Because they are simple, edge-based approaches are developed and applied widely.

The normal information is used to determine the sharp edges and curvatures are used for smooth edge detection in the approach developed by Benko and Varady [BV04]. Yang and Lee [YL99] detected the edge points based on curvature properties estimated from local geometry and then used an edge neighborhood chain coding algorithm to form the boundary curves. Huang and Menq [HM01] implemented a similar method for segmentation of unorganized point data. Zhang et al [ZPK02] identified the boundaries using Gaussian curvature and then processed the part decomposition accordingly. Sun et al [SPP02] proposed an edge-detection methods based on the Eigen-analysis of the surface normal vector field in a geodesic window. Razdan and Bae [RB03] presented a hybrid method to detect the boundary points combining the watershed algorithm with extraction of feature edges with the analysis of dihedral angles between faces, and then to create the regions with the complete feature loops. Recently, Demarsin et al [DVV07] detected the closed sharp edges in point clouds using normal estimation and graph theory.

Most of the edge-based methods suffer from the following problems. The scanned data from measurement sensors are generally noisy and unreliable in edge vicinities

[VMC97]. Finding edges is always unreliable as the computations of normal and higher derivatives are sensitive to the noise. Moreover, the number of points used for edge-based segmentation is usually small, only the points near the edge vicinities are used for calculation, which means it is difficult to generate reliable segmentation results.

Face-based methods, on the other hand, attempt to generate connected regions first. These approaches cluster the points into connected regions with similar properties and the boundaries are then derived by interactions or other computations from the surface patches. In principle, face-based methods work on the global size of point sets and so they are more robust to the noise than edge-based methods. Some face-based methods are listed below.

Besl and Jain [BJ88] developed a method based on variable order surface fitting which is a classical method that belongs to the face-based segmentation category. Sunil and Pande [SP08] presented a face-based method to segment the CAD mesh. In their method, the points are first clustered into 8 predefined surface types according to the signs of Gaussian and mean curvatures and then some region growing algorithm is implemented to generate the final connected regions. Chappuis et al. [CRB04] proposed a similar segmentation method, but clustered the points into 5 clusters of geometric primitives. Rabbani et al. [RSV06] developed a method for segmentation of point clouds using smoothness constraints (local surface normal and point connectivity) to find the smoothly connected areas in point clouds. In the method of Garland et al. [GWH01], the points are partitioned with different scales based on hierarchical face clustering algorithm. The face-based segmentation is still an active research area, however, the problems, such as time consumption of the region growing algorithm, under- or over- segmented regions, are not solved well yet.

The hybrid approaches combining the edge-based and face-based information have then been emphasized in recent years to overcome the limitations involved in edge-based and face-based approaches. In the segmentation method proposed in [LDB05], a k -means clustering algorithm is first used to produce redundant regions, and then, the region growing process reduces the number of regions with the help of sharp features. Finally the boundaries are updated in a contour-tracking process. Alrashdan et al. [AFM00] developed a hybrid segmentation method based on neural network. In their networks, the Gaussian and mean curvatures are used for face-based

segmentation, and the normal information is used for edge detection. Liu and Xiong [LX08] presented a method based on cell mean shift clustering algorithm and Gaussian map to segment point clouds. Woo et al. [WKW02] proposed an octree-based 3D grid method to handle the point sets, and used the normal vectors to subdivide the cells iteratively.

Most of the segmentation approaches are directly based on derivatives properties, such as normal, principle curvatures, Gaussian and mean curvatures. But they suffer from the following problems. The methods often fail for surface type recognition for noisy point data as the surface types are defined on Gaussian and mean curvatures, which are sensitive to noise. Complementarily, the methods for feature recognition assume that the polyhedral surface is noiseless. According to the author's knowledge, there is not an efficient solution to segment the noisy discrete models and to recognize the surface types of the segmented regions. As stated in [LX08], the segmentation problem hasn't been solved yet even for simple scenes containing only polyhedral objects.

5.3. Method overview

This thesis proposes a novel and efficient approach for segmentation and feature recognition. The noiseless assumption is not necessary. Our method is different from any other methods mentioned above, because it is the first approach to achieve the segmentation of engineering objects based on shape index and curvedness [KD92]. The method can produce reliable results for both segmentation and feature recognition.

In our method, we assume that the initial discrete shapes are in the form of dense sets of points sampled from the surfaces of the object and the discrete shape can be broken down into various components, which meet along the sharp or smooth edges. We also assume that the segmented components can be represented by mathematic functions with degree no higher than three, which covers most kinds of conditions in engineering applications. With the above assumptions, our method can always provide robust and accurate segmentation results.

The method can be decomposed into three main phases: discrete shape recognition, vertex clustering and connected region generation. Figure 5-1 shows the

process flow of the whole method.

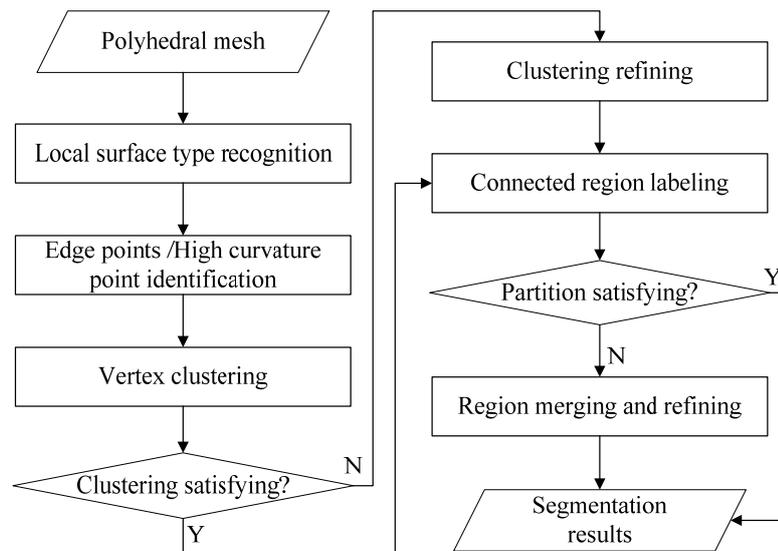


Figure 5-1: Procedures overview of shape recognition and segmentation

(1). *Local surface type recognition*

The most common local surface type definition of discrete shapes is directly based on the signs of Gaussian and mean curvatures [BJ88]. In this thesis, we formalize the definition of local surface type based on shape index [KD92]. In our definition, any point can be assigned to one of 10 basic surface types. In addition to the shape index, the curvedness is used to measure the intensity of the surface curvature. The pair of shape index and curvedness defines the complete shape information at a point, which provides a simple way for discrete shape recognition. Discrete shape recognition is also the basis of vertex clustering.

(2). *Vertex clustering*

First of all, the sharp/smooth edges points and high curvature points are detected based on the values of the curvedness. Because edge points don't define any region but only boundaries, the vertices except the sharp edge points are then initially classified into different clusters according to their surface types. Generally, if there are some imperfections in the discrete shape, such as the noise, the digital modeling dissertation (calculations for meshing, normal estimation, curvatures estimation, etc.), the initial clustering result is usually not satisfying. Hence, a vertex clustering refining procedure is performed in order to improve the quality of the clustering result.

(3). *Connected region generation*

After the vertex clustering, we will have a set of vertex clusters (the vertices belonging to the same cluster have the same surface type) and every vertices cluster includes multiply connected regions. In this phase, we aim at recovering these connected regions in each vertex cluster and decomposing the discrete shape into pieces accordingly. Based on the vertex clustering result, we perform the segmentation in two steps. The first step is the connected region labeling which is used to decompose the vertices into different connected regions initially. The second step is region merging and refining which is used to merge the small regions and refine the segmentation quality.

The main contributions of our work are listed below:

(a) We propose a new efficient segmentation method for discrete models based on shape index and curvedness. The pair of shape index and curvedness is more convenient than the traditionally used curvatures for the segmentation and recognition purposes. The method can recognize and extract the regions surrounded by real defined boundaries which can be sharp edges, smooth edges, inflexion edges, etc.

(b) We extend the work of Koenderink and Doorn [KD92] and formalize the definition of local surface types based on the shape index. According to the new definition of local surface types, a new clustering method is proposed to classify the vertices into the appropriate groups. The vertices within the same category have the same surface type.

(c) Unlike other segmentation methods in which the iterative region growing algorithm is always used, we developed a new approach to decompose the vertices into a group of regions which can be represented by the primary shapes. The approach uses the idea of the connected component labeling technique [HCS07]. It is simple and avoids the iteration operation for connected region generation, but the produced partition results are satisfying for engineering purposes.

(d) The reasonable measurement errors and computing errors are considered in our segmentation method. Therefore, our approach is robust and has wide applications, and is not limited to engineering applications.

5.4. Local surface type recognition

5.4.1. Surface types based on Gaussian and mean curvatures

As stated in most related literatures, the commonly used surface types are characterized by the signs of Gaussian curvature (K) and mean curvature (H). Besl and Jain defined eight fundamental surface types as shown in table 5-1 [BJ88].

Table 5-1: Surface types specified by signs of Gaussian and mean curvatures

	$K > 0$	$K = 0$	$K < 0$
$H < 0$	Peak	Ridge	Saddle ridge
$H = 0$	—	Flat	Minimal surface
$H > 0$	Pit	Valley	Saddle valley

This definition of surface types provides some insight into intrinsic differential geometry. However, it doesn't give full description of shape and it is inconvenient to understand a shape intuitively [LG09]. Koenderink and Doorn [KD92] then proposed a different surface descriptor, shape index to specify the local surface types. It is quite sufficient to specify the notion of surface types as the pair of Gaussian curvature and mean curvature do. Moreover, the shape index also provides the continuous intergradations between shapes, and thus has larger vocabulary to describe shape variations. It is more convenient to use the shape index to describe the local surface types than using the pair of Gaussian curvature and mean curvature.

In our work, we use the shape index to describe the surface type, but extend the work of Koenderink and Doorn [KD92] and formalize the definition of local surface types for segmentation and recognition purposes.

5.4.2. Surface type definition based on shape index

As noted in [BJ88], arbitrary smooth surface may be decomposed into a union of primitive surfaces that are approximated by lower order bivariate polynomials. In engineering applications, the orders of these bivariate polynomials are usually no higher than three, such as plane, sphere, cylinder, cone, etc. The goal of a

segmentation algorithm then is to decompose the discrete models based on these underlying primitive surface types [Bie85]. This is why we assume that the segmented regions can be approximated by polynomial functions with orders on higher than 3.

The shape index, defined in the section 3.3.5, is a quantitative indicator in terms of the principal curvatures to measure the local surface type of a given vertex.

There are nine fundamental surface types that can be characterized using only the shape index. On the shape index scales, the extrema of shape index interval represent minimal concave ($s = -1$) and maximal convex ($s = 1$) umbilical-like surfaces, which look like sphere surfaces. The concave and convex surfaces find their places on opposite sides. The concavities/convexities are separated by the saddle-like shapes located on the middle of shape index scales ($s = 0$).

One limitation of the initial shape-index-surface definition is that it has no definition on planarity. We extend this definition and assign a planar vertex with a shape index value of 2 ($s = 2$). Because the planar shape is defined by an isolated value not by the shape index interval, it is difficult to recognize the planar points only with this definition in practice. Fortunately, Curvedness provides an effective tool to identify the planar points. As mentioned in the section 3.3.5, the curvedness is always a positive number only except for the ideal planar points where it vanishes. Therefore, in practice, if the curvedness of a point (denoted as c_v) is less than a given threshold c_t (e.g.: 1×10^{-5}), the point can be identified as a planar vertex. A formal representation is given by the formula (5-1).

$$c_v < c_t \quad (5-1)$$

In many cases, a fast and unambiguous judgment of local shapes is important, it should be better if we are able to color the local shapes for visualization. In our cases, we conform to the color definition of local shapes with [KD92]. However, we assign the black color for the planar shape display. Till now, we have 10 complete local surface categories and their locations using shape index scale are shown in figure 5-2. The representative shapes from each category are illustrated in figure 5-3.

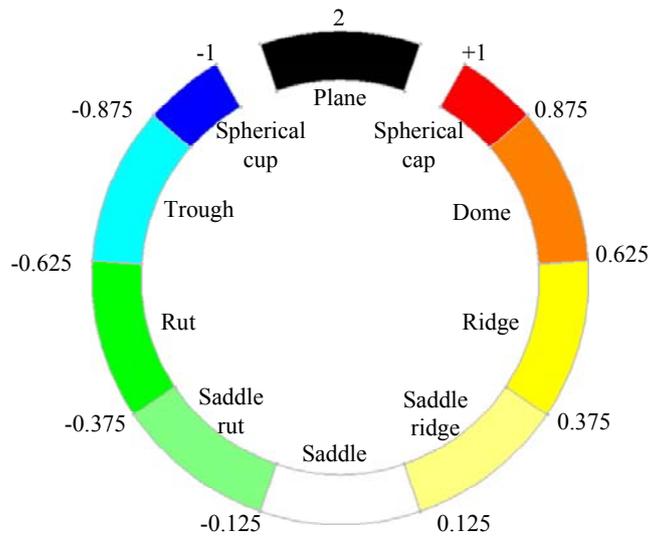


Figure 5-2: The locations of the surface types on shape index scales and their color scales

It is convenient to assign a particular integer to a point to indicate its local surface type for segmentation and surface fitting purpose. With appropriate surface type labels, the point data can be collected initially.

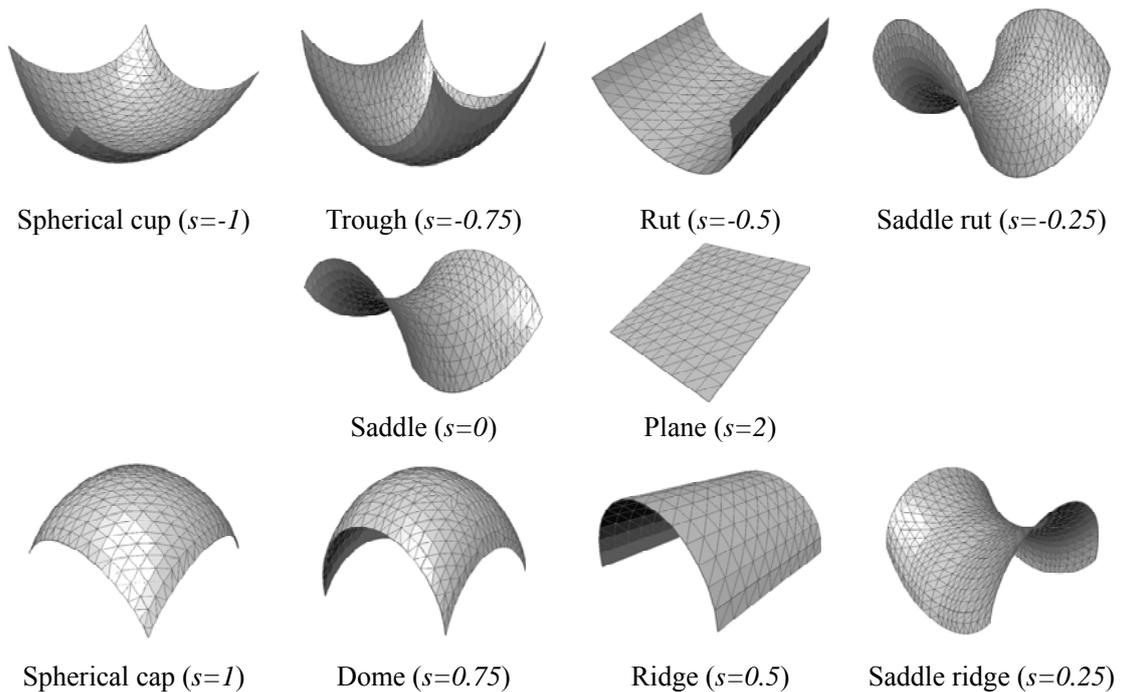


Figure 5-3: Representative shapes of the 10 defined surface types based on shape index

A formal definition of surface types based on Gaussian and mean curvature can be found in [BJ88] and [Sri08]:

$$T = 1 + 3(1 + \text{sgn}(H, \varepsilon)) + (1 - \text{sgn}(K, \varepsilon)) \quad (5-2)$$

Where, H and K indicate the mean curvature and Gaussian curvature respectively. ε denotes the tolerance threshold and $\text{sgn}(x, \varepsilon)$ is the tolerance signum function defined as:

$$\text{sgn}(x, \varepsilon) = \begin{cases} +1: x > \varepsilon \\ 0: |x| < \varepsilon \\ -1: x < -\varepsilon \end{cases} \quad (5-3)$$

Considering the opposite sides of concavities and convexities on the shape index scales, we define an integer range $\{-4, -3, \dots, 5\}$ to represent the local surface types. The mapping relationship from the shape index intervals to the local surface type labels is formulized as follows:

$$T = \begin{cases} \left\lfloor \frac{\lceil 8s \rceil}{2} \right\rfloor & s \in [-1, 1] \\ 5 & s = 2 \end{cases} \quad (5-4)$$

Where, $\lceil x \rceil$ denotes the minimum integer larger than x , while $\lfloor x \rfloor$ denote the maximum integer less than x .

According to the formula (5-4), the 10 fundamental surface types and their labels have already been defined in table 3-1 (section 3.3.5).

Some examples indicating the local surface types by color scales are shown in figure 5-4. The examples show that the 10 surface types' definition is reasonable and can be detected in applications.

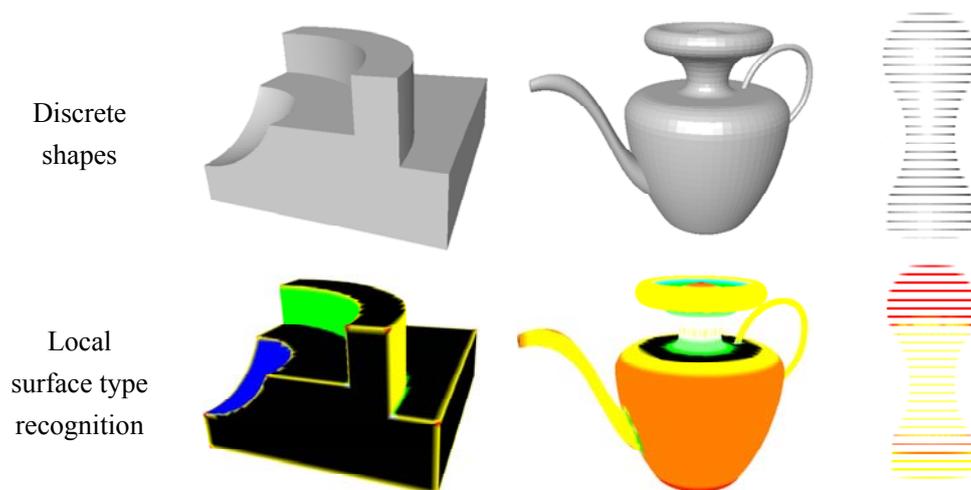


Figure 5-4: Discrete shapes and their local surface type visualization

5.5. Vertex clustering

Every vertex of the discrete shape is assigned a surface type label in the phase of local surface type recognition. At the vertex clustering step, the vertices will be classified into 10 basic clusters according to their surface types. In our method, the vertex clustering step consists of two stages: (a) sharp edges and high curvature regions detection; (b) vertex clustering and refining.

5.5.1. Sharp edges and high curvature regions detection

As a preliminary step, we should take into account and detect sharp edges and high curvature regions. The high curvature regions are identified to detect smooth transition regions within shapes. Edge points define only the boundaries which separate the surface regions, but not the regions. However, in the previous section, each of the vertices is assigned a surface type label according to its shape index indicator. It is inappropriate to label the sharp edge points with any local surface types.

In practices, the sharp edge points and high curvature points are usually mislabeled during the recognition phase. Considering the simple object shown in figure 5-5, the sharp edge e and the fillet region f which are the high curvature regions are assigned the same surface type label as the surface s (as shown in figure 5-5 (2)). In the later segmentation phase, they will be treated as the same connected region, which is obviously not correct. The mislabeled surface types may lead to wrong segmentation results. That is why the detection of sharp edges and high curvature regions is important for the downstream process.

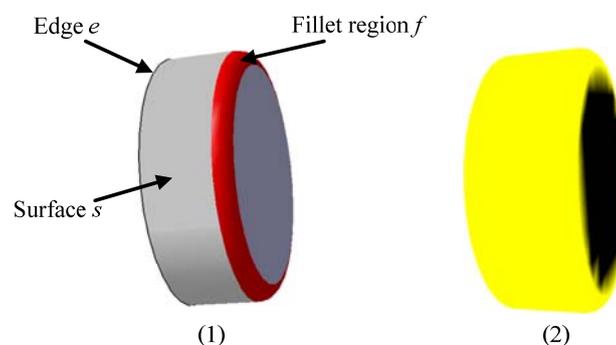


Figure 5-5: Example of mislabeling of sharp edge and high curvature region

The traditional methods for edge detection are mostly based on normal vectors [BV04, LDB05, SPP02]. These normal-based methods are highly sensitive to the noise and usually fail to detect the smooth high curvature regions without additional curvature information. Unlike these methods, we propose a robust detection method based on curvedness and don't use the normal information. This method is efficient and rather simple.

(1). *Curvedness map*

Curvedness is a shape descriptor to indicate the curved intensity of a point. In a discrete shape, the curvedness of the points on sharp edges and high curvature regions are usually much larger than the curvedness of the points in other places. It is reasonable to detect the sharp edges and high curvature regions using curvedness information.

In order to represent the curvature in an intuitive way for visualization, we describe the curvedness map of a discrete model using a continuous BGR color scales. Figure 3-3-6 shows two examples of curvedness maps.

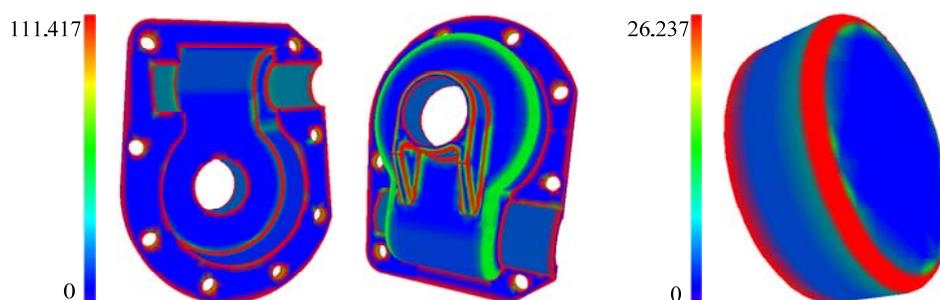


Figure 5-6: Curvedness maps of two discrete shapes

(2). *Detection and extraction*

According to the curvedness, a sharp edge is defined as an edge shared by two triangles in which only the two shared vertices have the curvedness values larger than a given threshold. A high curvature region is a set of points whose curvedness values are larger than the given threshold. The high curvature region detection is helpful to detect the smooth transition regions and their boundaries can be viewed as smooth edges.

The points in sharp edges and high curvature regions have the larger curvedness values than the points in other regions. Therefore, with an appropriate threshold, these

points can be extracted by the following criterion:

$$E(v) = \{v_i \mid c_i - c_{\min} > c_{th}\} \quad (5-5)$$

Where, $E(v)$ is the set of vertices which are located in the sharp edges and high curvature regions. v_i is an arbitrary vertex in $E(v)$. c_i is the curvedness value of the vertex v_i . c_{\min} is the minimum curvedness value of the given discrete model. And c_{th} is a given threshold set by the user. Experimentally, c_{th} is set as:

$$c_{th} = \delta \cdot \left(\frac{c_{\max} + c_{\min}}{2} \right) \quad (5-6)$$

Where, c_{\max} is the maximum curvedness of the given discrete model. δ is the coefficient and $\delta = 1$ by default. However, the coefficient δ can be adjusted by the user.

The curvedness ranges of the discrete models are always different for different shapes (see figure 5-6). For convenience in practice, the curvedness values of the given discrete shapes can be mapped into $[0, 1]$ and so the threshold can be adjusted easily within $[0, 1]$. Figure 5-7 shows two examples of the detection results.

Similar to the definition of concave and convex surface types, we can specify the identified sharp edge points and high curvature points as convex vertices or concave vertices according to their vicinities information. This classification is useful for the next connected region generation process.

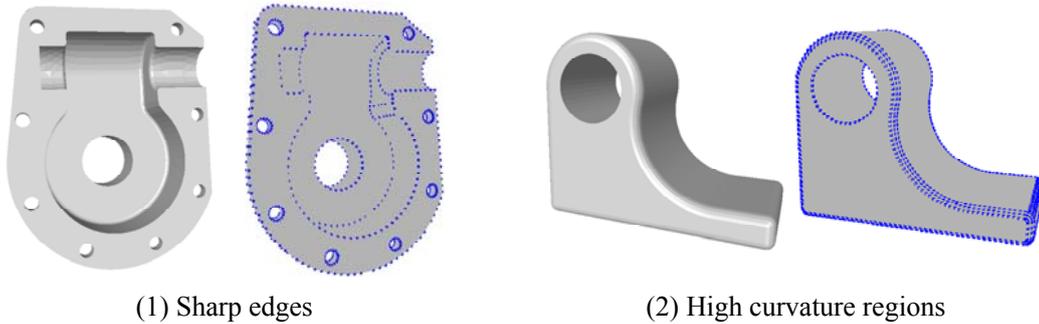


Figure 5-7: Detection of the points in sharp edges and high curvature regions

5.5.2. Vertex clustering and cluster refining

Vertices are first clustered into 10 different clusters according to their local surface types. If we view the vertices detected as sharp edges points and high curvature points as a vertex cluster, we can obtain a total 11 vertices clusters. The

vertices in the same cluster have the same surface type labels. We use the surface type label to index the vertices clusters.

This clustering algorithm can provide satisfying clustering results if the local surface types of the discrete shapes are ideally labeled. This may happen when clustering the tessellation mesh of CAD models. However, for the polyhedral mesh constructed from measured point data, due to the various errors, such as the manufacturing errors of the surfaces, the measurement errors of the acquired point data and the computing errors in the phases of polyhedral surface approximation, normal and discrete curvature evaluation, etc. the initial clustering results are often imperfect and need to be refined to improve the clustering quality.

During the cluster refining process, we don't consider the vertices labeled as sharp edges and high curvature regions in vertex clustering phase because they can be refined by adjusting the threshold c_{th} in formula (5-5).

The proposed cluster refining algorithm is based on an iterative process. In each iteration, given a vertex, we record all the appearing surface types of the vertices in its vicinity region (e.g. one-ring stencil). And then, based on a cluster distance, we evaluate the possibility of every appeared surface type that the given vertex should be refined to. And then the appeared surface types are ranked degressively according to their possibilities. Finally, the surface type of the given vertex is refined to the surface type with the maximum possibility. The iteration terminates when the surface type of each vertex doesn't change any more or a user-defined convergence condition occurs.

Considering the natural discontinuity of planar shape with other shapes in surface type definition, the refining of the planar cluster and other shape cluster are discussed respectively. The process of the clustering algorithm is described in figure 5-8.

(1). *Cluster refining for non-planar clusters*

Given a vertex v_i , its surface type is labeled as $T(v_i)$, the set of the vertices in its neighbor region is denoted as $N(v_i)$. For an arbitrary vertex in its neighborhood, $\forall v_j \in N(v_i)$, its surface type label is denoted as $T(v_j)$. The cluster distance between the two vertices v_i and v_j is defined as:

$$d_c(v_j, v_i) = |T(v_j) - T(v_i)| \cdot \xi_{ij} \quad (5-7)$$

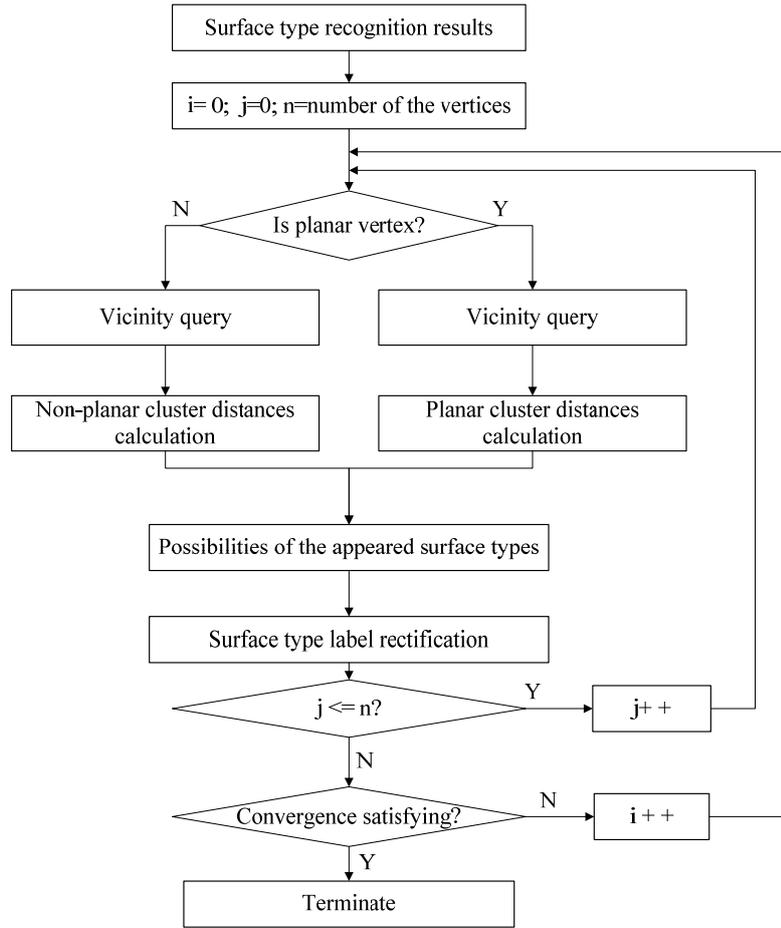
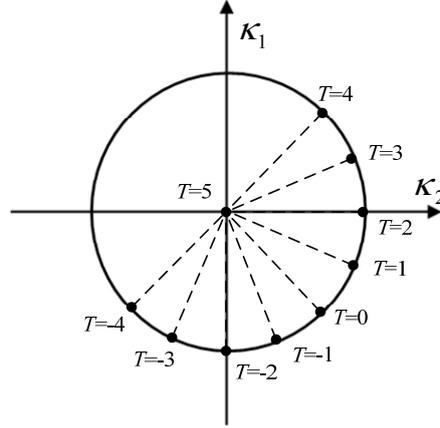


Figure 5-8: General flow of the proposed clustering algorithm

Where, ξ_{ij} are the weight coefficients to evaluate the similarities between two surface types. ξ_{ij} can be determined considering the position relationships of the defined surface types on the $\kappa_1 - \kappa_2$ plane (as shown in figure 5-9). A non planar surface type label T belongs to $\{-4, 4\}$. Hence, $\max(T)$ is equal to 4 while $\min(T)$ is equal to -4. If we define the distance between the two neighboring surface types as equal to 1, the normalized coefficients ξ_{ij} between two non-planar vertices can then be defined as:

$$\xi_{ij} = \frac{1}{\max(T) - \min(T)} = \frac{1}{8} \quad (5-8)$$

Figure 5-9: The defined surface types on the $\kappa_1 - \kappa_2$ plane

The cluster distance measures the similarity of local surface types between the two vertices. The value range of cluster distance is $[0, 1]$. The smaller cluster distance implies that the similarity between the surface types of the two vertices is larger, and the two vertices have the more possibility to be refined with the same surface type. Considering the inverse proportion between the similarity and the cluster distance, we define the possibility of the vertex v_i should be relabeled as the surface type of vertex v_j as:

$$P_i[T(v_j)] = \frac{[1 - d_c(v_j, v_i)] \cdot n_j}{\sum_{v_j \in N(v_i)} [1 - d_c(v_j, v_i)] \cdot n_j} \quad (5-9)$$

Where, n_j is the number of vertices with the surface type label $T(v_j)$ in the set of $N(v_i)$.

The possibility $P_i[T(v_j)]$ quantifies the reliability of the vertex v_i with respect to the surface type $T(v_j)$. The maximum value of $P_i[T(v_j)]$ is equal to 1 when all the vertices in $N(v_i)$ have the same surface type with the given vertex v_i . The minimum value is equal to 0 and happens only when the conditions that the surface type of v_i is spherical convex and the surface type of v_j is spherical concave, vice versa.

A large value of $P_i[T(v_j)]$ implies the high reliability to relabel the surface type of vertex v_i as the surface type $T(v_j)$. So it is reasonable to modify the surface type of vertex v_i as the surface type with the maximum possibility.

$$T(v_i) = \{T(v_k) | T(v_k) = \max_{v_j \in N(v_i)} \{P_i[T(v_j)]\}\} \quad (5-10)$$

The planar vertices are recognized based on the curvedness indicator. If the identification is not satisfying, we can adjust the curvedness threshold c_t mentioned in formula (5-1) to acquire better performance. We don't try to adjust a non-planar vertex to the planar cluster in the vertex clustering process. Therefore, if the vertex v_j is a planar vertex, we will ignore its influence on the cluster refining of the given vertex v_i .

(2). *Cluster refining for planar cluster*

The process of cluster refining for planar category is similar. However, we define the cluster distance in a different way. From the figure 5-9, we can see that the non-planar clusters ($T \in \{-4, \dots, 4\}$) locate around the planar cluster ($T = 5$). This distribution illustrates that a vertex has the same possibility to be adjusted into any of the other nine non-planar clusters. Accordingly, we define the cluster distance for planar vertex refining is:

$$d_c = \begin{cases} 0 & T(v_j) = 5 \\ \frac{1}{9} & T(v_j) \neq 5 \end{cases} \quad (5-11)$$

With the definition of planar cluster distance (5-11), the possibility of the vertex v_i to be relabeled as the surface type of vertex v_j has the same definition with equation (5-9). And the final refined surface type label of the given planar vertex can also be obtained using the equation (5-10).

In the clustering methods for both the planar and non-planar clusters, two surface types in $N(v_i)$ may be calculated as the same possibilities: $P_i[T(v_j)] = P_i[T(v_k)]$, where, v_j and v_k are two vertices with different surface types in $N(v_i)$. When this condition occurs, the algorithm will check whether there are other surface types which are different from $T(v_j)$ and $T(v_k)$ in $N(v_i)$. If there are still some other surface types in $N(v_i)$, the algorithm will keep the initial surface type $T(v_i)$ for v_i . If there is no other surface type existing in $N(v_i)$, the vertex v_i will be assigned the surface type with the smaller cluster distance. The surface types with smaller distance are closer. Because of the continuous gradation of the surface types defined by the shape index, they have the more possibility to be refined as the same surface type.

Figure 5-10 shows some examples of the vertex clustering results. From the examples, we can see that the quality of clustering results can be improved after cluster refining algorithm.

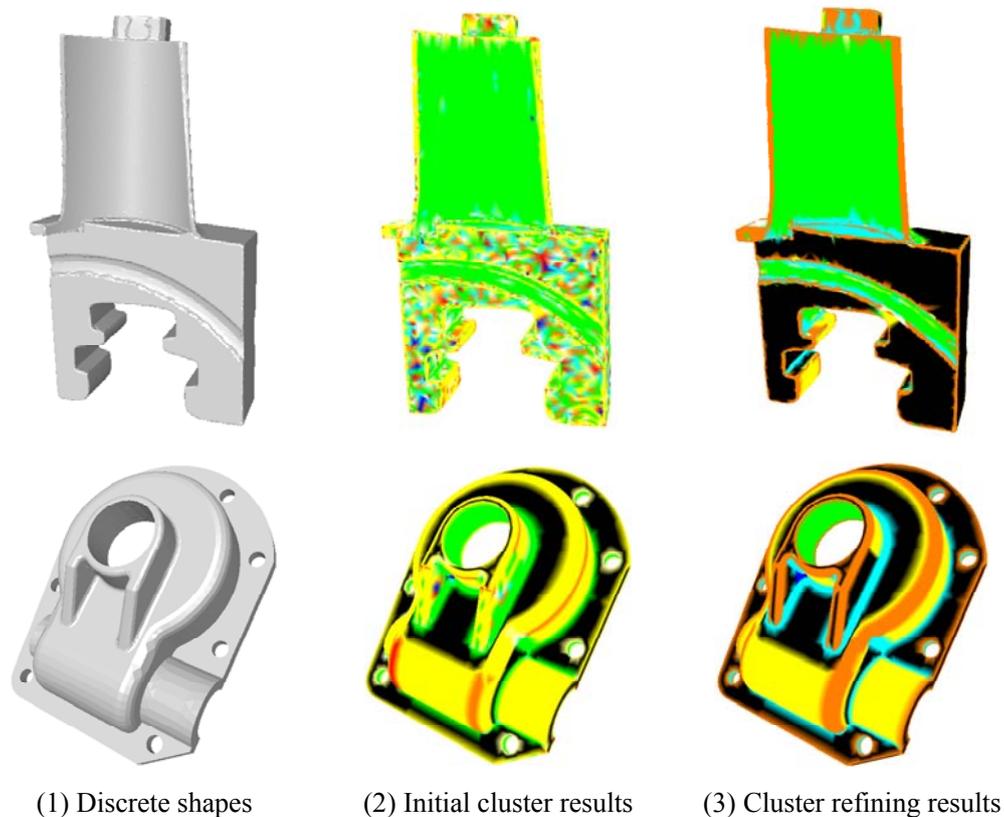


Figure 5-10: Examples of vertex clustering and cluster refining

5.6. Connected region generation

The main purpose of connected region generation procedure is to identify the connected regions from the clustered vertices and segment the discrete shape accordingly. Generally, every vertex cluster contains several connected regions, each of which can be represented as a single elementary shape. The whole discrete shape can be properly segmented based on the generation of all these connected regions in each vertex cluster. The proposed method for connected region generation is based on two main steps: (a) connected region labeling which is performed to generate the initial segmentation result; (b) region refining which is used to merge the small regions and improve the quality of the segmentation result.

5.6.1. Connected region labeling

(1). *General algorithm*

The connected region labeling is used to generate an appropriate region label for each vertex in the discrete shape. A region label is defined here as an integer to index the connected region it defines.

The connected region labeling is based on the connected component labeling technique which is widely used in image processing [HCS09]. When a vertex which hasn't been labeled is encountered, the vertices with the same surface type in the vicinity region of the given vertex are searched. Here, we only consider the one-ring stencil neighbor. According to the labeling conditions of these vertices in the vicinity region, the given vertex is marked with an appropriate region label. According to the associated region label, the given vertex could be integrated into an existing region or create a new region. The operation will terminate when all the vertices are associated with regions labels.

(2). *Connected region labeling and growing*

In the connected region labeling operation, we conform to the following denotations shown as table 5-2.

Table 5-2: The denotations used in connected region labeling

Symbol	Meaning
v_i :	An arbitrary vertex
$N(v_i)$:	The one-ring stencil neighbor region of v_i
$T(v_i)$:	The local surface type of v_i
$\Gamma(v_i)$:	The set of vertices which have the same surface type $T(v_i)$ in $N(v_i)$
$\Lambda(v_i)$	The set of the vertices which don't belong to $\Gamma(v_i)$ in $N(v_i)$

Obviously, it is impossible that the vertices in $\Lambda(v_i)$ belong to the same connected region with v_i . Thereby, we don't consider them in the following connected region labeling process at v_i .

Generally, when a vertex with vacant region label is encountered, there exist three

possible cases for connected region labeling, as shown in figure 5-11. In this figure, the vertex having the surface type as $T(v_i)$ but with vacant region label is shown as a circle with blank color. The vertex in $\Lambda(v_i)$ is represented as a circle with gray color. The vertices in $\Gamma(v_i)$ but with different region labels are colorized with different colors (red or green in figure 5-11).

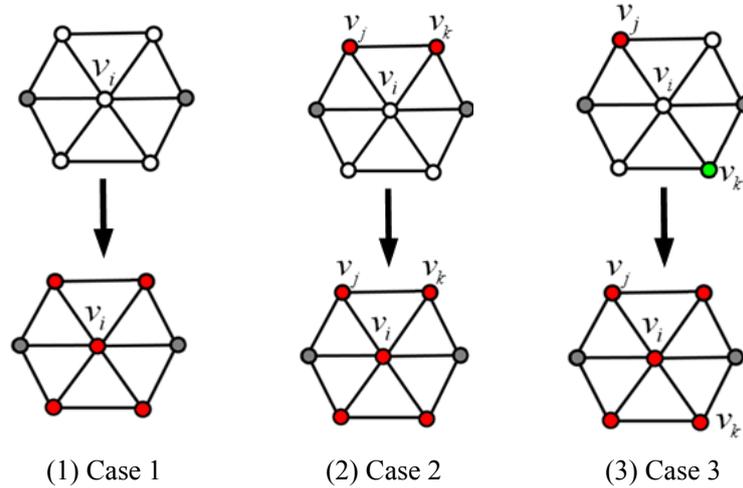


Figure 5-11: Studied cases for the connected region labeling

(a) The first case occurs when all the vertices in $\Gamma(v_i)$ are still not labeled yet (figure 5-11 (1)). A new region is then created and a new region label $l(v_i)$ is associated. The vertex v_i and all the vertices in $\Gamma(v_i)$ are labeled with $l(v_i)$.

(b) The second case happens when some or all of the vertices in $\Gamma(v_i)$ are already marked, but with the same region label, denoted as $l(v_j)$. In this case, the vertex v_i and the vertices in $\Gamma(v_i)$ which are not labeled yet are then associated with the same region label $l(v_j)$.

In figure 5-11 (2), the vertex v_j and the vertex v_k are two vertices in $\Gamma(v_i)$. They are marked with the same region label $l(v_j)$. Then, the vertex v_i and the other two vertices with vacant region labels in $\Gamma(v_i)$ are associated with the same region labels $l(v_j)$.

(c) The last case is observed when some or all of the vertices in $\Gamma(v_i)$ are labeled with different region labels. The set of the different region labels can be denoted as $L(v_i)$ ($v_i \in \Gamma(v_i)$) for convenience. Among $L(v_i)$, we assume that $l(v_j)$ is the region label with the minimum value which means $l(v_j) = \min(L(v_i))$. In this case, the vertex v_i and the vertices with vacant region label in $\Gamma(v_i)$ are associated with the region label $l(v_j)$. Other labeled vertices in $\Gamma(v_i)$, whose region

labels are not $l(v_j)$ are also updated to the minimum region label $l(v_j)$. In order to avoid generating over-segmented regions, all the vertices with the same surface type $T(v_i)$ are searched in the whole discrete shape. If they have been labeled with one of the appeared region labels in $L(v_i)$, their region labels will also be updated to the minimum region label $l(v_j)$.

In figure 5-11 (3), the vertex v_j and vertex v_k are associated with different region labels. If we assume that the region label of v_j is smaller, $l(v_j) < l(v_k)$, the vertex v_i and the other vacant vertices will be associated $l(v_j)$. Moreover, the vertex v_k will also be associated with the region label $l(v_j)$.

The connected region labeling algorithm will terminate when all the vertices in the discrete shape are associated with region labels. With different region labels, the discrete shape is actually already partitioned into several connected regions. However, because there is a relabeling operation in the connected region labeling algorithm (see case 2 and case 3), some region labels might be invalid, which contain null vertices. Thereby, the initially generated region labels should be optimized to delete invalid region labels and reorder the region labels sequence. The visualization of the segmented regions becomes convenience after the optimization of the region labels.

The connected region labeling operation provides the initial partition result. The performance of the connected region labeling algorithm depends on the clustering results. That's why the cluster rectification procedure is important in our segmentation method. If the clustering results are appropriate, the connected region labeling algorithm can always provide satisfactory segmentation results.

5.6.2. Region visualization

The connected region labeling operation segments the vertices of discrete model into different connected regions. However, the visualization of the segmentation results is also an important issue. A natural way uses different colors to display the different segmented regions. In our method, the partition is vertex based. However, the better visualization is based on triangle facet. Thereby, firstly, we assign a region label to each triangle according to its three vertices' region labels, and then, a series of different colors is implemented to render the different connected regions.

In this section, we conform to the following denotations. T_H denotes the cluster

of sharp edges and high curvature vertices. The vertices in other clusters are denoted as T_N .

When a triangle is encountered, one of the following four possible conditions occurs for the region label association to the triangle (as shown in figure 5-12).

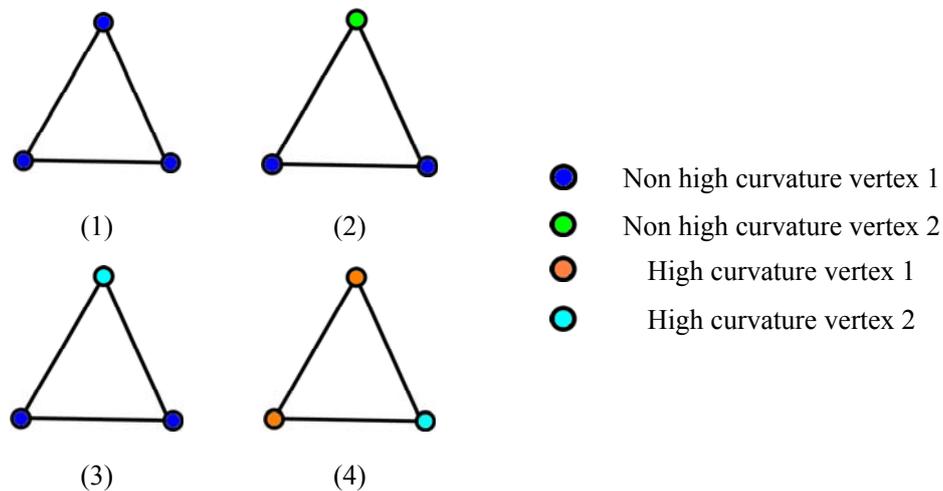


Figure 5-12: Four conditions a triangle possibly encountered for visualization

(a) All the vertices of the encountered triangle have the same region label, denoted as l_i (figure 5-12 (1)). In this condition, the region label l_i is assigned to the encountered triangle. It works whenever the vertices belong to T_H or T_N .

(b) The three vertices of the encountered triangle are associated with different region labels, but none of the vertices belongs to T_H (figure 5-12 (2)). This condition indicates that the encountered triangle locates on the border of different connected regions. In order to avoid a zigzag intersecting border and to get a smooth border, the minimal region label of $L(v)$ (the set of the appeared region labels) will be associated to the triangle.

(c) The region labels of the given triangle's vertices are different, but some vertices belong to T_H and others belong to T_N (figure 5-12 (3)). In this situation, if two T_N vertices appear, the triangle will be assigned with the minimal region labels of the T_N vertices for the same purpose of avoiding the zigzag border.

(d) All the three vertices of the encountered triangle belong to the category T_H . If the three vertices keep the same region label, as explained in condition (1), the triangle is associated with the same region label of the vertices. If the region labels of the vertices are different (figure 5-12 (4)), the triangle will be assigned with the minimal region label.

Even with the above criterion for triangle region label association, there may be some triangles assigned with inappropriate region labels, which need to be refined. One example is shown in figure 5-13. In this case, the three vertices of the yellow triangle are sharp edge points and so are associated with the same boundary region label. Hence, this triangle will be assigned with the same region label as its vertices (5-13 (2)), which is obviously inappropriate because the triangle and the adjacent triangle tr_1 should belong to the same planar region. Therefore, some post process should be done to correct the inappropriate region labeling.

When a triangle with three T_H vertices is encountered, such as the triangle tr_0 in figure 5-13 (1), the three adjacent triangles are queried. This query of adjacent triangles can be easily done based on the half-edge data structure.

Let tr_i ($i=1,2,3$) denote the three adjacent triangles of tr_0 . The normal vector of tr_0 is denoted as \bar{n}_0 and \bar{n}_i ($i=1,2,3$) denote the corresponding normal vectors of the three adjacent triangle respectively.

We first calculate the dihedral angles between each pair of the triangles $\langle n_0, n_i \rangle$ ($i=1,2,3$) and compare it to a predefined threshold (e.g. 1×10^{-4}). If the dihedral angle of one pair is smaller than the given threshold, the region label of tr_0 will be refined to the same label with the triangle in that pair. Figure 5-13, (2) shows the initial triangle labeling result and (3) shows the final triangle labeling result after label adjusting. The different colors in (2) and (3) indicate the different connected regions.

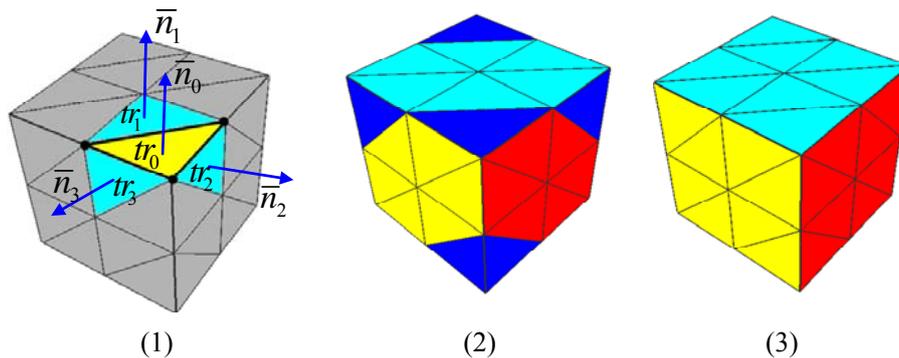


Figure 5-13: Region label adjusting for visualization

5.6.3. Region merging and refining

The connected region labeling process can provide satisfying decomposition

results of discrete models of engineering objects in general cases. However, the performance of connected region labeling method depends on the vertex clustering results. Even after the cluster refining, in some cases, the clustering results are not perfect enough to generate the expected segmentation results. Thereby, a region merging and refining process is necessary to reduce the over-segmented regions resulting from the previous steps and to improve the partition quality.

In our method, the region distance for each pair of adjacent regions is defined to measure the refining cost between them. The region merging is processed according to the region distance. For two adjacent regions, R_i and R_j , the region distance D_{ij} is defined as follows:

$$D_{ij} = d_m(R_i, R_j) \cdot p(R_i, R_j) \cdot n(R_i, R_j) \quad (5-12)$$

Where, $d_m(R_i, R_j)$ is a coefficient to measure the surface type similarity between the corresponding regions. $p(R_i, R_j)$ is a coefficient considering the boundaries' contribution for the region merging. The coefficient $n(R_i, R_j)$ aims at eliminating the region compared to relative smaller regions.

$d_m(R_i, R_j)$ is used to measure the shape similarity between the two adjacent regions R_i and R_j . In previous works, there are several methods [LDB05, LHM09] for the definition of $d_m(R_i, R_j)$. However, the surface type gives us a more natural way to define the similarity. Hence, in our method, the coefficient $d_m(R_i, R_j)$ is defined as:

$$d_m(R_i, R_j) = 1 - |T(R_i) - T(R_j)| \cdot \xi_{ij} \quad (5-13)$$

Where, $T(R_i)$ and $T(R_j)$ respectively denote the surface type labels of the connected region R_i and R_j . A segmented region is assigned the final surface type of the vertices it contains. ξ_{ij} is a coefficient to compute the surface type distance, by default, ξ_{ij} is set conforming to the definitions of ξ_{ij} in section 5.5.2:

$$\xi_{ij} = \begin{cases} 1/8 & T(R_i) \neq 5, \text{ and } T(R_j) \neq 5 \\ 1/9 & T(R_i) = 5, \text{ or } T(R_j) = 5 \end{cases} \quad (5-14)$$

The larger the value of $d_m(R_i, R_j)$ is, the better the similarity between the two adjacent regions is. It is more reliable to merge the two adjacent regions into one region.

$p(R_i, R_j)$ is a coefficient to consider the intersection border between the

corresponding regions. There are also several methods to define the coefficient $p(R_i, R_j)$, for example, in [LHM09], Lai et al specified a different definition of $p(R_i, R_j)$, which is defined as:

$$p(R_i, R_j) = \frac{P[e(R_i \cup R_j)]}{P[e(R_i \cap R_j)]} \quad (5-15)$$

Where, $P[e(R_i \cup R_j)]$ is the perimeter length of the boundary of $e(R_i \cup R_j)$, and $P[e(R_i \cap R_j)]$ is the perimeter length of the common boundary between the two regions.

In our method, we define $p(R_i, R_j)$ as:

$$p(R_i, R_j) = \frac{P[e(R_i \cap R_j)]}{\min(P[e(R_i)], P[e(R_j)])} \quad (5-16)$$

Where, $P(e)$ is the perimeter length of the edge e . $e(R_i)$ indicates the boundary of the region R_i , while $e(R_j)$ represents the boundary of the region R_j . The common boundary between the two adjacent regions is denoted as $e(R_i \cap R_j)$.

$p(R_i, R_j)$ is an important coefficient which considers the spatial positions of the regions in region merging decision. Adjacent regions with a larger common boundary are more likely to belong to the same region. Thus the value of $p(R_i, R_j)$ is larger.

$n(R_i, R_j)$ is a coefficient used to consider the influence of the area when merging two adjacent regions. If the area values are fairly distinct large, there is more opportunity to merge the two adjacent regions together. In our method, if $A(R_i)$ and $A(R_j)$ denote the region areas of R_i and R_j respectively, the coefficient $n(R_i, R_j)$ is defined as equation (5-17):

$$n(R_i, R_j) = \frac{|A(R_i) - A(R_j)|}{\max(A(R_i), A(R_j))} \quad (5-17)$$

Obviously, the larger the value of $n(R_i, R_j)$ is, the higher the opportunity to merge the two adjacent regions should be merged together.

From the above analysis, we can see that the three coefficients in equation (3-3-12) are increasing when merging configuration related to the related region pair is increasing. It is thus reasonable to believe that the region pair with much larger D_{ij} should be merged into one region during the region merging and refining process.

The region merging is performed as an iterative process. Firstly, all the adjacent regions pairs are searched and the region distance D_{ij} is then computed. The sequence of adjacent regions pairs are ranked into a priority queue degressively according to their region distances. In each iterative process, the region refining algorithm picks the adjacent regions pair with the maximum region distance, merges them into one region and updates the priority queue accordingly. Both the surface type and the region label of the region with larger areas are assigned to the new generated region. Finally, the region labels of the segmented region will be updated accordingly. The algorithm terminates when the number of the final regions reaches a predefined value or the region distance D_{ij} reaches a predefined threshold.

Figure 5-14 shows an example of the region merging and refining. From the figure (1), we can see there are some trivial small regions after the connected region labeling which are obviously inappropriate for segmentation and recognition purposes. After region merging, the final segmentation result of the object is shown in figure (2). In this case, the initial segmented region number is 33, and the final segmented region number after region merging is reduced to 14.

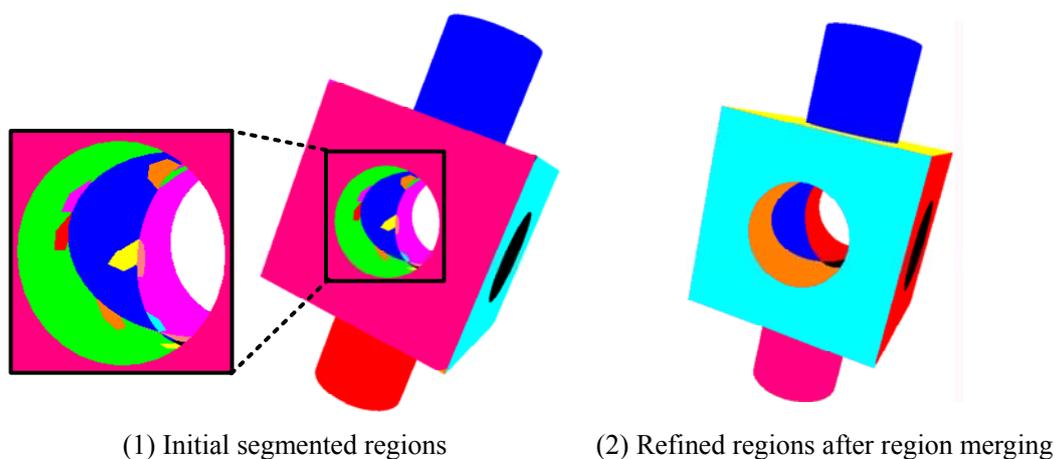


Figure 5-14: Connected region merging and refining

5.7. Experiments and Results

The algorithms mentioned in the previous sections have been implemented and considerable cases have been tested. The implementation is embedded in MS VC++8.0 supported by the CGAL libraries [CGAL]. The visualization result is based on Open/GL libraries.

The recognizable formats of the input files are STL and OFF. The specifications of the two formats are described in chapter 3.2.2. That means the mesh structure of the scanned point data has to be firstly built. Moreover, because the quality of the initial scanned point data is usually imperfect, the point data should be preprocessed for denoising and filtering to improve the quality before input for segmentation and recognition purposes.

The proposed method is based on two shape descriptors: shape index and curvedness, both of which are directly based on the principal curvatures. Hence, the principal curvatures at each vertex of the discrete shape have also to be estimated properly before starting the segmentation process.

5.7.1. Test cases

A region is usually surrounded by and connected with other regions by a set of boundaries. For the continuous surfaces, the commonly used boundaries in engineering applications can be classified into three types according to their continuity: G0 continuity, G1 continuity and G2 or higher continuity. The “G” in those terms refers to a mathematical concept called “Geometric continuity”

(a) G0 continuity is also called continuity of position. If one surface meets a continuous edge of another surface, then, the two surfaces are connected with the order of continuity G0. G0 continuity usually behaves as sharp edges in practice.

(b) G1 continuity, also named tangent continuity, is used to denote two smooth connected surfaces, whose first order derivatives are also continuous. In other words, G1 continuity means that the corresponding points on the shared boundaries of the two surfaces have the same normal vectors. The tangent connections between two shapes, such as the fillet, the inflexion connection, etc, are the examples of G1 continuity. The boundaries with G1 continuity can be denoted tangent edges for convenience.

(c) G2 continuity stands for continuity of curvature across two connected curves or surfaces. Previous conditions (G0 and G1 continuities) have to be fulfilled in order to guarantee G2 continuity. The boundaries with G2 continuity can be denoted curvature edges for convenience.

For discrete shapes, we can use a similar way to describe the boundaries of a

segmented region as sharp edges, tangent edges or curvature edges. Figures of the considered boundary types are illustrated in figure 5-15.

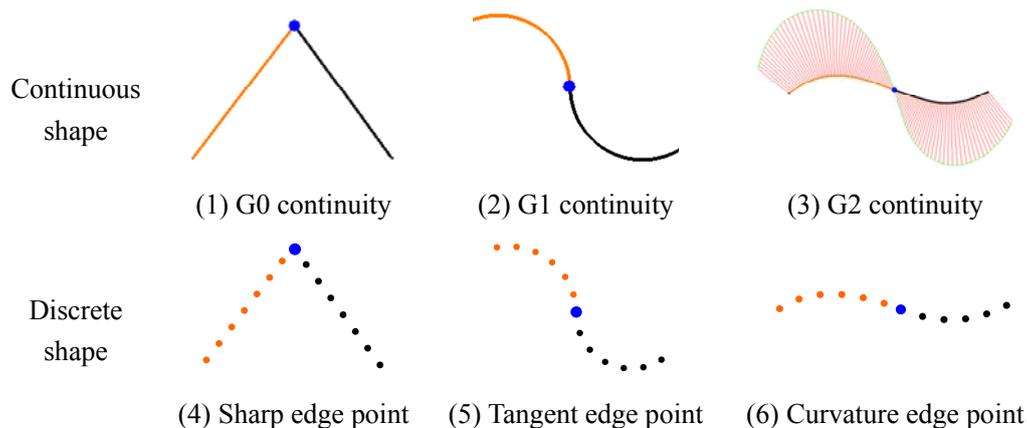


Figure 5-15: Boundary types in continuous and discrete shapes

The test cases in our experiments are organized according to the boundary types of the segmented regions.

(1). *Regions surrounded by sharp edges*

The points on the sharp edges can be recognized based on curvedness map. With the identified sharp edge points, two regions shared the sharp edges can be separated easily.

Figure 5-16 shows two segmentation results of the objects in this case. The sharp edges may occur between the regions with different surface types. The first case in figure 5-16 shows that the sharp edges may happen at the connection of the planar surface with other types of surfaces. The second case demonstrates the sharp edges occurring between the ridge surface and other types of surfaces. The results show that the sharp edges and the local surface types can be identified correctly. The final segmentation results are good enough. The first discrete shape is segmented into 11 regions and the second one is partitioned into 9 regions.

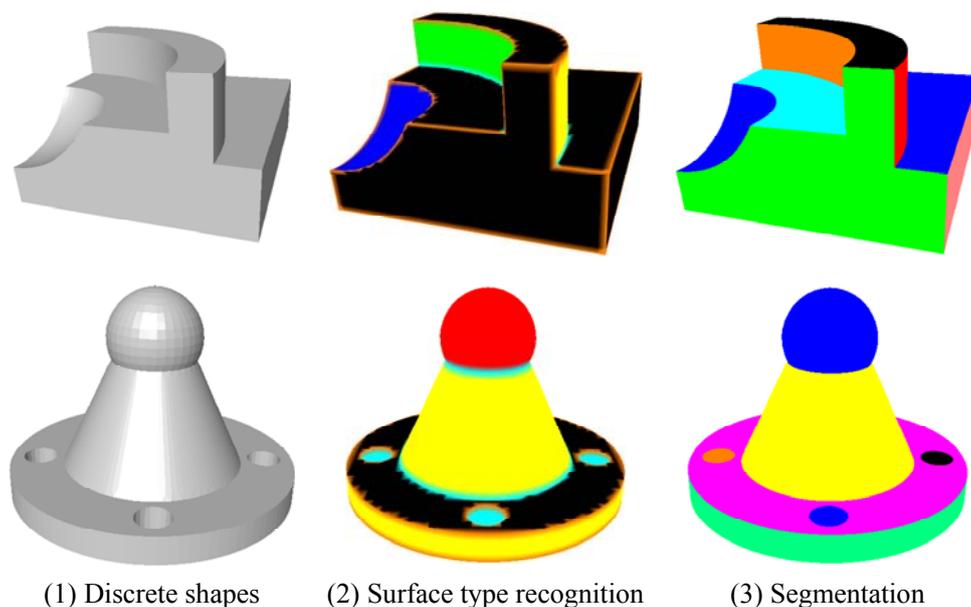


Figure 5-16: Segmentation of regions surrounded by sharp edges

(2). *Regions surrounded by tangent edges*

The tangent edges are quite often used in engineering applications, such as the fillet, the inflexion connection, the tangent connection between a plane and a cylinder or between a cylinder and a sphere, etc. The two tangent connected regions have continuous first order derivatives, but the second order derivatives are discontinuous. The shape index is defined as a second order derivative property and hence the shape index map is changed at the tangent edges, which means the two segmented regions sharing the tangent edge have different surface types. The proposed method can provide the reliable segmentation results of the regions surrounded by tangent edges.

Two examples in this case are shown in figure 5-17. The first case shows the segmentation result when the tangent edges happen between the planar and cylindrical surfaces. The second case shows the segmentation result of an axis-symmetric discrete shape which has a tangent edge between a spherical surface and a saddle-ridge surface. The two cases demonstrate that our method can provide reliable segmentation results when the discrete shapes have the regions surrounded by tangent edges. The numbers of the segmented regions of the two objects respectively are 22 and 4.

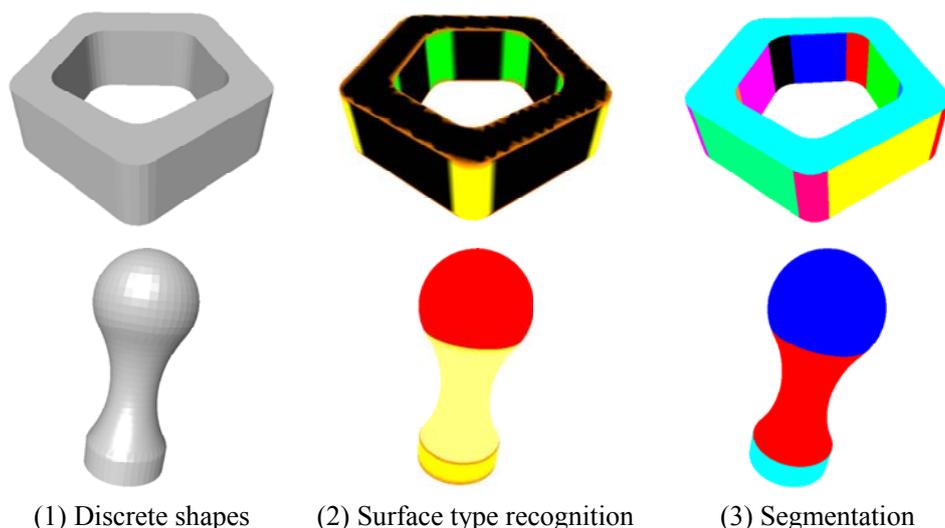


Figure 5-17: Segmentation of regions surrounded by tangent edges

(3). *Regions surrounded by curvature edges*

A curvature edge frequently happens between two freeform surfaces. The two surfaces shared a curvature edge means that the corresponding points on the curvature edge of the two surfaces have the same curvature properties. To identify the curvature edges needs higher derivatives. Because both the shape index and curvedness are second order properties in terms of the principal curvature, it is difficult to recognize the points on curvature edges. However, because the shape index provides a continuous gradation between shapes, the proposed method can also provide the reasonable segmentation results for some special cases, such as the case that the tangent edge happens at the critical values of the shape index intervals which separate the different surface types on the shape index scales; and the case where the points on the curvature edge are planar vertices which separate a convex and concave surfaces.

Figure 5-18 gives two examples of the segmentation in this case. In the first example, the curvature edge separates a convex surface and a concave one. The surface type of the points on the curvature edge is plane. The second example demonstrates the segmentation results of an axis-symmetric shape with curvature edges. The curvature edges happen at the critical values of the shape index intervals which separate the different surface types in our method (such as $s = \frac{5}{8}$ which is the critical value to separate the ridge and the dome, etc.). The numbers of the segmented regions of the two examples respectively are: 2 and 5.

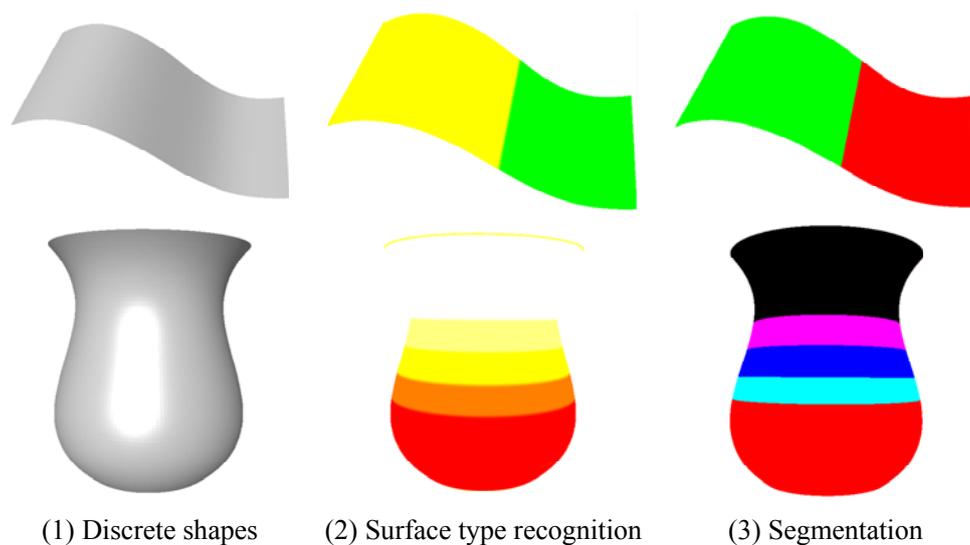


Figure 5-18: Segmentation of regions surrounded by curvature edges

(4). *Segmentation of complex shapes*

It is necessary to test the performance of the proposed segmentation method when it is used to decompose complex shapes, which contains several different types of edges at the same time. In order to test and evaluate our method, we used some complex discrete shapes from applications [Aim]. Figure 5-19 shows four examples of them. Some freeform surfaces, like the surfaces of the blade shape etc., are included in the four examples. From the segmentation results, we can see that the proposed method is robust and efficient to provide good results for the complex shapes which can be segmented by the mentioned three boundary types.

5.7.2. Noise effect

In many engineering applications, the point data are acquired directly from the physical objects by measurement sensors. Such point data always include noise. After the denoising procedure, the noise may be reduced. However, it is very difficult to eliminate the noise entirely. Considering that the noisy data are usually unavoidable, the segmentation method must be robust to such situations. This section presents the segmentation performance of our method when processing discrete shapes with noise.

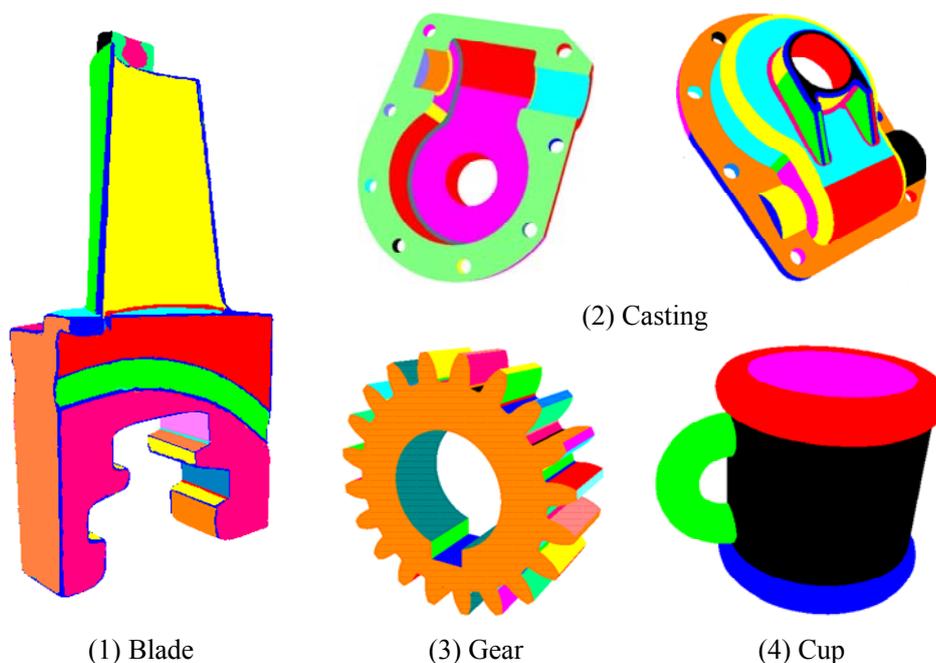


Figure 5-19: Segmentation of complex shapes

The proposed method tries to reduce the noise effect through two phases. One phase takes place during vertex cluster refining (see section 5.5.2) and the other one is during connected region merging and refining (see section 5.6.2). The vertex cluster refining operation attempts to adjust the surface type at a noisy point as the same type of the real points in its neighborhood, while the region merging and refining operation tries to merge the small regions generated by the noisy points into larger regions. We assume that most part of the noise can be filtered and reduced in the preprocessing procedure, and this section only considers the effect of the reasonable noise.

Figure 5-20 shows an example of segmentation of models with noise. Figure 5-20 (1) shows the initial discrete shape of the object. Figure 5-20 (2) shows the initial vertex clustering result based on its shape index map. The initial clustering result is “chaotic” and the boundaries are not clear because of the influence of the noise. Figure 5-20 (3) shows the final clustering result after cluster refining. Figure 5-20 (4) presents the final segmentation result of the discrete shape which contains 38 regions. The segmentation results prove the robustness of our method to the noise effect.

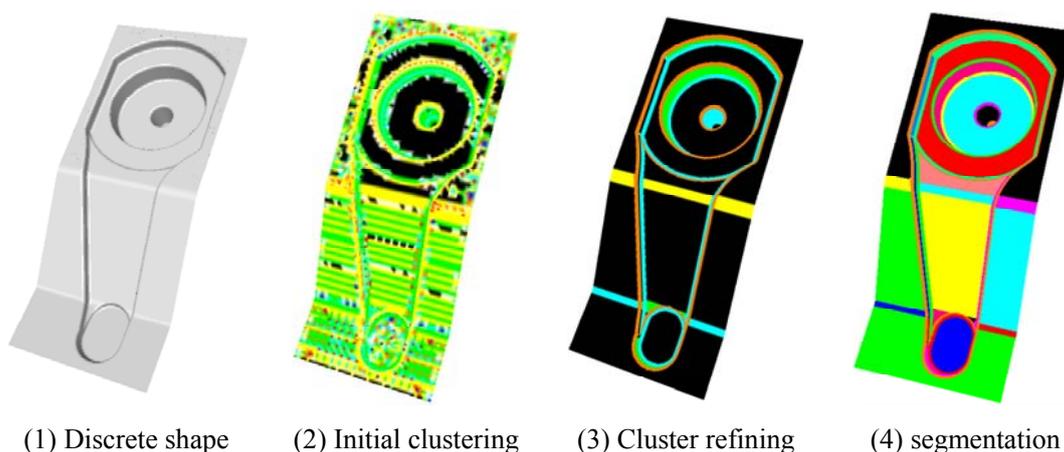


Figure 5-20: Segmentation of a discrete shape with noise

5.7.3. Time performance

Time performance is another important issue to evaluate the quality of our method. The proposed segmentation method includes mainly two stages: vertices classification and connected region generation. The time complexities of the two stages are: $O(n)$ for each iteration operation in clustering refining. The time complexity in region generation phase depends on the generated region number and the removal region number. However, the worst case for connected region labeling is $O(n^2)$ and $O(K)$ for each iterative process in regions merging and refining. Where, n is the number of the vertices of the given discrete model and K is the initial regions number before the regions merging operation. The whole time complexity falls to the similar level of other segmentation methods in engineering applications [LA08, LDB05]. In applications where the vertex number is not very large, the entire process is rather fast. The elapsed time for the test cases in this chapter (see figure 5-21) are listed in table 5-3

Table 5-3: Computing times for the tested cases

Discrete shapes	Vertices number	Clustering time (s)	Region labeling time (s)	Segmented region number
(1)	7041	0.015	2.828	11
(2)	4209	0.015	0.61	9
(3)	4239	0.016	1.578	22
(4)	4646	0.017	0.172	4
(5)	3330	0.016	0.687	2
(6)	8244	0.031	3.734	5
(7)	14315	0.032	17.735	57
(8)	5086	0.026	3.234	49
(9)	20713	0.047	20.093	126
(10)	5663	0.016	2.109	5
(11)	34200	0.078	86.391	38

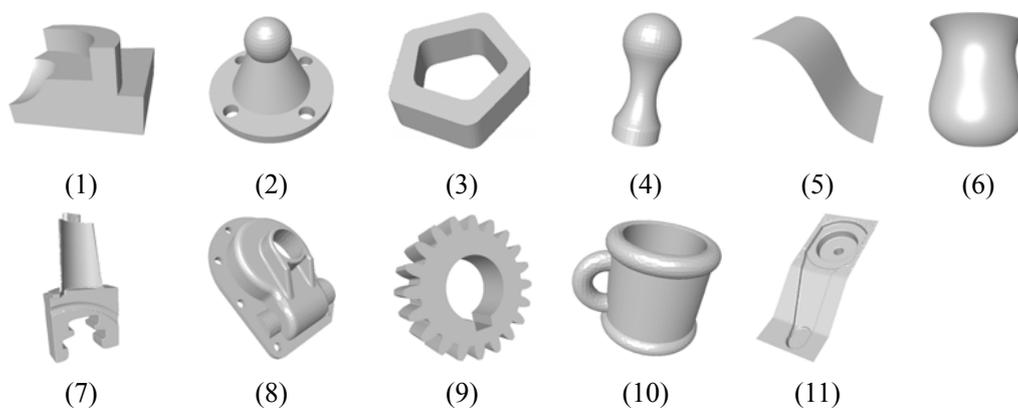


Figure 5-21: The discrete shapes tested in table 5-3

The table 5-3 shows that the time cost for clustering takes up only small part of the whole time cost. Time is mainly consumed in the phase of connected region labeling. The consuming time mainly depends on the vertices size of the discrete model, however, the segmented region number, is also an important factor that influences the time performance. Generally, the larger number of the segmented region is, the larger the computing time is. For example, the casting shape ((8) in

figure 5-21) contains less vertices than the cup shape ((10) in figure 5-21), but the casting model is decomposed into 49 regions which is much larger than the cup model which has only 5 regions. Thus, the casting object needs more time than the cup model during both clustering and the segmentation phases.

Even the final segmented region number influences the global computing time; the vertex number plays the most important role for the time performance. Figure 5-22 presents a rough trend of the relationship between the vertex number and the computing time. The model tested here (as shown in figure 5-22) is based a CAD model that we tessellate with different point densities so as to generate different discrete shapes with different vertex number (table 5-4). Note that the final segmented region number of these discrete models is 12. Therefore, table 5-4 and figure 5-23 describe the real relationship between the computing time and the vertices size which avoids the influence of the number of the segmented region.

From figure 5-23, we can find that the time performance of the method becomes worse when the vertex number is very large. However, as mentioned above, because the geometry order is not higher than 3 in the general engineering applications, a huge vertex number is usually not very necessary. So it is reasonable to believe that the proposed method has a satisfying time performance.

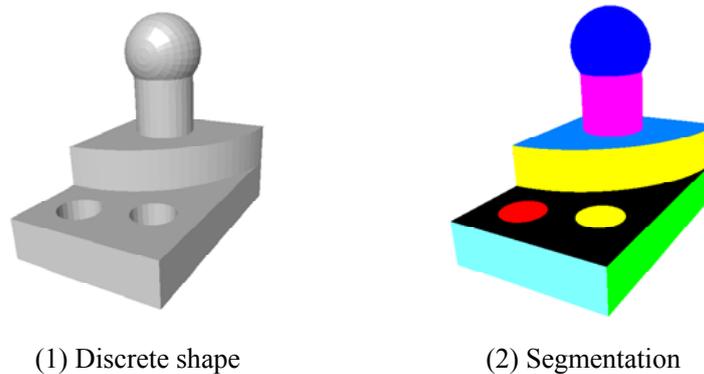
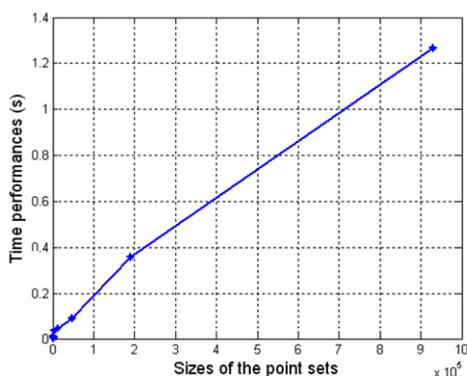


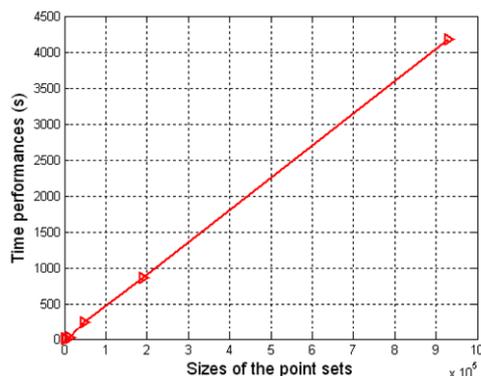
Figure 5-22: Shape used for testing the time performance

Table 5-4: Computing times for the same shape with different tessellations

Vertices number	Clustering time (s)	Region labeling time (s)
125	0.0015	0.0016
787	0.0016	0.047
1176	0.0020	0.109
2501	0.035	0.484
12376	0.047	16.812
48090	0.094	247.34
190520	0.359	865.15
928840	1.266	4176.42



(1) Clustering time



(2) Connected region labeling time

Figure 5-23: Time performance of the test cases with different tessellations

5.7.4. Limitations

Like other segmentation methods, the proposed method also has some limitations. The first limitation is that the method needs the input discrete model to be defined with real regular boundaries. The discrete shapes of complex objects, such as the shapes from biology, arts, etc. usually have geometry orders larger than 3, it is difficult to segment them appropriately. However, the local surface types of these shapes and some characteristic points, such as points with high curvatures can be

identified by our method. Figure 5-24 shows two identification results of the characteristic points from these complex shapes.

Another limitation of the proposed method is the non-smoothness of the boundaries of the segmented regions. The point data acquired from the measurement systems are unavoidably noisy. The noise is usually random and can come from the measurement errors of the whole measurement system. It is also difficult to approximate the exact local surface types of point data when they are sampled from rough surfaces. An example is shown in figure 5-25.

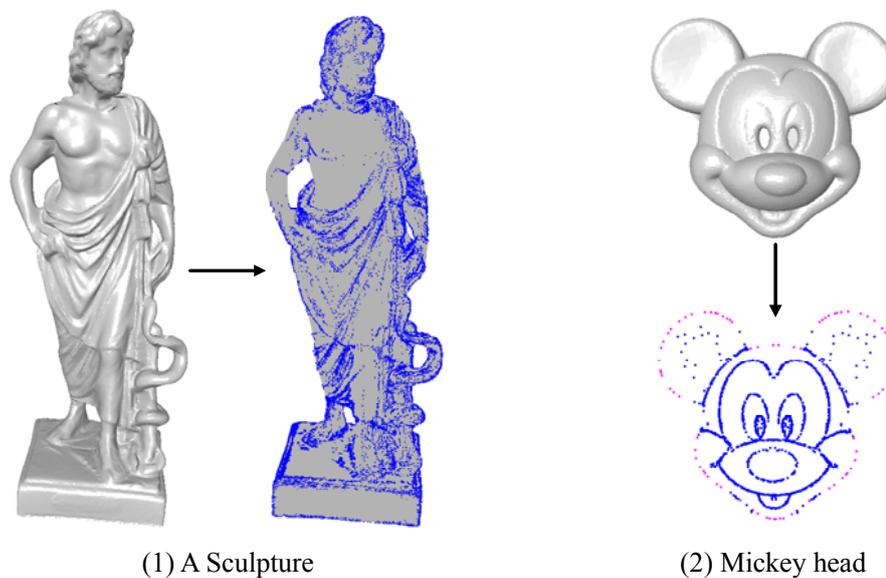


Figure 5-24: Characteristic points identification from complex shapes

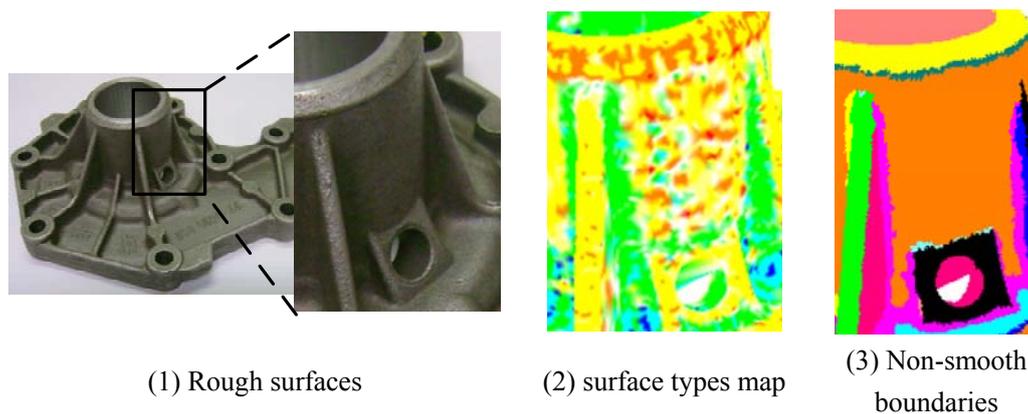
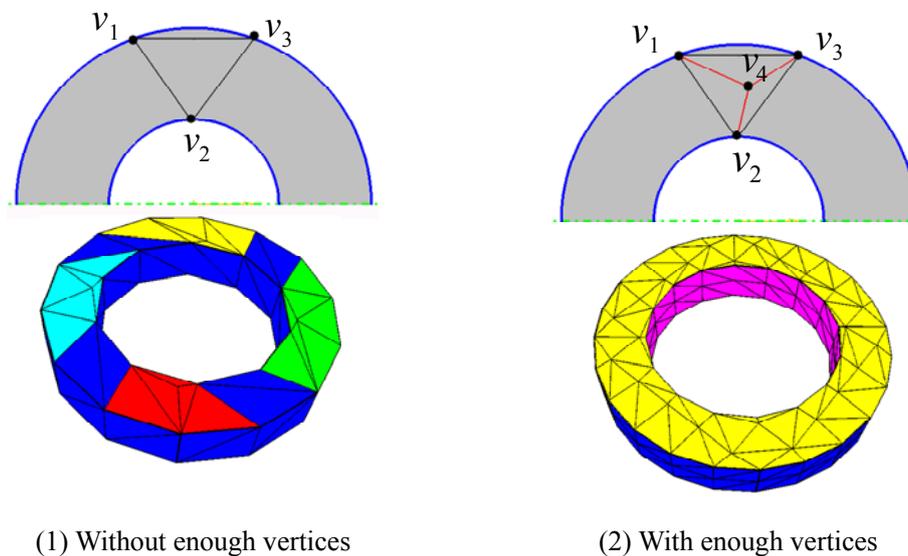


Figure 5-25: Limitations to the measured noisy point data

The third limitation is related to the vertex number. When the vertex number is over the magnitude order of 1×10^6 , the method takes a very long time for computing, generally more than 1 hour. This time performance indicates that the proposed method

is not suitable to process the discrete models with a large amount of vertices. The third limitation is for discrete shapes with sparse vertices. As we know, the discrete curvature of a vertex is defined from its neighborhood. When the vertices density is not enough, the local vicinity of a vertex may contain other vertices which don't really belong to the vicinity of the given vertex. The discrete curvature estimation would converge with large error. The segmentation result will not be accurate when the curvature estimation with large errors occurs. In order to avoid this limitation and provide good segmentation results, our method requires that the input discrete model must satisfy the following restrictions: when two vertices of a triangle facet belong to two disconnected boundaries, the third vertex of the triangle shouldn't belong to neither of the two disconnected boundaries.

An example is shown in figure 5-26 (1). Considering the triangle composed of v_1 , v_2 and v_3 , the vertices v_1 and v_3 are on the same boundary, and the vertex v_2 is on another disconnected boundary. The triangle will be treated as a high curvature region, which is not obviously right. The blue triangles in the segmentation result are corresponding to this condition. A different condition is shown in figure 5-26 (2), from which we can see that the vertices on the boundaries don't define a triangle directly. Thus, the segmentation result is more realistic.



(1) Without enough vertices

(2) With enough vertices

Figure 5-26: Segmentation with different sizes of vertices

5.8. Conclusion

This chapter presents new methods for discrete shape recognition and segmentation. Unlike the general methods in literature, mostly based on the principal curvatures, the proposed method is based on two different surface descriptors: shape index and curvedness.

The surface indicator, shape index, shows its convenience to represent the local surface types at a vertex. Based on the shape index, ten basic surface types are defined to recognize the local surface types of a given discrete shape.

The surface type recognition procedure also provides a simple and efficient way for vertex clustering. In the proposed method, after the initial clustering of the vertices, a new cluster refining process is performed to adjust the vertices into proper clusters. The cluster refining process improves the robustness to the noise and computing errors. The sharp edge points and high curvature points are also identified in the vertex clustering process.

A new algorithm, called connected region labeling is implemented to recover the connected regions in the discrete shape based on the vertex clustering result. The initial segmentation result is then generated accordingly. Finally, an iterative region merging process is performed to merge the small regions to obtain a better partition performance.

The proposed methods for discrete shape recognition and segmentation have been successfully tested for both engineering objects and freeform shapes. The limitations of the proposed method were also discussed.

Chapter 6

A Case Study

6.1 Introduction

The aim of this chapter is to further illustrate the methods described in the previous chapters by presenting a detailed case study of an industrial workpiece -automotive water pump cover.

The workpiece is fully digitized using the multisensor platform of LURPA. The multiple data are acquired by laser scanning and touch probing in both complementary and competitive multisensor configurations.

A software framework, DSP-COMS, has been developed as the testing platform for the developed methods of discrete geometry processing (normal and curvatures estimation, registration, recognition and segmentation) described above.

The acquired point data of the case study workpiece are then processed by the DSP-COMS software framework and the results of discrete geometry processing are discussed.

This chapter is organized as follows:

In section 6.2, we describe the measurement system configuration available LURPA and specify the main characteristics of the three used sensors. The point data acquisition process from the selected workpiece is presented in this section. Section 6.3 introduces the developed software DSP-COMS, which serves as the test platform for the algorithms developed in this dissertation. Section 6.4 presents the detailed processing results and section 6.5 concludes this chapter.

6.2. Multidata acquisition

6.2.1. Multisensor system configuration

(1). *System description*

The multisensor measurement system existing in LURPA integrated with three different sensors: a Kreon Zephyr KZ25 laser scanner [Kre], a Renishaw TP2 probe

[Ren] and a STIL CHR150-CL2 chromatic confocal sensor [Sti]. The measurement system contains two parts: (a) the dimensional measurement equipments, which include the CMM and the three sensors; (b) the measurement software packages, which support the measurement execution and data acquisition. Figure 6-1 gives a general overview of the measurement system. The three sensors configured on the CMM use with the Renishaw PH10M probe holder. The software platforms related with the computer are used for measurement planning and data acquisition, are “Prelude Inspection” supports the touch probing; “Polygonia” is used for laser scanning and “Mountain” used for STIL data acquisition.

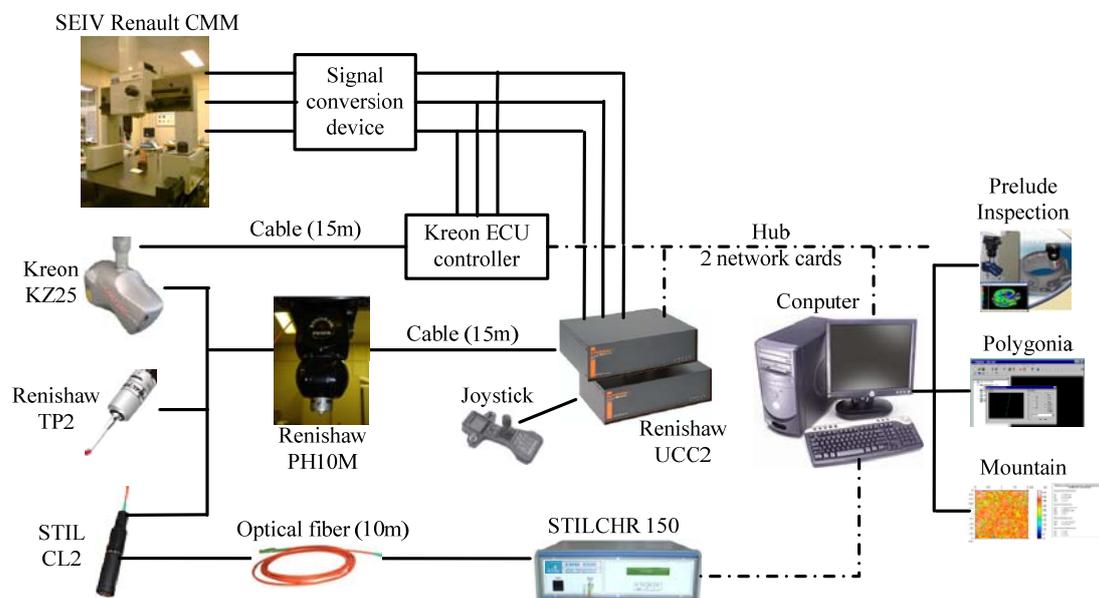


Figure 6-1: The multisensor measurement system at LURPA

The core of the measurement system is the sensors and their integration. The main characteristics of the three sensors are presented below.

(a) Kreon Zephyr KZ25 laser scanner.

Kreon Zephyr KZ25, as shown in table 6-1, is a high-accuracy range laser scanner. It works on the laser triangulation principle. The acquired data of KZ25 are 2D images which should be calibrated to obtain the 3D spatial coordinates of the measured points. The specifications of the KZ25 laser scanner are listed in table 6-1.

The KZ25 laser scanner can be used to measure 3D geometric deviations, for surface digitization with high point density, etc. When scanning, the surfaces to be measured should be positioned within the field of view (FOV) of the laser scanner.

Table 6-1: Specifications of the Kreon Zephyr KZ25 laser scanner

Schematic figure	Parameter	Value
	Standoff distance (H)	90 mm
	Depth of Field (P)	50 mm
	Stripe Length (L)	25 mm
	Acquiring speed	30,000 points per second
	Vertical resolution	3 μm
	Repeatability	3 μm
	Reproducibility	9/15 μm
	Weight	360 g

(b) Renishaw TP2 contact probe.

Renishaw TP2 contact probe is a kinematic resistive touch-trigger probe, which has been widely used in industrial applications. The TP2 touch trigger probe acquires the 3D coordinates (X, Y, Z) of the probing points directly. The specification of the TP2 probe is shown in table 6-2. The repeatability and the PTV (Pre-Travel Variation) are the parameters which affect the performance of the touch trigger probe [Ren].

Table 6-2: Specifications of the Renishaw TP2 touch trigger probe

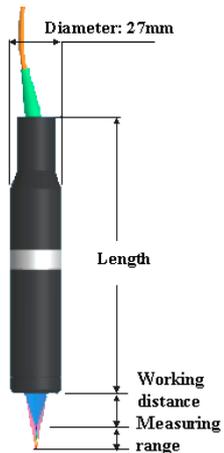
Schematic figure	Parameter	Value
	Max unidirectional Repeatability (2 σ)	0.35 μm
	PTV in XY plane	$\pm 0.8 \mu\text{m}$
	Sensing directions	$\pm X, \pm Y, Z$
	Stylus force range	0.07~0.15N
	Maximum extension on PH10	300mm
	Over-travel in XY plane	$\pm 14^\circ$
	Over-travel in +Z axis	4mm @ 0.07N; 3mm @ 0.15N

The Touch trigger probing technology is mature nowadays. It is simple and more adaptable to the environment. The accuracy of touch probes is higher than the non-contact scanning. However due to their slow probing speed, it is difficult to obtain dense point data. The applications of these probes are limited to the inspection of 3D prismatic parts and known surfaces [Ren]. The TP2 probe can also be used to capture the data of some special features which need measurement with higher accuracy to complement the laser scanning.

(c) STIL CHR150-CL2 chromatic confocal sensor.

The STIL CHR150-CL2 chromatic confocal sensor contains three parts: the CHR150 controller (including the light source), the CL2 optical pen (chromatic lens) and the optical fiber. The controller is the optoelectronic cabinet of the sensor. The same controller may be calibrated for several optical pens. The optical pen is a chromatic lens connected to the controller by an optical fiber. Optical pens are entirely passive in capturing the point data. The specifications of the CHR 150 controller with a CL2 optical pen are listed in table 6-3.

Table 6-3: Specifications of the STIL CL2 optical pen with CHR150 controller

Schematic figure	Parameter	Value		
	Nominal measuring range	400 μm		
	Working distance	11 mm		
	Axial resolution	0.012 μm		
	Max object slope	$\pm 28^\circ$		
	Mechanical diameter	27 mm		
	Magnifier model	MG210	MG140	MG210
	Lateral resolution (μm)	1.3	1.55	1.3
	Spot size diameter (μm)	2.6	3.1	2.6
	Length (mm)	243.3	208.9	243.3

The chromatic sensor CHR150-CL2 is suitable to perform the measurement tasks in micro-domain, like Micro-topography, film-thickness, etc. The chromatic sensor CHR150-CL2 is a 1D point sensor which can only capture the depth information. When performing the measurement tasks, the coordinates of X-axis and Y-axis can be

provided by CMM.

(d) Comparison of the three sensors.

The used three sensors are very different from each other. Each of them has its own distinct advantages and disadvantages. To measurement of a complex workpiece usually composed of various detailed features, the most suitable sensors must be specified for each particular feature. It is necessary to compare the characteristics of the three sensors for the sensor selection purpose. The table 6-4 presents the main characteristics comparison of the three sensors.

The three sensors can cover the measurement tasks in both macro and micro domains with 1D, 2D and 3D data acquisition with or without contact sensing techniques. The measurement system integrating the three sensors can be adaptable to the general applications in dimensional metrology, reverse engineering, etc.

(2). *System configuration*

The configuration of the three sensors in the CMM platform is a critical issue to achieve the physical integration. The multiple sensors should not interfere with each other during the system setup and also should not collide with each other or some components in the system during the measurement execution. There are several methods to combine the multiple sensors together. In order to facilitate the physical integration operation, a Renishaw probe holder PH10M is used to assemble the three sensors on the CMM platform.

The Kreon KZ25 laser scanner provides the interface with the Renishaw PH10M holder. Hence it is simple to integrate the Laser scanner on the CMM arm. Another convenience using the Kreon KZ25 scanner is that it also provides the interface with the contact probe. With the two interfaces, the Laser scanner and the contact probe can be easily integrated on the CMM platform.

However, there is still no standard component to integrate the STIL CL2 pen to the CMM platform. For this purpose, a joint component to fix the STIL CL2 pen on the CMM arm was designed in our laboratory. The component is shown in figure 6-2.



Figure 6-2: The component to install the STIL pen on CMM arm

If we combine the three sensors on the CMM arm synchronously, because of the limited measuring range (just 400 μm) and working distance (the maximal one is only 11 mm between the lens and the digitizing points), it is very difficult to avoid collision during the measurement execution. Fortunately, the STIL sensor focuses on the micro measurement, which has little overlap with the laser scanner or touch probe. Therefore, the macro and micro measurements generally are not required at the same time. Therefore, we can perform the sensor physical integration one after the other. In our case, when a measurement in micro domain is required, the STIL CL2 optical sensor will be installed, but before doing so, the laser scanner and the touch probe should be uninstalled to avoid collisions. The physical integration hence includes two configurations. One is the laser scanner and touch probe integration, the other is the STIL CL2 optical sensor integration. The two configurations are illustrated in figure 6-3.

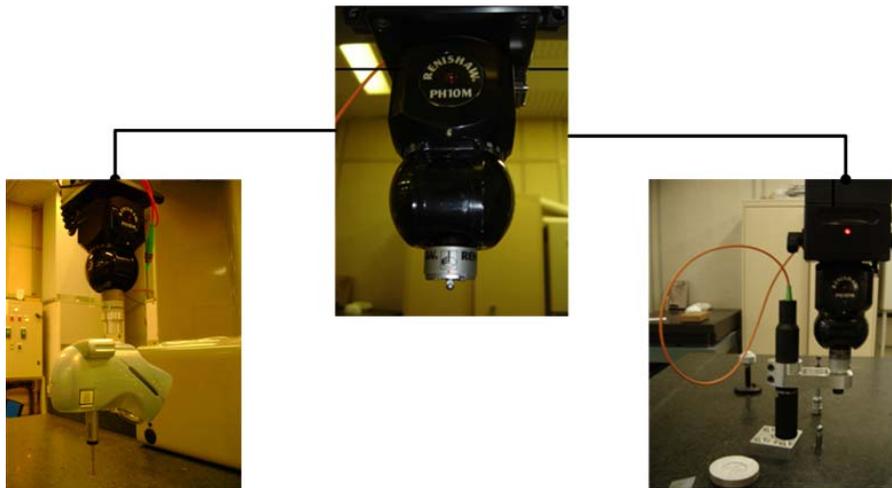


Figure 6-3: Two configurations of the sensor physical integration

Table 6-4: Comparison of the main characteristics of the three sensors

Sensor Attributes	Kreon KZ25 scanner	Renishaw TP2 probe	STIL CHR150-CL2 sensor
Principle	Laser triangulation	Kinematic touch trigger	Chromatic confocal imaging
Repeatability	3 μ m	0.35 μ m	0.012 μ m
Speed	30,000 point/second	1~2 point/second	Sampling frequency based (max. 1KHz)
Init. Data type	2D (R, C)	3D (X, Y, Z)	1D (Z)
Applied range	Macro	Macro	Micro (within 400 μ m)
Advantages	<ol style="list-style-type: none"> 1. Non contact 2. High capturing speed 3. Dense point data acquisition 4. Global information representation 5. suitable for the measurements of surfaces with soft/flexible materials 	<ol style="list-style-type: none"> 1. High accuracy/resolution 2. Direct 3D data acquisition 3. Simple and more adaptable to the environments 	<ol style="list-style-type: none"> 1. High axial resolution 2. Chromatic confocal setup 3. Coaxial without shadow 4. No limited to the materials and sample types
Disadvantages	<ol style="list-style-type: none"> 1. Lower resolution, noisy/redundant data 2. Limitations of occlusion and viewpoint 3. Sensitive to the surface optical conditions 	<ol style="list-style-type: none"> 1. Low data capturing speed 2. limitations to its own dimension sizes 3. Sparse density of the acquired points 	<ol style="list-style-type: none"> 1. Limitations to the micro/nano measurement ranges 2. No standard interface with CMM
Applications	<ol style="list-style-type: none"> 1. Global data acquisition 2. Complex surfaces or topography measure 3. Surfaces with soft/flexible materials 	<ol style="list-style-type: none"> 1. Primitive shapes 2. Features with known CAD models 3. Surfaces without large variations 	<ol style="list-style-type: none"> 1. Micro-topography, textures measurements 2. Thickness and roughness measurement; 3. Micro dimensional metrology.

6.2.2. Data acquisition

(1). *The case study workpiece (pump cover)*

An automotive water pump cover workpiece is selected as case study (figure 6-4). The pump cover has machined and rough surfaces. It should be an interesting case to test the performances of the proposed approaches for geometry processing.



Figure 6-4: Automotive water pump cover

(2). *Surface digitizing*

In the digitizing stage, the workpiece is digitized with the laser scanner and touch probe. The workpiece is located on the bed of the CMM when performing the measurement.

We digitized the workpiece using the laser scanner from 15 different views with two workpiece poses. 15 sets of point data are then obtained. The views of the laser scanner are adjusted by the Renishaw PH10M holder. The two poses of the workpiece are reversed in order to acquire entirely the point data from the top and the bottom of the pump cover.

Considering that it is difficult to scan the inner holes and the bottom plane. They require the measurements with more accuracy. We used the touch probe to acquire the touch point data from the bottom plane and the inner holes.

The measurement system configuration and some examples of the acquired data are shown in figure 6-5.

The point data acquired by the laser scanner contain many redundant data and noise. The imperfect quality of the initial acquired point data will deteriorate the performances of the proposed methods for geometry processing. Therefore, denoising

and filtering processes are necessary to improve the initial point data. Preprocessing procedures are beyond this thesis work. We use the existing functions provided by CATIA V5 and Geomagic software to do the preprocessing. The functions in CATIA, such as “Decimation”, “Optimize”, and the “denoising” function in Geomagic are helpful to improve the quality of the raw data.

The point data acquired by the STIL sensor are at micro scale. The point data acquired by the laser scanner and touch probe are at macro scale. There are still some difficulties in data registration phase because of the scale problem in our methods. In the studied case, we only take into account the processing of the data acquired by the laser scanner and the touch probe.

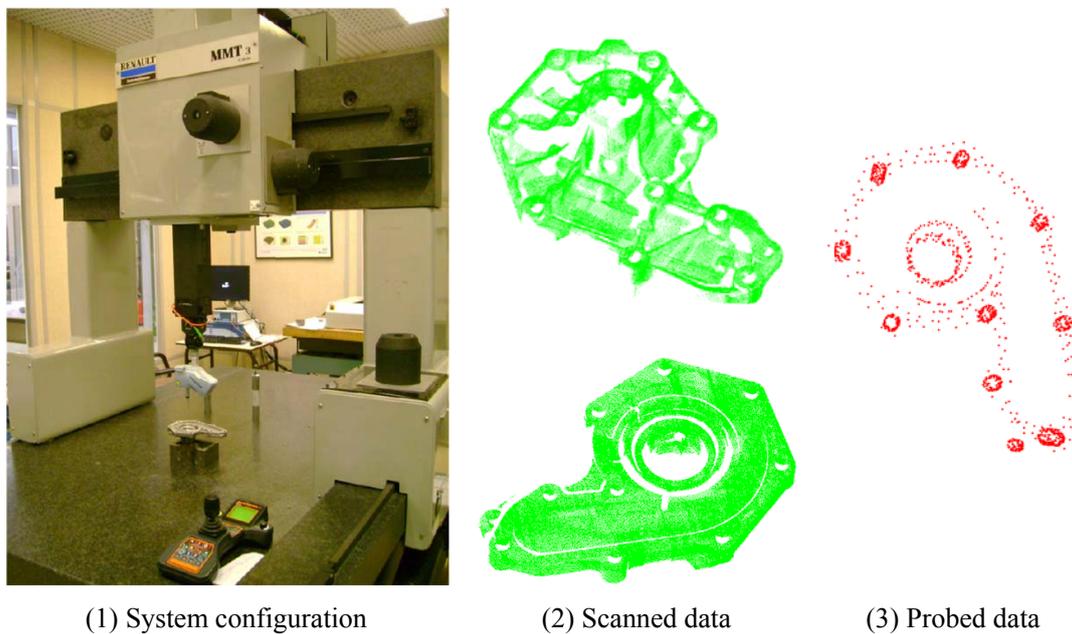


Figure 6-5: Measurement system configuration and examples of the acquired data

6.3. DSP-COMS platform overview

A software, called DSP-COMS (Discrete Shape Processing – for Coordinate Metrology Software), is developed as the test platform of all the algorithms proposed in the previous chapters. DSP-COMS is developed using MS VC++8.0. It provides a set of kernel functions for loading, registering, segmenting, and storing the mesh-based discrete shapes. Most of the functions are briefly discussed below.

6.3.1. Main interface

Figure 6-6 shows a screen shot of the main user interface, where the most important interaction tools are annotated:

- **Points** tool allows the user to show/hide the vertices of the discrete shape.
- **Mesh** tool allows the user to show/hide the polygon mesh of the discrete shape.
- **Triangle Facet** tool allows the user to show/hide the facet rendering of the discrete shape.
- **Shape index** tool is used to generate the shape index map of the discrete shape and control its visualization.
- **Curvedness** tool is used to generate the curvedness map of the discrete shape and control its visualization.
- **Vertex clustering** tool is used to cluster the vertices into different clusters and show/hide the clustering result.
- **Segmentation** tool is used to generate the initial segmentation result and show/hide the result.
- **Region merging** tool controls the merging process for segmented region refining and show/hide the merging results.
- **Load scene data** tool is used to load the scene data for registration. The model data is loaded using the button “Open”. It should be loaded before the scene data loading.
- **CFR fine registration** tool is used to execute the fine registration and show the final registration result.
- **Status bar** at the bottom is to show some detailed information at different process stages, such as the numbers of vertices, facets and edges contained in the discrete shape, the curvature estimation time, the vertex clustering time, the segmentation times and registration times, etc.

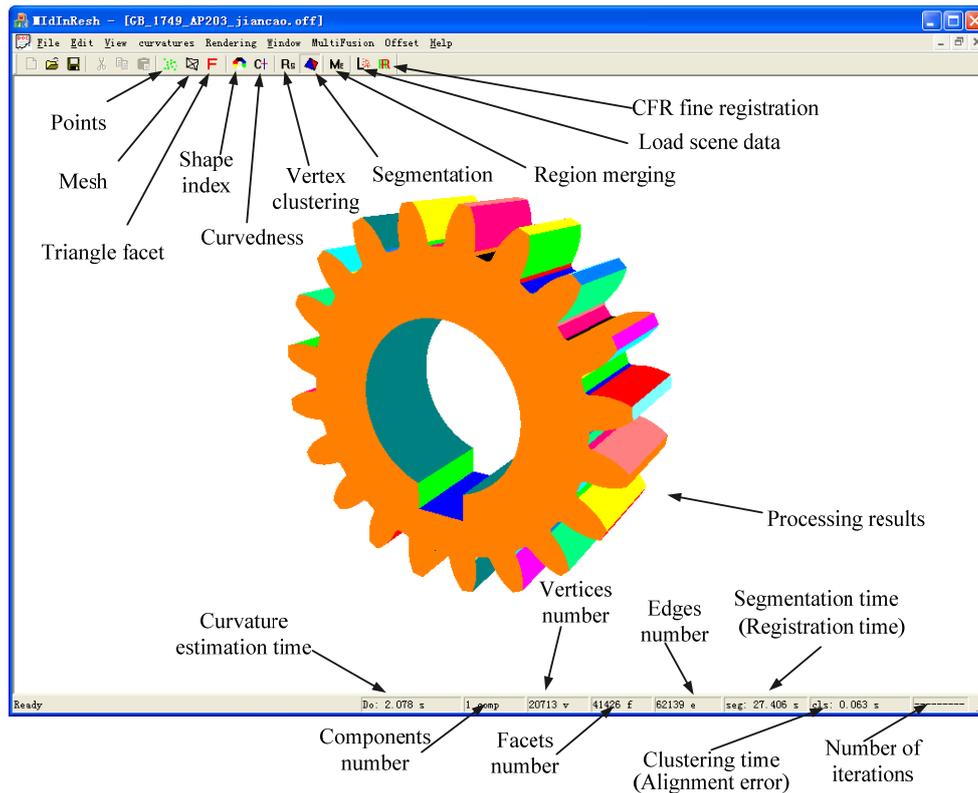


Figure 6-6: The main interface of DSP-COMS

6.3.2. Menu specification

The menu specification of DSP-COMS (figure 6-7) provides a more detailed description of the functions supported by the developed platform.

(a) **Curvatures.** The “Curvatures” menu is used for discrete curvature estimation of the discrete shape.

- **Estimation** submenu is used to estimate the two principal curvatures at each vertex of the discrete shape.
- **Filtering** and **Unshow filtering** submenus are used to control the high curvature points filtering results and their visualization.

(b) **Rendering.** The “Rendering” menu controls the visualization of curvature estimation results and the segmentation results.

- **Model** submenu is used to monitor the visualization of the discrete shapes as “points”, “mesh” or/and “triangle facet” ways.
- **Curvatures** submenu is used to show/hide the calculation results of the

parameters related to the principal curvatures, such as Gaussian curvature, mean curvature, shape index and curvedness, etc.

- **Normal** submenu controls the visualization of the normal vectors, the vertex-based normal or the facet-based normal.
- **Boundaries** submenu is used to estimate the boundary points in the discrete model.
- **Segmentation** submenu is used to execute the segmentation and then show/hide the results related to vertex clustering, connected region labeling and region merging.

(c) **MultiFusion**. The “MultiFusion” menu is used for registration purpose.

- **Load scene data** submenu is used to load the scene data and show the relative positions of the model data and the scene data.
- **Coarse registration** submenu is used to execute the coarse registration and show/hide the alignment results. “PCA based” controls the execution of principal poses alignment while “SC feature based” controls the execution of overlapping alignment.
- **Fine registration** submenu is used to execute the fine registration and show/hide the results. “CFR method” is used to execute the CFR method for registration and then show/hide the result. “Initial ICP” is used to execute the initial ICP algorithm and show/hide the registration result.

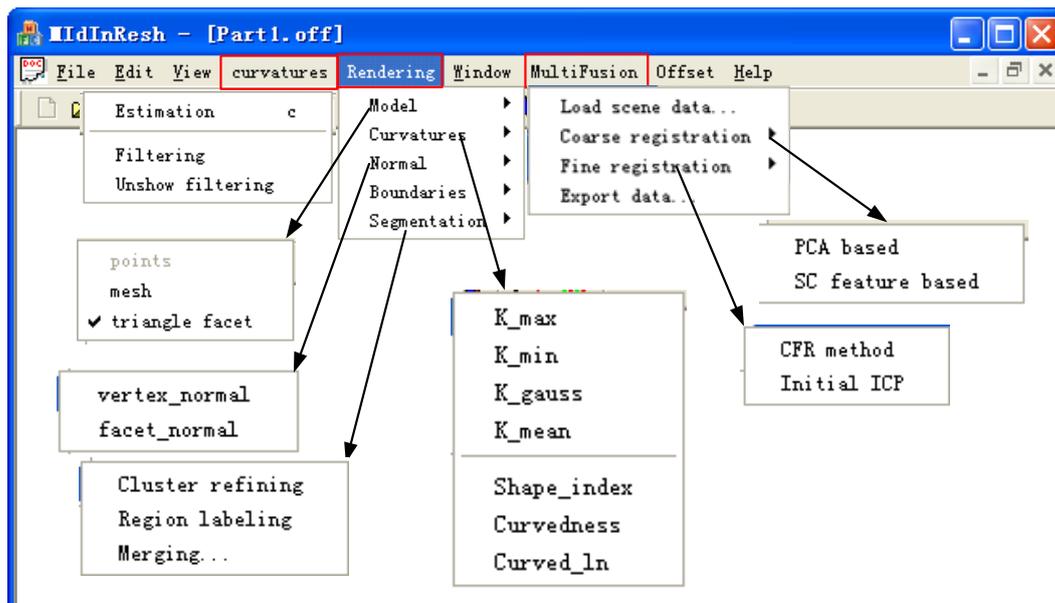


Figure 6-7: Detailed menu specification of the software

6.4. Discrete geometry processing

6.4.1. Registration

During data acquisition phase, we obtain range data by the touch probe and the laser scanner from different views. It is necessary to register them together to construct the complete discrete model of the workpiece. The registration process is performed based on the accumulative piecewise registration. In this section, we present the registration processing in three different cases.

(1). *Case 1: Different views with the same workpiece pose*

In this case, the two data are acquired by the laser scanner from different views, and the workpiece keeps the same pose during the scanning process. The shapes of the two input data and their relative positions are shown in figure 6-8. The data with gray color is viewed as the model data, while the other colorized by yellow-brown is viewed as the scene data. The model data contains 30868 vertices and the scene data contains 33550 vertices.

The results in coarse registration are shown in figure 6-9. And the fine registration results using our CFR method and the initial ICP algorithm are shown in figure 6-10. The CFR method takes 43962s and the alignment error is 0.319338mm. The initial ICP algorithm takes 127012s and the alignment error is 5.18128mm. From the comparison of the two methods, we can see that the CFR method provides the better performance.

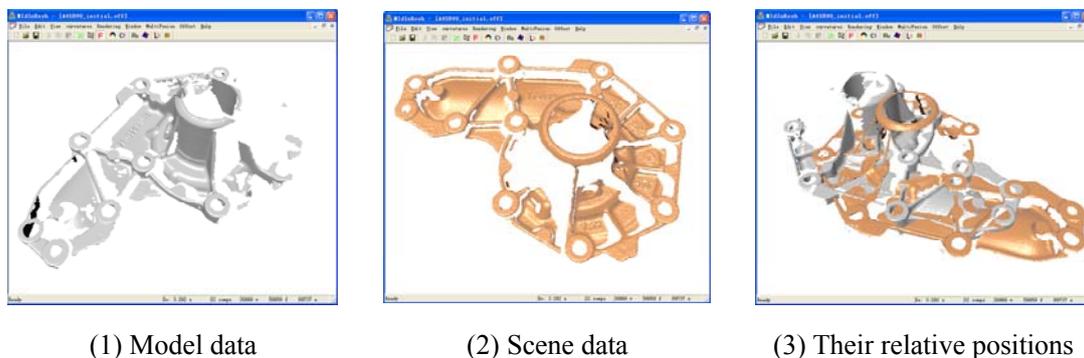


Figure 6-8: The scene data and the model data in case 1

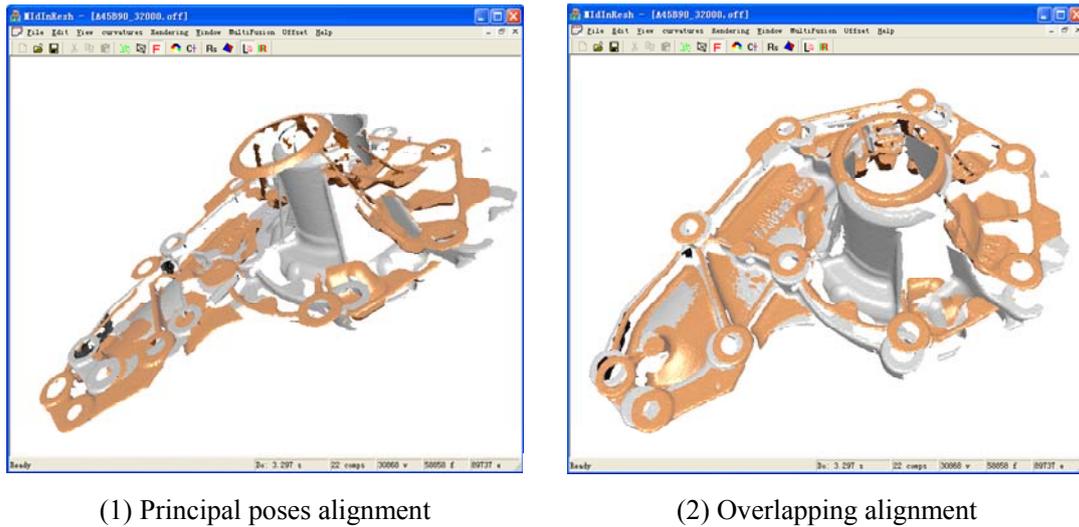


Figure 6-9: Coarse registration results in case 1

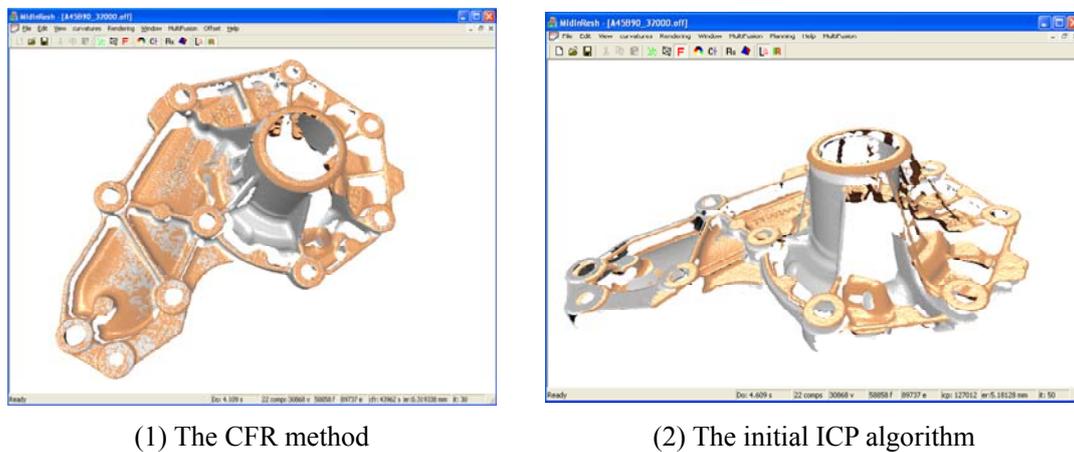


Figure 6-10: Fine registration results in case 1

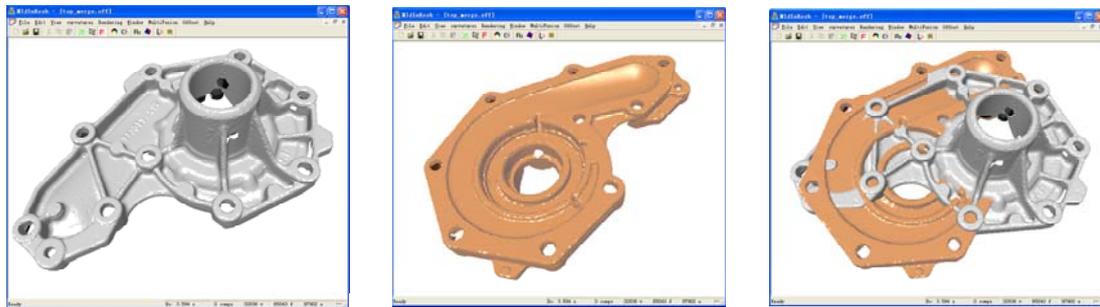
(2). *Case 2: Different poses of the workpiece*

During the measurement, the contact surfaces of the workpiece with the CMM bed are always inaccessible by the sensors. In order to obtain the complete data, the pose of the workpiece needs to be changed. In case 2, we present the registration results of the data acquired with different poses of the workpiece.

Figure 6-11 shows the two input data and their relative positions. The model data contains 32838 vertices while the scene data includes 21127 vertices.

Figure 6-12 shows the results of the coarse registration. Figure 6-13 illustrates the fine registration results using the CFR method and the initial ICP algorithm respectively. The CFR method does the registration in 99655.6s with 30 iterations. The alignment error is 0.408431mm. The initial ICP algorithm, however fails to

register the two data with satisfying accuracy (the alignment error is 1.2187mm) even within a time scale of 142987s with 50 iterations.

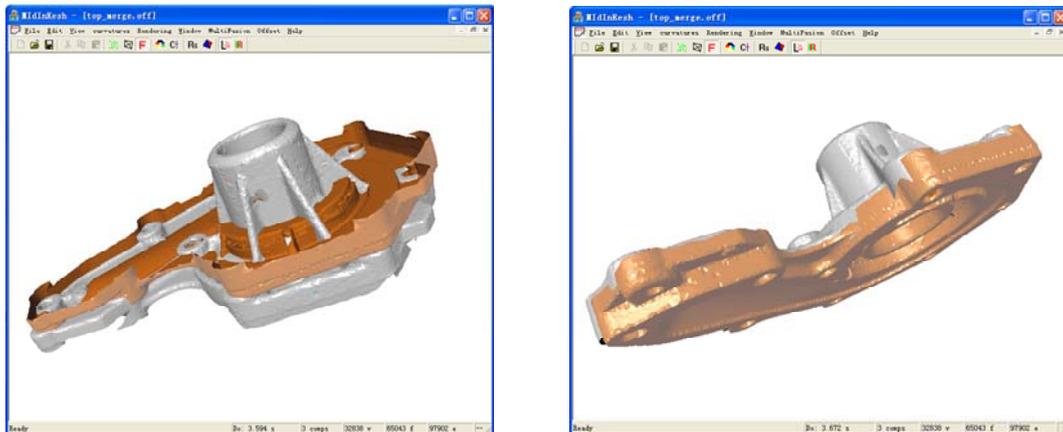


(1) Model data

(2) Scene data

(3) Their relative positions

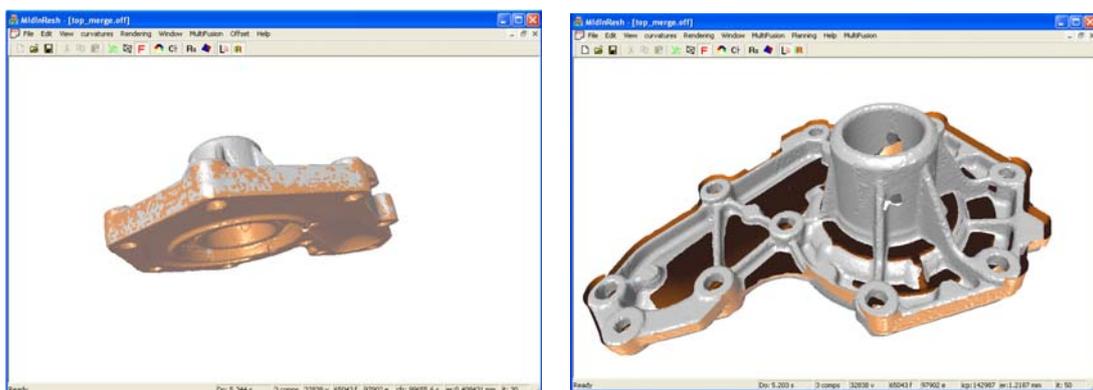
Figure 6-11: The scene data and the model data in case 2



(1) Principal poses alignment

(2) Overlapping alignment

Figure 6-12: Coarse registration results in case 2



(1) The CFR method

(2) The initial ICP algorithm

Figure 6-13: Fine registration results in case 2

(3). Case 3: Different sensors

Merging the data from different sensors is an important issue in surface digitizing. In order to show the performance of our method for such purposes, we probe the bottom plane and the holes of the workpiece by touch probe and register the probed data into the scanned data.

Figure 6-14 shows the two input data and their relative positions. The data acquired from the laser scanner, which contains 17314 vertices, is viewed as the model data. The probed data is viewed as the scene data, which contains 1200 vertices.

The coarse registration results of the two input data are shown in figure 6-15. The figure 6-16 shows the fine registration results. For fine registration, the CFR method takes 456.781s with 30 iterations. The alignment error is 0.0816mm. The initial ICP algorithm takes 806.026s with 50 iterations. The alignment error is 0.846mm.

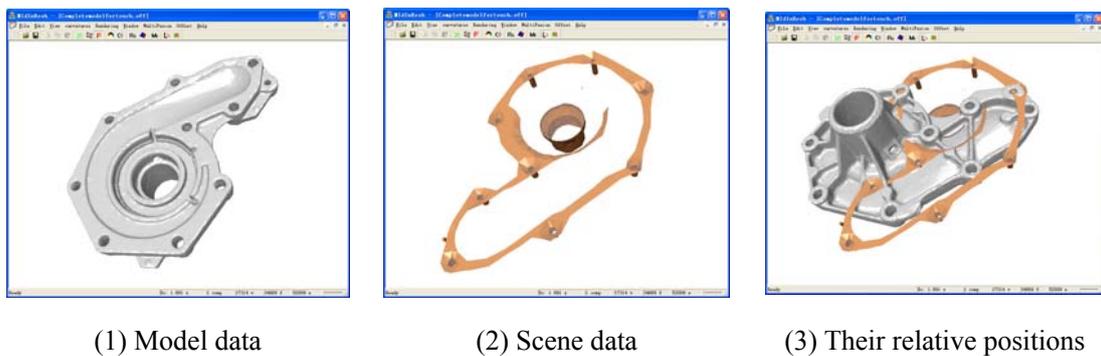


Figure 6-14: The scene data and the model data in case 3

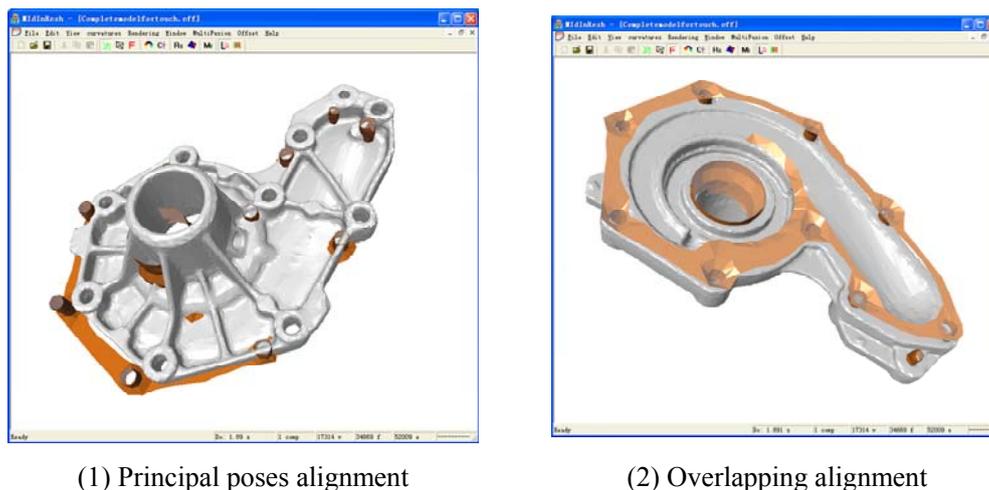
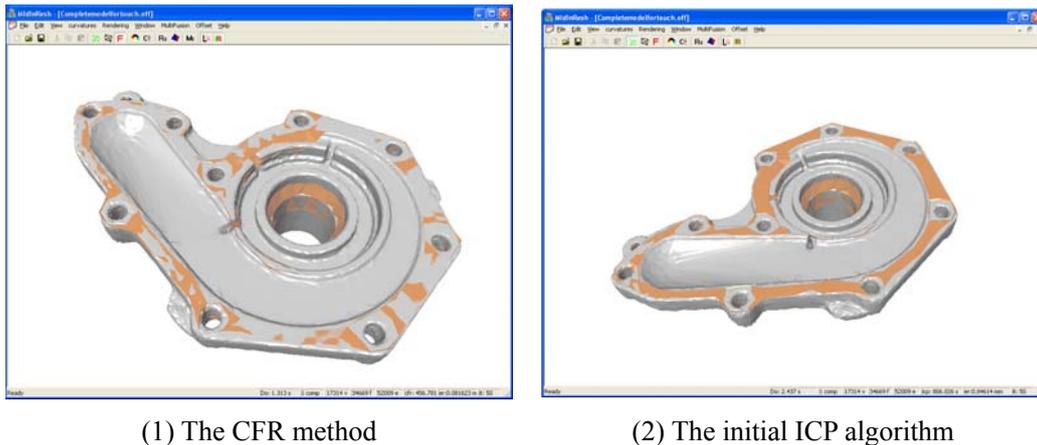


Figure 6-15: Coarse registration results in case 3



(1) The CFR method

(2) The initial ICP algorithm

Figure 6-16: Fine registration results in case 2

(4). *The final result*

With the proposed CFR method, we register all the data patches accumulatively piece by piece. Finally, we can get the registered model of the whole workpiece. After denoising and filtering, the final discrete shape of the workpiece is shown in figure 6-17. The discrete shape contains 141943 vertices.

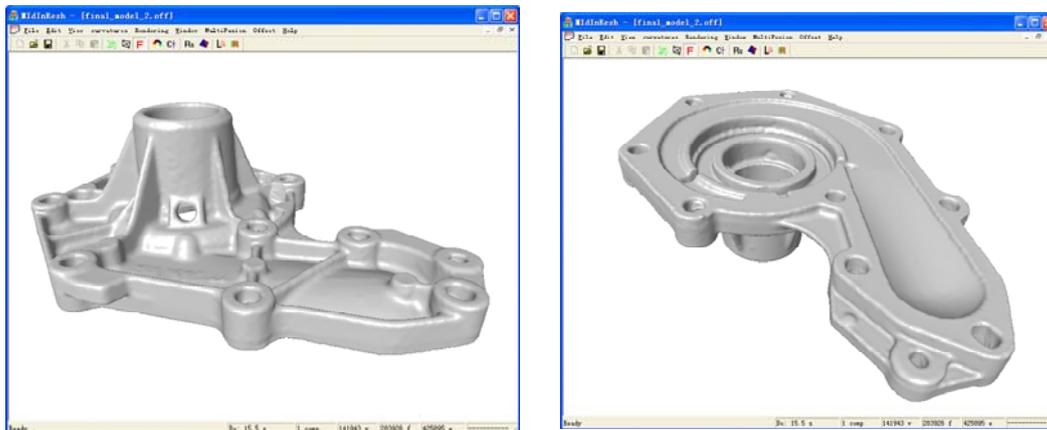


Figure 6-17: Final discrete shape of the workpiece

6.3.2. Segmentation

(1). *Discrete curvature estimation*

The discrete curvature estimation of the discrete shape can be estimated using the proposed method for discrete curvature estimation, the maximum and minimum principal curvature maps are shown in figure 6-18 and figure 6-19.

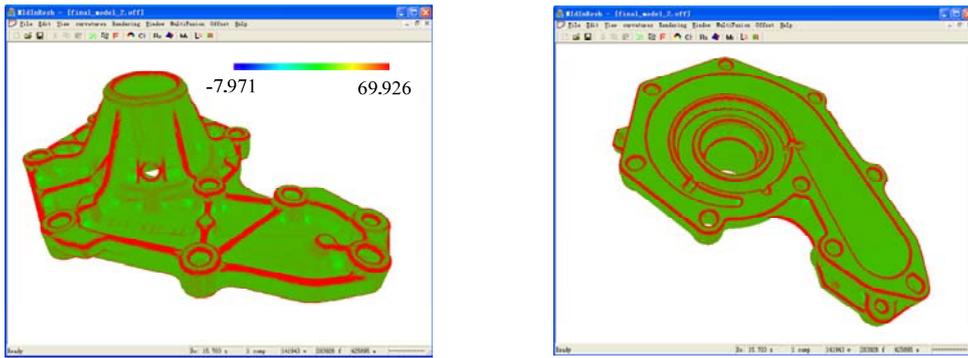


Figure 6-18: The maximum principal curvature map of the workpiece

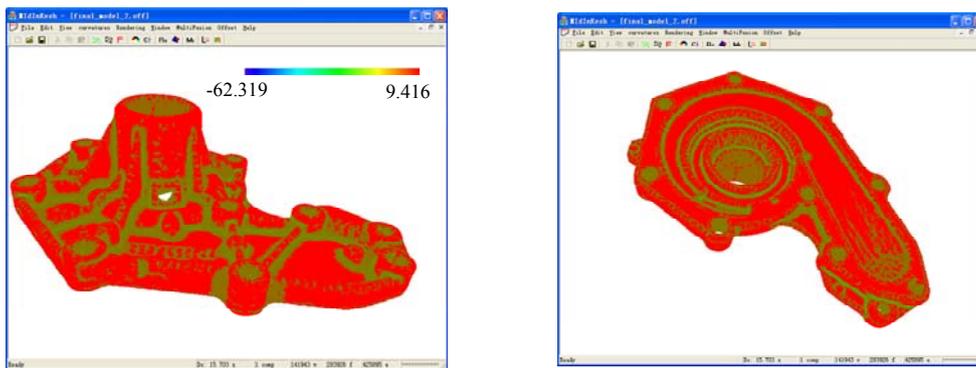


Figure 6-19: The minimum principal curvature map of the workpiece

(2). *Shape index and curvedness maps*

Based on the two principal curvatures, the two surface indicators, shape index and curvedness, for each vertex can be computed at each vertex. The results of the shape index map and the curvedness map are illustrated in figure 6-20 and figure 6-21.

From figure 6-4, we can see that the workpiece contains many rough surfaces with large form deviations. The point data acquired from these surfaces lead to a “chaotic” shape index map (figure 6-20).

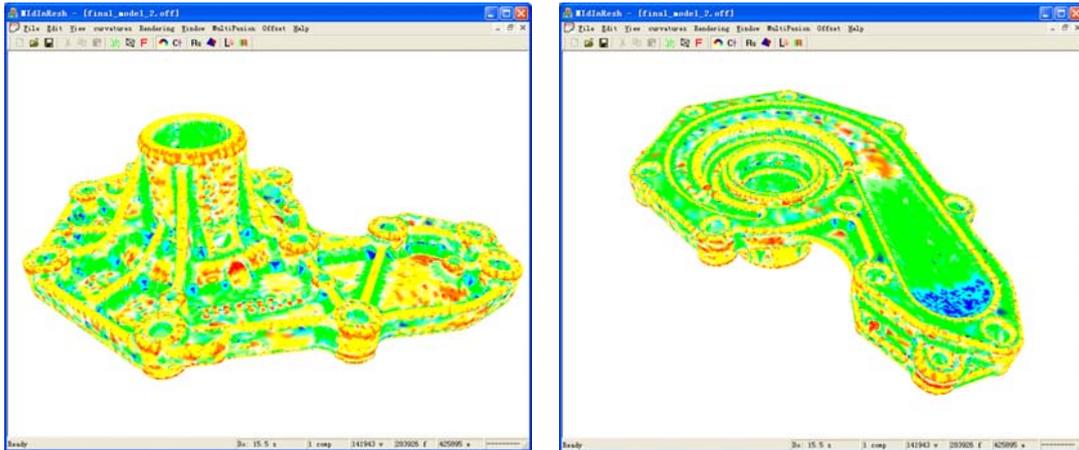


Figure 6-20: The shape index map of the workpiece

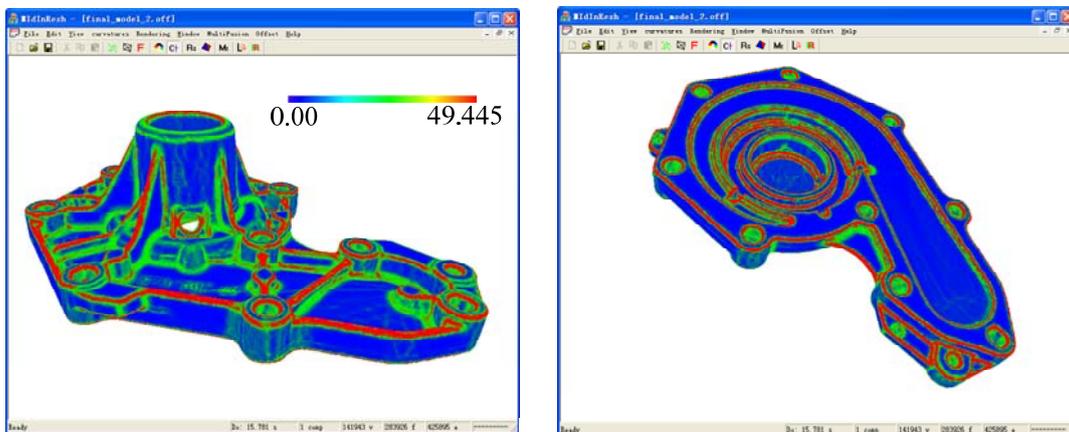


Figure 6-21: The curvedness map of the workpiece

(3). *High curvature points recognition*

Due to the rough surfaces of the workpiece and the uncontrolled random noise during the measurement the point data acquired from the real workpiece don't have good quality. There are still some limitations to generate robust results of local surface type recognition. However, we can recognize the high curvature points from the discrete shapes (figure 6-22) by adjusting the threshold of curvedness to $c_{th} = 0.55$.

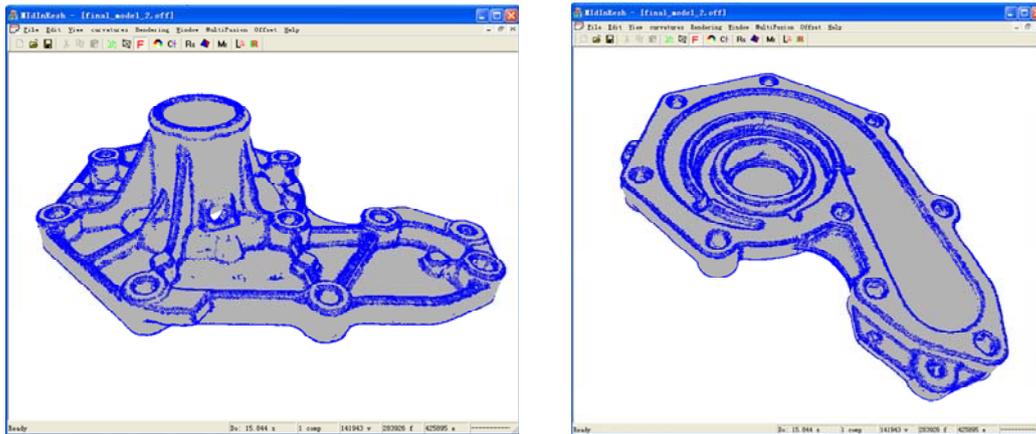


Figure 6-22: Recognition results of the high curvature points

(4). *Segmentation results*

The final segmentation result of the discrete shape of the workpiece is shown in figure 6-23. The segmentation process takes 2290.6s. We cluster the vertices basically according to the high curvature points identified in the previous phase. The noisy point data acquired from rough shapes make it difficult to obtain segmented regions with smooth boundaries. However, the proposed method can provide satisfying segmentation results.

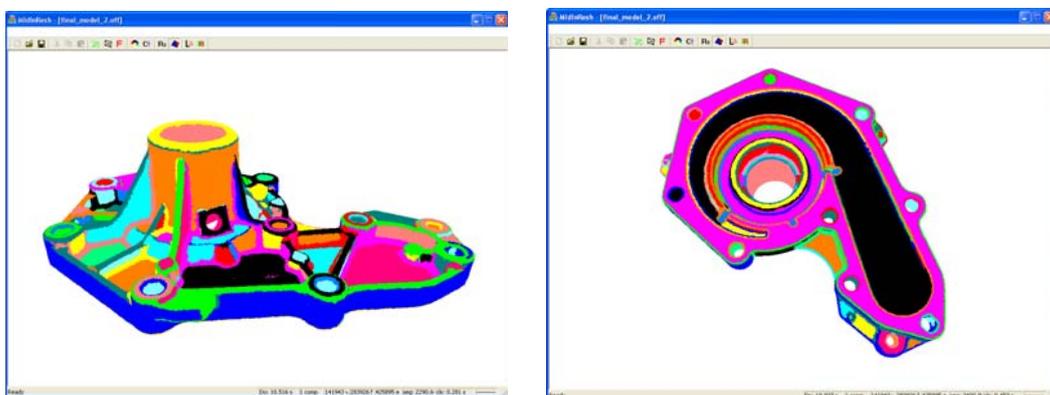


Figure 6-23: The final segmentation result

6.5. Conclusion

This chapter presents a detailed case study of an automotive water pump cover workpiece.

The multisensor platform of LURPA is first introduced. A comparative analysis of

the integrated sensors in the system is presented. Based on the multisensor configuration, we specify a measurement strategy to acquire the point data from the pump cover by two sensors (Kreon laser range scanner and Renishaw touch probe).

DSP-COMS software, designed as a modular software framework, has been developed to test and support the methods and algorithms developed in this thesis.

The detailed processing results of the studied case are based on DSP-COMS.

Three cases of the registration are discussed: (1) data from different views of laser scanner; (2) data with different poses of workpiece; and (3) data from different sensors. With registration, the acquired point data are aligned together and the complete shape of the workpiece is generated.

The principal curvature estimation results, the recognition of high curvature points and the final segmentation result are also presented.

We conclude that the developed methods in previous chapters provide satisfying results for this case study.

Conclusion

Multisensor integration is proven to realize measurements with holistic, more accurate and reliable information. Multisensor integration systems are increasingly used to fulfill the increasing requirements about the complexity and accuracy of coordinate metrology. This dissertation presented several significant contributions to achieve the multisensor integration in coordinate metrology from two parts: function and data modeling and discrete geometry processing.

A set of detailed hierarchical function models of the developed multisensor integration system has been specified using IDEF0 methodology. The system achieves the multisensor integration based on four activities. The information flow and resources in the system have been also specified clearly.

An ontology based method has been setup to model the various data in the system with structured representation using Protégé-OWL. According to different measurement stages defined in the function models, the data model of the systems are built with four basic data models. The hierarchical data and their inter-relationships are defined consistently.

Several methods have been developed for discrete geometry processing of the data acquired using multisensor integration system in coordinate metrology.

Estimation methods were developed to compute the normal and discrete curvatures of vertices of polygon mesh. The polygon mesh topological structure is associated to the scattered point data and the normal at each vertex is estimated from the normal vectors of the facets in its one-ring neighborhood region. A new method has been developed to estimate the discrete curvatures using a tensor based technique. In this method, the discrete shape operator matrix is computed at each vertex within its Voronoi cell. The principal curvatures and the principal directions are then derived from the eigenvalues and the eigenvectors of the discrete shape operator matrix. The shape index and curvedness, introduced for local shape description, are derived from the principal curvatures.

A robust method was developed to register the data captured by multiple sensors/views into one common coordinate system. The curvature information is used to improve the registration performances. Coarse registration and fine registration are combined in the method. In coarse registration, the principal poses of the two discrete shapes are estimated based on PCA and they are aligned accordingly. A new distance

based on shape index and curvedness is defined to search the overlapping regions on which the two shapes are aligned roughly. During fine registration, we defined a new distance, named geometric distance which combines the Euclidean distance and curvature ratio distance to measure the similarity between two points. An iterative method based on geometric distance is developed for fine registration with better performance. The complete shape of the measured workpiece can be generated based on the proposed registration method.

A set of new methods for discrete shape recognition and segmentation were proposed. Based on shape index, ten basic surface types are defined for shape recognition purposes. The sharp edge points and high curvature points are recognized based on curvedness. A new clustering algorithm has been developed to classify the vertices into appropriate clusters. A new method was developed to recover the connected regions based on the vertex clustering result and segment the discrete shape accordingly. The cluster refining and segmented region merging algorithms have been developed to improve the robustness of the proposed segmentation methods with respect to the noise.

We have also developed a modular software framework (DSP-COMS) to test and support the methods and algorithms developed in this thesis. DSP-COMS is developed based on MS VC++ 8.0 platform, supported by OpenGL and CGAL libraries [CGAL].

A detailed case study of an automotive water pump cover workpiece was presented. The workpiece was digitized to acquire the point data using Kreon laser range scanner and Renishaw touch probe based on the multisensor platform of LURPA. The acquired point data of the studied workpiece was processed based on DSP-COMS and the detailed processing results were presented.

The contributions of this dissertation to multisensor integration and discrete geometry processing for coordinate metrology are as follows:

- A detailed functional analysis for multisensor integration achievement in coordinate metrology has been developed. The specification of the requirements of the process activities, the information flow, the means and resources provide clear guidelines for system understanding and development. The function models

can be used to guide the development of other multisensor system in coordinate metrology.

- Ontologies were introduced to model the data for multisensor integration system in coordinate metrology. Ontologies provide effective means to structure and manage all the information in the system and make it easily accessible. The proposed ontology based method enables processing and managing the data and knowledge in the domain of multisensor integration for coordinate metrology.
- A set of new methods has been proposed to estimate the normal and discrete curvatures of discrete shapes with reliable results. A comparative analysis of different methods for discrete curvature estimation has been studied and the results show that the proposed method which is based on tensor-based technique and the Voronoi cell region provides more reliable curvature estimation results.
- The proposed methods for registration provide new way to combine the curvature and the Euclidean distance. This combination improves the registration performances both in coarse and fine registration. Different from other automatic methods for coarse registration, which just align the two data from the global view, the proposed method does the coarse registration based on their overlapping sections. The comparative results with the initial ICP algorithm show that our geometric distance based fine registration method provides better performances both in alignment accuracy and convergence speed.
- The developed methods for discrete shape recognition and segmentation use shape index and curvedness, which is different from the traditional methods that are mostly based on the principal curvatures or Gaussian/mean curvatures. The vertex cluster refining and segmented region merging processes improve the robustness to the noise. The proposed methods provide satisfying recognition and segmentation results for the objects with sharp, smooth and curvature smooth edges. The time performances are also satisfying.
- A software framework, DSP-COMS, has been developed to support the discrete geometry processing methods developed in this thesis. The framework is modular and it is easy to integrate new functions for discrete geometry processing.

The multisensor integration and discrete geometry processing for coordinate

metrology comprise many other topics which haven't been addressed in this thesis. Some promising future works are presented below:

- Multiscale measurement.

The advantages of different sensor measuring principles can be well combined by cooperative integration of inhomogeneous sensors with different resolutions in multiscale measurement [WJS09]. However, there are still some limitations to register the point data from different scales in our work. The multiscale measurement should be a promising perspective work. Some scales adjustments may be developed to make the proposed registration methods suitable for multiscale measurement.

- Multidata fusion.

The multidata fusion process is used to decide which measurement data is integrated into the final data set and how redundant data are combined. It should be done after registration. However, we haven't considered the multidata fusion in this dissertation. It should be another promising work in the future. Some quality criteria, such as measurement uncertainty, have to be taken into account in multidata fusion [WJS09].

- Data preprocessing.

Point data provided by multiple sensors are raw data which contain many noise points, outliers, incomplete and redundant data. It would be difficult to handle the raw data directly. It is necessary to preprocess the raw data to refine their quality.

- Ontologies construction.

There is little work on ontology development of multisensor integration system in coordinate metrology. Considering the data reusing and sharing purposes in a PLM context, ontologies is promising way to manage and to represent the data unambiguously. We have already constructed the ontology model of the developed multisensor system within this dissertation. It should be interesting to extend the ontologies in order to satisfy the new requirements when integrating new resources, like new sensors, new information systems, etc.

References

- [AB98] N. Amenta, M. Bern. Surface reconstruction by Voronoi filtering, Proceedings of SoCG'98, 14th Annual Symposium on Computational Geometry, 39-48, 1998
- [ABM06] M. Attene, S. Biasotti, M. Mortara, et al. Computational methods for understanding 3D shapes, Journal of Computer and Graphics, 30: 323-33, 2006
- [ABR05] S. Ajmal, M. Bennamoun, A. Robyn. Automatic multiview registration of range images acquired from unknown view points. Proceedings of ICMSAO/05, 1-6, 2005
- [ACD03] P. Alliez, D. Cohen-Steiner, O. Devillers, et al. Anisotropic polygonal remeshing, Transactions of ACM on Graphics, 22 (3): 485-93, 2003
- [ACT07] P. Alliez, D. Cohen-Steiner, Y. Tong, et al. Voronoi-based variational reconstruction of unoriented point sets, Symposium of Eurographics on Geometry Processing, 39-48, 2007
- [AHB87] K. Arun, T. Huang, S. Blostein. Least squares fitting of two 3-D point sets. Transactions on Pattern Analysis and Machine Intelligence, 9 (5): 698-700, 1987
- [Aim] <http://shapes.aimatshape.net/viewmodels/>
- [ALA99] V. Albuquerque, F. Liou, S. Agarwal, et al. Automatic coordinate measuring machine inspection, Proceedings of ASME Design Engineering Technical Conferences, DETC99/DAC-8700, 1-11, 1999
- [ALD07] A. Almhdie, C. Leger, M. Deriche, et al. 3D registration using a new implementation of the ICP algorithm based on a comprehensive lookup matrix: application to medical imaging. Pattern Recognition Letters, 28: 1523-33, 2007
- [ALG95] C. Ang, M. Luo, R. Gay. Knowledge-based approach to the generation of IDEF0 models, Journal of Computer Integrated manufacturing System, 8 (4): 279-90, 1995
- [ALT05] J. Aguilar, M. Lope, F. Torres, et al. Development of a stereo vision system for non contact railway concrete sleepers measurement based in holographic optical elements, Journal of Measurement, 38: 154-165
- [AMF00] A. Alrashdan, S. Motavalli, B. Fallahi. Automatic segmentation of digitized data for reverse engineering applications, IIE Transactions, 32: 59-69,

2000

- [ASG04] J. Aguilar, J. Santolaria, D. Guillomia, et al. Accuracy analysis of laser scanning probes used in coordinate measurement: simulation and experiments, *Journal of VDI Berichte*, 1860: 739-44, 2004
- [ATS04] I. Ashokaraj, A. Tsourdos, P. Silson, et al. Sensor based robot localization and navigation: using interval analysis and unscented Kalman filter, *Proceedings of IEEE/RSJ, International Conference on Intelligent Robots and Systems*, 7-12, 2004
- [AV04] O. Andersen, G. Vasilakis. Building an ontology of CAD model information, In *Book of Geometric Modeling, Numerical Simulation, and Optimization*, Springer Berlin Heidelberg, Part I: 11-40, 2004
- [BA00] P. Baker, Y. Aloimonos. Complete calibration of a multi-camera network, *Proceedings of IEEE workshop on Omnidirectional Vision*, 134-41, 2000
- [BC02] J.D. Boissonnat, F. Cazais. Smooth surface reconstruction via natural neighbor interpolation of distance functions, *Journal of Computational Geometry: Theory and Applications*, 22 (1): 185-203
- [BCD93] M. Brunet, E. Cosnard, B. Deltour. Procédé d' étalonnage d'un système d'acquisition tridimensionnelle de forme et du système pour la mise en oeuvre du dit procédé, Brevet 0452422B1, KREON Ind. 1993
- [BDW04] G. Bendels, P. Degener, R. Wahl, et al. Image-based Registration of 3D range data using feature surface elements, *Proceedings of 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2004)*, 115-24, 2004
- [Ber07] P. Bergstrom. A method suitable for reiterated matching of surface. *Proceedings of 10th SIAM Conference on Geometric Design and Computing*, 2007
- [Bie85] L. Biederman. Human image understanding: recent research and a theory, *Journal of Computer Vision, Graphics and Image Process*, 32 (11): 29-73, 1985
- [BJ88] P.J. Besl, R.C. Jain. Segmentation through variable order surface fitting, *Transactions of IEEE on Pattern Analysis and Machine Intelligence*, 10(2): 167-92, 1988

- [BL95] G. Blais, M. Levine. Registering multiview range data to create 3D computer object. *Transaction of IEEE on Pattern Analysis and Machine Intelligence*, 17(8): 820-4, 1995
- [BLV03] J. Barreiro, J. Labarga, A. Vizan, et al. Functional model for the development of an inspection integration framework, *Journal of Machine Tools & Manufacture*, 43: 1621-32, 2003
- [BM92] P. Besl, N. McKay. A method for registration of 3D shapes. *Transactions of IEEE on PAMI*, 14 (2): 239-56, 1992
- [BMT06] S. Brandt, J. Morbach, M. Miatidis, et al. Ontology-based information management in design processes, *Symposium of 16th ESCAP and 9th PSE*, 2021-6, 2006
- [BRM92] W. Blair, T. Rice, B. McDole, et al. Least-squares approach to asynchronous data fusion, *Proceedings of SPIE on Acquisition, Tracking, and Pointing VI*, 1697 (130): 130-41, 1992
- [BS99] R. Benjemma, F. Schmitt. Fast global registration of 3D sampled surface using a multi-z-buffer technique. *Journal of Imaging and Vision Computing*, 17: 113-23, 1999
- [Bur93] M. Burns. *Automated fabrication: Improving productivity in manufacturing*. Prentice Hall Press, 1993
- [BV04] P. Benko, T. Varady. Segmentation methods for smooth point regions of conventional engineering objects, *Journal of Computer-Aided Design*, 36: 511-23, 2004
- [CBV01] V. Chan, C. Bradley, G. Vickers. A multi-sensor approach to automating coordinate measuring machine-based reverse engineering, *Journal of Computers in Industry*, 44: 105-15, 2001
- [CDS97] B. Chazelle, D.P. Dobkin, N. Shouraboura, et al. Strategies for polyhedral surface decomposition: An experimental study. *Journal of Computational Geometry*, 7: 327-42, 1997
- [CGAL] <http://www.cgal.org/>
- [CH82] R. Chin, C. Harlow. Automated visual inspection: a survey, *Transactions of IEEE Pattern Analysis and Machine Intelligence*, 4 (6): 557-73, 1982

- [Cha95] J. Challis. A procedure for determining rigid body transformation parameters, *Journal of Biomechanics*, 28 (6): 733-737, 1995
- [CHC98] C. Chen, Y. Hung, J. Cheng. A fast automatic method for registration of partially-overlapping range images, *Proceedings of 6th International Conferences on Computer Vision '98*, 242-8, 1998
- [CL97] L. Chen, G. Lin. A vision-aided reverse engineering approach to reconstructing free-form surfaces, *Robotic and Computer-Integrated Manufacturing*, 13 (4): 323-36, 1997
- [CLB09] M. Chuang, L. Luo, B.J. Brown, et al. Estimating the Laplace-Beltrami operator by restricting functions, *Proceedings of Eurographics Symposium on Geometry*, 28 (5): 1475-84, 2009
- [CM03] D. Cohen-Steiner, J.M. Morvan, Restricted Delaunay Triangulation and Normal Cycle, *Proceedings of 19th Annual ACM Symposium on Computational Geometry*, 312-21, 2003
- [CM91] Y. Chen, G. Medioni. Object modeling by registration of multiple range images, *Proceedings of IEEE International Conference on Robotics and Automation*, 2724-9, 1991
- [CM95] W. Cheng, C. Menq. Integrated laser/CMM system for the dimensional inspection of objects made of soft material, *Journal of Advanced Manufacturing Technology*, 10 (1): 36-45, 1995
- [CRB04] C. Chappuis, A. Rasineux, A. Breiktopf, et al. Improving surface meshing from discrete data by feature recognition, *Journal of Engineering with Computers*, 20(3): 202-9, 2004
- [CN04] R. Christoph, H. Neumann. *Multisensor coordinate Metrology: Measurement of form, size, and location in production and quality control*, verlag moderne industrie, 2004
- [Cos07] R. Costadoat. *Implantation d'un capteur Stil dans une cellule de mesure multi-capteurs*, Rapport de Master Sciences de L'Ingénieur, 2007
- [CSS02] D. Chetverikov, D. Svirko, D. Stepanov, et al. The trimmed iterative closest point algorithm, *Proceedings of IEEE on ICPR'02*, 3: 545-8, 2002
- [CVY02] D. Cohen-Steiner, E.C. de Verdiere, M. Yvinec. *Conforming Delaunay*

- triangulations in 3D, Proceedings of SoCG'02, 199-208, 2002
- [CWL03] H. Chen, B. Wang, X. Luo, et al. Multi-sensor integrated automated inspection system, Proceedings of SPIE, 5253: 528-31, 2003
- [CWL05] H. Chen, B. Wang, X. Luo, et al. Automatic Free-Form surface inspectoin system based on conoscopic holography and CAD database, Symposium of ISTM on Test and Measurement, 6: 5247-50, 2005
- [CZ08] Carl Zeiss documentation, <http://www.zeiss.com/>, 2008
- [DAS95] Deputy Assistant Secretary of Defense of the United States, Reader's guide to IDEF0 function models,
<http://www.archives.gov/era/pdf/rmsc-19951006-dod-rm-function-and-information-models.pdf>, 1995
- [Dey06] T.K. Dey. Curve and surface reconstruction: Algorithm with mathematical analysis. Cambridge University Press, 2006
- [Dey07] T.K. Dey. Delaunay mesh generation of three dimensional domains, Technical Report, OSU-CISRC-9/07-TR64, October, 2007
- [DG06] T.K. Dey, S. Goswami. Provable surface reconstruction from noisy samples, Journal of Computational Geometry: Theory and Applications, 35 (1): 124-41, 2006
- [DJ97] C. Dorai, A.K. Jain. COSMOS-A representation scheme of 3D free-form object, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (10): 1115-30, 1997
- [Dims] <http://www.dmisstandards.org/>
- [Dml] <http://www.dmisstandard.com/DML/>
- [DoC76] M. Do Carmo. Differential geometry of curves and surfaces, Prentice-Hall, Englewood Cliffs, New Jersey, 1976
- [DSH01] R. Duda, P. Hart, D. Stork. Pattern classification, John Wiley & Sons (2ed), 2001
- [Dub] <http://dublincore.org/>
- [Dur88] H. Durrant-Whyte. Sensor models and multisensory integration, Journal of Robotics Research, 7: 97-113, 1988

- [DVD02] J. Demongeot, G. Virone, F. Duchene, et al. Multi-sensors acquisition, data fusion, knowledge mining and alarm triggering in health smart homes for elderly people, *Journal of Comptes Rendus Biologies*, 325 (6): 673-82, 2002
- [DVV07] K. Demarsin, D. Vanderstraeten, T. Volotine, et al. Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Journal of Computer-Aided Design*, 39: 276-83, 2007
- [EC01] W. EIMaraghy, C. Rolls. Design by quality product digitization, *Journal of CIRP Annals-Manufacturing Technology*, 50 (1): 93-6, 2001
- [ESW05] J. Esteban, A. Starr, R. Willetts, et al. A review of data fusion models and architectures: towards engineering guidelines, *Journal of Neural Computing & Applications*, 14 (4): 273-81, 2005
- [EY03] H. EIMaraghy, X. Yang. Computer-Aided planning of laser scanning of complex geometries, *Journal of CIRP Annals-Manufacturing Technology*, 52 (1): 411-4, 2003
- [EW96] R. Edgeworth, R. Wilhelm. Uncertainty management for CMM probe sampling of complex surfaces, *Proceedings of ASME Manufacturing Science and Engineering, MED*, 4: 511-8, 1996
- [Fen94] S. Feng. Dimensional inspection planning based on product data standards, *Proceedings of Concurrent Engineering: Research and Applications Conference and Application*, 1994
- [FH86]. O. Faugeras, M. Hebert. The representation, recognition and locating of 3-d objects. *International Journal of Robotic Research*, 5(3): 27-52, 1986
- [FM09] S. Fourey, R. Malgouyres. Normal estimation for digital surfaces based on convolutions, *Journal of Computer Graphics*, 33: 2-10, 2009
- [Fra08] Fraunhofer-Allianz Vision, <http://www.vision.fraunhofer.de/>
- [GAF00] M. Gruninger, K. Atefi, M. Fox. Ontologies to support process integration in enterprise engineering, *Journal of Computatoinal & Mathematical Organization Theory*, 6: 381-94, 2000
- [GAG09] S. Gundimada, V. Asari, N. Gudur. Face recognition in multi-sensor images based on a nvel modular feature selection technique, *Journal of Information Fusion*, doi: 10.1016/j.inffus. 05. 002, 2009

- [GC96] P. Gu, K. Chan, Generative inspection process and probe path planning for coordinate measuring machines, *Journal of Manufacturing Systems*, 15 (4): 240-55, 1996
- [GCB07] N. Van Gestel, S. Cuypers, P. Bleys, et al. A performance evaluation test for laser line scanners on CMMs, *Proceedings of OPTIMESS2007 Workshop*, 2007
- [Gfo]<http://www.onto-med.de/ontologies/gfo/>
- [GFM08] <http://www.gfmesstechnik.com/>
- [GGR06] E. Grinspun, Y. Gingold, J. Reisman, et al. Computing discrete shape operators general meshes, *Symposium of Eurographics on Computer Graphics Forum*, 25 (3): 547-56, 2006
- [GPL04] J. Geisler, E. Peinsipp-Byma, M. Litfab, et al. Smart recognition assistance for multi-sensor-image-based reconnaissance, *Proceedings of the 6th Joint International Military Sensing Symposium MSS*. 2004
- [Gru93] T. Gruber. A translation approach to portable ontologies, *Journal of Knowledge Acquisition*, 5 (2): 199-220, 1993
- [Gru95] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing, *Journal of Human-Computer Studies*, 43 (4-5): 907-928, 1995
- [GWH01] M. Garland, A. Willmott, P.S. Heckbert. Hierarchical face clustering on polygonal surface, *Proceedings of ACM Sym. Interactive 3D Graphics*, 49-58, 2001
- [Gol85] D.B. Goldgof. Motion estimation from three-dimensional data, *Master thesis*, University of Illinois at Urbana Champaign, 1985
- [GRB94] G. Godin, M. Rioux, R. Baribeau. Three-dimensional registration using range and intensity information, *Proceedings of SPIE*, 2350: 279-88, 1994
- [GY03] M. Greenspan, M. Yurick. Approximate k-d tree search for efficient ICP, *Proceedings of IEEE 4th International Conference on 3-D Digital Imaging and Modeling*, 442-8, 2003
- [HCS07] L. He, Y. Chao, K. Suzuki, et al. Fast connected-component labeling, *Journal of Pattern Recognition*, 42: 1977-87, 2007

- [HD06] O. Hjelle, M. Daehlen. Triangulations and applications, Mathematics and Visualization, Springer-Verlag, Berlin, Germany, 2006
- [HDD92] H. Hoppe, T. DeRose, T. Duchamp, et al. Surface reconstruction from unorganized points, Journal of Computer Graphics, 71-8, 1992
- [HL03] M. Heizmann, F. P. Leon. Imaging and analysis of forensic striation marks, Journal of Optical Engineering, 42 (12): 3423-32
- [HKR04] M. Horridge, H. Knublauch, A. Rector, et al. A practical guide to building OWL ontologies using the Protege-OWL plugin and CO-ODE tools Edition 1.0, <http://protege.stanford.edu/> 2004
- [HM01] J. Huang, C. Menq. Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinates points, IEEE Transaction on Robotics and Automation. 17(3):268–79, 2001
- [Hor05] J. Horst. NIST support for dimensional metrology interface standards, Symposium of IA – CMM International Conference, 2005
- [Hor87] B. Horn. Closed-form solution of absolute orientation using unit quaternion, Journal of the Optical Society of America, 4: 629-42, 1987
- [HP04] K. Hildebrandt, K. Polthier. Anisotropic filtering of non-linear surface features, Symposium of Eurographics on Computer Graphics Forum, 23 (3): 391-400, 2004
- [HPM07] R. Hunter, J. Perez, J. Marquez, et al. Modeling the integration between technological product specifications and inspection process, Journal of Materials Processing Technology, 191: 34-8, 2007
- [HQ07] Y. Huang, X. Qian. A dynamic sensing-and-modeling approach to three-dimensional point- and area-sensor integration, Journal of Manufacturing Science and Engineering, 129: 623-35
- [HQC09] Y. Huang, X. Qian, S. Chen. Multi-sensor calibration through iterative registration and fusion, Journal of Computer-Aided Design, 41 (4): 240-55, 2009
- [IDE93] IDEF0, Draft Federal Information Processing Standards Publication 183, announcing the Standard for Integration Definition for Function Modeling (IDEF0), <http://www.idef.com/pdf/idef0.pdf> 1993

- [IDEF5] <http://www.idef.com/IDEF5.html>
- [Ins] <http://www.inspexbuildinginspections.com/>
- [JM02] T. Jost, H. Hugli. A multi-resolution scheme ICP algorithm for fast shape registration, Proceedings of the first International Symposium on 3DPVT'02, 540-3, 2002
- [JM07] A. Jagannathan, E. Miller, Three-dimensional surface mesh segmentation using curvedness based region growing approach. IEEE Transactions on pattern analysis and machine intelligence. 29(12): 2195-204, 2007
- [JOM06] J. Jamshidi, G. Owen, A. Mileham. A new data fusion method for scanned models, Transactions of ASME, 6: 340-8, 2006
- [KD03] J.J. Koenderink, R.J. van Doorn. Local structure of Gaussian texture, IEICE Transactions on Information and Systems, E86-D (7): 1165-71, 2003
- [KD92] J.J. Koenderink, R.J. van Doorn. Surface shape and curvature scales, Journal of Imaging and Vision Computing, 10 (8): 557-65, 1992
- [Ket99] L. Kettner. Using generic programming for designing a data structure for polyhedral surfaces, Journal of Computational Geometry: Theory and Applications, 13: 65-90, 1999
- [KH09] T. Kramer, J. Horst. User manual for version 2.1.5 of the NIST DMIS test suite, http://www.isd.mel.nist.gov/projects/metrology_interoperability/ 2009
- [KHM01] T. Kramer, H. Huang, E. Messina, et al. A feature-based inspection and machining system, Journal of Computer-Aided Design, 33 (9): 653-69, 2001
- [KKW04] S. Kim, K. Kim, W. Woo. Projection -based registration using color and texture information for virtual environment generation. Springer, PCM, LNCS3331, 434-43, 2004
- [KMY00] N. Khalili, F. Mokhtarian, P. Yuen. Recovery of curvature and torsion features from free-form 3-D meshes at multiple scales, Proceedings of the Asian Conference on Computer Vision, 1070-5, 2000
- [Kre] <http://www.kreon3d.com/>
- [KSL09] T. Kim, Y. Seo, S. Lee, et al. Simultaneous registration of multiple views with markers. Journal of Computer-Aided Design, 41: 231-9, 2009

- [KT03] S. Katz, A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts, *Transactions of ACM on Graphics*, 22(3): 954-61, 2003
- [KZD05] V. Kaftandjian, Y. Zhu, O. Dupuis, et al. The combined use of the evidence theory and fuzzy logic for improving multi-modal nondestructive testing systems, *Transactions IEEE on Instrumentation and Measurement*, 54 (5): 1968-77, 2005
- [LA08] J.M. Lien, N. M. Amato. Approximate convex decomposition of polyhedral and its applications. *Journal of Computer Aided Geometric Design*, 25: 503-522, 2008
- [LBM04] C. Lartigue, P. Bourdet, L. Mathieu, et al. Algorithms for the calibration of laser-plane sensors on CMMs, *Proceedings of Advanced Mathematical and Computational Tools in Metrology VI*, 82-97, 2004
- [LBS07] T. Langer, A. Belyaev, H. Seidel. Extract and interpolatory quadrature for curvature tensor estimation, *Journal of Computer Aided Geometric Design*, 24 (9): 443-63, 2007
- [LC01] Z. Lin, J. Chow. Integration planning model of IDEF0 and STEO production data representation methods in a CMM measuring system, *Journal of Advanced manufacturing technology*, 17: 39-53, 2001
- [LCB02] C. Lartigue, A. Contri, P. Bourdet. Digitised point quality in relation with point exploitation, *Journal of Measurement*, 32: 193-203, 2002
- [LCE04] E. Lomonosov, D. Chetverikov, A. Ekart. Fully automatic, robust and precise alignment of measured 3D surfaces for arbitrary orientations, *Proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition*, 39-46, 2004
- [LDB05] G. Lavoue, F. Dupont, A. Baskurt. A new CAD mesh segmentation method, based on curvature tensor analysis, *Journal of Computer-Aided Design*, 37: 975-87, 2005
- [LDW09] F. Lv, H. Du, Z. Wang, et al. The study of the fusion estimation and the fault diagnosis method based on multi-speed sensor, *Symposium of World Congress on Computer Science and Information Engineering*, 7: 176-80, 2009
- [LG04] Y. Li, P. Gu. Free-form surface inspection techniques state of the art review,

- Journal of Computer-Aided Design, 36: 1395-417, 2004
- [LG09] M. Lillholm, L.D. Griffin. Statistics and category systems for the shape index descriptor of local 2nd order natural image structure. *Journal of Image and Vision Computing*, 27: 771-781, 2009
- [LHM09] Y. Lai, S. Hu, R.R. Martin, et al. Rapid and effective segmentation of 3D models using random walks. *Journal of Computer Aided Geometric Design*, 26: 665-79, 2009
- [LL00] Z. Lin, C. Lin, IDEF0 model of the measurement planning for a workpiece machined by a machining centre, *Journal of Advanced manufacturing Technology*, 16: 656-67, 2000
- [LL03] Y. Li, Z. Liu. Method for determining the probing points for efficient measurement and reconstruction of freeform surfaces, *Journal of Measurement Science and Technology*, 14: 1280-8, 2003
- [LLS06] H. Lee, S. Lee, Y. Seo, et al. Registering 3D scanned point cloud using markers. *Proceedings of SICE-ICCAS'06*. 4634-8, 2006
- [LLZ07] X. Liu, A. Li, X. Zhao, et al. Model-based optical metrology and visualization of 3D complex objects, *Journal of Optoelectronics Letters*, 3 (2): 115-8, 2007
- [LMM01] Y. Lin, R. Mahabaleshwarkar, E. Massina. CAD-based CMM dimensional inspection path planning – a generic algorithm, *Journal of Robotica*, 19 (2): 137-48, 2001
- [LNL08] B. Liu, S. Ning, J. Lin, et al. Transferring landmarks to individual foot using deformable template model, *Proceedings of IEEE. 9th International Conference for Young Computer Scientist*, 677-82, 2008
- [LPZ06] S. Liu, K. Peng, X. Zhang, et al. The study of dual camera 3D coordinate vision measurement system using a special probe, *Proceedings of SPIE 6357 II: 63574H-7*, 2006
- [LR09] Y. Liu, K. Ramani. Robust principal axes determination for point-based shapes using least- median of squares, *Journal of Computer-Aided Design*, 41: 293-305, 2009
- [LX08] Y. Liu, Y. Xiong. Automatic segmentation of unorganized noisy point clouds

- based on the Gaussian map, *Journal of Computer-Aided Design*, 40: 576-94, 2008
- [MB05] J. Min, K.W. Bowyer. Improved range image segmentation by analyzing surface fitting patterns, *Journal of Computer Vision and Image Understanding*, 97: 242-58, 2005
- [McG00] M. McGuire. The half-edge data structure, website address: http://www.flipcode.com/archives/The_Half-Edge_Data_Structure.shtml/, 2000
- [MD02] L. Morency, T. Darrell. Stereo tracking using ICP algorithm and normal flow constraint, *Proceedings of 16th International Conference on Pattern Recognition*, 4: 367-72, 2002
- [MDS02] M. Meyer, M. Desbrun, P. Schroder, A.H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds, *Proceedings of Visualization and Mathematics*, 35-57, 2002
- [Met08] <http://us.metris.com/>
- [MFF07] H. Munim, A. Farag, A. Farman. A new variational approach for 3D shape registration, *Proceedings of the 4th IEEE. International Symposium on Biomedical Imaging, from Nano to Macro*, 1324-7, 2007
- [MGC05] F. Martins, J. Garcia-Bermejo, E. Casanova, et al. Automated 3D surface scanning based on CAD model, *Journal of Mechatronics*, 15: 837-857, 2005
- [MGP04] N. Mitra, N. Gelfand, H. Pottmann, et al. Registration of point cloud data from a geometric optimization perspective, *Proceedings of Eurographics Symposium on Geometry Processing*, 22-31, 2004
- [Mil00] L. Miller. Ontologies and Metadata: A draft discussion of issues raised by the Semantic Web Technologies Workshop, <http://www.ilrt.bristol.ac.uk/discovery/2000/11/lux/index.html> 2000
- [MKY01] F. Mokhtarian, N. Khalili, P. Yuen. Multi-scale free-form 3D object recognition using 3D models, *Journal of Image and Vision Computing*, 19 (5): 271-81, 2001
- [ML08] C. Mehdi-Souzani, C. Lartigue. Contactless laser plane sensor assessment: toward a quality measurement, *Proceedings of IDMME-Virtual Concept 2008*, CD Rom.

- [Mor08] J.M. Morvan. Generalized curvatures, Geometry and Computing, Springer Press, 2008
- [MW00] D. Meek, D. Walton. On surface normal and Gaussian curvature estimations given data sampled from a smooth surface, Journal of Computer-Aided Geometry Design, 17: 521-43, 2000
- [MTL06] C. Mehdi-Souzani, F. Thiebaut, C. Lartigue. Scan planning strategy for a general digitized surface, Journal of Computing and Information Science in Engineering, 6: 331-9, 2006
- [MW99] A. Mangan, R. Whitaker. Partitioning 3D surface meshes using watershed segmentation. IEEE Visual Computer Graphics, 1999, 5(4): 308–21
- [MWW09] M. McCabe, E. Wood, R. Wojcik, et al. Hydrological consistency using multi-sensor remote sensing data for water and energy cycle studies, Journal of Remote Sensing of Environment, 112 (2): 430-44
- [Myc08] <http://www.mycrona.com/>
- [MZL03] X. Ma, G. Zhang, S. Liu, et al. Measuring information integration model for CAD/CMM, Chinese Journal of Mechanical Engineering (English version), 16 (1): 59-61, 2003
- [Nist] <http://www.nist.gov/mel/isd/cs/idme.cfm>
- [NLH07] A. Nuchter, K. Lingemann, J. Hertzberg. Cached k-d tree search for ICP algorithm, Proceedings of 6th International Conference on 3-D Digital Imaging and Modeling, 419-26, 2007
- [NYH97] M. Nashman, B. Yoshimi, T. Hong, et al. Unique sensor fusion system for coordinate measuring machine tasks, Proceedings of SPIE on Sensor Fusion and Decentralized Control in Autonomous Robotic Systems, 3209: 145-56, 1997
- [OYF05] D. OuYang, H. Feng. On the normal vector estimation for point cloud data from smooth surface, Journal of Computer-Aided Design, 37: 1071-79, 2005
- [One97] B. O’neill. Elementary Differential Geometry, Academic Press, 1997
- [OWL] W3C recommendations, <http://www.w3.org/TR/owl-features/> 2004
- [Pcd] <http://www.pcdmis.com/>

- [Pet02] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes, *ACM Computing Surveys*, 34 (2): 211- 62, 2002
- [PD05] L. Patil, D. Dutta. Ontology formalization of product semantics for Product Lifecycle Management, *Proceedings of IDETC/CIE*, 24-8, 2005
- [PHY06] H. Pottmann, Q. Huang, Y. Yang, et al. Geometry and convergence analysis of algorithms for registration of 3-D shapes, *International Journal of Computer Vision*, 67(3): 277-96, 2006
- [Pie03] G. Piella. A general framework for multiresolution image fusion: from pixels to regions, *Journal of Information Fusion*, 4 (4): 259-80, 2003
- [PK06] F. Puente Leon, S. Kammel. Inspection of specular and painted surfaces with centralized fusion techniques, *Journal of Measurement*, 39 (6): 536-46, 2006
- [PS03] S. Park, M. Subbarao. An accurate and fast point-to-plane registration technique, *Journal of Pattern Recognition Letters*, 24: 2967-76, 2003
- [Pul99] K. Pulli. Multiview registration for large data sets, *Proceedings of IEEE 2ed International Conference on 3-D Digital Imaging and Modeling*, 160-8, 1999
- [Pun99] O. Punska. Bayesian approaches to multi-sensor data fusion, *Master Dissertation of Cambridge University*, 1999
- [PZT02] X. Peng, Z. Zhang, H. Tiziani. 3-D imaging and modeling- Part I: acquisition and registration, *Journal of Optik*, 113 (10): 448-52, 2002
- [Qdas] <http://q-das.com/>
- [RB03] A. Razdan, M. Bae. A hybrid approach for feature segmentation of triangle meshes. *Journal of Computer-Aided Design*, 35, 783-9, 2003
- [Ren] <http://www.renishaw.com/>
- [RL01] S. Rusinkiewicz, M. Levoy. Efficient variants of the ICP algorithm, *Proceedings of 3rd International Conference on 3D Digital Imaging and Modeling*, 145-52, 2001
- [Ros86] R. Rost. OFF – A 3D object file format. *Workstation Systems Engineering*, <http://www.martinreddy.net/gfx/3d/OFF.spec>, 1986 (updated 1989)
- [Rou84]. P. Rousseeuw. Least median of squares regression, *Journal of American Statistical Association*, 79 (388): 871-880, 1984

- [RRT00] C. Reich, R. Ritter, J. Thesing. 3-D shape measurement of complex objects by combining photogrammetry and fringe projection, *Journal of Optical Engineering*, 39 (01): 224-31, 2000
- [RSV06] T. Rabbani, F.A. van den Heuvel, G. Vosselman. Segmentation of point clouds using smoothness constrains, *Proceedings of IAPRS*. 36(5): 248-53, 2006
- [RT07] D. Reniers, A. Telea. Skeleton-based hierarchical shape segmentation, In *Proceedings of IEEE International Conference on Shape Modeling and Application (SMI'07)*, 179-88, 2007
- [RW92] S. Rock, M. Wozny. Generating topological information from a “bucket of facets”, *Proceedings of Solid freeform fabrication symposium*, 251-9, 1992
- [SB95] N. Sapidis, P. Besl. Direct construction of polynomial surfaces from dense range images through region growing, *Transactions of ACM on Graphics*, 14 (2): 171-200, 1995
- [SCK06] C. She, C. Chang, Y. Kao, et al. A study on the computer-aided measuring integration system for the sheet metal stamping die, *Journal of Materials Processing Technology*, 177: 138-41, 2006
- [SDS07] E. Savio, L. De Chiffre, R. Schmitt. Metrology of freeform shaped parts, *Journal of CIRP Annals – Manufacturing Technology*, 56 (2): 810-35, 2007
- [Sha08] A. Shamir. A survey on mesh segmentation techniques, *Transactions of Computer Graphics Forum*, 27(6): 1539-56, 2008
- [She01] A. Sheffer. Model simplification for meshing using face clustering, *Journal of Computer-Aided Design*, 33:925-34, 2001
- [SHM00] T. Shen, J. Huang, C. Menq. Multiple-sensor integration for rapid and high-precision coordinate metrology, *Transactions of IEEE/ASME on Mechatronics*, 5 (2): 110-21, 2000
- [SHM01] T. Shen, J. Huang, C. Menq. Multiple-sensor planning and information integration for automatic coordinate metrology, *Journal of Computing and Information Science in Engineering*, 1: 167-79, 2001
- [SJH98] C. Schutz, T. Jost, H. Hugli. Multi-feature matching algorithm for free-form 3D surface registration, *Proceedings of 14th ICPR*, 2: 982-4, 1998
- [SKL03] S. Son, S. Kim, K. Lee. Path planning of multi-patched freeform surfaces for laser scanning, *Journal of Advanced Manufacturing Technology*, 22: 424-35,

2003

- [SKT05] D. Sokolov, D. Kasantzev, J. Tyrrell, et al.. Combined confocal and scanning probe sensors for nano coordinate metrology, Nano-scale calibration standards and methods, Weinheim: Wiley-VCH: 131-43, 2005
- [SLW02] G. Sharp, S. Lee, D. Wehe. ICP algorithm using invariant features, Transactions of IEEE on Pattern Analysis and Machine Intelligence, 90-102, 2002
- [SLY04] K. Shi, P. Li, S. Yin, et al. Chromatic confocal microscopy using supercontinuum light, Journal of Optical Express, 12 (10): 2096-101, 2004
- [SNK02] H. Schwenke, U. Neuschaefer-Rube, H. Kunzmann, et al. Optical methods for dimensional metrology in production engineering, Journal of CIRP Annuals, 51 (2): 685-99, 2002
- [SP08] V. Sunil, S. Pande. Automatic recognition of features from freeform surface CAD models, Journal of Computer-Aided Design, 40: 502-17, 2008
- [SPC04] H. Shin, J. Park, B. Choi, et al. Efficient topology construction from triangle soup, IEEE Proceedings of the Geometry Modeling and Processing, GMP'04, 359-65
- [SPM03] R. Schmitt, T. Pfeifer, C. Mersmann, et al. A method for the automated positioning and alignment of fibre-reinforced plastic structures based on machine vision, Journal of CIRP Annuals-Manufacturing Technology, 57: 501-4, 2003
- [SPP02] Y. Sun, D.L. Page, J. Paik, et al. Triangle mesh-based edge detection and its applications for surface segmentation and adaptive surface smoothing, Proceedings of Int. Conf. on Image Processing, 3: 825-8, 2002
- [Sri08] T. Srinark. Lecture note on differential geometry, 2008
- [Sri08-1] V. Srinivasan. Standardizing the specification, verification, and exchange of product geometry: research, status and trends, Journal of Computer-Aided Design, 40 (7): 738-49, 2008
- [SRM06] S. Soll, B. Roither, H. Moritz, et al. Three-dimensional surface test with Shape from Shading, Photonik international, (originally published in German), 4: 74-6, 2006

- [SSC04] A. Shamir, L. Shapira, D. Cohen-Or, et al. Geodisc mean shift, Proceedings of Korea-Israel Conference on Geometric Modeling and Computer Graphics. 51-6, 2004
- [ST09] P. Stelldinger, L. Tcherniavski. Provable correct reconstruction of surfaces from sparse noisy samples, Journal of Pattern Recognition, 42: 1650-59, 2009
- [Stal] <http://protege.stanford.edu/>
- [Stan] <http://graphics.stanford.edu/data/3Dscanrep/>
- [Step] <http://www.steptools.com/library/standard/>
- [Sti] <http://www.stilsa.com/>
- [Stla09] <http://people.sc.fsu.edu/~burkardt/data/stla/stla.html/>
- [STK02] S. Shlafman, A. Tal, S. Katz. Metamorphosis of polyhedral surfaces using decomposition, Proceedings of Computer Graphics Forum, 21 (3):219–229, 2002
- [Sumo] <http://www.ontologyportal.org/>
- [SW07] P. Singh, Y. Wu. Multimodel industrial inspection and analysis, Journal of Computing and Information Science in Engineering, 7: 102-7, 2007
- [Tau95] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 902-7
- [TDK04] J. Tyrrell, C. Dal Sovoi, R. Kruger-Sehm, et al. Development of a combined interference microscope objective and scanning probe microscope, Journal of Review of Scientific Instruments, 75 (4): 1120-6, 2004
- [TGC04] J. Tang, J. Gu, Z. Cai. Data fusion with different accuracy, Proceedings of IEEE on Robotics and Biomimetics, 811-5, 2004
- [Uml] <http://www.uml.org/>
- [VHB08] M. Vanco, B. Hamann, G. Nrunnett. Surface reconstruction from unorganized point data with quadrics, Symposium of Computer Graphics Forum, 27 (6): 1593-606, 2008
- [VMC97] T. Varady, R.R. Martin, J. Coxt. Reverse engineering of geometric models- An introduction, Journal of Computer-Aided Design, 29 (4): 255-68, 1997

- [VSH04] J. Vergeest, Y. Song, D. Hartge. Freeform object positioning by 3D shape matching without artificial feature points, *Journal of WSCG'2004*, 12 (3): 483-90, 2004
- [Wab99]http://rpdrc.ic.polyu.edu.hk/old_files/stl_introduction.htm/
- [WBH07] M. Wardetzky, M. Bergou, D. Harmon, et al. Discrete quadratic curvature energies. *Journal of Computer Aided Geometric Design*, 24: 499-518, 2007
- [WEP04] A. Wechenman, T. Estler, G. Peggs, et al. Probing system in dimensional metrology, *Journal of CIRP Annuals-Manufacturing Technology*, 53 (2): 657-84, 2004
- [Wil01] J. Williams. A system for multiple view 3D acquisition and registration incorporating statistical error models, Ph.D thesis, Queensland University of Technology, Australia, 2001
- [WJS09] A. Weckenman, X. Jiang, K.-D. Sommer, et al. Multisensor data fusion in dimensional metrology, *Journal of Annals of the CIRP*, manuscript, 58 (2), 2009
- [WKW02] H. Woo, E. Kang, S. Wang, et al. A new segmentation method for point cloud data, *International Journal of Machine Tools and Manufacturing*, 42: 176-78, 2002
- [WM99] A. Mangan, R. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *Transactions of IEEE Visual Computer Graphics*, 5(4): 308–21, 1999
- [WN03] A. Weckenmann, K. Nalbantlic. Precision measurement of cutting tools with two matched optical 3D-sensors, *Journal of CIRP Annuals-Manufacturing Technology*, 52 (1): 443-6, 2003
- [WRS00] K. Wolf, D. Roller, D. Schafer. An approach to computer-aided quality control based on 3D coordinates metrology, *Journal of Material Processing Technology*, 107: 96-110, 2000
- [WWH08] A. Weckenmann, J. Weickmann, W. Hartmann. Model and simulation of fringe projection measurements as part of an assistance system for multi-component fringe projection sensors, *Proceedings of SPIE*, 7102:

- 71020N-0N12, 2008
- [XMQ04] H. Xie, K.T. McDonnell, H. Qin. Surface reconstruction of noisy and defective data sets, Proceedings of IEEE Conference on Visualization'04, 259-66, 2004
- [XWZ05] Z. Xie, J. Wang, Q. Zhang. Complete 3D measurement in reverse engineering using a multi-probe system, Journal of Machine Tools & Manufacture, 45: 1474-86, 2005
- [XZJ06] Z. Xie, Z. Zhang, M. Jin. Development of a multi-view laser scanning sensor for reverse engineering, Journal of Measurement Science and Technology, 17: 2319-27, 2006
- [YL99] M. Yang, E. Lee. Segmentation of measured point data using a parametric quadric surface approximation, Journal of Computer-Aided Design, 31: 449-457, 1999
- [YLL05] H. Yamachi, S. Lee, Y. Lee, et al. Feature sensitive mesh segmentation with mean shift, Proceedings of International Conference on Shape Modeling (SMI'05), 236-43, 2005
- [ZAB09] H. Zhao, N. Anwer, P. Bourdet. A PLM-based multiple sensor integration measurement system for geometry processing, Proceedings of CAT, CD-Rom, 2009
- [ZAW00] S. Zhang, A. Ajmal, J. Wootton, et al. A feature-based inspection process planning system for coordinate measuring machine (CMM), Journal of Materials Processing Technology, 107: 111-8, 2000
- [ZGX05] X. Zhuang, L. Gu, J. Xu. Medical image alignment by normal vector information, Proceedings of International Conference on Computational Intelligence and Security, 3801: 890-5, 2005
- [Zor03] D. Zorin. Curvature-based energy for simulation and variational modeling, Proceedings of Int. Conf. on Shape Modeling and Applications (SMI'05), 198-206, 2005
- [ZPK02] Y. Zhang, J. Paik, A. Koschan, et al. A simple and efficient algorithm for part decomposition of 3D triangulated models based on curvature analysis. Proceedings of Int. Conf. on Image Processing, 3: 273-6

- [ZPM04] Z. Zhang, C. Pan, S. Ma. An automatic method of coarse registration between multi-source satellite images, Proceedings of IEEE Conference on Intelligent Sensors, Sensors Network and Information, 205-9, 2004
- [ZSN03] T. Zinsser, J. Schmidt, H. Niemann. A refined ICP algorithm for robust 3-D correspondence estimation, Proceedings of IEEE on ICIP, 2: 695-8, 2003
- [ZSR08] H. Zheng, D. Saupe, M. Roth, et al. Efficient 3D shape acquisition and registration using hybrid scanning data, Proceedings of 3DPVT'08, 251-8, 2008
- [ZWM01] X. Zhao, R. Wilhelm, O. Martin, et al. Software frameworks for integrated measurement processes, Proceedings of ASPE, 2001
- [ZWW06] H. Zhao, J. Wang, B. Wang, et al. A PLM-based automated inspection planning system for Coordinate Measuring Machine. Proceedings of SPIE, 6358: 6358291-6

Abstract:

Multisensor integration systems are used to fulfill the increasing requirements on the complexity and accuracy of coordinate metrology. This thesis investigates on the achievement of multisensor integration in coordinate metrology from two main parts: function and data modeling; and discrete geometry processing. Function and data modeling allow obtaining clear understanding of the system and guiding the system development. Hierarchical function models of the system are built using IDEF0 methodology. An ontology based method is employed to model the various data and their relationships in the system with structured representation. Efficient methods for discrete geometry processing to support multisensor integration are the main focus of this dissertation. Reliable estimation of the curvatures is essential for discrete shapes processing. A tensor based method is proposed for discrete curvature estimation. The proposed method provides better performance according to a comparative analysis. Registration is one of the most important and decisive steps of multisensor integration. Registration algorithms enable to align the different point data captured in the respective sensor's coordinate systems. A registration method is proposed to align different discrete shapes with unknown correspondences. A combination of curvature information and Euclidean distance is used to improve the registration performance. A new method is developed for shape recognition and segmentation based on shape index and curvedness. The local surface type of a vertex is recognized according to its shape index and curvedness. The vertices are then clustered according to their local surfaces. A connected region generation algorithm is then implemented to generate the segmentation results. Considerable cases are tested and the method provides satisfying results. A new software framework DSP-COMS is developed as the testing platform of the developed methods. A detailed case study of an industrial workpiece digitized by laser scanning and touch probing is presented. The measured point data are then processed using DSP-COMS. The results show that the proposed methods can provide satisfying results for the studied case.

Résumé:

L'intégration de systèmes multi-capteurs est la réponse la plus adaptée aux besoins croissants en matière de complexité et d'exactitude de mesures en métrologie par coordonnées. Cette thèse a pour objectif l'intégration de systèmes multi-capteurs en métrologie par coordonnées selon deux aspects: la modélisation des fonctions et des données; et le traitement de la géométrie discrète. La modélisation des fonctions et des données permet une meilleure compréhension des exigences des fonctions du système et des spécifications des données pour l'intégration de systèmes multi-capteurs. Des modèles hiérarchiques fonctionnels du système sont construits en utilisant la méthodologie IDEF0. Une démarche à base d'ontologies est utilisée pour modéliser les différentes données et de leurs relations par leur représentation structurée. Des méthodes efficaces et robustes pour le traitement de la géométrie discrète dans le cadre de l'intégration multi-capteurs sont l'objet principal de cette thèse à travers le développement de méthodes robustes. Une méthode basée sur le tenseur de courbure est utilisée pour l'estimation des courbures discrètes. Une analyse comparative montre que la méthode proposée fournit de meilleures performances par rapport à d'autres méthodes existantes. Le recalage est l'une des étapes les plus importantes de l'intégration multi-capteurs. Les algorithmes de recalage permettent de recalibrer les ensembles de points mesurés issus de différents capteurs ou de plusieurs vues. Une méthode de recalage sans correspondance connue est proposée dans cette thèse. Une combinaison de la courbure et de la distance euclidienne est utilisée pour améliorer les performances de l'algorithme classique de recalage ICP. Une nouvelle méthode basée sur l'indice de forme (shape index) et l'intensité de courbure (curvedness) est développée pour la reconnaissance et la segmentation des formes discrètes. Le type de surface locale d'un sommet est reconnu par l'intermédiaire de son indice de forme. Les sommets sont ensuite groupés en fonction de leurs surfaces locales. Un algorithme de connexité par région est mis en œuvre pour produire les résultats de segmentation. Plusieurs exemples ont été testés et la méthode proposée fournit des résultats satisfaisants. Ces différentes méthodes de traitement géométriques ont été implémentés dans un nouveau système informatisé appelé DSP-COMS. Une étude de cas détaillée d'une pièce industrielle numérisée par laser scanning et par palpation est présentée dans cette thèse. Les points mesurés sont ensuite traités dans DSP-COMS. Les résultats obtenus montrent bien la robustesse des méthodes développés dans cette thèse.