Discrete shape modeling for geometrical product specification : contributions and applications to skin model simulation

Min Zhang

N° ENSC-2011/2012

"Version Rapporteurs"

THESE DE DOCTORAT

DE L'ECOLE NORMALE SUPERIEURE DE CACHAN

Présentée par

ZHANG Min

pour obtenir le grade de

DOCTEUR DE L'ECOLE NORMALE SUPERIEURE DE CACHAN

Domaine:
MECANIQUE - GENIE MECANIQUE - GENIE CIVIL

Sujet de la thèse:

Discrete Shape Modeling for Geometrical Product Specifications: Contributions and Applications to Skin Model Simulation

M.		Président
M. S. Samper	Professeur - Université de Savoie	Rapporteur
M. A. Ballu	Professeur - Université de Bordeaux	Rapporteur
M. L. Mathieu	Professeur - Université Paris Sud 11	Directeur de thèse
M. N. Anwer	Maître de conférences- Université Paris Nord	Encadrant

Laboratoire Universitaire de Recherche en Production Automatisée
61, avenue du Président Wilson - 94 235 Cachan cedex

Abstract:

The complex shapes with tight tolerances which are increasingly adopted in industrial applications make it become a great challenge for geometrical product specification to control the geometrical variations along the product lifecycle. Significant efforts are being devoted to develop systems that can support geometrical product specifications. Among them, GeoSpelling [Ballu 95] [Mathieu 03] is a classical one, which has already standardized as ISO 17450-2005.

This thesis investigates on the achievement of GeoSpelling system for geometrical product specification from two aspects: discrete geometry processing supported for GeoSpelling and skin model simulation.

GeoSpelling is proposed as a coherent and complete model to manage the geometrical variations along the product lifecycle. GeoSpelling is based on six geometrical operations to handle both ideal and non-ideal features. The discrete representations of features are first introduced and a new framework is developed to implement the operationalization of GeoSpelling based on discrete geometry processing techniques.

The Skin model, a non ideal feature, is a kernel concept of GeoSpelling. Robust discrete shape modeling methods for skin model simulation are the main concern of this work. The proposed simulation of the skin model is composed of nominal model, random deviations and systematic deviations. Different methods are developed to add random and systematic deviations to the nominal model.

Random errors are statistic fluctuations in either direction and typically obeyed to Gaussian distribution. Three methods are developed to simulate the skin model considering the random deviations: 1-D Gaussian based method, multi-Gaussian based method and Gibbs based method. The specified associated tolerances are used as constrains to locate the simulated skin model within the specified tolerance zone. A comparative analysis of the developed three methods is discussed

Comparing to random errors, systematic errors are reproducible inaccuracies that are consistently in the same directions. They usually keep similar values or follow some specific functions which can be calculated or simulated. A group of second order shapes are used as constraints for systematic deviations' generation. The complete skin model can be generated by adding random and systematic deviations. A method based on statistic shape models is proposed for skin model analysis and trend prediction.

The above methods have been implemented in a new computerized system. An application with a detailed case study is discussed. The simulated skin model by the proposed methods is compared to the measured data and the simulated results by another FEA-based method. The comparative results testify that our methods can provide a satisfying performance.

Contents

Figures

Tables

Introduction

With the economy globalization, complex shapes are increasingly adopted to improve company's competition. Product is desired to be machined as accurate as possible to match its functional requirements. However, the more accurate it is manufactured, the more cost it will take both time and money. Therefore, geometric tolerances are designed to make a compromise between the functional requirements and production cost during the design phase. Geometrical variations are inevitably generated during the manufacturing stage because of the accuracy of applied machining methods, machining tools, etc. variations also generate during the measurement stage considering the measurement uncertainty always exists. How to design the geometric tolerance to control the geometrical variations is a critical issue in geometrical product specification (GPS). Some Computer Aided Tolerancing platforms have already developed to assist geometrical tolerancing process [Salmon 96] [Ballu 03]. However, the techniques for computational-supported geometrical product specification haven't solved.

It is necessary to manage geometric variations and to share a coherent and complete tolerancing process along the product live cycle [Mathieu 05]. According to Mathieu and Ballu [Mathieu 05], Geometrical tolerancing process can be synthesized into four main tasks: tolerance specification, tolerance analysis, and tolerance synthesis and tolerance verification. They proposed a proper solution called Geospelling to cover these four stages for coherent and complete tolerancing process, which has already standardized as ISO17450. GeoSpelling is based on geometrical operations and is able to describe both ideal and non-ideal geometric features [Mathieu 05]. However, till now, GeoSpelling is only a "paper-based" concept and realized with computer. There is no GeoSpelling-based CAT (Computer-Aided Tolerancing) system developed to optimize tolerancing process. It is thus very useful to research and find proper solutions to develop GeoSpelling-based system with computer and the advantages of GeoSpelling can then be completely fulfilled.

In this dissertation, we investigate the solutions to develop the GeoSpelling system for Geometrical Product Specification (GPS). Solutions are proposed based on discrete shape modeling and statistic analysis for skin model simulation and applications.

GeoSpelling, as an ISO standardized representation of geometrical tolerances, proposes a coherent and complete representation scheme to manage the geometrical

variations in the context of product lifecycle. The critical characteristics of GeoSpelling are reviewed for comprehensive understanding. Consider shape with and/or without geometrical variations are the basis of GeoSpelling, the available shape modeling techniques are analyzed for ideal/non-ideal geometrical shape construction.

GeoSpelling is proposed to handle both ideal geometries (e.g. CAD model or geometries without geometrical variations) and non-ideal geometries (e.g. actual parts or geometries with geometrical variations) based on 6 basis operations: partition, extraction, filtration, association, collection and construction. The common CAD software packages are mature to handle ideal geometries, but they are lack to handle non-ideal geometries. However, the discrete shape modeling techniques which are based on discrete geometry provides solutions for non-ideal geometry processing. The mapping relationships between GeoSpelling and discrete geometry are established and analyzed.

Skin model is a kernel concept in GeoSpelling, which acts as an approximation of the actual shapes of real products. To simulate and shape the skin models is a fundamental task for GeoSpelling based tolerancing systems. An appropriate skin model is also the basis for the implementations of the 6 operations for geometry processing in GeoSpelling.

Geometrical variations may come from random errors (e.g. random errors in manufacturing and inspection) and from systematic errors (e.g. the imperfect performances of the applied instruments for manufacturing). In order to simulate and represent the actual shapes of real products, both of them should be considered for skin model simulation and shaping. It is necessary to develop methods to analyze the skin models, to predict the trend of real shapes, to guide the geometrical production specification in applications.

Contributions

The objective of the thesis is to enhance the digital processing capability of GeoSpelling model using discrete geometry technique. The main contributions of the thesis are:

- A new framework for geometrical operationalization of GeoSpelling based on association between discrete geometry and GeoSpelling.

- A comparative analysis of common-used discrete normal estimation methods.
- Three different methods to simulate the skin model by adding the random deviations in different ways.
- Set of methods to simulate the skin model considering systematic errors.
- A method to generate statistical shape model for skin model analysis and trend prediction.
- A developed software framework for discrete geometry processing especially for the skin model simulation.

Outline

The thesis is organized as follows:

Chapter 1 introduces the tolerance process especially focus on the tolerance representation techniques, and the shape modeling approaches are also presented.

Chapter 2 presents the association between GeoSpelling and discrete geometry. Both of the characteristics of GeoSpelling and discrete geometry are introduced. The method to associate them together is discussed for GeoSpelling operationalization based on discrete geometry techniques.

Chapter 3 presents the developed three methods of skin model simulation considering the random errors: 1-D Gaussian-based, multi-Gaussian-based, and Gibbs-based. The methods to generate the skin models considering specified tolerances are presented. A comparative analysis of the three methods is discussed.

Chapter 4 presents methods to simulate the skin models with systematic errors. A group of predefined parametric second order shapes are implemented to constraint the systematic deviations. A method to compute the statistical shape model of a set of training skin models is developed for skin model analysis and trend prediction.

Chapter 5 presents a detailed case study of an industrial workpiece which is a cross-shaped model. The developed software, which serves as the test platform for the algorithms developed in this thesis. The detailed processing results of the studied workpiece are presented. The comparative analysis between the simulation models and the measurement models is discussed.

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

Chapter 1

Geometrical Variations within the Whole Product Lifecycle

1.1. Introduction

In modern production engineering, complex shapes with tight tolerances are increasingly adopted to improve the products' competition. Geometrical variations are inevitably generated during the manufacturing stage due to the limited accuracy of applied machining methods, machining tools, etc. Variations also generate during the measurement stage considering the measurement uncertainty always exists. The complex shapes with high quality requirements make it as a great challenge for tolerancing process and geometrical variations' control.

Within the context of Product Lifecycle Management (PLM), the information communication and sharing requires to manage the geometrical variations along the whole product lifecycle. The geometrical variations should be considered at the beginning of tolerancing process in the design stage. Many Computer Aided Tolerancing tools [Mathieu 05] can help designer for functional tolerance specification, but limited to control the geometrical variations for the view of product lifecycle. Some authors [Ballu 05] [Salomons 96] proposed concepts to build coherent and complete tolerancing process. As a perfect example of these concepts, GeoSpelling proposed by Ballu and Mathieu [Ballu 95] has already standardized as ISO 17450. However, till now there is no reported system yet has capacity to manage geometrical variations along the product lifecycle.

This chapter surveys the popular techniques for tolerancing process, especially in tolerance presentation and specification. The GeoSpelling model is introduced to manage the geometrical variations. The shape modeling techniques supported for geometrical production specification are reviewed.

This chapter is organized as follows:

Section 1.2 presents the popular techniques for tolerancing process, especially techniques for tolerance representation. Section 1.3 surveys the common-used shaping modeling techniques in literature and industrial applications. Section 1.4 is the conclusion of this chapter.

1.2. Tolerance modeling

1.2.1 Tolerancing process

In automotive, aerospace and other industrial applications, the management of geometrical variation has become an important issue in product design, concurrent engineering and product lifecycle management [Anwer 03]. The computer aided tolerancing technique can assist the designers in functional tolerance specifications. In generally, product design, process plan, manufacturing and inspection activities are concerned in today's CAx context. In this point of view, some researchers [Salomons 96] [Ballu 05] proposed using a coherent and complete tolerance process to manage the geometrical variations in mechanical engineering.

Based on [Mathieu 05], the tolerancing process is defined through all the activities involved by geometric product variations management. In [Salomons 96] the tolerance process is divided into four classical activities: tolerance representation, tolerance specification, tolerance analysis, and tolerance synthesis. In [Mathieu 05], Mathieu and Ballu improve this classification into five activities while adding tolerance verification.

Tolerance representation links to the mechanisms and algorithms of expressing geometric variations in computerized systems. It is the basic foundation for a coherent and complete tolerance process activity. To be coherent and complete, the tolerance process has to employ a uniform model to express tolerancing for each participators involved in the process. Section 1.2.1 will focuses more on the tolerance representation technique.

Tolerance specification focus on determining the tolerance types and values needed on geometrical features to satisfy the functional requirements. Tolerance specification is used to convert the functional requirements into geometric requirements on parts and assemblies. It solves the problems such as: tolerance type, tolerance value, datum reference framework and tolerance principles, etc. In traditional tolerance practice, the tolerance specification is designed by designers according to the tolerancing standards and their experiences. However, the standards do not give the methods to specify the tolerance specifications. The main contribution on this topic is the work of A. Clement [Clement 91] [Clement 93], who proposed the

TTRS (Topologically and Technologically Related Surfaces) model using 13 signed constraints for the geometrical specification. A constraint is a mathematical relationship between variable elements. TTRS is at one disposal an integrated model which allows completion, consistence and explicit definition in the dimensioning, tolerancing and metrology fields [Clement 97].

Tolerance analysis is used to verify the functional requirements after tolerances have been specified on an assembly. Traditionally, the tolerance analysis is a manual work, which relies on dimension chain and tolerance chart. Bjork [Bjorke 89] proposed the automation of tolerance analysis of assemblies based on the development of the data structure for their representation in geometric modeling, and the computerization of procedures. Mort often, tolerance analysis is performed by verifying two aspects [Salonmons 96]: one is verifying assemblability of the assembly: the feasibility of assembly (fit); one is verifying if specified clearances between parts are still met: the quality of assembly (clearance). Many efforts have been attempted on this topic by the researchers. The analysis approaches can be roughly classified into worst-case and statistical method. Worst-case analysis determines the extreme displacement of each ending geometric entity resulting from the limits specified on the contributors while statistical analysis determines the full frequency distribution from frequency distributions of the contributors [Giordano 01] [Anselmetti 10].

Tolerance synthesis focuses on tolerance allocation and tolerance optimization functions in the point of manufacturing and inspection views. Traditional tolerance synthesis methods are implemented separately in the design and the process planning stages [Campatelli 11]. In 1970's, Michael [Michael 82] and Speckhart [Speckhart 72] proposed tolerance synthesis models allocate the tolerances in the design stage, avoiding consideration of the manufacturing processes, and these approaches are focused only on the evaluation of the tolerance chain. In 1990's, the evolution based on optimization of a cost function has been developed by many researchers [Nassef 93] [ElMaraghy 93] [Roy 97]. The design synthesis method is a mapping relationship from the functional requirements to artifacts, with multi-stage constrained optimization during stages of deign proposed by [Pramanik 03] [Dantan 05], etc.

Tolerance verification defines inspection planning and metrological procedures for the functional requirements, functional specifications and manufacturing specification. It is very important to consider tolerance verification early in the design

activities to be able to evaluate uncertainties [Mathieu 03]. Tolerance verification permits to close the process loop, to check the product conformity and to verify assumptions made by the designer as illustrated in figure 1-1 [Mathieu 05].

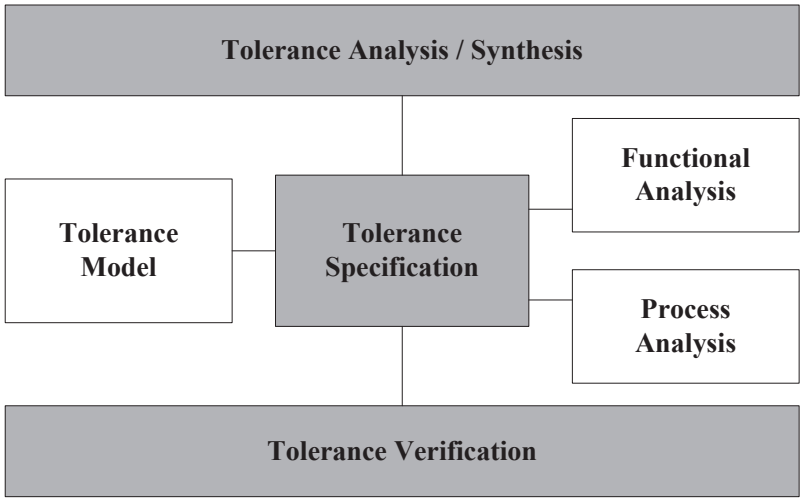


Figure 1-1: Tolerance process

1.2.2 Tolerance model

The tolerance model scheme in current CAD/CAM systems includes two research domains: tolerance modeling and tolerance representation. The first one is used to establish the mathematical expression of tolerance specification, and the latter one focus on how the express the tolerance specification in computerization context. This section provides a survey on tolerance model technique based on this classification.

(1). Tolerance modeling technique

The difficulty in translating the information language (customer level) into a formal language (industrial level) through a mathematic model of geometric deviations is well known. When the designer cannot interpret the shape variations model in mathematic expression, the annotation writing manner is adopted even it is difficult for the metrologist to measure. Some efforts on this topic are discussed as follow.

Hillyard and Braid [Hillyard 78] developed the concept of variational geometry that is a dimension-driven, constraint-based technique. This approach adopts constraint equation to reflect the tolerance information. In this approach, the solid

objects are deemed as the physical frames. At the initial state, the physical frame is linked incompactly by each part. The notes and the connecting bars of the physical frames are the corresponding points and edges of the solid objects. Accordingly, the dimension information of the objects is the fixed constraint of the physical frame, while the tolerance information is the minor changes of constraints. Therefore, the dimension tolerance can be reflected by the displacement of parameter vectors, which are derived from the physical frame model. For the orientation and position variation of the object, each point of the surface should be parameterized thereby infinite dimension vectors are need to express the variation and this is impossible for calculation. Thus, his approach can reflect the dimension tolerance well but it can not deal with the geometric tolerance information.

Another early example of previous work of tolerance modeling technique is the solids offset approach proposed by Requicha [Requicha 83], and its improvement in [Requicha 84] [Requicha 86]. In this method, the tolerance zones of the objects are obtained by "offsetting" the nominal boundaries. Conventional tolerances are considered as a special case of geometric tolerances, and no attempt is made to characterize the nominal geometry using dimensions. This approach considers both ideal and non-ideal features to respect the specification when its boundary is within tolerance zones. The solids offset method differs from the tolerancing standards, since the individual pairs of offset surface are combined to obtain a composite tolerance zone of the entire solid and the individual tolerances cease to be independent constraints [Salomons 96].

Based on the solids offset approach, Jayaraman [Jayaraman 89] and Srinivasan [Srinivasan 89] proposed Virtual Boundary Requirements (VBRs) and Conditional tolerances (CTs) approach. The purpose of their work is that the functional requirements can be captured in a specific form of tolerances designated as their models. The VBRs approach is proposed based on the study the problems form the perspective of two classes of functional requirements, which directly relatable to the geometry of mechanical parts [Jayaraman 89]. The first involves positioning parts with respect to one another in an assembly, and the second maintaining material bulk in critical portions of parts. The CTs approach on a set of features specifies geometric variations that are dependent on geometric parameters derived from fitting fo the actual set of features. The Maximum Material Condition (MMC) and Least Material Condition (LMC) tolerances used in practice are in fact CTs specifications. Typically,

VBRs when translated into allowable variations in geometric parameters of features give rise to CTs. There are some reasons for derive alternative specifications (as nearly equivalent to the VBRs as possible) based on the concept of CTs, such as: part fabrication process planning problem, part fabrication process planning problem, part fabrication process control problem, and statistical tolerancing problem, and the details of these reasons are indicated in [Srinivasan 89].

Considering both the solids offset approach [Requicha 83] and the virtual boundary requirements [Srinivasan 89] approach provide a strict tolerance constraint. Based on solids offset approach, Etesmi [Etesmi 93] formalize the solids offset model, and proposed using Tolerance Specification Language (TSL) to describe tolerance constraints. At the same time he discussed some practical implementation cases.

Hoffman [Hoffman 82] proposed using point vector to express the tolerance requirements in three-dimensional Euclidean space. The principle of this approach is that the geometrical images are deemed as a set of point vectors and thus the tolerance requirements are converted to the tolerance function, in which the parameters are the point vectors. The tolerance requirements can be explained by an inequation $L \le f(x) \le U$, where x is the parameter, f is the tolerance function, and L, U is the limitation of the tolerance zone.

Based on Hoffman approach, Turner [Turner 87] proposed a higher degree polynomial function to model form tolerance for the nominal planar part feature. In this approach, the independent geometric variables defined as the basic vectors of the vector space. This feasibility space approach is not suitable for three-dimension tolerance representation because of a too high complexity and doubtful used fullness in three-dimension space. Based on the higher degree polynomial approach, Turner and Wozny [Turner 87] proposed to model straightness tolerance by breaking up the boundary line with additional vertices. The number of these interior vertices determines the form variation frequencies of the boundary line. Later, Gupta and Turner [Turner 87] proposed to break a planar surface into triangular patches and fit Bezier triangles to each patch. In this approach, the boundary segment is represented by a curved surface (in two-dimension), which is more natural than the straight line (in one-dimension) used in previous approach.

Wirtz [Wirtz 93] proposed another approach based on vectorial theory. In this approach, the tolerance is represented as a limit constraint on the elements of a vector,

which has a correspondence between the toleranced feature and the reference feature. This approach also has the limitations that it is not close to the standards and is oriented too much towards older dimensioning and tolerancing practices, for example, it can not deal with the form tolerances specification problems.

Roy [Roy 98a] [Roy 99] presented a mathematical scheme for interpreting dimensional and geometric tolerances for polyhedral parts in a solid modeler. Using the interrelated surface-based variational modeling technique, the geometric tolerance specifications have been interpreted as sets of algebraic constraints that define tolerance zones for model variables. The main object of this approach is to define the resultant tolerance zones of a part (from its tolerance specifications) in terms of appropriate sets of algebraic relations. Each face of the part is described by these model variables and those variables are ultimately constrained by the algebraic relations. This approach considers both the issues of initial tolerance zone formation and final representation of the variant boundary surfaces. Based on this approach, Tech [Tech 01] studied the modeling method of tolerance zone of planar shapes, and applied it to the tolerance analysis process.

Wang and Pramanik [Wang 06] proposed a method that can convert tolerance specifications as per maximum material condition (MMC)/least material condition (LMC)/regardless of feature size (RFS) material conditions for standard mating features (such as: planar, cylindrical, spherical) into a set of inequalities in a deviation space. The constant boundary generated by the collective effects of a size feature's specified MMC or LMC material condition and the geometric tolerance for the material condition. The principle of this approach is that the variations of size or shape could be thought of as deviations of a set of generalized coordinates defined at some convenient point on a feature, and any tolerance specification for a feature imposed kinds of restrictions on its deviation parameters. This approach is also used for tolerance synthesis and analysis process.

Shah [Shah 98] proposed a dimension and geometric model, which is based on the relative degrees of freedom of geometric entities: feature axes, edges, faces and features-of-size. In this approach, the dimension graphs are created based on the degrees of freedom for each control direction, and the datum reference frames and the standard tolerance classes are incorporated into the graph. This approach allows dimension specification, dimension scheme modification, and dimension scheme

validation. The principle of this approach is that the deviations of the toleranced features follow the direction of the freedoms of the features, thus the deviations can reflect the semantics of the tolerance. Later, Davidson, Mujezinovic and Shah proposed T-MAP [Davidson 02] based on the Shah approach. In T-MAP approach, the variations of the toleranced features are reflected by the volumes of the point sets. Combining Minkowski calculation to the T-MAP approach, the accumulative calculation of the tolerance zones can be obtained.

The concept of the Small Displacement Torsor (SDT) has been developed by Bourdet and Clement [Bourdet 87] in order to solve the general problem of the fit of a geometrical surface model to a set of points using rigid body movements [ElMaraghy 05]. The great advantage of the torsors for the displacement modelisation is due to the fact that the set of torsors is a vectorial space. The composition of small displacements is obtained by a sum of torsors. This operation is commutative and simple to compute, while the composition of two displacements in the general case (great displacements) is a non commutative operation [Bourdet 96]. For their operations of tolerance transfer, the deviations have to be composed. The sum of torsors allows modelising these operations [Giordano 07].

For an effective integration of CAD and CAM, the tolerance specifications should be taken into account in the context of computerization procedure. How to express the tolerance information in the computer point of view is an important topic. Some contributions on this topic are discussed below.

(2). Tolerance representation technique

According to the relationship with the solid modeling technique, the tolerance representation techniques are divided into solid modeling dependent tolerance representation method and solid modeling independent tolerance representation method [Xu 08]. The first one is an attribute model, since it provides an object-oriented programming environment in which specifications of each part, such as tolerances, functional requirements, cost function, machining process, have been treated as classes with their attributes [Roy 98b]. The attribute models can be classified into: CSG based method, the B-Rep based method and the hybrid CSG/B-Rep method. While the other models can be divided into: TTRS, tolerance primitive, UML based method and GeoSpelling, etc.

(a) B-Rep based model.

Johnson [Johnson 85] proposed a tolerance representation approach, which integrates dimensioning and tolerancing modelers with the geometric modelers. Based on this proposal, a dimensioning and tolerancing modeler, named Evaluated Dimensions and Tolerances (EDT) model, is applied in CAM-I system. As the EDT model uses boundary representation, the B-Rep solid modeling part needed to be created first, before the tolerancing specification templates applied to the EDT model. The tolerancing specification templates are used to check the features validity to establish the DRF and to compute the tolerancing values.

The schema of EDT model consists of four types of node: a D/T (Dimension and Tolerance) node; an EL (Entity-Linking) node; a DRF (Datum Reference Frame) note; and an ED (Evaluated Data) note. An example of an EDT model for slot tolerancing is illustrated in figure 1-2.

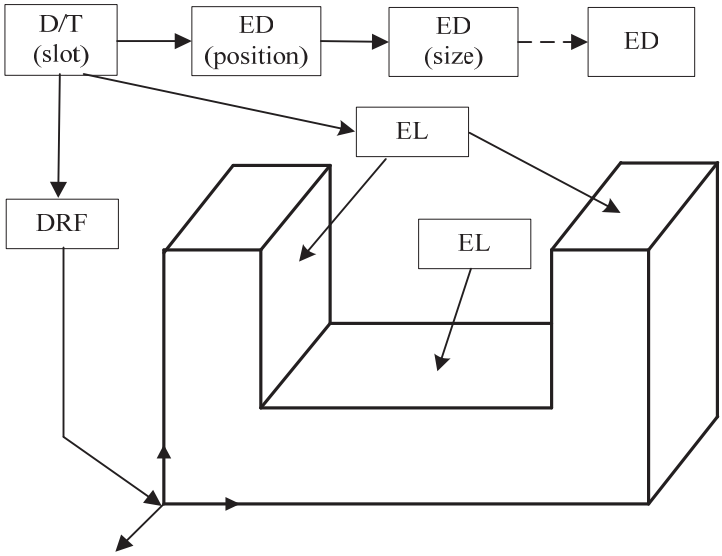


Figure 1-2: The structure of EDT

This B-Rep based model is applicable only for location and size tolerances, and it is limited to geometric entities such as planar faces, cylindrical faces, conical faces and spherical faces.

(b) CSG based model.

Requicha [Requicha 86] proposed the CSG based tolerancing representation model, which named PADL-I and PADL-II modeler. Based on the VGraph (Variational Graph) data structure, the tolerancing information treated as properties or

attributes of an object's features. This VGraph data structure is associated with the PADL modeler by NFace (belonged to nominal geometry), which are associated with the faces of objects constructed by CGS structure. The logical structure of a VGraph is illustrated in figure 1-3. The lowest layer of the VGraph is the NFace notes, which are the nominal geometry of the object. VFace notes linked to the Nface notes and they point to the face on the object's boundary, and VEdge notes represent the intersection of two associated VFaces. SFeature notes and CFeature notes define groups of VFace and VEdge respectively, to form the attribute lists. The DatumSystem note is used to establish the datum systems.

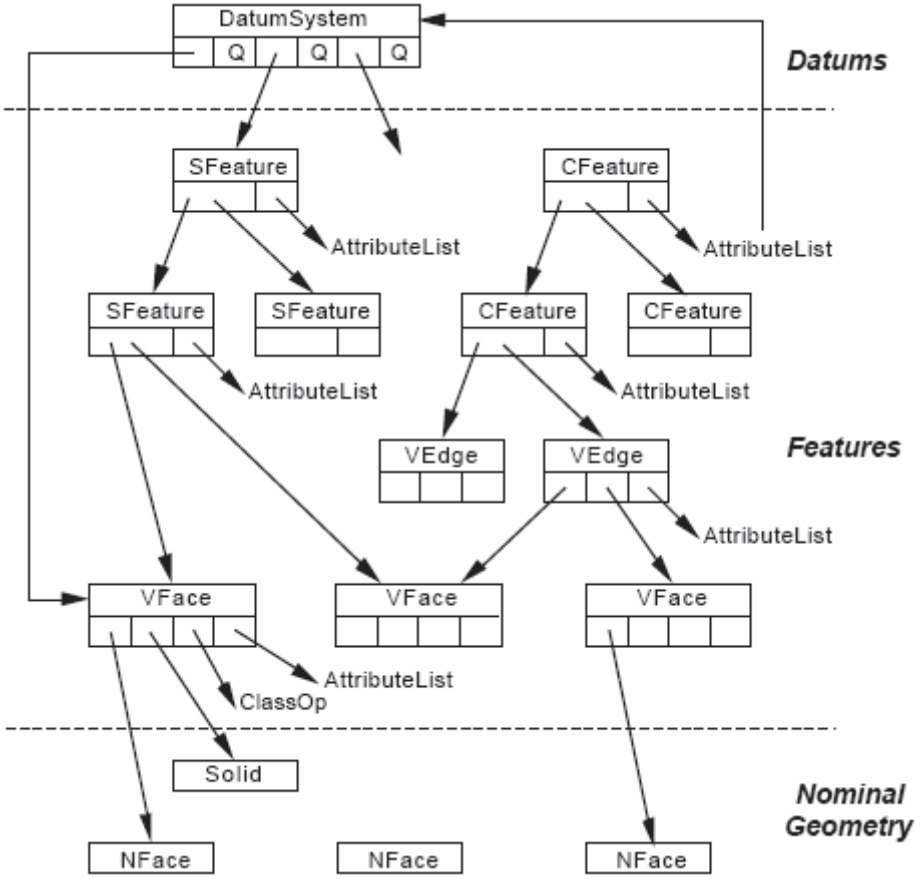


Figure 1-3: The structure of VGraph

The system does not use the B-rep for direct access to these NFaces. An indexing scheme for the faces of each instance of the primitive solids is consistently adopted in the system for the proper identification of any requirement.

As the handling of dimensions and tolerances in the general case requires the ability to access the bounded entities of objects, the CSG-based tolerance theory of Requicha raises some manipulation problems during implementation [Roy 90]. The

principal disadvantage of CSG based approach is that all non primitive faces derived from the same primitive face receive the same variations.

(c) Hybrid CSG/B-Rep model.

The tolerance representation is feature-based, and these geometric features can be classified lower-level features (such as points, lines, arcs and surfaces) and higher-level features (such as holes, slots, pockets or other complex features). Lower-level features are basic topological entities, and the higher-level features are combinations of the lower-level features. Based on the investigation that the features information must be extracted from the solid model, Roy and Liu [Roy 88] proposed a hybrid CSG/B-Rep data structure for the tolerance representation (see figure 1-4).

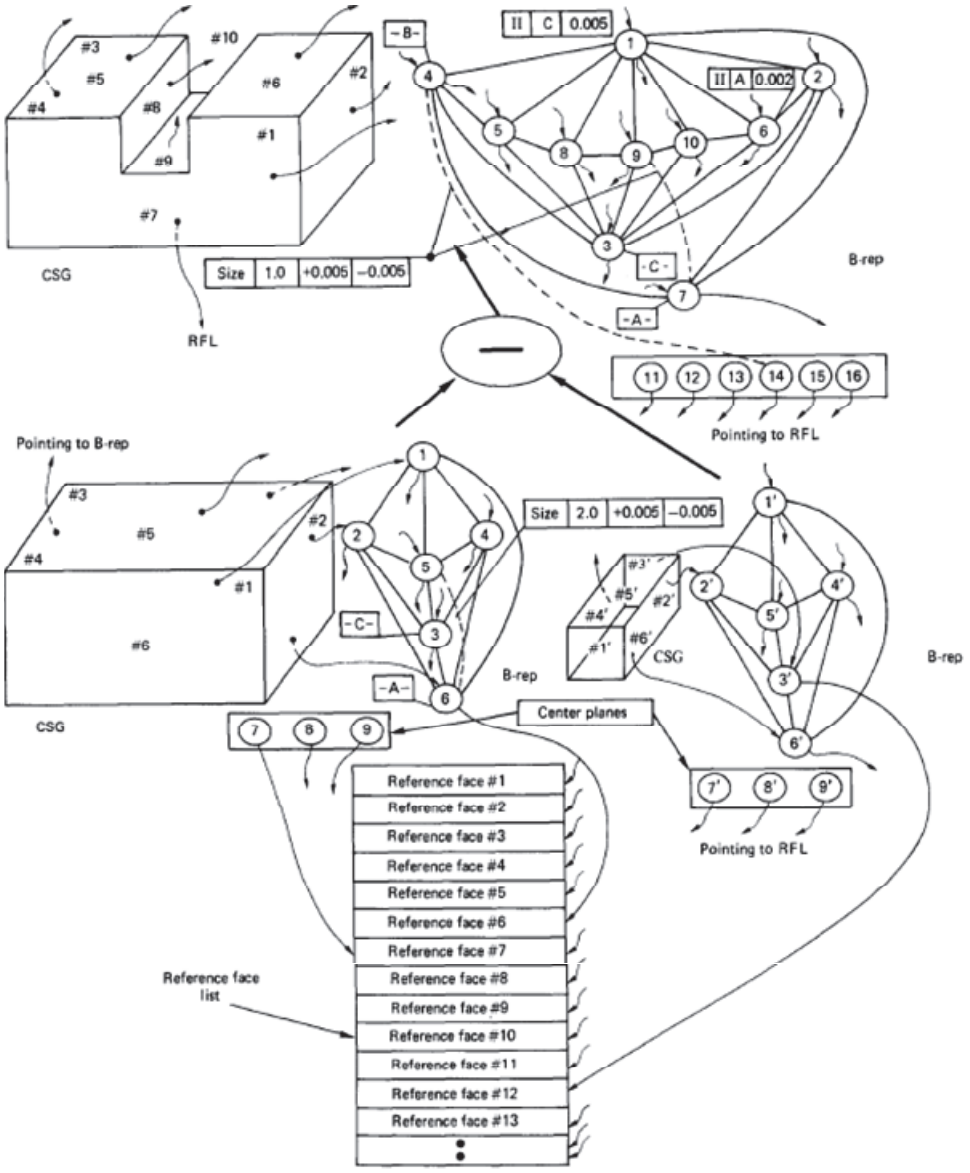


Figure 1-4: The structure of hybrid CSG/B-Rep method [Roy 91]

This approach can integrate the advantages both CSG and B-Rep models. Figure 1-4 shows how the specified tolerance information integrated into the hybrid CSG/B-Rep modeler by the reference face list, which is the ligament between CSG and B-Rep data structure.

Gossard [Gossard 88] proposed a similar feature-based design system, which uses B-Rep solid model and combines GSG representational scheme. This approach can be employed on a polyhedral solid model, but it is limited to the conventional tolerance representation only.

(d) TTRS model

TTRS model is proposed by Clement [Clement91] [Clement 93] to discuss the tolerance representation both in mathematical and theoretical point of view. Clement proved that seven basic surface types can construct the complex objects. These basic surface types are: spherical surface, planar surface, cylinder surface, helical surface, rotational surface, prismatic surface and the any surface. To combine these 7 surface types, 28 cases of combinations can be obtained, and these combinations are named TTRS (Technologically and Topologically Related Surfaces). This method employs the reclassifications of TTRS to denote the different tolerance cases, and the relationship between MGDE (Minimum Geometric Datum Element) and the basic surfaces is shown in table1-1.

Elementary surface	MGDE element	MGDE symbol	DoF
Sphere	Point		3T
Plane	Plane		1T, 2R
Cylinder	Line		2T, 2R
Helical	Point and line or line and plane		2T, 2R, 1TR
Rotational	Point and line		3T, 2R
Prismatic	Line and plane		2T, 3R
Any	Point and line and plane		3T, 3R

Table 1-1: MGDE and the degrees of freedom

Figure 1-5 [Salomons 96] illustrates the reclassification of TTRS and the types of tolerance specification can be determined.

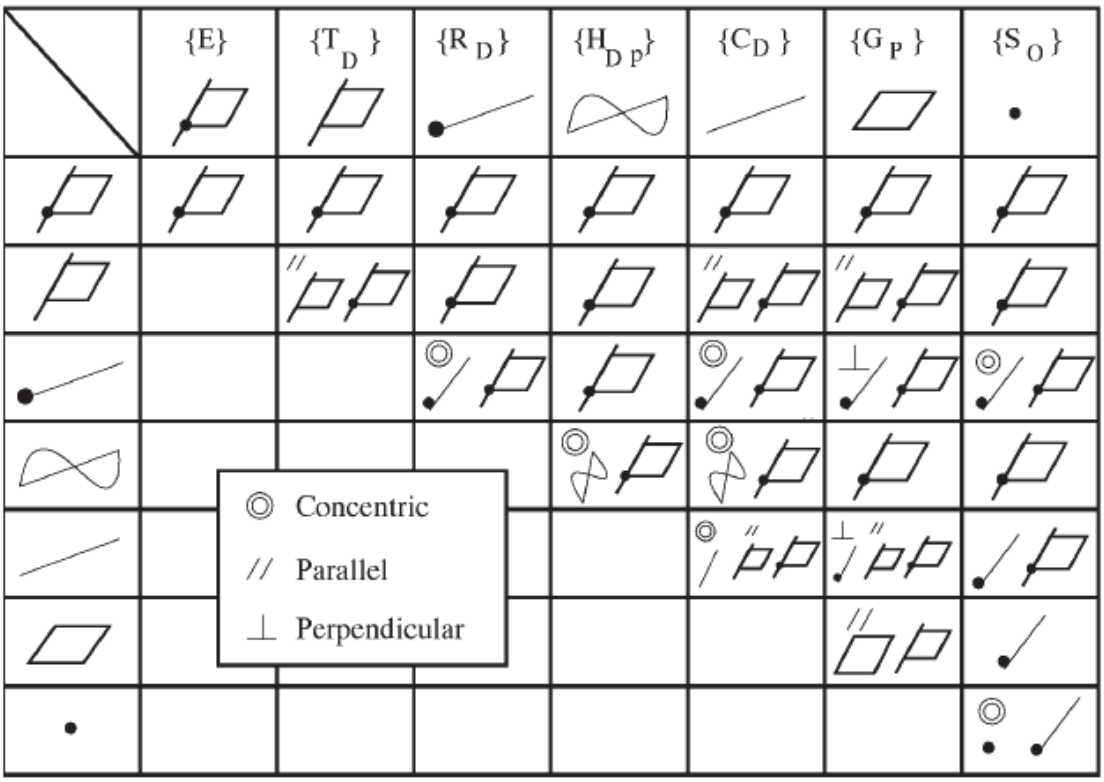


Figure 1-5: Cases of tolerancing between surfaces with MGDEs results

In this approach, tolerances are represented vectorially, in so–called torsors, allowing for tolerance analysis. This has been addressed in [Roy 90] [Gaunet 94] and [Rivière 94].

(e) GeoSpelling model.

GeoSpelling is a geometric product specification model proposed by Mathieu and Bulla used to describe ideal and non-ideal geometry [Mathieu 03] [Ballu 95]. Indeed, it allows express the specification from the function to the verification with a common language. This model is based on geometrical operations which are applied not only to ideal features, defined by the geometrical modelers in a CAD system, but also to the non-ideal features which represent a real part. These operations are themselves defined by constraints on the form and relative characteristics of the features. GeoSpelling is proved to be a unique specification model used in the global product lifecycle in mathematical and theoretical view [Ballu 95]. The details of Geospelling model is presented in section 2.2.

1.3. Shape modeling

Shape is any multi-dimensional media that is primarily characterized by form or spatial extent in a space of two, three or higher dimension [Biasotti 08]. Considering that shape records all the geometrical information that remains when location, scale and rotational effects are filtered out from an object [Kendall 84], it is an important information media and focal resource in many scientific and applied scenarios: mechanical engineering, computer graphics and vision, biological and medical applications, etc.

According to Requicha [Requicha 89], a shape model is a computational structure that captures the spatial aspects of the objects of interest for an application. Requicha's work which shaped the whole geometric modeling field introduces four levels of abstraction, called universes. The first level is used to clarify the problem from the point of view of the universe to be modeled, (the physical universe); the second level is used to analyze the problem from the mathematical point of view (the mathematical model); the third allows to understand the various issues of discretizing the elements from the mathematical universe (the representation or computational model); and the fourth is used to map the discretized elements from the representation universe into the structures of a computer language (implementation model) [Robbiano 10].

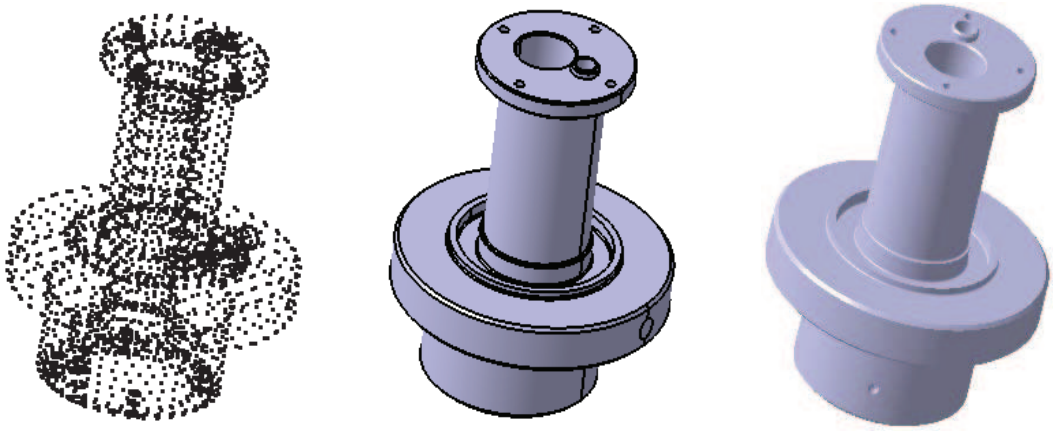
In mechanical engineering, shape modeling is a consistent set of principles for mathematical and computer modeling of three-dimensional geometric shapes and/or solids. It covers the domains as geometric shape design, shape representation, shape analysis and etc. The computer representation of shapes which provides the foundations for other issues are the basis for shape modeling.

In this section, three issues are discussed for shape modeling: shape structural expression, shape representation scheme and shape description.

1.3.1. Shape structural expression

Shape represents 3D object using a collection of various geometric entities in 3D space, such as points, triangles, curves, surfaces, etc. According to their appearance, shape structural expression can be divided into three categories: point-based,

surface/shell-based and volume-based. Figure 1-6 gives an example of these three expressions of the same object.



(a) point-based (b) surface-based (3) volume-based

Figure 1-6: Three shape structural expressions

(1). Point-based

A point-based shape, also called point cloud, is a set of vertices in a three-dimensional coordinate system. These vertices are usually defined by X, Y, and Z coordinates, and is typically intended to be representative of the external surface of an object. There are still some kinds of point clouds which are volumetric data, like the point cloud acquired by Computer Tomography (CT) used in medical and metrological applications.

Point-based data are usually created by 3D scanners. In metrology/inspection purpose, the acquired point data can be used directly to compare with the CAD model, but, more commonly, they themselves are not direct usable in many applications and therefore are converted to polygon or triangle mesh models, NURBS surface models, or CAD models through a process commonly referred to as surface reconstruction.

(2). Surface/shell-based

These models represent the surface, e.g. the boundary of the object, not its volume (like an infinitesimally thin eggshell). Because the appearance of an object depends largely on the exterior of the object, surface representations are common in computer graphics. Two dimensional surfaces are a good analogy for the objects used in graphics, though quite often these objects are non-manifold. For discrete surface, a

discrete digital approximation is required since surface is not finite. Polygonal meshes (and to a less extent subdivision surfaces) are by far the most common representation

(3). Volume-based

These models define the volume of the object they represent (like a rock). These are more realistic, but more difficult to build. The material characters, like mass, elastic modulus, etc. can be specified for a volume-based shape. Volume-based models are mostly used for non-visual simulations such as medical and engineering simulations, for CAD and specialized visual applications. Recently, Computer-Tomography (CT) technology is increasingly applied in dimensional metrology, which acquires volumetric point clouds, the volume-based shape modeling is becoming more and more popular.

1.3.2. Shape representation

Representations of shape or geometry have a very strong connection to mathematics. These include Set Theory, Graph Theory, Algebraic Topology, etc., in addition to the various varieties of shape/geometry: projective, analytic, algebraic, differential, and combinatorial [Naylor 96]. Based on the mathematical foundations, a shape representation scheme can be defined as a mapping from a computer structure to a well-defined mathematical model which defines the notion of the physical object in terms of computable mathematical properties and is independent of any particular representation scheme [Shapird 01]. For mechanical engineering applications, we focus on the different methods or schemes to represent shapes in computer, the details about the mathematical model can be referred to [Biasotti 08] [Shapird 01]. Considering various kinds and properties of shapes, many representation methods are developed, and the same shape may also be represented by different schemes. The modeling space of any representation methods is finite, and any single representation scheme may not completely suffice to represent all types of solids. Therefore, the most suitable representation methods should be considered for a specific shape.

In this thesis, the shape representation schemes are classified by their principles and used data structures but organized according to their suitable applications. Any single pure representation in its pure form will be limited in the range of operations for which it is ideally suited [Naylor 96]. It is interesting and useful to identify their

suitable applications considering the shape properties. The considering shape properties include shape type (discrete or continuous) and their structural expression (point-based, surface/shell-based or volume-based). Figure1-7 presents a general taxonomy of categories of shape representation schemes for this purpose.

For each category, only several representative schemes are presented. Not all representation schemes used today are presented in this classification. In the following, some of the widely used schemes are discussed in further details.

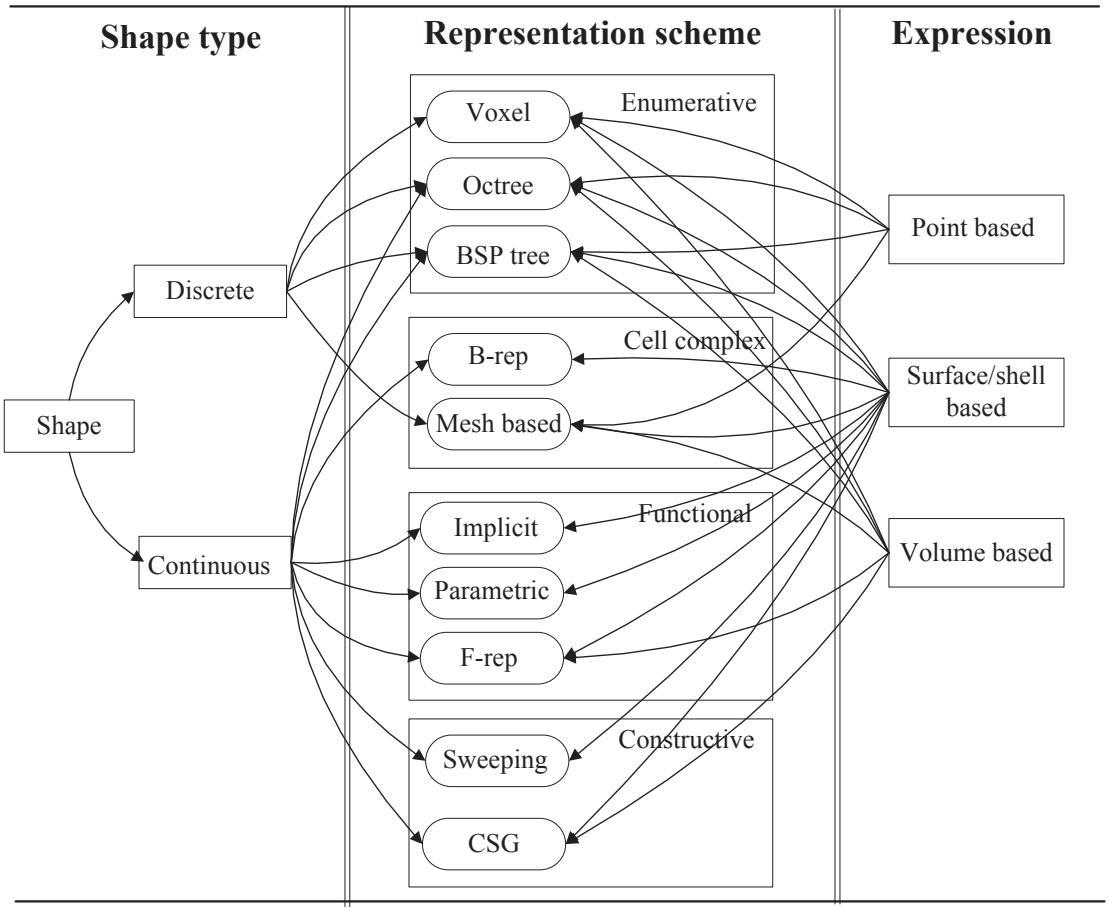


Figure 1-7: Taxonomy of categories of shape representations

(1). CSG (Constructive Solid Geometry)

Constructive solid geometry (CSG) is a technique used in solid modeling. Constructive solid geometry allows a modeler to create a complex surface or object by using Boolean operators to combine primitive objects. CSG is often used in CAD and 3D computer graphics for procedural modeling. It can also be performed on Polygon mesh based models.

In CSG, The simplest solid objects used for the representation are called primitives. Typically they are the objects of simple shape: cuboids, cylinders, prisms, pyramids, spheres, cones. It is said that an object is constructed from primitives by means of allowable operations, which are typically Boolean operations on sets: union, intersection and difference. The construction for the solids bounded by planar and second degree surfaces have been solved completely [Shapird 01]. Figure 1-8 gives an example of CSG representation of a mechanical part.

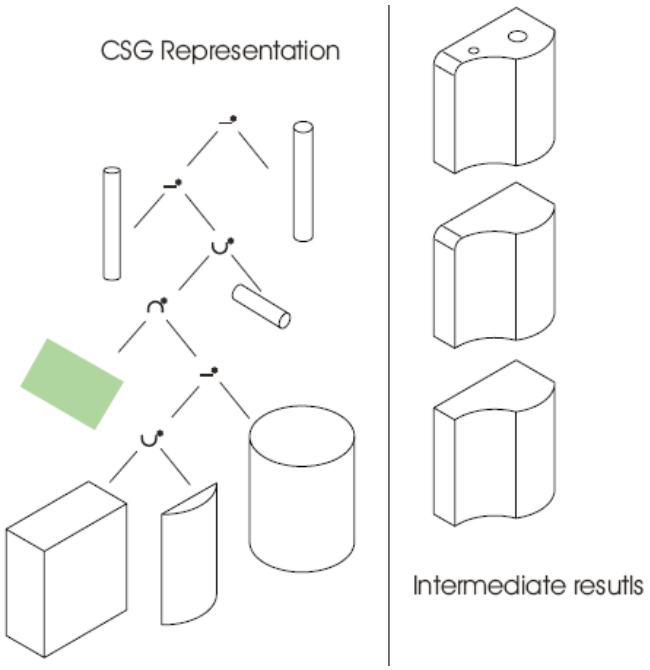


Figure 1-8: CSG representation [Shapird 01]

The attractive properties of CSG include conciseness, guaranteed validity (by definition), computationally convenient Boolean algebraic properties, and natural control of the solid's shape in terms of high-level parameters defining the solid primitives and their positions and orientations. The relatively simple data structures and the elegant recursive algorithms further contributed to the popularity of CSG in academia and early commercial systems [Shapird 01].

Beside CSG, there are many other constructive methods, like offsetting [Rossigna 86], blending, convolutions [Bloomenthal 91], sweeping [Sourin 96], etc. they have popular applications in production engineering, mechanical analysis, and computer graphics, etc.

(2). B-rep (Boundary representation)

Boundary representation is a method for representing shapes using the limits. A

shape/solid is represented as a collection of connected surface elements, the boundary between solid and non-solid. A B-rep shape model is composed two parts: topology and geometry. The main topological items include faces, edges, vertices and the geometrical items are surface, curves and points. A face is a bounded portion of a surface; the bounding curves of a face are represented as edges. An edge is a bounded piece of a curve; the portion curve that forms the edge is represented by two vertices. A vertex lies at a point. A simple example of B-rep shape using 6 faces is given in figure1-9.

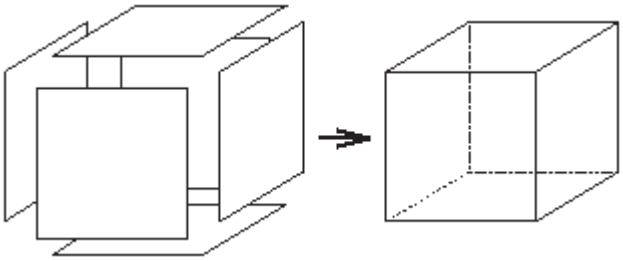


Figure 1-9: An example of B-rep shape model

Comparing to CSG, B-rep is much more flexible and has richer operations. Therefore, the B-rep shapes are widespread adopted in modern CAD systems, like Parasolid and ACIS, etc. The worldwide data exchange standard STEP also defines some data models for boundary representation and figure 1-10 shows the topological structure defined by STEP 10303-42.

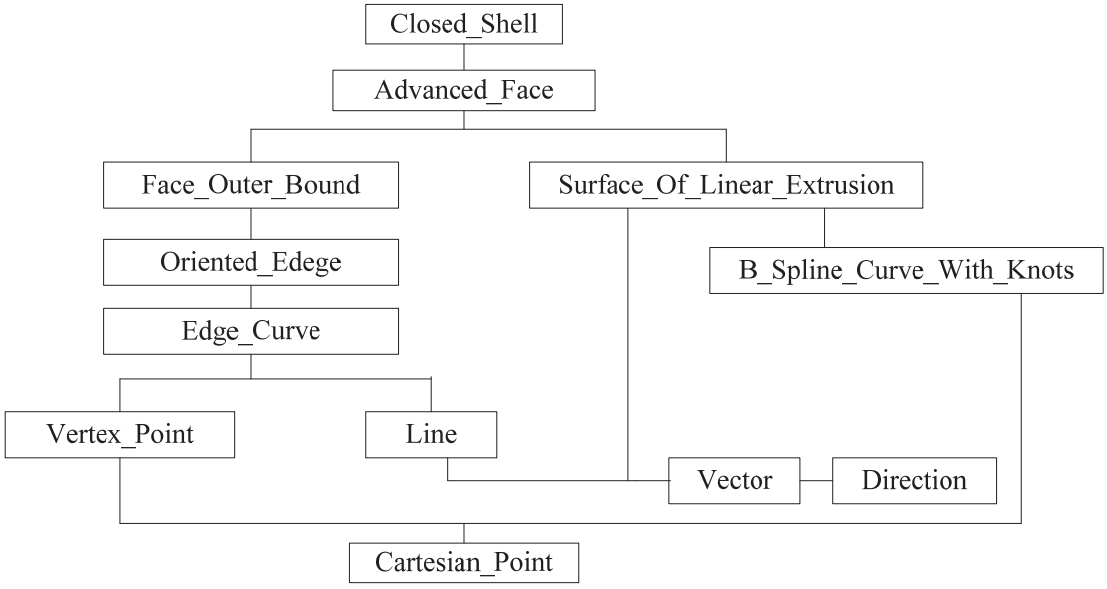
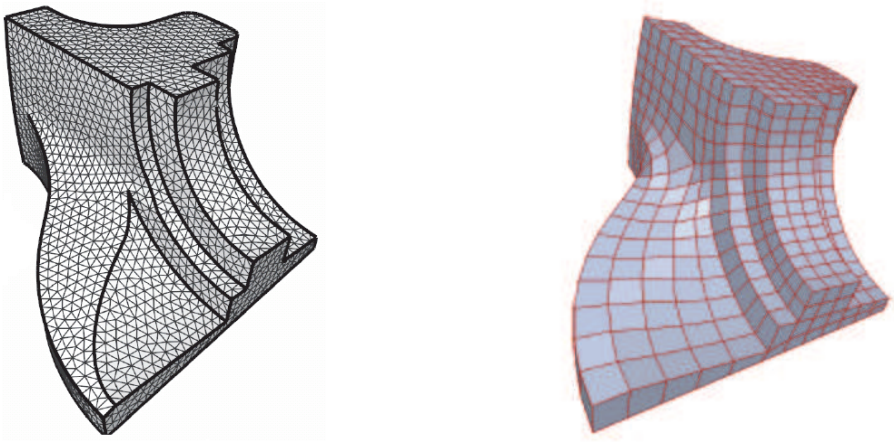


Figure 1-10: The topological data structure defined in STEP standard

(3). Polygon mesh

A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling. The faces usually consist of triangles (see figure 1-11 (a)), quadrilaterals (see figure 1-11 (b)) or other simple convex polygons.



(a) triangle mesh (b) quadrilateral mesh

Figure 1-11: Examples of polygon mesh representation

Polygon mesh is suitable to construct the topological structure of discrete point-based surfaces. It is extremely used in computer graphics for geometry-based applications. In engineering, Applications of polygon mesh focus on FEA (Finite Element Analysis) and manufacturing/metrology purpose. In manufacturing purpose, the polygon mesh is generated from the continuous surface usually as STL format [Stla 09] which is more convenient for computing by the manufacturing machines. In metrology, an object is scanned and the point cloud is acquired by the 3D scanners firstly, the point cloud is then needed to be modeled as polygon mesh based on which the neighbor information can be queried for further analysis.

The basic elements for mesh modeling are vertices, edges, facets and polygon. Different mesh representation structures store and organize these elements in different way. Typically, the common used data structures for mesh representation include facet-based and boundary based. 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 [Kettner 99], 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. There are various types of boundary representations, such as Winged-edge, Quad-edge, Half-edge, etc and a more sophisticated Half-edge data structure. A good survey of data representations for polygon mesh can be found in [Kettner 99].

The 3D mesh, also called volumetric mesh, which not only represents the surface, but also represent the volume information, is also used in some application, but this is out of this thesis's content.

(4). Voxel-based representation

A voxel is a volume element, representing a value on a regular grid in three-dimensional space. This is analogous to a pixel, which represents 2D image data. As with pixels in a bitmap, voxels themselves do not typically have their position (their coordinates) explicitly encoded along with their values. Instead, the position of a voxel is inferred based upon its position relative to other voxels. In contrast to pixels and voxels, points and polygons are often explicitly represented by the coordinates of their vertices.

Voxel-based representation is a kind of spatial-partitioning technique. With this technique, a solid is decomposed into small cubic voxels. An example of voxel-based representation is given in figure 1-12.

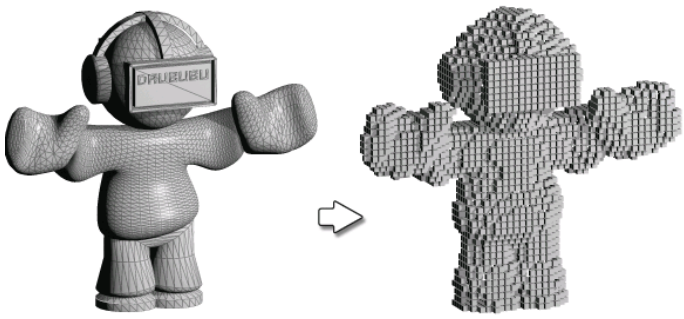


Figure 1-12: An example of B-rep shape model

The accuracy of voxel-based shape depends on the size of the voxel unit. It usually used to speed up the geometry computations. Typically voxel-representation is used for volumetric representation, such as the point data acquired by CT scanner used in medical and metrology applications. But it can be used to represent polygon/surfaces with lots of empty-filled cells.

(5). F-rep (Function representation)

F-rep [Pascal 04] was introduced in "Function representation in geometric modeling: concepts, implementation and applications" as a uniform representation of multidimensional geometric objects (shapes). F-rep defines a geometric object by a single real continuous function of point coordinates as formula 1-1

$$F(X) \geq 0 \tag{1-1}$$

$X = (x_1, x_2, ..., x_n)$ is a point in E^n, and F is the defining function defined on space E^n. The above inequality defines a closed n-dimensional object in En space with the following characteristics:

$F(X) > 0$ for points inside the object;

$F(X) = 0$ for points on the object's boundary;

$F(X) < 0$ for points outside the object.

F-rep combines many different models like algebraic surfaces, skeleton based implicit surfaces, set-theoretic solids or CSG, sweeps, volumetric objects, parametric models, procedural models, etc. the main features of F-rep are:

(a) Defining function evaluation procedures traversing the constructive tree structure.

(b) Leaves of the constructive tree are primitives with known types of defining functions.

(c) Nodes of the tree contain operations and relations.

(d) Extensibility of primitives, operations, and relation.

(6). Parametric representation

Parametric representation is the most general way to specify a surface. A parametric surface is defined by a parametric form with two parameters in the Euclidean space R^3. The two parameters are often labeled as u and v. The parameterization of a surface can then be formed as a vector valued function as formula 1-2

$$s = \mathrm{r}(u, v) \tag{1-2}$$

The regular surfaces, such as sphere, cylinder, parabolic, hyperboloid, etc. their parametric form are quite known. Take the spherical surface for example, which has the parametric formula 1-3

$$\begin{cases} x = r \cdot \sin(u)\cos(v) \\ y = r \cdot \sin(u)\sin(v) \\ z = r \cdot \cos(u) \end{cases} \tag{1-3}$$

Where, $u \in [0, \pi]$ and $v \in [0, 2\pi]$.

In general, for 3D freeform surface representation, The Bezier surface, B-Spline and NURBS (Non-Uniform Rational B-Spline) surfaces are the most commonly implemented techniques in CAD and computer graphics applications. Take NURBS surface which is the most popular techniques in CAD for parametric surface modeling for example, the parametric form can be represented as formula 1-4

$$\mathrm{S}(u, v) = \frac{\sum_{i=0}^{p}\sum_{j=0}^{q} B_{i,m}(u) \cdot B_{j,n}(v) \cdot P_{i,j} \cdot w_{i,j}}{\sum_{i=0}^{p}\sum_{j=0}^{q} B_{i,m}(u) \cdot B_{j,n}(v) \cdot w_{i,j}} \tag{1-4}$$

Where, $P_{i,j}$ is a control point, and $w_{i,j}$ is the weight being associated to it. $B_{i,m}(u)$ and $B_{j,n}(v)$ are the basis functions along u, v directions respectively, and they have similar recursive definitions. The definition of $B_{i,m}(u)$ is decribed by formula 1-5

$$B_{i,m}(u) = \begin{cases} B_{i,0}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & else \end{cases} & (m = 0) \\ B_{i,k}(u) = \dfrac{u - u_i}{u_{i+k} - u_i} B_{i,k-1}(u) + \dfrac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} B_{i+1,k-1}(u) & else \end{cases} \tag{1-5}$$

$B_{i,m}(u)$ and $B_{j,n}(v)$ can also be expressed through a polynomial for convenient derivatives calculation. Figure 1-13 gives an example of a NURBS surface representation.

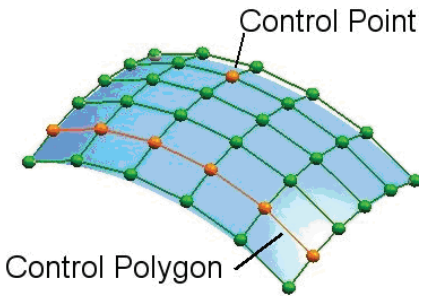


Figure 1-13: An example of NURBS surface

(7). Implicit representation

Implicit surfaces are widely used in computer graphics. It provides powerful primitives for geometric modeling. They have applications in ray-tracing and are also used to determine whether a point is inside or outside of an object. Implicit surface are the surfaces which are contours (isosurfaces) through some scalar field in 3D. A typical implicit surface is defined by formula 1-6

$$s: f(x,y,z)=0 \tag{1-6}$$

Where $f:\Omega\subseteq\mathbb{R}^3\to\mathbb{R}$, for well-behaved functions f, this defined surface is indeed a manifold surface. A simple example is given as the implicit representation of a sphere as formula 1-7

$$f(x,y,z)=x^2+y^2+z^2-R^2=0 \tag{1-7}$$

One of the most important properties of implicit representation is that connectivity dose not need to be determined for the discretization. A uniform Cartesian grid can be used along with straightforward generalizations of the technology from two spatial dimensions. Possibly the most powerful aspect of implicit surface is that it is straightforward to go from two spatial dimensions to three spatial dimensions.

Implicit interface representations include some very powerful geometric tools. For example, since we have designated the $f(\vec{p})=0$ iso-contour as the interface. We can determine which side of the interface a point is on simply by looking at the local sign of $f(\vec{p})$. $f(\vec{p})<0$ indicates the point is inside the interface while

$f(\vec{p}) > 0$ indicates outside. The point is on the surface when $f(\vec{p}) = 0$. Figure 1-14 illustrates these cases in 2D representation.

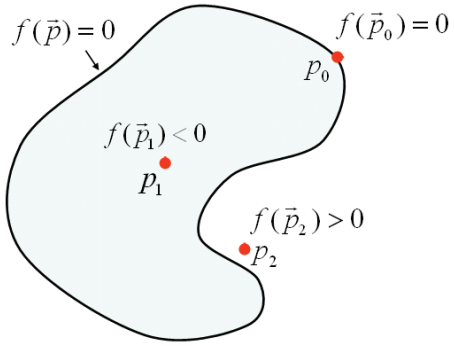


Figure 1-14: Implicit representation in 2D space

1.3.3. Shape description

Even through shape representation methods have been researched extensively, more recently, research interests are shifting from methods to represent shapes towards methods to describe shapes. The distinction between representation and description can be expressed as follows: an object representation contains enough information to reconstruct (an approximation to) the object, while a description only contains enough information to identify an object as a member of some class [Nackman 84]. Shape representation that contains more details and accurate information of an object is quantitatively similar to the object; description which is more concise and conveys and elaborate and composite view of the object class is only qualitatively similar.

Shape descriptors which simplify the shape representations but carry most of the important information and easier to handle are used to describe shapes. Various descriptors are designed for shape description, including shape signature, signature histogram, shape invariants (e.g. curvature), moments, shape context, shape matrix spectral features, etc. [Zhang 04a]. In this section, the shape descriptors we consider are of two types: TTRS (see section 1.2) and curvature based shape descriptors.

Moments are used to transform a shape into a set of features or 1D function and carry invariant properties with respect to some group of spatial transformation. They are robust in shape identification and shape comparison tasks [Taron 07].

Curvature is a shape invariant parameter and has been classically used as a shape

descriptor. Curvature is one of the most critical differential characters of a shape (both continuous and discrete). There are many curvature-based descriptors, such as the principal curvatures, Gaussian curvature, Mean curvature, shape index, curvedness, etc. Typically, three or four numbers of curvature are sufficient to characterize the shape for an infinitesimally small region [Zhang 04a].

Curvatures are defined to measure the local bending of the surface. Take a given point p for example, for each unit direction $\vec{u}$ on its tangent plane $T_{\Sigma}(p)$, 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 formula 1-8 [Oneill 97]:

$$\kappa_n(\vec{u}) = S_p(\vec{u}) \cdot \vec{u} \tag{1-8}$$

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 formula 1-9:

$$S_p(\vec{u}) = -\nabla_{\vec{u}}\, \vec{n}(p) \tag{1-9}$$

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. The principal curvatures can also be derived directly from the Gaussian and Mean curvatures which can computed directly based on the first and second fundamental forms of surface.

For discrete shape, the discrete shape operator can be constructed based on the neighbor information of a given point and the principal curvatures can then be approximated (see figure 1-13) [Taubin 95] [Cohen-Steiner 03] [Zhao 10].

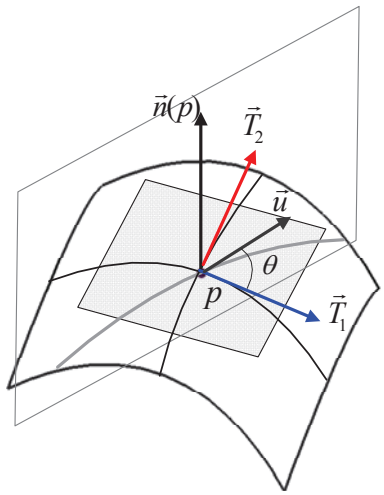


Figure 1-15: Normal and principal curvature directions of a smooth surface at point p

Other curvature indicators can be derived from the principal curvatures. If κ_1 and κ_2 represent the maximum and the minimum principal curvatures respectively. Some important curvatures can be represented as formula 1-10 to formula 1-13:

$$K = \kappa_1 \cdot \kappa_2 \tag{1-10}$$

$$M = \frac{1}{2}(\kappa_1 + \kappa_2) \tag{1-11}$$

$$s = -\frac{2}{\pi}\arctan(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}) \quad (\kappa_1 \geq \kappa_2) \tag{1-12}$$

$$c = \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}} \tag{1-13}$$

Where, K indicates Gaussian curvature which is an intrinsic measure of curvature. The parameter M indicates Mean curvature which is an extrinsic measure of curvature. The parameter s indicates shape index which is a single value within the range [-1, 1] to measure the local surface types [Koenderink 92]. The parameter c is the curvedness which is a positive number to specify the amount of the surface curvatures [Koenderink 92].

1.4. Conclusion

A coherent and complete tolerancing process is necessary to control the geometrical variations along the whole product lifecycle. This chapter surveys the techniques for geometrical variation management within product lifecycle.

Firstly, the tolerance modeling methods which are common-used in CAT and Geometrical product specification are reviewed considering the tolerance specifications and representations.

Secondly, the shape modeling techniques are introduced considering three issues: shape structural expressions, shape representations and shape descriptions. The different representation methods are discussed according their principles. The mapping relationship between shape structural expressions and representations are constructed. Discrete shape modeling is highlight for the further processing of geometrical variations.

In the Chapter 2, we will derive the main characteristics of discrete geometry and Geospelling in order to provide a new framework for the operationalization of Geospelling. The Skin model or non-ideal geometry will benefit from our studies as described in Chapter 3 and Chapter 4.

Chapter 2

A Discrete Geometry Framework for GeoSpelling

2.1. Introduction

GeoSpelling is proposed as a coherent and complete model to manage the geometrical variations along product lifecycle. GeoSpelling is defined based on geometrical operations to handle both ideal (e.g. CAD model) and non-ideal (real part) features. The methods to handle ideal features are extensively researched and are mature in the popular CAD software packages. But the methods to handle the non-ideal features (skin-model which is the kernel concept in GeoSpelling) haven't solved yet.

With widely applied 3D scanners (e.g. Laser scanner), 3D point clouds acquired by 3D scanners is becoming popular in many applications, such as reverse engineering, quality control, rapid-prototyping, etc. 3D point clouds which are acquired from the real part can represent the real features. Therefore, the skin model can be constructed and simulated based on 3D point clouds. The discrete geometry processing technologies can then be used to handle the discrete skin models and the related geometrical operations.

This chapter represents the characteristic concepts of Geospelling and discrete geometry. The corresponding relationships between GeoSpelling and discrete geometry are constructed and a new framework is defined for the geometrical operationalization of GeoSpelling from the view of discrete geometry.

This chapter is organized as follow:

Section 2.2 presents the characteristic concepts and geometrical operations of GeoSpelling. Section 2.3 introduces the basic principles of discrete geometry. Section 2.4 discusses the new framework for operationalization of GeoSpelling using discrete geometry methods. Section 2.5 is a case study and 2.6 is the conclusion.

2.2. GeoSpelling

GeoSpelling is a geometrical specifications model proposed by Mathieu and Ballu in 1990s. This model is designed as a mean of communication, in which designers, manufacturers and metrologists exchange product geometrical information concerning the function requirements of products unambiguously. GeoSpelling that

has been adopted by ISO TC 213 experts can describe both ideal and non-ideal geometrical features, and it allows expressing th94e geometrical specification from the function to the verification with a common language [Mathieu 07]. While, as the actual GeoSpelling has not been defined completely, it can not be fully integrated in CAD/CAM/PLM chain directly. The objective of improving the actual GeoSpelling is to enhance its computational capability. In this chapter, t discrete geometry concepts and methods are discussed in the context of GeoSpelling. Discrete geometry mainly focuses on discrete geometrical objects (such as point and mesh), and it also has a large overlap with computational geometry and discrete differential geometry. These characteristics of discrete geometry could provide a consistent and rigorous framework for enhancing digital processing capability of GeoSpelling. GeoSpelling model is defined by the following concepts.

2.2.1 Specification

A specification is a **condition** on a **characteristic** defined form **geometrical features**, and these geometrical features are created from the model of the real surface of the part (**skin model**) by different **operations** (figure 2-1) [Mathieu 05].

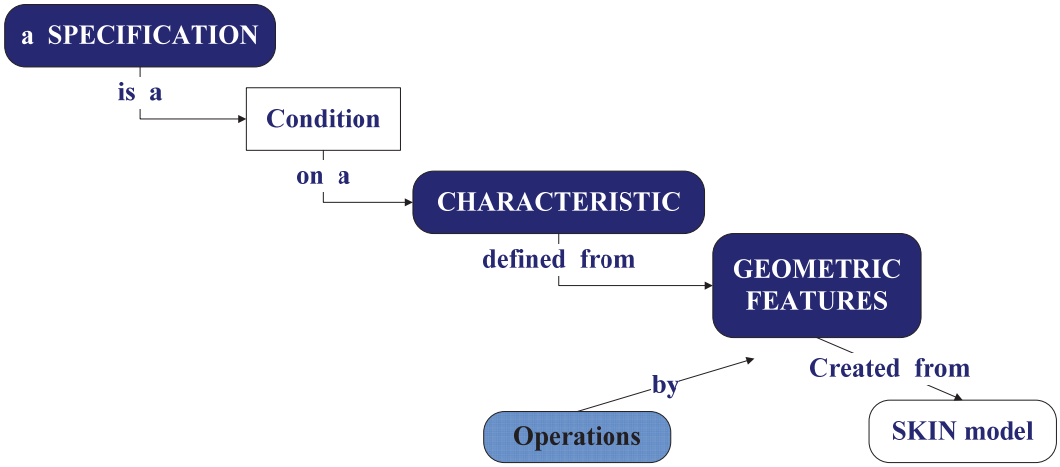


Figure 2-1: Specification defined in GeoSpelling

2.2.2 Geometrical Features

In GeoSpelling, the features are classified into three categories: ideal features, non-ideal features and limited features. Today the definitions of these features are roughly described as follow.

Let us consider ideal features such as a plane of the nominal model of the part or a cylinder fitted to a real feature nominally cylindrical. In the first case, the plane of the nominal model is a nominal feature; while in the second case, the cylinders fitted to real surfaces are fitted features. Therefore, the ideal features can be divided into two classes: nominal features and associated features. The nominal features are those features that completely independent of the real surface of the part and also can constitute the nominal model defined by the CAD systems. The associated features can be obtained by association operation from a skin model. Actually, an ideal feature can be defined by an equation, and they are characterized by their invariance class [Clement 94] and their type.

Non ideal features such as a surface portion of the model of the real surface of a part or a real axis. The non ideal features can not be defined by equations. They depend completely on the real surface of the part.

2.2.3 Operations

The operations are used to obtain geometrical features. They are classified into six categories (figure 2-2).

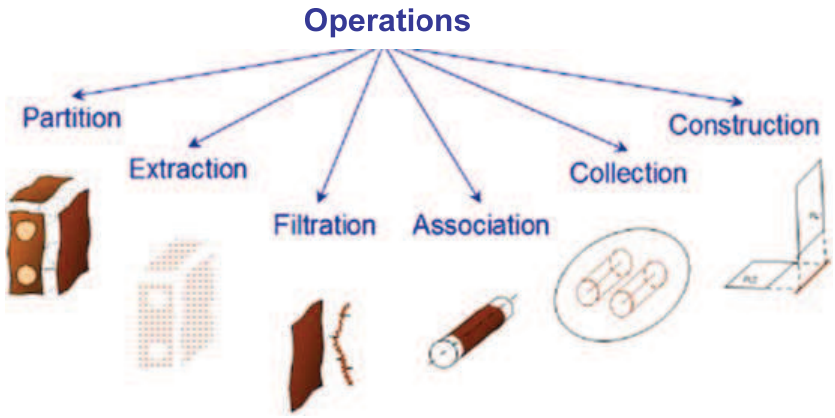
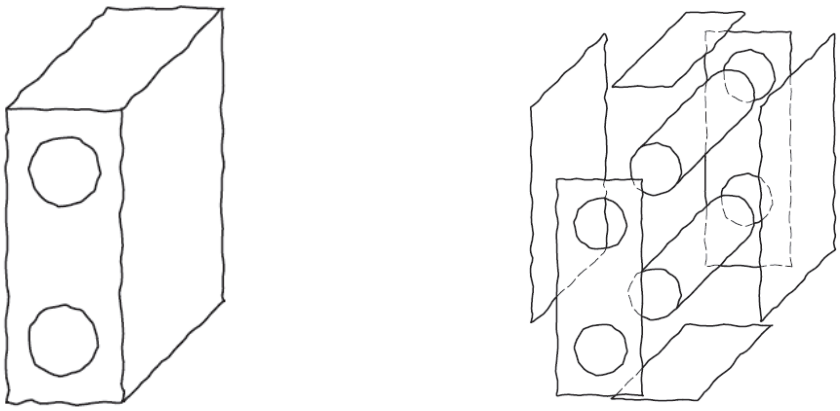


Figure 2-2: Classifications of operations

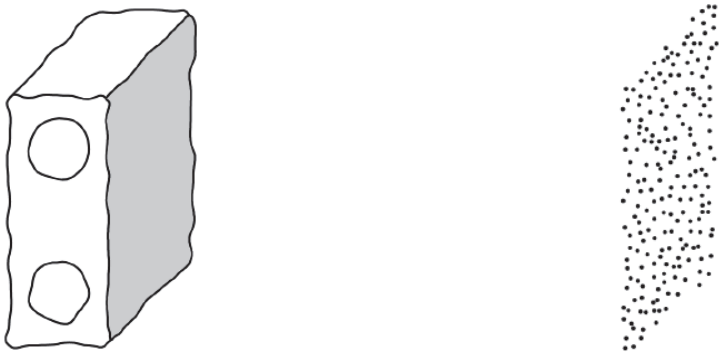
A **partition** is an operation used to identify bounded feature(s) from non ideal feature(s) or from ideal feature(s) (figure 2-3).



(a) Skin model (b) Partition

Figure 2-3: Partition operation

An **extraction** is an operation used to identify specific points from a non ideal feature (figure 2-4).



(a) Skin model (b) Filtration

Figure 2-4: Filtration operation

A **filtration** is an operation used to create a non ideal feature by reducing the level of information of a non ideal feature (figure 2-5).

Figure 2-5: Extraction operation

A **collection** is an operation used to consider more than one features together (figure 2-6).

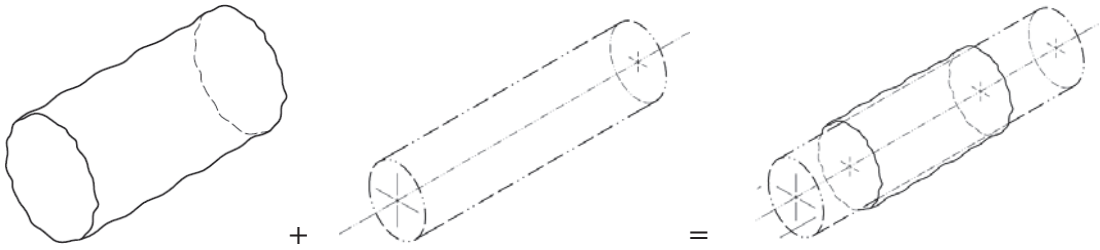


Figure 2-6: Collection operation

An **association** is an operation used to fit ideal feature(s) to non ideal feature(s) according to a criterion (figure 2-7).

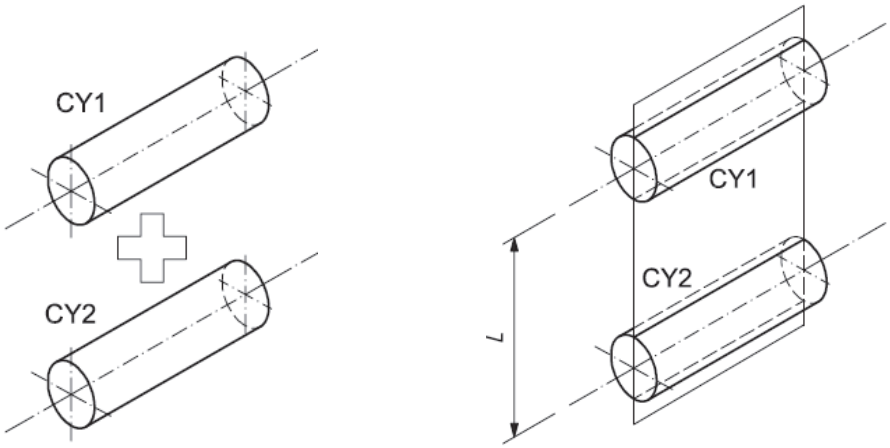


Figure 2-7: Association operation

A **construction** is an operation used to build ideal feature(s) from other ideal features with constraints (figure 2-8).

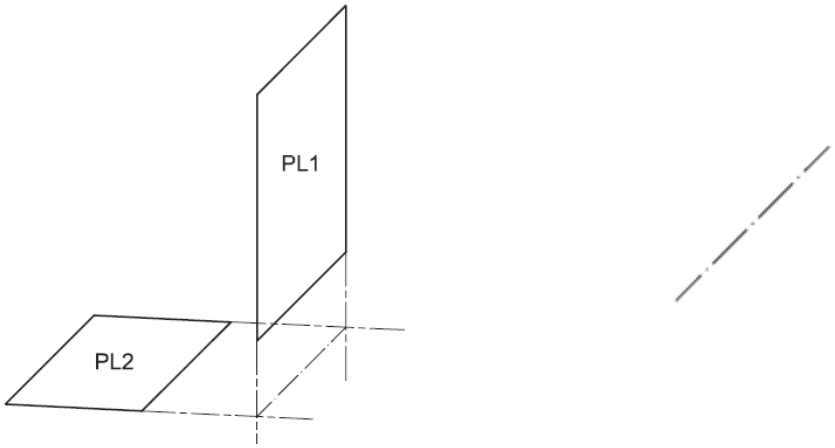


Figure 2-8: Construction operation

2.2.4 Characteristics

A characteristic is a parameter of one feature or between two features to express a length or an angle. The characteristics are used to define specifications, and they belong to five families.

(a) Intrinsic characteristic. They are the characteristic of an ideal feature.

(b) Situation characteristic between ideal features. The characteristic defines the relative position between two ideal features.

(c) Situation characteristic between ideal and limited features. The characteristic defines the relative position between an ideal feature and a limited feature.

(d) Situation characteristic between ideal and non ideal features. The characteristic defines the relative position between a non ideal feature and an ideal feature.

(e) Situation characteristic between non ideal features. The characteristic defines the relative position between two non ideal features.

2.2.5 Skin model

Based on the GPS standard [ISO 2005], the surface model is a closed surface which can represent the interface between workpiece and its environment. The concept of skin model is the basic foundation of GeoSpelling, and all the operations are performed on the skin model. The skin model is a kind of surface model, and it is opposite to nominal model.

The nominal surface model is a surface model of perfect shape defined by CAD software. Normally, the nominal surface model is deemed as an ideal feature. It used to reflect the design intend and the function requirements. A comparison between the nominal model and the skin model is illustrated in figure 2-9.

The skin model also called the specification surface model is a surface model of non-perfect shape imagined by the designers to reflect the specifications and verification requirements. Considering the form errors of workpieces involved during the manufacturing and inspection process can be classified into two categories: random errors and systematic errors [Henke 99] [Desta 03], the skin model need to

reflect these two kinds of errors. The definition of skin model proposed in our work is that a surface model combined both random and systematic errors on a nominal model. The skin model simulation methods will be introduced in chapter 3 and chapter 4.

(a) Nominal model (b) Skin model

Figure 2-9: Surface model

Discrete geometry

Along with the development of computer science and information technologies, discrete geometry emerged as a new growing area of research that uses concepts from classical and non-classical geometry, topology, combinatorics, probability theory, computer science and engineering in order to design structures and algorithms for the acquisition, representation, processing and transmission of complex 3D models. Discrete geometry as a discipline is on the cutting edge of modern applications such as geographic information systems, computer graphics, solid modeling, and computational biology.

Discrete geometry deals with the structure and complexity of discrete geometric objects as well as with the design of efficient computer algorithms for their manipulation. Discrete refers to the fact that the objects of interest are discrete data (point sets, triangle meshes, tetrahedron meshes, etc.) and is used to contrast with "continuous" geometry, for example, smooth surfaces.

Discrete geometry has large overlap with Computational geometry that arose as a new field within the past thirty years. The term "discrete" fit well with the term "computational" as the geometry must be discretized in preparation for computations [Devadoss 11].

In our work we consider that discrete geometry covers both the old topics of discrete geometry and the new issues of computational geometry.

Research themes in discrete geometry cover the geometry processing pipeline that include studying transition mechanisms between physical (real), mathematical (abstract), and discrete (digital) representations of complex shapes [Botsch 10].

Alliez [Alliez 09] enumerates and structures the related algorithms throughout the following geometry processing pipeline:

(a) Acquisition. There exists a variety of geometric measurement devices based on different technologies such as contact probing, photogrammetry, laser scanning, structured lighting and tomography. The output of such devices is often composed of 2D or 3D point sets with in some cases color attributes and normal estimation.

(b) Registration. The output of shape acquisition devices is a series of point set corresponding to different acquisition viewpoints. A registration is required to align the data in a common coordinate frame. Many approaches employ rigid transforms and variants of Iterative Closest Point (ICP) techniques [Besl 92] [Rusinkiewicz 01] and new researches deal with non-rigid transforms [Li 08] for compensating calibration errors, noise, and irregular sampling.

(c) Reconstruction. The process of shape reconstruction is determining a surface that approximates an unknown shape from samples. This problem is ill-defined since many surfaces approximating the samples can be retrieved. Moreover, the point set can be characterized with variable density as well as noise and outliers due to the acquisition process. The main challenge here is to guarantee that the topology of the original surface is preserved, while sharp features and surface boundaries are reproduced accurately in the reconstructed surface [Chang 07]. The research literature is very rich and various techniques have been developed [Bolle 91] [Alexa 01] [Tamal 06]. The earliest techniques used NURBS or B-splines to fit and stitch together local surface patches but showed their limits when dealing with complex physical object, large data and noise. Voronoi-based methods provide a provable reconstruction guarantee. Implicit surface fitting techniques produce high-quality surfaces that accommodate sharp features and boundaries while moving least squares surfaces are robust to noise.

The terms “surface fitting” is in reference to two distinct classes of problems [Hoppe 92]: surface reconstruction and function reconstruction. The goal of function

reconstruction is to determine a function that "best" approximates a known domain surface. Function reconstruction methods can be used for surface reconstruction in special cases, where the surface to be reconstructed is known such as quadric surfaces [Zhang 11].

Fitting surfaces to measured data algorithms estimate surface parameters by minimizing the shortest distances between the surface and the measurement points. There has been much recent algorithmic development based on conventional best-fit technique, mainly Orthogonal Distance Fitting (ODF) [Ahn 04]. Three ODF techniques are of particular interest for coordinate metrology applications. The classical ODF L_2 norm (Euclidean distance), ODF L_1 norm which is robust regarding outliers and $L_{infinite}$ norm called also the Tchebychev norm or minimum norm which is related to the minimum-zone problem. In general two approaches are proposed to solve the problem of surface best-fit [Aranda 10]. The first approach considers the best-fit as a purely geometric problem and generates a Gaussian distribution of uncertainty. The second considers the best-fit as a statistical approach similar to regression analysis and focus on the representation of the real surface and not only the sampled surface.

(d) Analysis. The efficient analysis of shapes is important for modeling and processing. Estimation of normal, curvatures and other differential properties, as well as topology characterization and feature extraction are the main common analyses reported in the literature [Botsch 10].

(e) Processing. It refers to existing approaches, which modify a shape to best prepare it for subsequent operations along the pipeline [Alliez 09]. The main common operations are denoising, smoothing, fairing, reduction, idealization and preparation of quality meshes for simulation.

Other research themes as reported by Alliez are related to visualization and rendering [Alexa 03], editing [Botsch 05], transmission and storage [Alliez 05], protection and watermarking [Cayre 04] and searching and browsing [Funkhouser 05]. For further details on discrete geometry and other emerging topics, refer to [Gross 07].

2.3. Discrete geometry and GeoSpelling

GeoSpelling proposed by Ballu and Mathieu [Ballu 95] is used to describe both ideal and non-ideal geometric features. Indeed, it allows the expression of product specifications from function to verification with a common language. This model is based on geometrical operations which are applied not only to ideal features defined by CAD systems, but also to the non-ideal features which can represent a real part. These operations include partition, extraction, filtration, association, collection and construction items.

Discrete geometry research focuses on basic discrete geometrical objects, such as points, segments, triangles and other convex discrete shapes, and it is quite efficient to implement digital processing techniques. Therefore, discrete geometry theories and techniques are suitable for enhancing the data processing capabilities of GeoSpelling. Based on the standard [ISO 2005], a specification is defined as a condition on a characteristic defined from geometric features which are created from a skin model by different operations. The related concepts and algorithms are presented as blow.

2.3.1 Geometrical Features

There are two kinds of geometrical features defined in GeoSpelling: ideal features and non ideal features. These two kinds of features are related to the surface models (skin model). The features that are obtained from the nominal surface models defined by the CAD software can be approximately deemed as ideal features, while the non ideal features are derived form the skin model which integrated the imaginations of the real workpiece by the designers. In discrete geometry field, ideal features can be acquired by reconstruction and fitting operations, while ideal features can be describe by discrete geometrical features such as point, segment, polyline, mesh etc.

2.3.2 Operations

Operation defined in [ISO 2005] including partition, extraction, filtration, association, collection and construction. These operations play an important part in Geospelling [Mathieu 07], so in this section we want to find out the interrelated

techniques in discrete geometry.

(1) . Partition

The operation called partition is used to identify bounded features [ISO 2005]. In Discrete geometry, there are abundant contributions on this area. The majority of partition methods can be classified into three categories: edge-detection methods, region-growing method and hybrid methods [Woo 02].

Edge-detection method can be described that, after identified the boundary points, connect each point to a continue boundary and consider the area inside the boundary as the segmentation. Yang and Lee [Yang 99] introduced a partition technique for extraction edge points by estimating the local surface curvature, and the parametric quadric surface approximation method was applied. Ke and Fan [Ke 04] extracted the edge points by divided dimension grids according the density of points. Wu and Wang [Wu 07] identified edge points based on the differential geometry theory.

The edge-based methods always have two stages; one is edge detection stage, and the other is linking stage. This method finds boundaries between surfaces in point data firstly, and an edge-linking process followed, in which disjoint edge points are connected to form continuous edges. Then infer the surfaces as the areas inside the boundaries. Usually, detect edge points based on the behavior of some features such as normal or curvatures e.g. [Yang 99] [Wu 07]. The main problem to the edge-based technique is the points near sharp edges are unreliable, result in a relatively high sensitivity to occasional spurious data points.

The region-growing methods proceed with segmentation by detecting continuous surfaces that have homogeneity or similar geometrical properties. Ke and Chen [Ke 06] proposed a partition method based on the differential geometry and mathematical statistic theory. Jagannathan [Jagannathan 07] presented a graph-morphology-based 3D mesh segmentation algorithm to classify vertices into different categories based on their intensities of curvatures. Liu and Xion [Liu 08b] addressed the approach mainly based on the Gaussian map, the Clustering, the region Recognition and the region Rectification. Benkö and Várady [Benkö 04] studied how to improve the connections of partition regions more smoothly.

The region-growing methods detect continuous surfaces firstly, and then can get the homogeneity or similar geometrical segmentation, e.g. [Ke 06] and [Jagannathan

07]. For this method, it always selects a series of seed points firstly, based on a certain threshold, the seed points growing to a related range, and then can segment the cloud points. The advantage to face-based techniques is that it work on a larger number of points to reduce the risk of sensitivity to occasional spurious data points and can identified which points belong to which surface.

Hybrid methods have been developed by combining the edge-based and region-based methods together to yield an approach that can overcome the limitations involved in the above two kinds of methods. Woo [Woo 02] proposed segmentation that uses an octree-based 3D-grid splitting process that uses the iterative subdivision of cells based on the normal values of points, and the region-growing process to merge the divided cells into several groups.

Based on the hybrid methods, Zhao [Zhao 10] proposed a segmentation method based on shape index and curvedness. This method is based on two assumptions: the initial discrete shapes can be broken down into various components, and these segmented components can be represented by mathematic functions with degree no higher than 3. Based on the shape index and curvedness information, the local surface type can be determined, and the edge points are detected with the high curvature values. After clustering the edge points and merging the local surfaces with the same type, the segmentation result is obtained. The main contribution of this method is that it can recognize and extract the regions surrounded by real defined boundaries, which can be sharp edges, smooth edges, inflexion edges, etc.

To sum up, region-growing methods have less influence by noise than edge-based methods, but it is quite sensitive to threshold and the selection of seed point is also a technical problem. The hybrid methods combining the virtue of edge-based and region-based methods can overcome the two kinds of limitations.

(2) . Extraction

The operation called extraction is used to identify a finite number of points from a feature, with specific rules [ISO 2005]. In discrete geometry terms, the technique of sampling can deal with this kind of problem. It links two aspects, one is the extraction location and the other is the extraction size. There is flock of the contributions, for geometrical specification Zhang et al. [Zhang 07] has proposed a method, and for CMM detection Kim [Kim 00] et al. introduced the corresponding approach.

Zhang [Zhang 07] classified the extraction strategy into four kinds: grid extraction, stratified extraction, special curve and point extraction, and depending on the relationship between invariant class and extraction strategy to give the determination. For extraction size, it is based both on the sine wave function and Nyquist sampling theory and the result of point size for each cut-off wavelength is 7.

In CMM measurement, four kinds of extraction methods were investigated: Hammersley sequence sampling, Halton-Zaremba sequence, Aligned systematic sampling, and Systematic random sampling [Kim 00]. British Standard BS7172 recommended that the sample size can vary from 5~15 points for various individual shapes and features [Zhang 00]. Zhang et al. [Zhang 96] introduced neural network approach to determine the sample size for inspection of holes using process method, size of hole and tolerance band as factors. Li [Li 98] adopted rate-distortion theory to determine the number of measuring.

(3) . Filtration

The filtration operation in GPS [ISO 2005] is the operation that used to distinguish between roughness, waviness, structure and form etc.

While in discrete geometry, filtration mainly concerned about outlier moving based on sample points. Outlier is the erroneous point that outside the surface and should be removed. There are some methods can achieve this object, such as plane fit criterion, mini-ball criterion and nearest-neighbor reciprocity criterion [Gross 07].

(a) Plane fit criterion.

The plane fit criterion considers a plane H that minimizes the squared distances to a sample point p's neighbors, it can be described by formula 2-1 (see figure 2-10):

$$\min_{H} \sum_{q \in N_k(p)} dist(p, H)^2 \tag{2-1}$$

Let d be the distance of p to H, and $\overline{d}$ the mean distance of points from $N_k(p)$ to H. The plane fit criterion is defined as formula 2-2

$$x_{pl}(p) = \frac{d}{d + \overline{d}} \tag{2-2}$$

Normalization by $\overline{d}$ relates d to possible noise and surface deviations. Figure 2-10

illustrates that the plane fit criterion compares p's distance d to a least squares plane H with the average distance of its neighbors to H, and p's k-neighbors are denoted in blue.

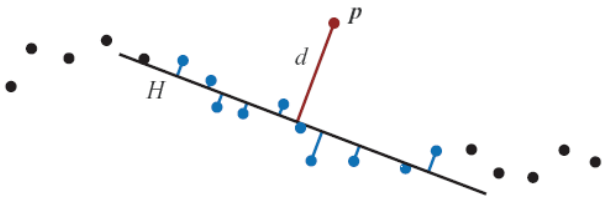


Figure 2-10: The plane fit criterion

(b) Mini-ball criterion.

A point comparatively distant to the cluster built by its k-nearest neighbors is likely to be an outlier. This observation leads to this criterion.

For each point p consider the smallest enclosing sphere S around $N_k(p)$ (see figure 2-11). S can be seen as an approximation of the k-nearest neighbor cluster. Comparing p's distance d to the center of S with the sphere's diameter yields a measure for p's likelihood to be an outlier. The mini-ball criterion is defined as formula 2-3

$$x_{pl}(p) = \frac{d}{d + 2r/\sqrt{k}} \tag{2-3}$$

Normalization by $\sqrt{k}$ compensates for the diameter's increase with increasing number of k-nearest neighbors at the object surface. The principle of the Mini-ball criterion can be described by figure 2-11. The mini-ball S approximates the cluster of p's neighbors, and the criterion compares p's distance to S with the diameter of the sphere.

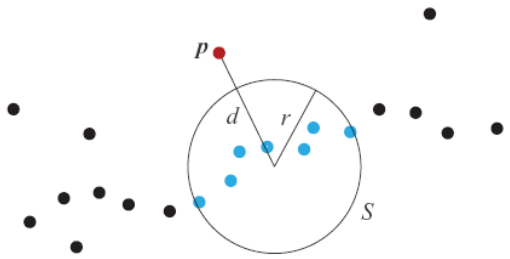


Figure 2-11: The Mini-ball criterion

(c) Nearest-neighbor reciprocity criterion.

This criterion is based on the following observation: potential outliers draw their k-nearest neighbors from a larger vicinity than points in a well-sampled environment. In particular, a valid point sample q may be in the k-nearest neighbor of an outlier, but the outlier will most likely not be part of q's k-nearest neighbors.

This relationship can be expressed by means of a directed graph G of k-nearest neighbor relationships (see figure 2-12). Outliers are assumed to have a high number of unidirectional exitant edges (i.e., asymmetric neighbor relationships). Consequently the criterion considers the ratio between unidirectional and bidirectional exitant edges in G. The classifier is expressed as formula 2-4

$$x_{bi}(p) = \frac{\left\|N_{k,uni}(p)\right\|}{\left\|N_{k,bi}(p)\right\| + \left\|N_{k,uni}(p)\right\|} = \frac{\left\|N_{k,uni}(p)\right\|}{k} \tag{2-4}$$

The final outlier classification is computed using weights w1,…,w3, $\sum_i w_i = 1$, interactively defined by the user as formula 2-5

$$x(p) = w_1 x_{pl}(p) + w_2 x_{mb}(p) + w_3 x_{bi}(p) \tag{2-5}$$

As all outlier criteria are based on the k-nearest neighbor graph, x_{pl}, x_{mb}, and x_{bi} are computed once and cached during the computation of x. The principle of Nearest-neighbor graph method can be describe by figure 2-12. Depicted are the five nearest-neighbor relations for p and its five neighbors $q_0,\ldots,q_4$, since that only q_2 shares a reciprocal neighbor relationship with p.

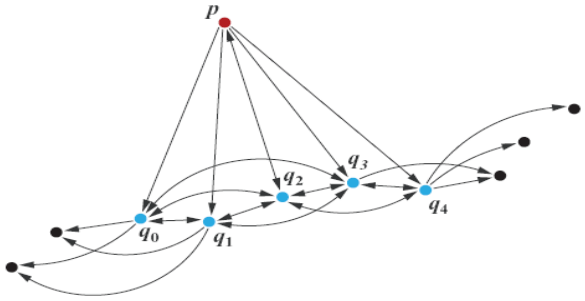


Figure 2-12: The Nearest-neighbor graph criterion

(4) . Association

The methods of association defined in GPS can be classified as Least Squares, Minimum Zone, One Side Chebyschev, Maximum inscribed and Minimum Circumscribed, and all of them can be illuminated by a uniform formula 2-6

$$L_{p-norm} = [\frac{1}{n}(\sum_{i=1}^{n}|r_i|^p)^{\frac{1}{p}}]_{n\to\infty} \quad (2\text{-}6)$$

Where i is the index of point on the non-idea feature, p is the index of the function, n is the number of points of the non-ideal feature, r_i is the residuals between non-ideal feature and ideal feature (see figure 2-13).

In discrete geometry, we evaluate the association operation instead of continuous features are discrete ones. Therefore, the three of common used methods of association in discrete geometry will be introduced in this section. There are Least Squares (LS), Weighted Least Squares (WLS) and Moving Least Squares (MLS) [Nealen 92]. The Least Squares method has be adopted in GPS, while the other two are something new.

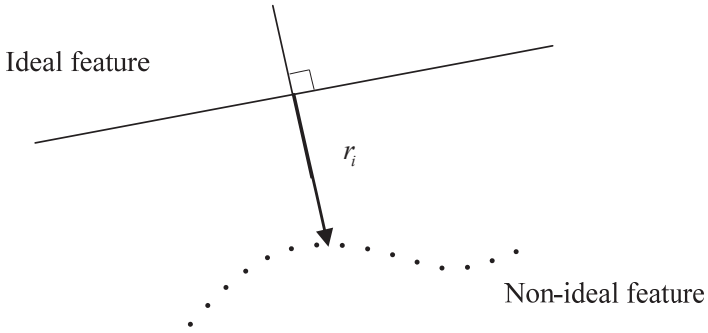


Figure 2-13: Lp norm of association

(a) Least Squares method. Given N points located at positions x_i in Rd where $i \in [1 \cdots N]$. In order to obtain a globally defined function $f(x)$ that approximates the given scalar values f_i at point x_i in the least-squares sense with the error functional $J_{LS} = \sum_i \|f(x_i) - f_i\|^2$. Thus the problem can be described as formula 2-7

$$\min_{f \in \prod_m^d} \sum_i \|f(x_i) - f_i\|^2 \quad (2\text{-}7)$$

Where f is taken from $\prod_m^d$, the space of polynomials of total degree m in d spatial

dimensions (see figure 2-14).

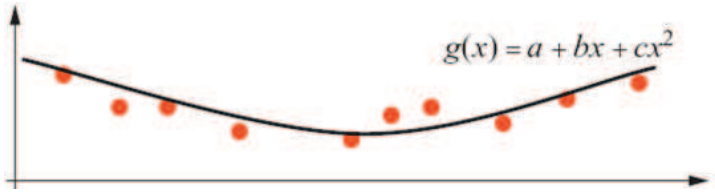


Figure 2-14: Least Squares method

(b) Weighted Least Squares method. Use the error functional $J_{WLS} = \sum_i \theta(\|\bar{x} - x_i\|)\|f(x_i) - f_i\|^2$ for a fixed point $\bar{x} \in R^d$, in the weighted least squares formulation. Then the problem can be described as formula 2-8

$$\min_{f \in \Pi_m^d} \sum_i \theta(\|\bar{x} - x_i\|)\|f(x_i) - f_i\|^2 \qquad (2\text{-}8)$$

Similar to above, only that now the error is weighted by $\theta(d)$ where d_i are the Euclidian distances between $\bar{x}$ and the positions of data points x_i.

(c) Moving Least Squares method. The MLS method was proposed by Lancaster and Salkauskas [Lancaster 81] for smoothing and interpolating data. The idea is to start with a weighted least squares formulation for an arbitrary fixed point in R^d, and then move this point over the entire parameter domain, where a weighted least squares fit is computed and evaluated for each point individually. It can be shown that the global function $f(x)$, obtained from a set of local functions is continuously differentiable if and only if the weighting function is continuously differentiable. We can see (figure 2-15): in each location x a polynomial is computed using the least squares method, however, weighting the influence of the data points based on their distance. The value of this polynomial at x yields the functional approximation (left); the set of locally approximated function values, together, forms the approximated curve (right).

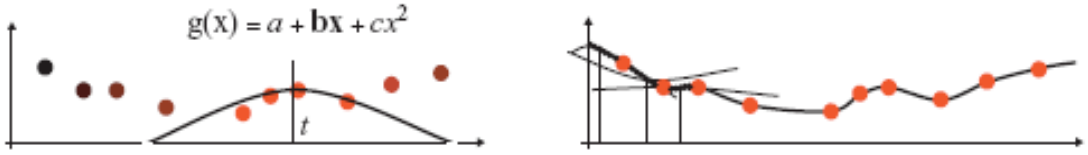
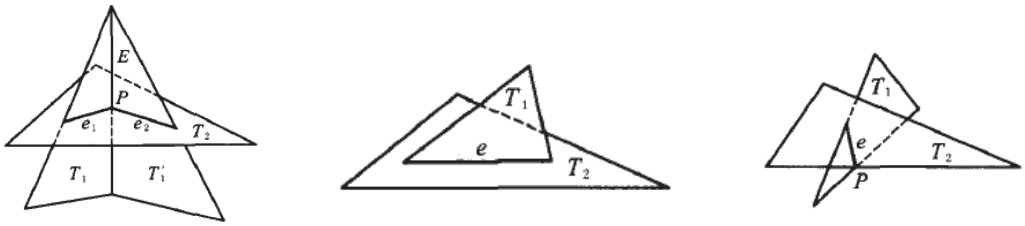


Figure 2-15: Moving least squares

(5) . Collection and Construction

Construction and collection operation in GPS can be depicted by Boolean operation technique used in discrete geometry. Collection should be the union operation of Boolean operation, while the construction is the intersection operation in Boolean operation. Boolean operation can be classified into mesh-based and point-based methods, and the later one can be subdivided into MLS (Moving Least Squares), RBF (Radial Basis Function) and MPU (Multi-level Partition of Unity Implicit) categories based on the method to construct a surface.

(a) Mesh-based method. Mesh-based method can deal with the Boolean operation between mesh and mesh, Bi [Bi 08] proposed a method which based on the orientability of manifold meshes and the topological relationship of the meshes. The process can be mainly divided into two steps. First is to estimate the intersections (see figure 2-16), and second is to divide the polygons into triangles (see figure 2-17).



(a) intersection is a segment (b) intersection is one edge (c) intersection is a point

Figure 2-16: Intersection cases

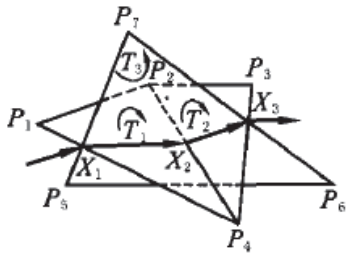


Figure 2-17: Dividing the polygon into triangles

(b) Point-based method. The common used methods of point-based Boolean operation can be classified into MLS, RBF and MPU three categories, and this sort is relay on the approach that reconstruct a implicit surface.

The process of MLS Boolean operation described as follow [Gross 07]. Before

going into detail, let's have some definitions. Given two point set P_1 and P_2, since Boolean operations typically produce sharp creases at the intersection of the two surfaces S_1 and S_2, P_3 consists of two subsets $Q_1 \subseteq P_1$ and $Q_2 \subseteq P_2$ plus a set of newly generated sample points that explicitly represent the intersection curves. Thus, in order to perform a Boolean operation for point-sampled geometry, the following techniques are required: a classification predicate to determine the two sets Q_1 and Q_2, an algorithm to find samples on the intersection curve and a rendering method that allows the user to display crisp features curves using point primitives.

For classification: the goal of the classification stage is to determine which points of P_1 are inside or outside the volume enclosed by the surface S_2 and vice versa. For this purpose a classification predicate Ω_p is defined as formula 2-9

$$\Omega_p(x) = \begin{cases} 1 & x \in V \\ 0 & x \notin V \end{cases} \tag{2-9}$$

Where V is the volume bounded by the MLS surface S defined by the point cloud P.

Given the classification predicate Ω, the subsets Q_1 and Q_2 can be computed as shown in table 2-1.

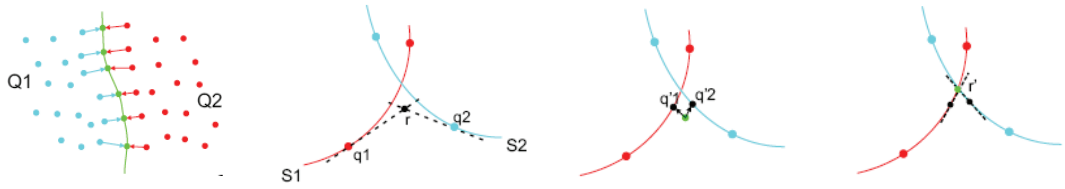
	Q_1	Q_2
$\mathcal{S}_1 \cup \mathcal{S}_2$	$\{\mathbf{p} \in \mathcal{P}_1 \mid \Omega_{\mathcal{P}_2}(\mathbf{p}) = 0\}$	$\{\mathbf{p} \in \mathcal{P}_2 \mid \Omega_{\mathcal{P}_1}(\mathbf{p}) = 0\}$
$\mathcal{S}_1 \cap \mathcal{S}_2$	$\{\mathbf{p} \in \mathcal{P}_1 \mid \Omega_{\mathcal{P}_2}(\mathbf{p}) = 1\}$	$\{\mathbf{p} \in \mathcal{P}_2 \mid \Omega_{\mathcal{P}_1}(\mathbf{p}) = 1\}$
$\mathcal{S}_1 - \mathcal{S}_2$	$\{\mathbf{p} \in \mathcal{P}_1 \mid \Omega_{\mathcal{P}_2}(\mathbf{p}) = 0\}$	$\{\mathbf{p} \in \mathcal{P}_2 \mid \Omega_{\mathcal{P}_1}(\mathbf{p}) = 1\}$
$\mathcal{S}_2 - \mathcal{S}_1$	$\{\mathbf{p} \in \mathcal{P}_1 \mid \Omega_{\mathcal{P}_2}(\mathbf{p}) = 1\}$	$\{\mathbf{p} \in \mathcal{P}_2 \mid \Omega_{\mathcal{P}_1}(\mathbf{p}) = 0\}$

Table 2-1: Boolean operations

For intersection curves: taking the union of Q_1 and Q_2 can not produce a point cloud that accurately describes the surface S_3, since the intersection curve of the two MLS surfaces S_1 and S_2 is not represented adequately. Therefore, a set of sample points that lie on the intersection curve is explicitly computed and added to $Q_1 \cup Q_2$, to obtain

the point cloud P_3.

From all closest pairs ($q_1 \in Q_1, q_2 \in Q_2$) of these points a new sample q on the intersection curve is computed using a Newton type iteration. This is done as follows: Let r be the point on the intersection line of the two tangent planes of q_1 and q_2 that is closest to both points (i.e., that minimizes the distance $\|r-q_1\|+\|r-q_2\|$). The point r is the first approximation of q and can now be projected onto S_1 and S_2 to obtain two new starting points q_1' and q_2' for the iteration. This procedure can be repeated iteratively until the points q_1 and q_2 converge to a point q on the intersection curve (figure 2-18).



(a) closest point pairs (b) first estimate (c) re-projection (d) second estimate r'

Figure 2-18: Finding the intersection curve of two MLS surfaces

2.3.3 Characteristics

In [ISO 2005] characteristics are defined using some instances: ideal features also called intrinsic characteristics, or between ideal features and called situation characteristics, or between ideal and non-ideal features and also called situation characteristics.

Correspondingly, in discrete geometry characteristics maybe can be defined as: either on ideal features called intrinsic characteristics, or between ideal features called situation characteristics, or between ideal and non-ideal features called situation characteristics, or between non-ideal feature and non-ideal feature and similarly called situation characteristic.

(1). Intrinsic characteristics of ideal features

The intrinsic characteristics are functions of features, so they are denoted as

functions of these features [ISO 2005], and it is the same as in discrete geometry.

(2). Situation characteristics between ideal features

The definition of ideal feature in discrete geometry has been discussed in section 2.1, and the type of ideal features in discrete geometry can be divided into point, straight line and plane as the same in GPS. Therefore, the situation characteristics, including location and orientation characteristics, both in GPS and discrete geometry can be described as what has been defined in [ISO 2005] (table 2-2).

Location	Orientation
point – point distance	straight line – straight line angle
point – straight line distance	Straight line – plane angle
point – plane distance	plane – plane angle
straight line – straight line distance	
plane – plane distance	

Table 2-2: Situation characteristics between ideal features

(3). Situation characteristics between non ideal and ideal features

Both in GPS and discrete geometry, these situation characteristics are only distances, and used for association operation. The difference is that, in GPS the non-ideal features is continuous, but in discrete geometry the non-ideal feature represent in discrete features such as point, segment or triangles. The contrast can be illustrated in figure 2-19.

(a) non-ideal feature is continuous (b) non-ideal feature is discrete

Figure 2-19: The difference of distance between ideal and non-ideal feature

(4). Situation characteristics between non-ideal and non-ideal features

In GPS, there is not exist this kind of situation characteristics, whereas, in discrete geometry it should be take into account, and these situation characteristics are also only distance. Corresponding to the location characteristic between ideal feature (see table 2-2), the location characteristic between non-ideal and non-ideal feature should be present as table 2-3 and the mathematical notation as table2-4. The most used non-ideal features in discrete geometry are denoted by two letters, see table 2-5.

Point-point distance
Point-segment distance
Point-triangle distance
Segment-segment distance
Segment-triangle distance
Triangle-triangle distance

Table 2-3: Location characteristic

Distance (point-point)
Distance (point-segment)
Distance (point-triangle)
Distance (segment-segment)
Distance (segment-triangle)
Distance (triangle-triangle)

Table 2-4: Mathematical notation

Type	Symbol
Point	PT
Segment	SE
Triangle	TR

Table 2-5: symbols

The calculation of each kind of location characteristics between non-ideal and non-ideal features in discrete geometry will be discussed as follow [Schneider 03].

(a) Point to point distance. As we all know, the fundamental to any distance algorithm is the distance between two points, so we will discussed the distance in 3D case firstly. Let two points are $X = (x_0, x_1, x_2)$ and $Y = (y_0, y_1, y_2)$. The sketch map and formula are provided, see table 2-6.

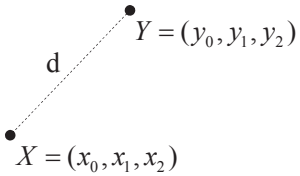
Sketch map	Formula
	$\text{Distance}^2(X,Y) = \|X-Y\|^2 = (x_0-y_0)^2 + (x_1-y_1)^2 + (x_2-y_2)^2$ *Note: the square root calculation is much more time consuming than squared one, so all the distance discussed below will be squared calculation.*

Table 2-6: Point to point distance

(b) Point to segment distance. Given a point Y and a line L parameterized as $X(t) = P + t\vec{d}$, the closest point on the line to Y is the projection $X(\bar{t})$ of Y onto the line for some parameter value $\bar{t}$. The distance between a point and a segment is more

like this case. However, the projection of Y onto the line might not lie on the line segment S. The projection could lie behind the initial point or ahead of the final point of the segment. Table 2-7 shows the three possibilities. The direction vector is $\vec{d} = P_1 - P_0$, the difference of end points for the line segment. The parameter interval is [0, 1]. The value $\bar{t}$ is just as for the line, but for the segment it is tested against the parameter interval [0, 1].

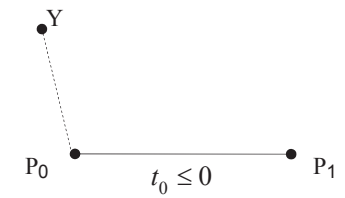
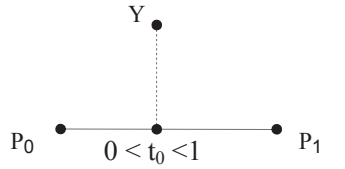
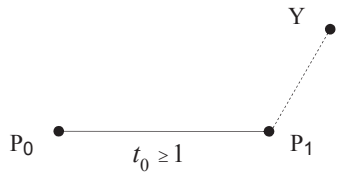
Sketch map	Formula
	$\text{Distance}^2(Y,S) = \|Y - P_0\|^2$ *Note: P_0 closest to Y.*
	$\text{Distance}^2(Y,S) = \|Y - (P + t_0\vec{d})\|^2$ *Note: $X(\bar{t})$ closest to Y.*
	$\text{Distance}^2(Y,S) = \|Y - P_1\|^2$ *Note: P_1 closest to Y.*

Table 2-7: Closest point on a segment to a given point. Where $t_0 = \vec{d} \cdot (Y - P_0) / \left\|\vec{d}\right\|^2$

(c) Point to triangle distance. Given a triangle T with vertices {V0, V1, V2}, described by parametrical expression as formula 2-10

$$T(s,t) = B + s\vec{e}_0 + t\vec{e}_1 \tag{2-10}$$

For $(s,t) \in D = \{(s,t) : s \in [0,1], t \in [0,1], s + t \le 1\}$, $B = V_0$, $\vec{e}_0 = V_1 - V_0$, and $\vec{e}_1 = V_2 - V_1$. The distance between a point P and the triangle T can be deemed to the minimum distance, which is computed by locating the values $(\bar{s},\bar{t}) \in D$ corresponding to the point P`, the closest point on the triangle to P. The squared distance from any point on the triangle to P is as formula 2-11

$$\text{Distance}^2(s,t) = Q(s,t) = \left\| T(s,t) - P \right\|^2 \tag{2-11}$$

For $(s,t) \in D$, this formula also can be denoted as formula 2-12

$$\text{Distance}^2(s,t) = Q(s,t) = as^2 + 2bst + ct^2 + 2ds + 2et + f \tag{2-12}$$

Where: $a = \vec{e}_0 \cdot \vec{e}_0$, $b = \vec{e}_0 \cdot \vec{e}_1$, $c = \vec{e}_1 \cdot \vec{e}_1$, $d = \vec{e}_0 \cdot (B - P)$, $e = -\vec{e}_1 \cdot (B - P)$, $f = (B - P) \cdot (B - P)$.

In calculus terms, the goal is to minimize $Q(s,t)$ over the domain D. Since $Q(s,t)$ is a continuously differentiable function, the minimum occurs either at an interior point of D where the gradient $\nabla Q = 2(as + bt + d, bs + ct + e) = (0,0)$ (that is, inside the triangle) or at a point on the boundary of D.

The gradient of Q is zero only when $\bar{s} = \frac{be - cd}{ac - b^2}$ and $\bar{t} = \frac{bd - ad}{ac - b^2}$. If $(\bar{s}, \bar{t}) \in D$, then the minimum of Q is obtained. Otherwise, the minimum value must occur on the boundary of the triangle. To find the correct boundary, consider figure 2-20.

The center triangle labeled 0 is the domain of Q, that is $(s,t) \in D$. If $(\bar{s}, \bar{t})$ is in region 0, the P`, the point on the triangle closest to P, is on the interior of the triangle. The three dimensional illustration is be provided in figure 2-21. The Q with quadratic function represents as paraboloid, and $\nabla Q(\bar{s}, \bar{t}) = (0,0)$ occurs at $\bar{s}, \bar{t}$ in region 0 case.

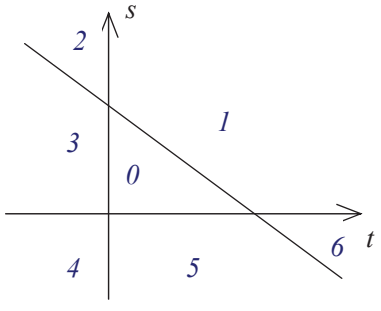


Figure 2-20: triangle domain D

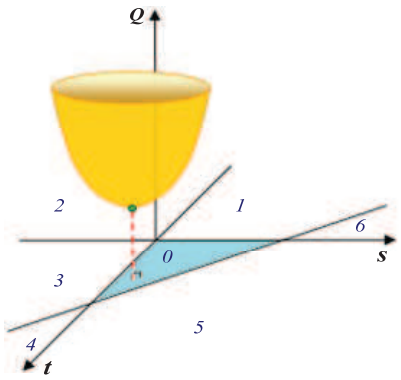


Figure 2-21: when $\nabla Q(\bar{s}, \bar{t}) = (0,0)$, $\bar{s}, \bar{t}$ in region 0

If $(\bar{s},\bar{t}) \notin D$, that means the minimum occurs on the boundary of the triangle, take $(\bar{s},\bar{t})$ is in region 1 for example. The level curves of Q are those in the st-plane for which Q is a constant. Since the graph of Q is a paraboloid, the level curves are ellipses (see figure 2-22). At the point at where $\nabla Q = (0,0)$, the level curve degenerates to a single point $(\bar{s},\bar{t})$. The globe minimum of Q occurs there, call it Vmin. As the level values V increase from Vmin, the corresponding ellipses are increasingly further away from $(\bar{s},\bar{t})$. There is a smallest level value V0 for which the corresponding ellipse (implicitly defined by $Q = V_0$) just touches the triangle domain edge $s+t=1$ at a value $s = s_0 \in [0,1]$, $t_0 = 1 - s_0$. For level values $V < V_0$, the corresponding ellipses do not intersect D. For level $V > V_0$, portions of D lie inside the corresponding ellipses. In particularly, any points of intersection of such an ellipse with the edge must have a level value $V > V_0$. Therefore, $Q(s,1-s) > Q(s_0,t_0)$ for $s \in [0,1]$ and $s \neq s_0$. The point (s_0,t_0) provides the minimum squared distance between P and the triangle. When found the value of (s_0,t_0), then can obtain the Q. The triangle point is an edge point. Figure 2-23 illustrates the idea by showing various level curves.

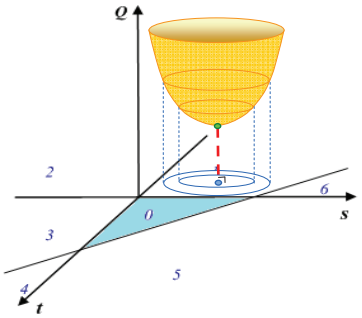


Figure 2-22: Various of level curves

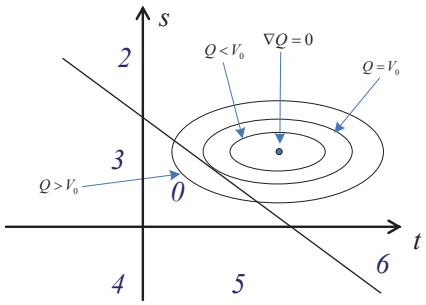


Figure 2-23: When $\nabla Q(\bar{s},\bar{t}) = (0,0)$, $\bar{s},\bar{t}$ in region 0

(d) Segment to segment distance. To find the distance between two segments, first we find it between two lines. It equivalent to computing s and t such that the length of vector $\vec{v} = Q_1 - Q_0$ is a minimum, and can be rewrite as formula 2-13

$$\left\|\vec{v}\right\|^2 = \vec{v}\cdot\vec{v} = ((P_0 - P_1) + s\vec{d}_0 - t\vec{d}_1)\cdot((P_0 - P_1) + s\vec{d}_0 - t\vec{d}_1)) \quad (2\text{-}13)$$

This is a quadratic function in s and t, that is, it is a function $f(s,t)$ whose shape is a paraboloid. For the case of lines, the domain of s and t is unrestricted, and the solution corresponds to the point where f is minimized (that is, the bottom of the paraboloid).

However, if the linear components are both segment, then the domain of s and t is restricted, which can be expressed by $0 \le s \le 1$ and $0 \le t \le 1$ (see figure 2-24).

In general, the global minimum of the quadratic function may not be within the restricted domain. In such case, the minimum will be at some point along one of the boundary edges of the domain. The method to solve this problem is more like the algorithm that classifies the location of (s_c, t_c), and then we can determine the closest points and their distance.

Analogous to the approach of categorizing the region in which (s_c, t_c) lies, this approach considers which edges of the bounded domain are visible to (s_c, t_c).For example, in the case of segment and segment distance, the domain is restricted to $[0,1]\times[0,1]$. Figure 2-25 shows two of the possible visibility conditions for a solution: on the left, only the boundary $t = 1$ is visible, and on the right, both $t = 1$ and $s = 1$ are visible.

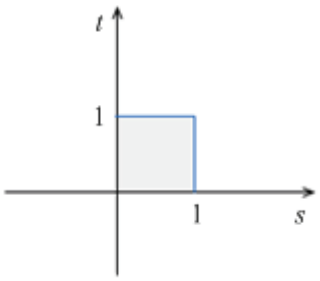
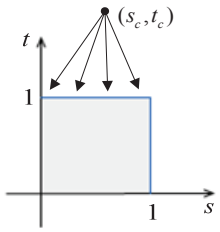
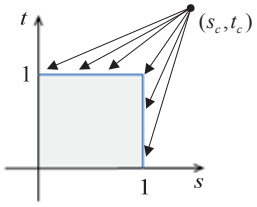


Figure 2-24: Domain for segment



(a) $t = 1$ visible



(b) $t = 1$ and $s = 1$ visible

Figure 2-25: Definition of visibility of domain boundary

By simply comparing the values of s_c and t_c, it can be easily determine which domain boundary edges are visible. For each visible boundary edge, compute the point on it closest to (s_c, t_c). If only one boundary edge is visible, then that closest

solution point will be in the range [0, 1], otherwise, the closest solution point will be outside that range, and thus need to check the other visible edge. The basic idea is to first compute the closest points of the infinite lines on which the segments lie, that is s_c and t_c. To compute the distance between segments, then the domains of s and t are restricted. So we need to find the points that minimize the squared distance function over the restricted domain, and these points will have parameter values that correspond to points on the boundary.

It can easily compute the closest points on the boundary edges by employing a little calculus. For the case of the edge $s = 0$, it has $\left\|\vec{v}\right\|^2 = (\vec{u} - t\vec{d}_1) \cdot (\vec{u} - t\vec{d}_1)$, where $\vec{u} = P_0 - P_1$. The derivative of this with respect to t gives as formula 2-14

$$0 = \frac{d}{dt}\left\|\vec{v}\right\|^2 = -2\vec{d}_1 \cdot (\vec{u} - t\vec{d}_1) \tag{2-14}$$

Then the minimum can be calculated as formula 2-15

$$t' = \frac{\vec{d}_1 \cdot \vec{u}}{\vec{d}_1 \cdot \vec{d}_1} \tag{2-15}$$

If $0 \le t' \le 1$, then this is the actual solution, otherwise, the actual solution is 1 if $t' > 1$ or $t' < 0$.

In the case of the edge $t = 0$ is exactly analogous. In this case, it is $\left\|\vec{v}\right\|^2 = (-\vec{u} - s\vec{d}_0) \cdot (-\vec{u} - s\vec{d}_0)$. The derivative of this with respect to s as formula 2-16

$$0 = \frac{d}{ds}\left\|\vec{v}\right\|^2 = -2\vec{d}_0 \cdot (-\vec{u} - s\vec{d}_0) \tag{2-16}$$

Then the minimum can be calculated as formula 2-17

$$s' = \frac{-\vec{d}_0 \cdot \vec{u}}{\vec{d}_0 \cdot \vec{d}_0} \tag{2-17}$$

If $0 \le s' \le 1$, then this is the actual solution, and otherwise, the actual solution is 1 if $s' > 1$ or 0 if $s' < 0$.

In the case of edge $s = 1$, it has $\left\|\vec{v}\right\|^2 = (\vec{u} + \vec{d}_0 - t\vec{d}_1) \cdot (\vec{u} + \vec{d}_0 - t\vec{d}_1)$. Taking the derivative with respect to t, then the deviation can be calculated as formula 2-18

$$0 = \frac{d}{dt}\left\|\vec{v}\right\|^2 = -2\vec{d}_1(\vec{u} + \vec{d}_0 - t\vec{d}_1) \tag{2-18}$$

Then the minimum can be calculated as formula 2-19

$$t' = \frac{\vec{d}_1 \cdot \vec{u} + \vec{d}_0 \cdot \vec{d}_1}{\vec{d}_1 \cdot \vec{d}_1} \tag{2-19}$$

If $0 \le t' \le 1$, then this is the actual solution, otherwise, the actual solution is 1 if $t' > 1$ or 0 if $t' < 0$.

In the case of edge $t = 1$, it has $\left\|\vec{v}\right\|^2 = (-\vec{u} + \vec{d}_1 - s\vec{d}_0) \cdot (-\vec{u} + \vec{d}_1 - s\vec{d}_0)$. Taking the derivative with respect to s, then the deviation can be calculated as formula 2-20

$$0 = \frac{d}{ds}\left\|\vec{v}\right\|^2 = -2\vec{d}_0(-\vec{u} - s\vec{d}_0 + \vec{d}_1) \tag{2-20}$$

Then the minimum can be calculated as formula 2-21

$$s' = \frac{-\vec{d}_0 \cdot \vec{u} + \vec{d}_1 \cdot \vec{d}_0}{\vec{d}_0 \cdot \vec{d}_0} \tag{2-21}$$

If $0 \le s' \le 1$, then this is the actual solution, and otherwise, the actual solution is 1 if $s' > 1$ or 0 if $s' < 0$.

Figure 2-26 made the above concept more clearly, and the figures are schematic, that means the two linear components are not necessary be perpendicular.

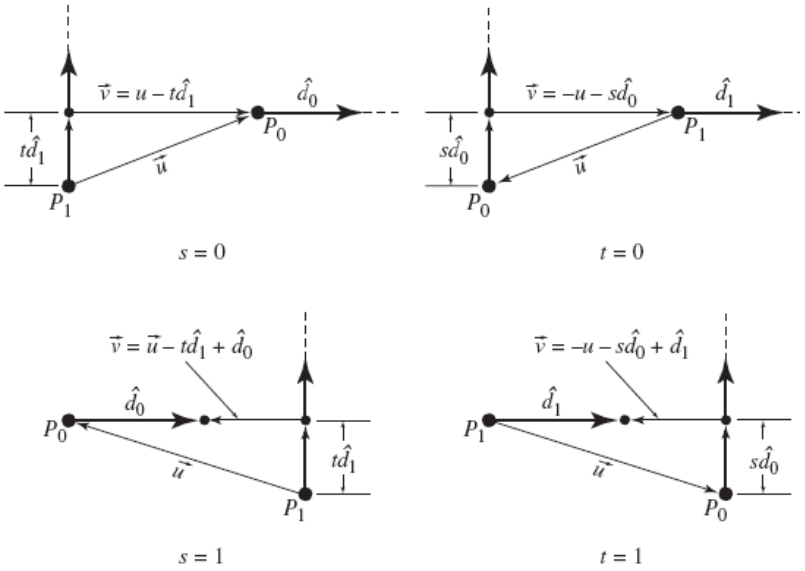


Figure 2-26: Cases for the four edges of the domain

Figure 2-27 shows two segments, which we want to find the distance between, and the restricted domain for the solution. In this case, the domain of the solution is

restricted to $[0,1]\times[0,1]$, the domain is bounded on all four sides: $s=0$, $s=1$, $t=0$ and $t=1$. If either s_c or t_c lies outside this region, then have to find the point on the boundary edge of the domain that is closest to the solution point (s_c,t_c). The domain is bounded on all four sides, but as discussed above, it only to find the boundary point on at most two of the edges.

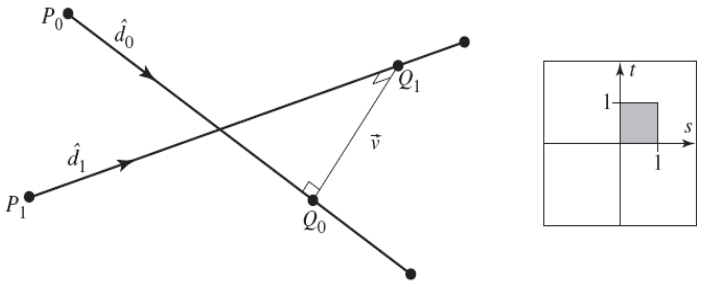


Figure 2-27: Distance between two segments

(e) Segment to triangle distance. To compute the distance between a segment and a triangle, as show in figure 2-28, the segment defined by a point and a direction vector, and can be expressed as formula 2-22

$$L(t) = P + t\vec{d} \tag{2-22}$$

Where $0 \le t \le 1$, the point P_0 and P_1 are the end points of the segment, then $\vec{d} = P_1 - P_0$. Triangles are represented as three vertices V_0, V_1 and V_2. For deal with this problem, use the parametric representation as formula 2-23

$$T(u,v) = V + u\vec{e}_0 + v\vec{e}_1 \tag{2-23}$$

Where V is a vertex of the triangle, and $\vec{e}_0 = V_1 - V_0$ and $\vec{e}_1 = V_2 - V_0$. Any point in the triangle can then be described in terms of the two parameters u and v, with $u \le 0$, $v \le 1$ and $u + v \le 1$ (see figure 2-29). Computing the distance between the segment and the triangle means that find the values of t, u, and v that minimize the squared distance can be calculated as formula 2-24

$$\text{Distance}^2(u,v,t) = Q(u,v,t) = \left\| T(u,v) - L(t) \right\|^2 \tag{2-24}$$

Expanding terms and multiplying, can get formula 2-25

$$
\begin{aligned}
Q(u,v,t) &= \left\| T(u,v) - L(t) \right\|^2 \\
&= (\vec{e}_0 \cdot \vec{e}_0)u^2 + (\vec{e}_1 \cdot \vec{e}_1)v^2 + (\vec{d} \cdot \vec{d})t^2 \\
&\quad + 2(\vec{e}_0 \cdot \vec{e}_1)uv + 2(-\vec{e}_0 \cdot \vec{d})ut + 2(-\vec{e}_1 \cdot \vec{d})vt \\
&\quad + 2(\vec{e}_0 \cdot (V-P))u + 2(\vec{e}_1 \cdot (V-P))v + (-\vec{d} \cdot (V-P))t \\
&\quad + (V-P) \cdot (V-P)
\end{aligned}
\tag{2-25}
$$

Where, $a_{00} = \vec{e}_0 \cdot \vec{e}_0$, $a_{11} = \vec{e}_1 \cdot \vec{e}_1$, $a_{22} = \vec{d} \cdot \vec{d}$, $a_{01} = \vec{e}_0 \cdot \vec{e}_1$, $a_{02} = -\vec{e}_0 \cdot \vec{d}$, $a_{12} = -\vec{e}_1 \cdot \vec{d}$, $b_0 = \vec{e}_0 \cdot (V-P)$ $b_1 = \vec{e}_1 \cdot (V-P)$, $b_2 = -\vec{d} \cdot (V-P)$, $c = (V-P) \cdot (V-P)$.

As the solution to this consists of three values (u_0, v_0, t_0), then the solution space can be deemed as a three spatial dimensions, which is the domain of a finite triangular prism (figure 2-30).

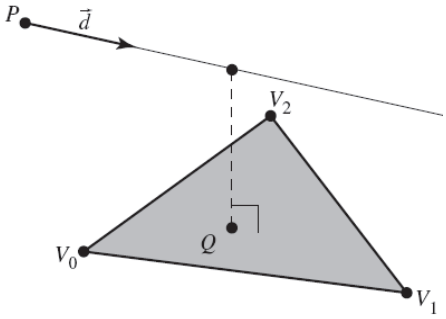


Figure 2-28: Distance between a segment and a triangle

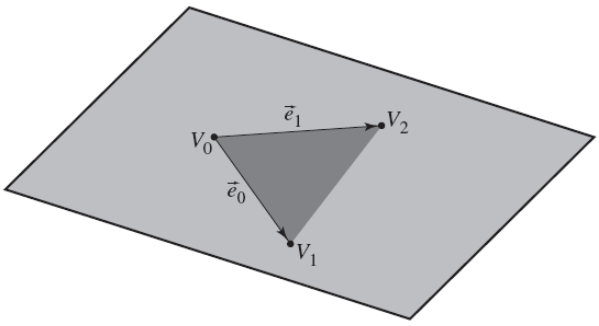


Figure 2-29: Description of a point in the triangle

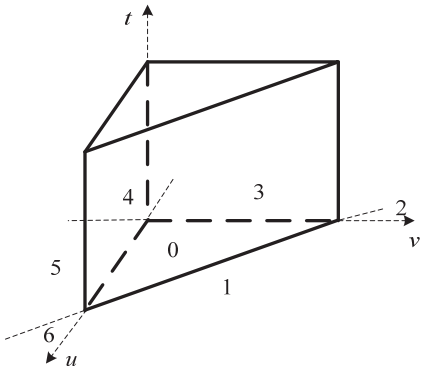


Figure 2-30: Domain of a triangular prism

If the configuration of the triangle and linear component are such that the point on the plane containing the triangle that is nearest to the linear component actually lies within the triangle, and the point on the linear segment nearest the triangle is within the bounds of the linear segment, then the solution $(\overline{u},\overline{v},\overline{t})$ will lie within region 0 of the solution domain's triangular prism, and the distance between the linear segment and the triangle will be $\|T(\overline{u},\overline{v},\overline{t})-L(t)\|$. Otherwise, the minimum of $\nabla Q=(\overline{u},\overline{v},\overline{t})$ will be on a face separating the regions.

(f) Triangle to triangle distance. The calculation between two triangles must be very complicated, because there are so many possible conditions. In order to obtain the result we can turn this problem into calculate the distance between point and triangle. That means the minimum distance among each point of one triangle to the other triangle is the actual distance between two triangles.

To sum up, the main contribution of this chapter is that we established a mapping relationship between the algorithms existed in discrete geometry domain and the concepts defined in GeoSpelling model. The corresponding relationship is summarized in table2-8.

	GeoSpelling	Discrete geometry
Features	Non-ideal feature	point, segment, triangle, point set, polyline, mesh
	Ideal feature	geometric shapes: plane, cylinder, sphere, …
Characteristics	Distance	point to segment, point to triangle, segment to segment, segment to triangle
	Angle	segment to segment, segment to plane, plane to plane
Operations	Partition	segmentation
	Extraction	sampling
	Filtration	outlier removal, filtering
	Association	approximation, interpolation
	Collection	Boolean operation (Union)
	Construction	Boolean operation (Intersection)

Table 2-8: Concepts mapping between GeoSpelling and discrete geometry

Since the fundament of applying discrete geometry theory to GeoSpelling is established, the next part of our work focuses on implement these algorithms to GeoSpelling model. Considering skin model is the basis concept defined in GeoSpelling and each operation needs to rely on it to carry out, the next chapter will discuss how to simulate the skin models.

2.4. Conclusion

GeoSpelling provides a coherent and complete model for ideal and non-ideal features' representation and operation. The real features are usually acquired by measuring instruments and represented as discrete point data. In this chapter, a new framework based on discrete geometry is developed for tolerance representation and operationalization of GeoSpelling.

GeoSpelling defines ideal and non-ideal features to represent the shapes with and without geometrical variations. And it also defines six operations to process these

features: partition, extraction, filtration, association, collection and construction. The corresponding shape representation and processing techniques in discrete geometry are investigated to handle the geometrical features and operation defined in GeoSpelling.

The developed framework provides new convenient and executable solutions for feature representations and operations in GeoSpelling. As a representation of real features which include both random and systematic errors, Skin model is the kernel concept of GeoSpelling. The simulation of skin model based on discrete geometry and statistic analysis is thus investigated in the next two chapters.

Chapter 3

Skin Model Simulation and Visualization

3.1. Introduction

As discussed in previous chapter, Skin model in GeoSpelling represents the non-ideal features associated with geometrical variations. Since the skin models are the geometrical objects the defined operations (partition, extraction, filtration, association, collection and construction) to process, skin model construction is one of the critical issues for operationalization of GeoSpelling.

Skin model is a simulation of real shapes. The characteristics of real shapes should be considered for skin model shaping. Typically, the geometrical variations of the real surface to the nominal one are caused by two kinds of errors: random error and systematic error. Both of the two kinds of errors should be simulated in order to construct the complete skin model. A designed shape is always specified with geometrical tolerances. The simulated skin models should be located within the specified tolerance zones which can be constraints for skin model generation methods.

This chapter presents the developed methods to simulate the skin model with random errors. The methods for skin model shaping with systematic errors will be discussed in the next chapter.

In our research, we follow this hypothesis: the deviations caused by random errors follows Gaussian (normal) distribution, which is proven reasonable for mechanical applications [Bhu 85]. The Based on discrete representations of skin models and their discrete geometrical characteristics, three different methods are developed to shape the skin models with random errors: 1-D Gaussian-based, multi-Gaussian-based, and Gibbs-based. A comparative analysis of the three methods is discussed.

This chapter is organized as follows:

Section 3.2 discusses the discretization of continuous nominal shapes and the discrete characteristics of discrete shapes. Section 3.3 presents the developed methods for shaping skin models with random errors. Section 3.4 proposed the simulating methods of the skin model when considering the tolerance constraints. Section 3.5 gives the comparative analysis of the proposed methods, and section 3.6 is the conclusion.

3.2. Basic foundations

The skin model simulation procedure proposed in our work follows the geometrical product design context. The skin model created using our method inherits the nominal model generated by CAD software. The procedure of creating skin model is that after tessellating the CAD model a discrete geometry product nominal model is obtained, which construct by mesh structure. We import this tessellate nominal model to our own system to generate the skin model. In order to satisfy product life cycle sequence and fulfill the technological and manufacturing demands, there are some requirements are considered in our system. Firstly, it can import tessellate model created by CAD software. Secondly, it can generate skin model, and thirdly, it can export the skin model. At last, it can support other operations defined in GeoSpelling. The basic concepts and foundations concerned will be introduced in this chapter.

3.2.1 Discrete representations of the surface model

As discussed above, the skin models are created from nominal CAD models. This means that our proposed approach considers the CAD model as the reference model (or master model) in the CAD environment. CAD models are typically generated to create a product shape satisfying functional requirements without prior knowledge of their effects on all downstream applications. A CAD system database contains details of the geometry of all features in the model as initially created. In some systems, if a subsequent modification changes the topology of an existing feature, the geometry of the modified feature is also recorded [Kim 07].

In order to obtain the discrete surface model, a tessellation process to convert CAD models into polyhedral ones is considered here. Normally there are two main categories of this processing integrating the tessellation process into CAD modelers and conversion using tessellation criteria [Ruppert 95]. When the initial CAD model is topologically consistent [Hamri 10], i.e. when the CAD model is created in the same software as the polyhedral model and the shape of this model matches the tessellation requirements, it is easy to set up a tessellation process from the user point of view [Hamri 10]. Such cases only occur for simple shapes handled in a so-called integrated environment. However it is no longer efficient when the CAD model has an inconsistent topology due to data exchange between different softwares or to the

differences of shape requirements between tessellated mesh and CAD models, in such case the tessellation criteria should be taken into account.

In the perspective of a full CAD and downstream applications at the level of integration and interoperability, the polyhedral model tessellated from CAD model directly is adopted in our work.

3.2.2 Tessellation techniques

Tessellation techniques are often used to manage datasets of polygons and divide them into suitable structures for rendering and numerical analysis. A tessellation of the object that contains not only the coordinates and supporting triangle indices but other data such as the surface parameters (for the points) as well as the connectivity of the triangles assists the investigator in traversing through and dissecting the CAD representation of a part [Haimes 98].

The use of tessellation techniques for solid models has been studied from different kinds of point of view. The hardware support for adaptive tessellation of Bezier surfaces is proposed by [Espino 07], and the high quality triangulations from CAD models are discussed in [Haimes 98]. In most cases, the solid model is tessellated for satisfying the rendering capabilities 98in the software application environment. In our work, we adopt tessellation technique to obtain the discrete structure of the solid CAD model.

Based on the shape of primitives distributed on a tessellation mesh, the tessellation can be roughly classified into two categories: regular and irregular one. A regular tessellation is a highly symmetric tessellation made up of regular polygons, which are triangles, squares or hexagons. The most common irregular tessellation is composed by Delaunay triangular mesh, in which the size of triangles is different from each other. In our work, we consider both regular and irregular tessellations.

3.2.3 Polyhedral surface approximation

The built-in tessellation algorithm creates a boundary representation that covers the surface of the solid model with triangles, and this mesh data structure is a common way of describing the geometry and the topology of objects in computer graphics. Generally the mesh data structure is the polygon mesh. A polygon mesh consists of

vertices connected to form the faces that in turn form an object. It's known that a proper data structure of polyhedral model is quite important for geometry processing and the data structure of mesh has been studied for a while.

(1). Mesh data structure

As discussed in [Zhao 10], a boundary representation (B-Rep) of mesh considers more adjacent relationships among the neighborhood than face-based representation by which a polygon mesh is represented only as a shared list of vertices and a list of faces. Many boundary representation research efforts have been reported, such as Winged-edge [Baumgart 75], Quad-edge [Leonidas 85], Half-edge [Eastman 82], etc. A synthetic survey of data representation of meshes can be found in [Kettnert 99]. In our method, the Half-edge method is adopted.

As its name implies, the Half-edge Data Structure (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 exactly one triangle. The half-edges of one triangle can be oriented as clockwise or counter-clockwise around the facet.

The half-edge data structure is an extension of the simple polygon mesh. Not only does it contain information of vertices and faces, it also stores information about edges in the mesh, which simplifies analysis of the mesh. An edge is defined as a vertex, a face and three connected edges known as *next*, *prev* and *pair*, see figure 3-1. Using these connections it is possible to access any other edge on the mesh.

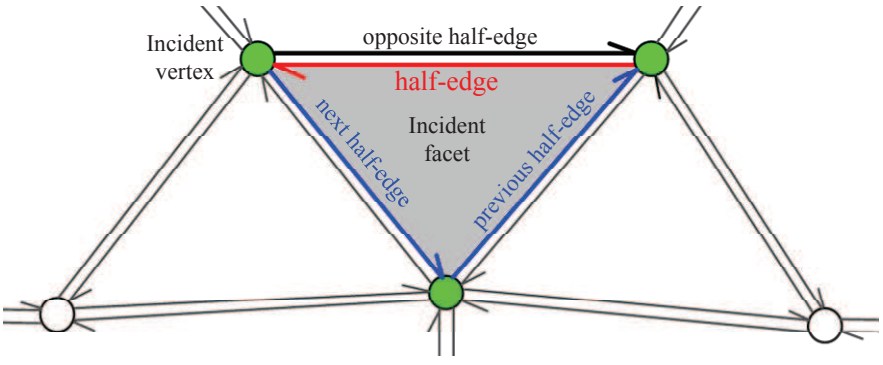


Figure 3-1: Half-edge data structure

Half-edge data structures have a high degree of variability. They may allow

several representations, as well as holes in facets, infinite edge, boundary edges, etc. The representation itself may allow various accesses to the data structure, such as clockwise or counter clockwise traversal of a facet boundary [Bronnimann 01]. Many publicly available implementations of Half-edge data structures are now very popular such as CGAL [CGAL], OpenMesh [OpenMesh], and MeshLab [MeshLab]. In our work, the clockwise data structure is built using the CGAL geometric library.

(2). Mesh file formats

The exchange of polyhedral mesh model between different CAD systems and to downstream applications is an important practical requirement for interoperability and integration. Until recently [Zhao 10] there are many file formats used to store and exchange the mesh data in computer graphics and engineering applications

From the very basic ones, such as Stanford Triangle Format (.ply) which represent just an indexed face set or Stereo-lithography or Stand Triangulation Language (.stl) an Object File Format (.off) which represent just a triangle soup, to such advanced ones, such as Virtual Reality Modeling Language or VRML (.wrl) which enable to represent single object as well as a complete scene with complete interaction.

However our applications work only with set of points and meshes, and therefore, we are only interested in the formats to read and to store the mesh information. Two file formats are considered in our work.

(a) STL file format (.stl)

An STL file is a triangular representation of a 3D surface geometry. The surface is tessellated logically into a set of oriented triangles (facets). Each facet is described by the unit outward normal and three points listed in counterclockwise order representing the vertices of the triangle.

STL is a file format native to many CAD softwares such as CATIA V5. It is widely used for rapid prototyping and computer-aided manufacturing. STL files describe only the surface geometry of a three dimensional object without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations.

The first line in the ASCII format is a description line that starts with the word "solid" in lower case, followed eventually by additional information as the file name, author, date etc. The last line should be the keyword "endsolid". The file continues with any number of triangles, and each one represented as follows figure 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
```

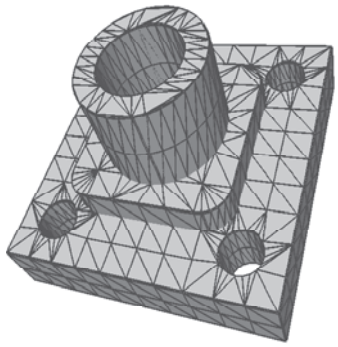


Figure 3-2: STL file format and the STL model

The STL format is easy to generate. However, there are many drawbacks of the STL files among them the "bucket of facets" or the lack of topology and ill-formatted values and errors. Therefore, when a STL file is encountered, model healing and topology construction should be performed.

(b) OFF file format (.off)

OFF files are used to represent the geometry of a model by specifying the polygons of the model's surface. The polygons can have any number of vertices. OFF files are all ASCII files beginning with the keyword OFF. The next line states the number of vertices, the number of faces, and the number of edges. The number of edges can be safely ignored.

The vertices are listed with x, y, z coordinates, written one per line. After the list of vertices, the faces are listed, with one face per line. For each face, the number of vertices is specified, followed by indices into the list of vertices. See the examples below, figure 3-3.

```
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
••• •••
```

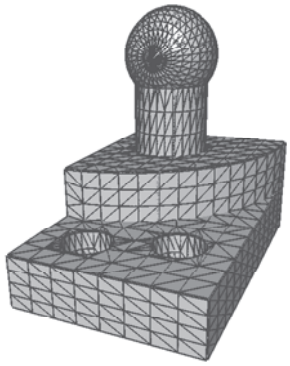


Figure 3-3: OFF file format and the OFF model

Comparing to STL files, OFF files have more compact structure. Considering the representations of a shape model both in STL file and OFF file, the OFF file always has less file size than the STL file. However, the OFF file can describe composite attributes of a shape model, such as vertex, facet, color, and texture etc., but it ignores any vertex normal estimation attribution.

Thanks to their popularity and their availability on CAD systems and public software, STL and OFF file formats are both used in this work.

3.2.4 Normal vector estimation

The derivation of discrete analogs to differential properties of smooth surfaces has been an active area of research for many years [Petitjean 02] [Hildebrandt 06] [Botsch 10]. Two main groups of methods have been developed: fitting and discrete methods. Fitting is a very common approach to estimating local surface properties using a local reconstruction of the surface followed by analytic evaluation on the reconstructed smooth surface patch [Botsch 10]. Differential geometry is used then and employs methods of differential calculus to describe local properties of smooth curves and surfaces [Carmo 76].

Since triangular meshes are piecewise linear surfaces, the classical concepts of differential geometry cannot be applied directly. The challenge is then to compute approximations of the differential properties of this underlying surface directly from the mesh data. Different approaches have been proposed in recent years [Zhao 10].

(1). Neighbor region

Many operations in geometry processing and computer graphics require normal vectors estimation and discrete curvatures computation. A 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 made for a reliable estimation of normal vectors from point sets.

To calculate the normal vector form a point set, the most common way is to construct a local neighborhood through meshes to obtain the local relationship among each point. In this work, the local averaging region is a one-ring neighborhood as discussed in [Zhao 10].

Mathematically, a mesh can be described by the couple $M = \{V, F\}$, where the vertex list is noted as $V = \{v_i : 1 \le i \le n_V\}$, and the facet list is noted $F = \{f_j : 1 \le j \le n_F\}$. In figure 3-4, the set of neighbor points to p_i is noted P^i. If $p_j \in P^i$, p_j is the neighbor of p_i. $e_{j,j+1}$ is the edge between facets f_j and f_{j+1}.

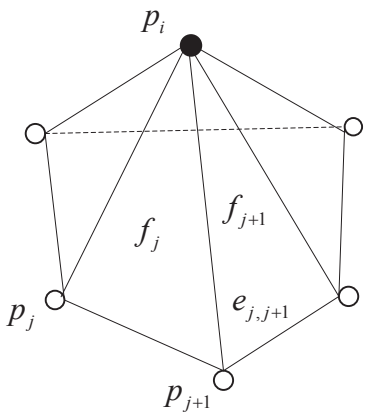


Figure 3-4: The neighborhood structure of point

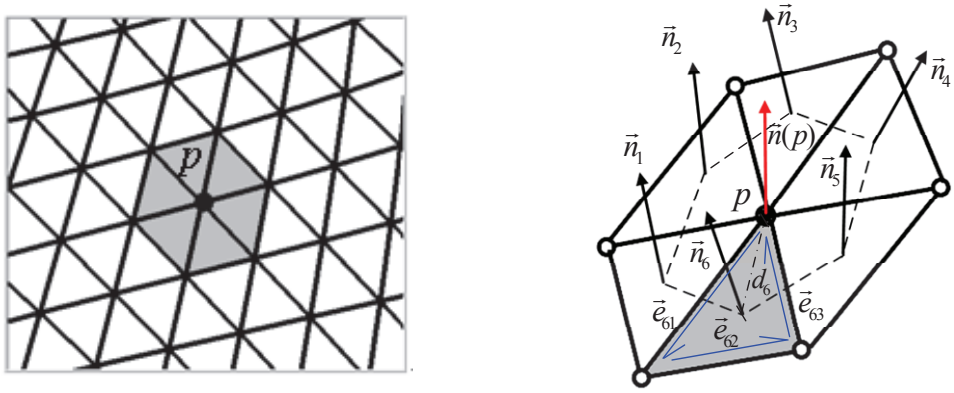
Reliable estimation of the normal vector at a point is much dependent on the identification of its valid neighbor points. Many researchers have used a fixed number of Euclidean nearest neighbor points to estimate the normal vector at the given point [Zhao 10]. Too many neighbor points may degrade the local characteristics of the normal vector. And too few neighbor points may be not sufficient to evaluate the real local geometry. Moreover, possibly, the number of neighbor should be different for different points in implementation. Here we focus on the approach of vertex estimation instead of the mesh structure.

(2). Vertex normal estimation methods

Given a polyhedral mesh structure (see figure 3-5 (a)), to calculate the point P vertex normal, generally, the neighborhood of point P should be found out (see figure 3-5 (b)). The vertex normal of point P is a weighted average value of the normal of triangles, which compose the neighborhood of point P. In figure 3-5 (b), for the i th triangle, the normal $\vec{n}_i$ can be estimated by its edges, see formula 3-1.

$$\vec{n}_i = \frac{\vec{e}_{i1} \times \vec{e}_{i2} + \vec{e}_{i2} \times \vec{e}_{i3} + \vec{e}_{i3} \times \vec{e}_{i1}}{\left\| \vec{e}_{i1} \times \vec{e}_{i2} + \vec{e}_{i2} \times \vec{e}_{i3} + \vec{e}_{i3} \times \vec{e}_{i1} \right\|} \tag{3-1}$$

Where, $\vec{n}_i$ ($i = 1, \ldots N$) indicate the unit normal vector of the i-th triangle facet respectively. $\vec{e}_{ij}$ ($j = 1, 2, 3$) are the three unit edge vectors of the i-th triangle facet in counter-clockwise.



(a) Mesh structure (b) Vertex normal

Figure 3-5 Vertex normal estimation basics

Since the vertex normal is a weighted average value of its neighborhood, a number of researches of evaluating the weight coefficient have been proposed. We present here a survey and comparison of these algorithms in order to find out the most suitable methods for our work.

(a) Gouraud's method [Gouraud 71].

In this method, the vertex normal $\vec{n}(p)$ is deemed as the average vertex normal of all the triangles, which compose point p 's neighborhood. Mathematically, it can be described by formula 3-2

$$\vec{n}(p) = \frac{\sum_{i=1}^{N} \overrightarrow{n_i}}{\left\| \sum_{i=1}^{N} \overrightarrow{n_i} \right\|} \tag{3-2}$$

(b) Thumer's method [Thumer 98].

Based on the Gouraud's method, Thumer and Wuthrich proposed a new method which takes into account the influence of the area of triangles. In Thumer's method, the area of triangle is bigger the influence to the vertex normal is greater. The weight coefficient can be calculated by formula3-3.

$$\omega_i = \frac{A_i}{\sum_{i=1}^{N} A_i} \tag{3-3}$$

ω_i $(i = 1,...N)$ are the weight coefficients corresponding to the normal vectors of facets f_i. A_i $(i = 1,...N)$ represent the area of the i-th triangle facet.

(c) Shen's method [Shen 05].

Shen improved Thumer's method by introducing centroid distance into his method. Instead of estimating the influence of triangle's area, the centroid distance is invested. The weight coefficient can be calculated by formula3-4

$$\omega_i = \frac{d_i}{\sum_{i=1}^{N} d_i} \tag{3-4}$$

(d) Zhao's method [Zhao 10].

Given a polyhedral mesh surface, the normal vector at a 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 using formula 3-5.

$$\vec{n}(p) = \frac{\sum_{i=1}^{N} \overrightarrow{n_i}}{\left\| \sum_{i=1}^{N} \overrightarrow{n_i} \right\|} \tag{3-5}$$

Where, $\boldsymbol{n}_i\ (i=1,...N)$ indicate the unit normal vector of the i th triangle facet. $\omega_i\ (i=1,...N)$ are the weight coefficients corresponding to the normal vectors of facets f_i.

The method used here for the weight coefficients computation considers the influence of the area of each adjacent triangle facet and the distance between the given vertex and the barycenter of each adjacent facet. Parameter ω_i can be calculated by formula 3-6.

$$\omega_i = \frac{A_i / d_i^2}{\sum_{i=1}^{N} A_i / d_i^2} \tag{3-6}$$

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

(3). Comparison techniques

In order to test the above methods, three kinds of mesh structure of the unit sphere are discussed, as showed in figure 3-6. The first two spheres (figure 3-6(a)(b)) are constructed by regular triangle mesh, and the third one (figure 3-6(c)) is constructed by irregular triangle mesh. The comparison among different vertex normal estimation methods is to evaluate the error deviation between the theoretical and the computed value of vertex normal. Theoretically, given a random vertex $v_i(x_i, y_i, z_i)$ of a sphere, and then its vertex normal N_i can be calculated using formula 3-7

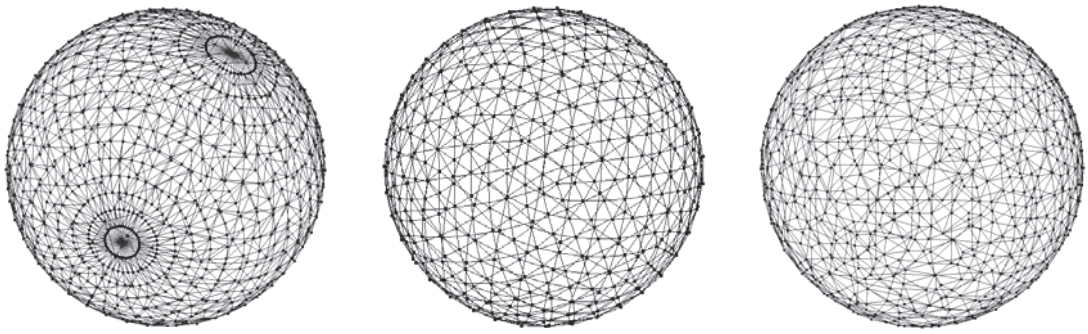
$$N_i = x_i\boldsymbol{i} + y_i\boldsymbol{j} + z_i\boldsymbol{k} \qquad (\|N_i\| = 1) \tag{3-7}$$

In our testing, given the algorithmic and theoretical normals, we estimate the angular cosine discrepancy as formula 3-8

$$d_{discrepancy} = 1 - N_{algorithmic} \cdot N_{theoretical} \tag{3-8}$$

The value of the discrepancy at which the fraction is 0, it means that algorithmic and theoretical normals are in the same direction. The value of the discrepancy is big then the difference between the two vertex normals is distinct.

To evaluate the accuracy of each vertex normal algorithm, we performed two different ways to compare these methods based on formula 3-8: one is comparing the vertex normal of marked points called landmark and the other one is comparing the average vertex normal and its standard deviations.



(a) Regular mesh (b) Uniform triangle mesh (c) Non-regular mesh

Figure 3-6: Mesh structure

The marked point testing method marks several points on different target models to compare the vertex normal, and it is based on the sampling strategy. This kind of testing method is propitious to the experiments in which the model is composed by numerous points. In our method, we adopt three landmarks to test and the result is illustrated in table 3-1.

Landmark	Case	Gouraud	Thumer	Shen	Zhao
Landmark 1	Case (a)	1.3×10^{-5}	3.09×10^{-4}	1.0×10^{-6}	1.19×10^{-4}
	Case (b)	6.4×10^{-5}	8.0×10^{-5}	3.0×10^{-5}	5.5×10^{-5}
	Case (c)	6.5×10^{-5}	7.1×10^{-5}	3.5×10^{-5}	3.7×10^{-5}
Landmark 2	Case (a)	1.58×10^{-1}	6.32×10^{-2}	6.69×10^{-2}	3.45×10^{-2}
	Case (b)	2.3×10^{-2}	5.5×10^{-3}	5.65×10^{-3}	1.17×10^{-2}
	Case (c)	7.39×10^{-3}	3.51×10^{-2}	2.19×10^{-2}	1.82×10^{-2}
Landmark 3	Case (a)	9.0×10^{-6}	1.4×10^{-4}	2.0×10^{-6}	4.5×10^{-5}
	Case (b)	9.2×10^{-5}	1.07×10^{-4}	5.4×10^{-5}	8.3×10^{-5}
	Case (c)	5.1×10^{-5}	5.6×10^{-5}	4.7×10^{-5}	5.6×10^{-5}

Table 3-1: Vertex normal of one marked point

For the result we can see that Shen's method provide the closest vertex normal

value in different sphere models and Zhao's method has a good behavior in irregular mesh model, while Gouraud's and Thumer's methods can offer the closet value in regular mesh case. From this testing we can roughly infer that Shen's method offers the smallest error in most cases.

The average vertex normal testing method takes into account the mean vertex normal value of all the points. It reflects the global deviation between theoretical and computed vertex normal value. In this testing, we evaluate both mean value and standard deviation, and the first item reflects the average deviation while the second item reflects the distribution of this deviation. The testing result is illustrated in table 3-2. From table 3-2 we can see, in case (a) and case (b), Shen method has the closest vertex normal value comparing to the other three methods, and this reflect that Shen's method is more suitable to the regular mesh structure case. In table 3-2 we can see, in case (c) Zhao's method provides the closest mean value, and this illustrates that Zhao's method has a good performance on the irregular mesh structure case.

Case	Item	Gouraud	Thumer	Shen	Zhao
Case (a)	Mean	7.7×10^{-5}	2.85×10^{-4}	2.2×10^{-5}	1.72×10^{-4}
	StDev	1.5×10^{-4}	3.76×10^{-4}	4.3×10^{-5}	2.81×10^{-4}
Case (b)	Mean	3.6×10^{-5}	3.8×10^{-5}	1.4×10^{-5}	3.7×10^{-5}
	StDev	4.3×10^{-5}	5.0×10^{-5}	1.8×10^{-5}	4.5×10^{-5}
Case (c)	Mean	5.93×10^{-4}	7.91×10^{-4}	6.19×10^{-4}	2.62×10^{-4}
	Stdev	5.99×10^{-3}	1.04×10^{-3}	1.04×10^{-2}	3.92×10^{-4}

Table 3-2: Average vertex normal of point set

Combining the two evaluation methods of vertex normal estimation discussed above, we infer that Shen's method is good at regular mesh structure case while Zhao method is suitable to the irregular mesh structure case. Although Shen's method has the closet value in regular testing experiment, the difference comparing to Zhao method is small. In irregular experiment, the difference between these two methods is also small. In order to use a unique vertex normal estimation method, we adopt Zhao's method in our work.

3.2.5 Discrete curvatures

Considerable research activities in discrete differential geometry have been ensued in recent years and many techniques have been developed [Taubin 95] [Meyer 02] [Steiner 03]. Based on the investigation of [Meek 00], these techniques can be classified into three main categories: local surface approximation based methods, discretization of continuous differential operators and formulas and tensor based methods.

The first method approximates a given mesh by a local quadric surface, and then it applies the first and second derivatives to obtain the curvatures [Meyer 02]. This method is based on surface parameterization and calculates mean curvature and Gaussian curvature using formula 3-9

$$r(u,v) = (x(u,v), y(u,v), z(u,v)) \tag{3-9}$$

Where $r(u,v)$ correspond to semi-geodesic coordinates [Mokhtarian 01], Gaussian curvature and mean curvature could be derived from formula (3-9) using formula 3-10 and formula 3-11

$$K = \frac{b_{uu}b_{vv} - b_{uv}^2}{x_v^2 + y_v^2 + z_v^2} \tag{3-10}$$

$$H = \frac{b_{vv} + (x_v^2 + y_v^2 + z_v^2)b_{uu}}{2(x_v^2 + y_v^2 + z_v^2)} \tag{3-11}$$

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

Principal curvatures are then derived as follows (formula 3-12).

$$\begin{cases} \kappa_1 = H + \sqrt{H^2 - K} \\ \kappa_2 = H - \sqrt{H^2 - K} \end{cases} \tag{3-12}$$

Where, κ_1 and κ_2 are the maximal and the minimal principal curvatures respectively.

The second techniques discretize the mathematical formulas that compute the

curvatures of continuous surface and then extend the underlying notions to discrete domains [Srinark 08]. This method begins with the calculations of mean curvature and Gaussian curvature based on the discrete version of the Laplace-Beltrami operator.

Similar to the various discrete versions of the Laplace-Beltrami operator, many methods have been proposed for directly estimating the curvature tensor on polygonal surfaces. The method introduced by Cohen-Steiner and Morvan [Cohen-Morvan 03] has been successfully applied for discrete shape processing. In differential geometry, the shape operator or Weingarten endomorphism [Oneill 97] determines the principal curvatures and the principal directions of a surface. In the case of discrete shapes, the discrete shape operator matrix is estimated at each vertex and then the principal curvatures are derived.

The method proposed by Zhao [Zhao 10] is based on the work reported in [Cohen-Morvan] which has been improved by adding weights to the discrete shape operator matrix and modifying neighbour region.

Considering an arbitrary vertex p on a polyhedral mesh surface Σ, the discrete shape operator matrix at p can be estimated as formula 3-13

$$W(B)=\frac{1}{A}\sum_{e\in E}\lambda_e\cdot\beta(e)\cdot length(e\bigcap B)\cdot(\vec{e}\times\vec{e}^{\,T}) \tag{3-13}$$

Where, B is the local region around the vertex and A is its area. In our case, B is defined based on the Voronoi cell, (figure 3-7 (a)).

λ_e is the coefficient associated with the edge e and is defined as formula 3-14

$$\lambda_e=\frac{\lambda_e^*}{\sum_{e\in E}\lambda_e^*},\quad \lambda_e^*=\arccos\frac{\vec{n}(p)\cdot\vec{n}(e)}{\|\vec{n}(p)\|\cdot\|\vec{n}(e)\|} \tag{3-14}$$

Where $\vec{n}(p)$ is the unit normal vector of vertex p. $\vec{n}(e)$ is the edge normal of the edge e, which is equal to the average normal vectors of the two triangles incident to the edge. E is the set of all the mesh edges in B. $length(e\bigcap B)$ is the length of the edge e in B. $\beta(e)$ is the dihedral angle between the two normal vectors of the

triangles incident with e. In figure 3-7 (b), these two normal vectors are denoted as $\vec{n}_1$ and $\vec{n}_2$. $\vec{e}$ is the unit vector along the edge e.

(a) Voronoi cell (b) Normal vectors

Figure 3-7: Discrete curvature estimation parameters

The two principal curvatures can be estimated as the first two eigenvalues of the matrix $W(B)$. The two principal directions can be also estimated by their corresponding eigenvectors.

3.3. Skin model simulation

The 3D topography of the surface of a manufactured part looks similar to a geographical terrain [Stout 10]. Prediction and modeling of surfaces which emulate complex geographical and geological shapes have been widely developed in geostatistics and spatial statistics applications [Cressie 93].The topic of rough surface simulation is also fundamental in modeling and empirical practices of the wear and friction literature. Two categories of techniques have been investigated by the wear and friction research community. The first entails the use of statistical, stochastic, or Monte Carlo methods to distribute characteristic values over a surface (point height, curvatures, etc.) [Patir 78]. 3D surface topography modeling mainly concentrates on random surfaces of Gaussian ordinate distribution [Reizer 11] or Gaussian Processes [Xia 08].

The second category employs fractal models [Blackmore 97]. There are also hybrid approaches that mix both statistical and fractal algorithms [Zou 07]. However, Surfaces simulated by fractals are different from their physical counterpart due to their homogeneous quality [Bakolas 03].

Statistical approaches are physically intuitive and represent surfaces in a manner

consistent with observation and measurement as well and physical based modeling [Peridier 11]. However, they fail to consider the multi-scale nature of rough surfaces

Due to the discrete nature of shapes we are considering here and the discrete differential geometry characteristics of points such as normal and curvatures, statistical approaches are well adapted to model the skin model and the non-ideal features thanks to the rich analytical and sampling techniques of random variables.

3.3.1 Random simulation techniques

Random variables' simulation is based on the random number generation techniques, which is used to design and generate a sequence of numbers without any patterns. Based on the application field, the random simulation techniques can be classified into physical and computational methods. A physical random number generator is based on a random atomic physical phenomenon whose unpredictability can be traced to the laws of quantum mechanics [Tsoi 07]. The computational random simulation numbers are obtained by computational algorithms, which produce long sequences of apparently random results. Actually, the computational random numbers are pseudorandom, and they are determined by the initial value of the algorithm [Fog 01]. Although pseudorandom number is not a true random number, the result sequences are quite closer to truly random ones by using hardware random number generators. The careful mathematical analysis in [Peterson 98] proved that a pseudorandom generation numbers are sufficiently random to suit the intended use.

Since the pseudorandom sequence is dependent on the initial value of the algorithm, these values have a periodicity. There are some investigations on the periodicity of the pseudorandom simulation. According to [Luby 96] if the internal state of a pseudorandom number generator is n bits, its period is no longer than 2^n results. However, in practice the output sequence is shorter than expected periods for some initial values, and this is an open problem for today's pseudorandom number generator.

The result of the pseudorandom number generator can follow a certain distribution types of random variables, such as Gaussian distribution, uniform distribution, Perlin distribution [Wijgerse 07], Poisson distribution [Ronald 88], etc.

In our work, we focus mainly on Gaussian distribution. There are three different

methods to obtain the Gaussian random distribution numbers [Rubinstein, 81] [Taygeta 95] and these methods can be used also in the case of other distributions.

The first method is the inversion method. The basic idea is based on inverse function of the cumulated density function (figure 3-8). The variable $X = F^{-1}(U)$ has the cumulated density function F and U is a uniform random variable in the interval [0,1]. For the standard normal distribution, the inverse cumulated distribution function uses the error function erf(x) (see formula 3-15). The inverse of the error function can only be approximated.

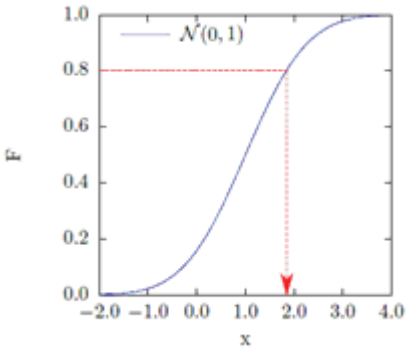


Figure 3-8: Density function

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{u=-\infty}^{x} e^{-\frac{u^2}{2}} du = \frac{1}{2}[1 + erf(\frac{x}{\sqrt{2}})]$$
$$erf(x) = \frac{2}{\sqrt{\pi}} \int_{u=0}^{x} e^{-u^2} du \qquad (3\text{-}15)$$

The second method uses the fundamental transformation law of probabilities. The idea is that the transformation takes random variable from one distribution as inputs and the outputs are in a new distribution function. Box-Muller transformation [Box 58] allows us to transform uniformly distributed random variables to a new set of random variables with a Gaussian distribution.

The basic foundation of the transformation is as follows (formula 3-16)

$$y_1 = sqrt(-2\ln(x_1))\cos(2\pi \cdot x_2)$$
$$y_2 = sqrt(-2\ln(x_1))\sin(2\pi \cdot x_2) \qquad (3\text{-}16)$$

This algorithms start with two independent random numbers x_1 and x_2, which are uniformly distributed in the interval [0,1]. The transformation formula (formula 3-16) creates two new independent random numbers which have a Gaussian distribution

with zero mean and a standard deviation of one.

The third method is rejection sampling. This method belongs to the general field of Monte Carlo techniques, including Markov chain Monte Carlo algorithms that use acceptance/rejection to achieve simulation from the target distribution [Kalos 09].

3.3.2 1D-Gaussian method

In probability theory, the Gaussian distribution is a continuous probability distribution that is often used as a first approximation to describe real-valued random variables that tend to cluster around a single mean value. The graph of the associated probability density function is "Bell"-shaped.

The principle of the 1D-Gaussian method can be described in figure 3-9. The "bell" shape reflects the scope of the 1D-Gaussian distribution acting on one point in the direction of the vertex normal.

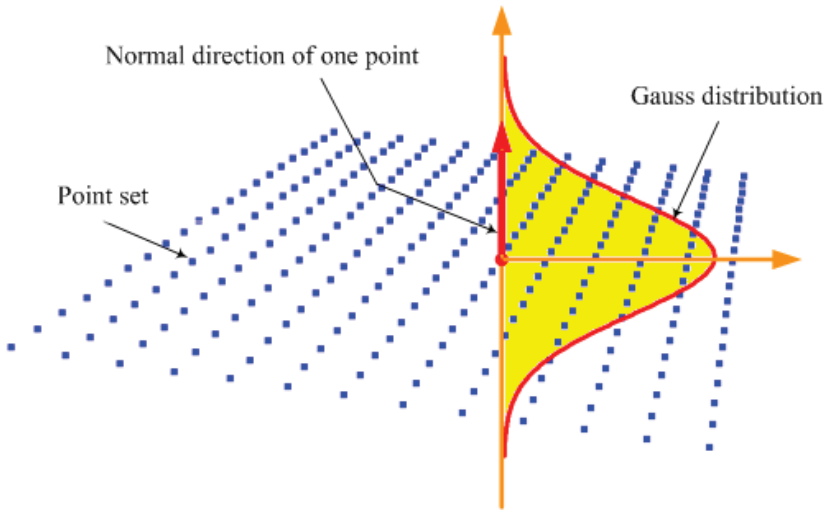


Figure 3-9: Principle of 1D-Gaussian method

The random distribution value can be calculated by the probability density function of the 1D-Gaussian (formula 3-17) method.

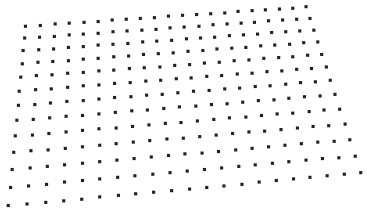
$$X = N(\mu,\sigma^2) : f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3-17}$$

$$u = \frac{x-\mu}{\sigma} = N(0,1)$$

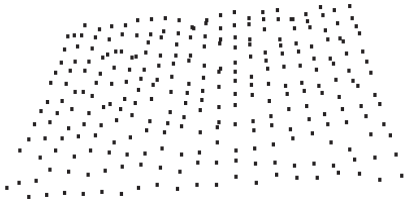
Where μ is the mean value, σ^2 is the variance, x is the Gaussian variable.

In our case, in formula 3-17, the mean value is the input points' coordinates, and the variance determines the width of the Gaussian distribution. This method uses the Gaussian variable as the deviation value in the direction of the vertex normal (see section 3.2.1) of CAD model, and then applies this calculation to each point.

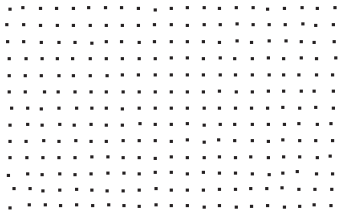
The simulation result in figure 3-10 shows the 1D-Gaussian method in the case of a skin model of a plane sampled with 273 points.



(a) Nominal plane



(b) skin model view-1



(c) skin model view-2

(d) skin model view-3

Figure 3-10: Skin model generated by 1D-Gaussian method

3.3.3 Multi-Gaussian method

The multivariate Gaussian random function model is commonly used in physical modeling and geostatistics to model spatial variability. Its distribution is a generalization of the one-dimensional normal distribution to higher dimensions. A random vector is said to be multivariate normally distributed if every linear combination of its components has a univariate normal distribution. The multivariate normal distribution is often used to describe random variables which cluster around a mean value. Suppose that 1D-Gaussian is a verso method, and then the multi-Gaussian would be a recto method. While 1D-Gaussian method is based on the assumption that the corresponding points of the real workpieces could be found out from the CAD model in the vertex normal directionmulti-Gaussian method enables to simulate how the points of a CAD model would distribute after a manufacturing process. The variables generated in a multivariate Gaussian method are constrained by a covariance matrix.

Since multi-Gaussian method generates the random variables directly from the CAD model, independent from the vertex normal estimation, it would be a more reliable and more general method.

In our case, a spatial random vector is defined as $X = (X_1, X_2, X_3)^T$. The probability density function of multivariate Gaussian distribution can be expressed as in formula 3-18.

$$f(x) = \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) \quad (3\text{-}18)$$

Where Σ is the covariance matrix, $|\Sigma|$ is the determinant value, and μ is the mean vector.

The principle of this method is described in figure 3-11. The ellipsoid reflects the scope of trivariate-Gaussian distribution acting on one point. It is a random vector and can be calculated by the probability density function of the 3D-Gaussian (formula 3-18).

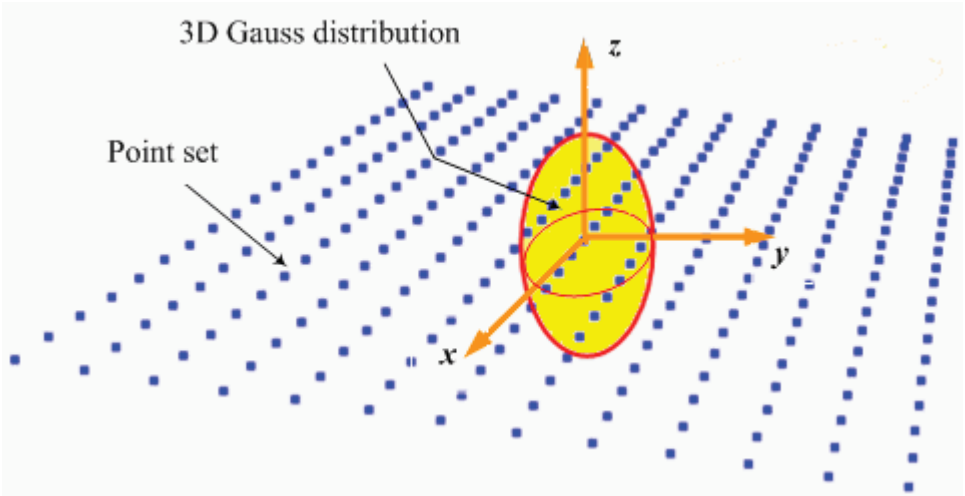


Figure 3-11: Principle of 3D-Gaussian method

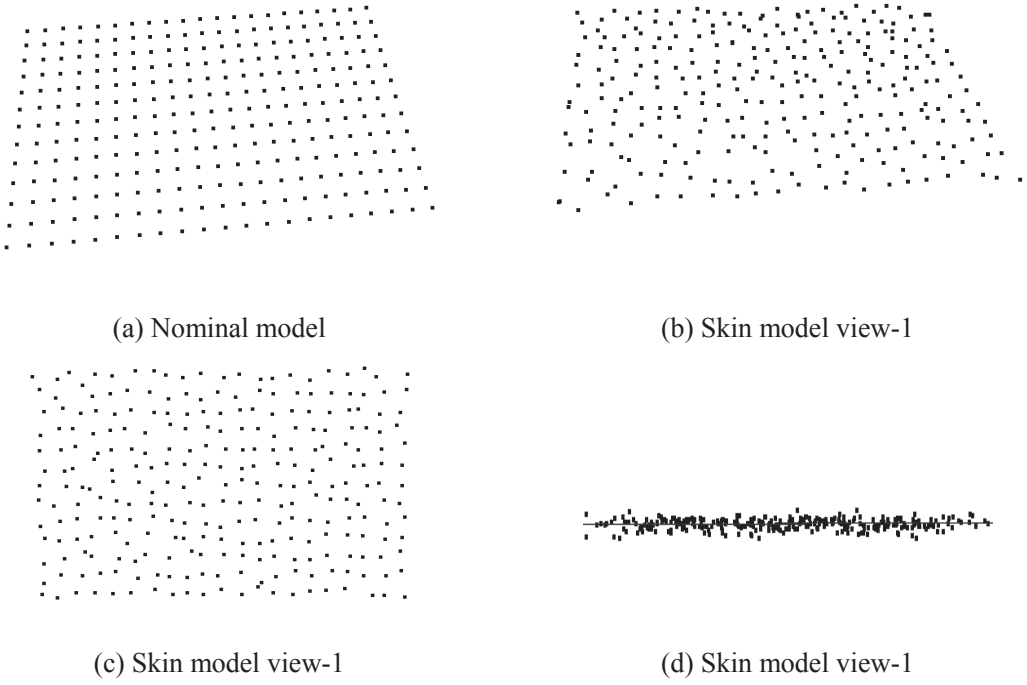
In our case (formula3-18), the mean value is the point's coordinates, and the relationships among each axis are constrained by the covariance matrix. To generate a multivariate normal random vector X~N(μ, Σ), we use the following algorithm [Kroese 11].

(a) Derive the Cholesky decomposition of Σ = AAT where A is the n × n lower triangular matrix of the Cholesky factor of the matrix Σ.

(b) Generate Z1, . . . , Zn~N(0, 1) and let Z = (Z1, . . . , Zn)

(c) Output X = μ + AZ.

Here is an experiment of this method, when considering a CAD plane composed of 273 points. The simulation result with different views is illustrated in figure 3-12. Compared to the simulation result of 1D-Gaussian, the random distribution of multi-Gaussian is in three dimensions.



(a) Nominal model

(b) Skin model view-1

(c) Skin model view-1

(d) Skin model view-1

Figure 3-12: Skin model generated by 3D-Gaussian method

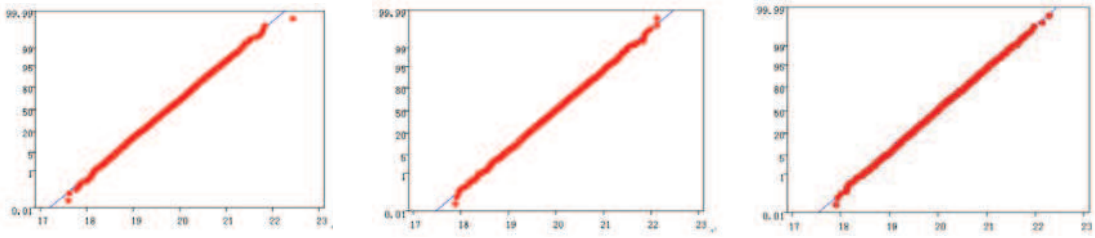
3.3.4 Gibbs method

Gibbs method defined here is used to simulate the multi-Gaussian distribution of random numbers using Markov Chain Monte Carlo (MCMC) methods. MCMC method is an example of Monte Carlo methods, which relies on repeated random sampling to compute their results. Since Monte Carlo simulation methods do not always require truly random numbers, the pseudorandom sequences are enough in most cases [Davenport 92]. Considering that MCMC methods are based on constructing a Markov chain that has the desired distribution and only depends on current state instead of the entire past data, the Gibbs method provides a reliable Gaussian distribution result.

The Gibbs sampling algorithm is used to generate a sequence of samples from

the joint probability distribution of two or more random variables. The algorithm is in reference to an analogy between the sampling algorithm and statistical physics. Gibbs method requires an initial value of the parameters. At each iteration each parameter of interest is sampled a given value from the other parameters and data. Once all the parameters of interest are sampled, the nuisance parameters are sampled given the parameters of interest and the observed data. This characteristic of the Gibbs method can make the random distribution of point set of skin models approximates a Gaussian distribution.

When constructing a Markov chain for Gibbs method to obtain a target distribution, the determination of how many steps are needed to converge to the stationary distribution within an acceptable error is an important problem. Since the convergence performance is determined by the iterative times, the probability distribution of the normality assumptions of Gibbs should be considered. The experiment of normal distributions is simulated by Minitab® software for different iterative runs. In our experiment, we test the interactive time N equal to 100, 1000 and 10000 respectively. The convergent normality distributions are illustrated in figure 3-13, and we can see that the convergent probability of the first two conditions is around 99% and in the third condition can be achieved at 99.9%.



(a) N =100 (b) N =1000 (c) N =10000

Figure 3-13: Convergence test of Gibbs method

After determining the interactive time of Gibbs method, we focus on how to construct a Markov chain, which has the Gaussian distribution random samples. Based on [Rouchka 97] Gibbs sampling requires an initial starting value for the parameters.

In Gibbs sampling, a vector of parameters of interest Ψ is required, and also a nuisance parameter Λ with observed data B, from which a converging distribution can be obtained. Suppose $(\Psi_1(0), \Psi_2(0), ..., \Psi_d(0))$ is an initial starting random value

in R^d. Note that the conditional density function of $\Psi_j \left| \Psi_{-j} \right.$ is $f(\Psi_j \left| \Psi_{-j} \right.)$, from which the Gibbs sampler selects candidate points.

The initialization of the process is at $t = 0$ and the initial vector value is then $\Psi(0)$. When t is increasing ($t = 1,2,...,T$), then $\Psi(t)$ follows the density function $f(\Psi_j \left| \Psi_{-j} \right.)$ to generate new point to replace the old one and iterative calculation is performed until it converges to the target value. The corresponding pseudo-code is described as follow.

For $t = 1,2,...,T$

(a) Let $\psi_1 = \Psi_1(t-1)$.

(b) Let j a variable between $[1, d]$. For $j = 1,2,...,d$, using $f(\Psi_j \left| \psi_{-j} \right.)$ to get candidate point $\Psi_j^*(t)$, and then update $\psi_j = \Psi_j^*(t)$. During the sampling process the variations of each vector are as follow:

- Sample $\psi_1(t)$ from $f(\psi_1 \left| \psi_2(t-1),...,\psi_d(t-1),\Lambda(t-1),\mathrm{B} \right.)$;
- Sample $\psi_2(t)$ from $f(\psi_2 \left| \psi_1(t),\psi_3(t-1),...,\psi_d(t-1),\Lambda(t-1),\mathrm{B} \right.)$;
- ……
- Sample $\psi_d(t)$ from $f(\psi_d \left| \psi_1(t),...,\psi_{d-1}(t),\Lambda(t-1),\mathrm{B} \right.)$;
- Sample $\Lambda(t)$ from $f(\Lambda \left| \psi_1(t),...,\psi_d(t),\mathrm{B} \right.)$;

(c) Let $\Psi(t) - = (\Psi_1^*(t),...,\Psi_d^*(t))$, and then increase t.

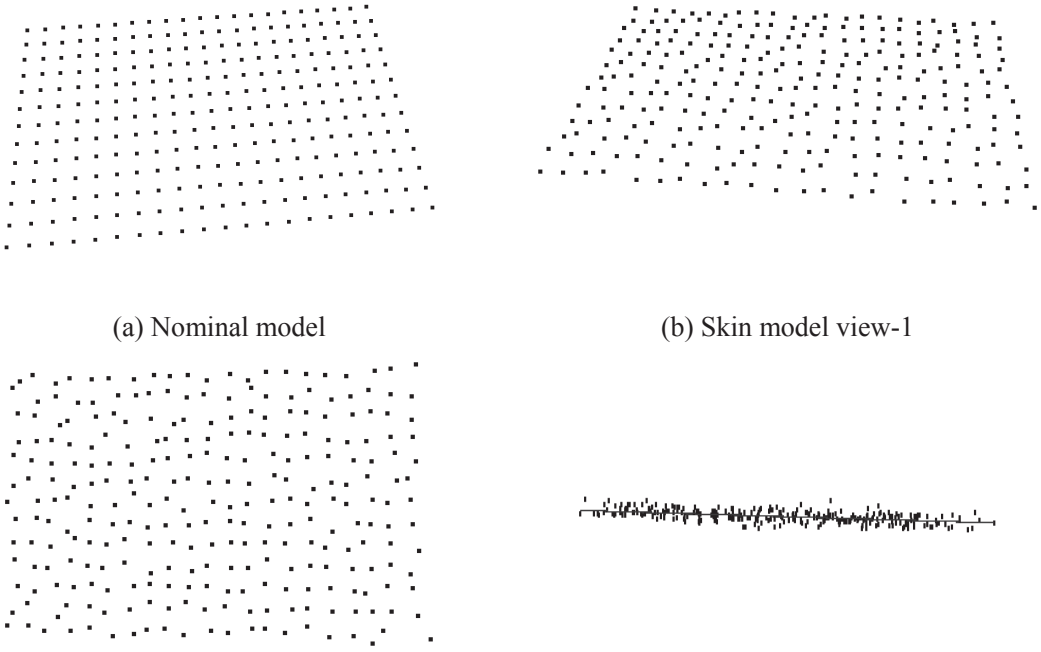
The vectors $\psi(0),\psi(1),...,\psi(t)$ represent the realization of a Markov chain, where the transition probability from $\psi(\cdot)$ to ψ is defined by formula 3-19

$$F(\psi(\cdot),\psi) = f(\psi_1 \left| \psi_2(\cdot),...,\psi_d(\cdot),\Lambda(\cdot),\mathrm{B} \right.) \cdot f(\psi_2 \left| \psi_1,\psi_3(\cdot),...,\psi_d(\cdot),\Lambda(\cdot),\mathrm{B} \right.) \cdots$$

$$f(\psi_d \left| \psi_1,...,\psi_{d-1},\Lambda(\cdot),\mathrm{B} \right.) \qquad (3\text{-}19)$$

The joint distribution of $(\psi_1(i),...,\psi_d(i),\Lambda(i))$ converges geometrically to $f(\psi_1,...,\psi_d,\Lambda|\mathrm{B})$ when $i \to \infty$.

In our case, the stationary objective function follows a Gaussian distribution. Based on Gibbs interactive method, sampling random variables from Gaussian random numbers enables to simulate the skin model as in figure 3-14.



(a) Nominal model

(b) Skin model view-1

(c) Skin model view 2

(d) Skin model view 3

Figure 3-14: Skin model generated by Gibbs method

3.3.5 Time performance

To compare the time performance of skin model simulation methods including 1D-Gaussian method, 3D-Gaussian method and Gibbs method, we experiment five different CAD models with different vertex numbers for testing. For each test model, we generate the corresponding skin models using the three different skin model generation methods described above, and the results are illustrated in table 3-3.

Time performance is obtained based on the following configuration platform: MS Visual C++ 2005 on Windows operation system; 2.10GHz Core II with 1.00GB RAM.

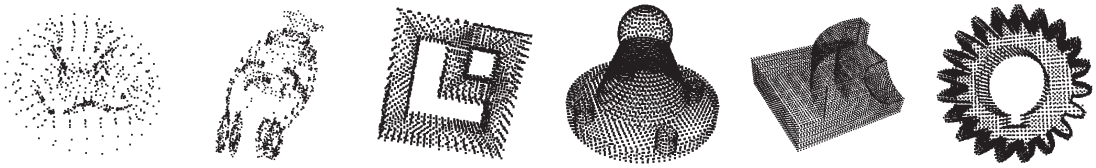
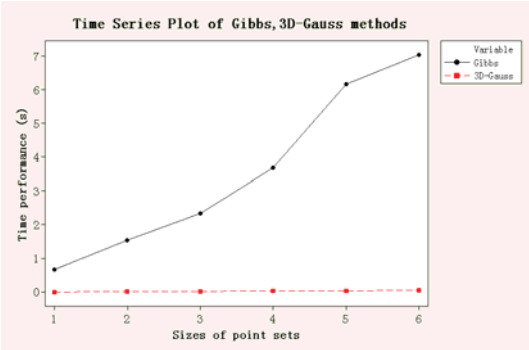
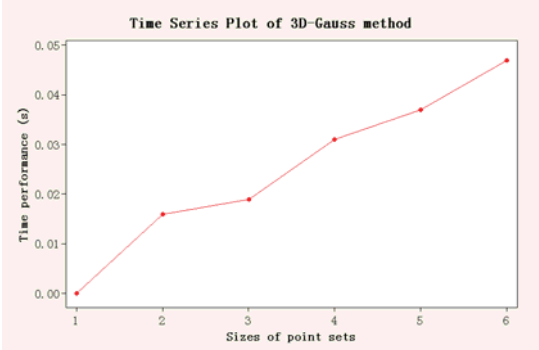
Index	Number of vertices	Time (s) 1D-Gaussian	Time (s) 3D-Gaussian	Time (s) Gibbs
（1）	769	<0.001	<0.001	0.672
（2）	1752	<0.001	0.016	1.532
（3）	2686	<0.001	0.019	2.343
（4）	4209	<0.001	0.031	3.672
（5）	7041	<0.001	0.037	6.156
(6)	8104	<0.001	0.047	7.031
				
(1)	(2)	(3)	(4)	(5)	(6)

Table 3-3: Time performance

From table 3-3 we can deduce that the 1D Gaussian method has better performances, followed by the 3D-Gaussian than the Gibbs method. The variation trends of the Gibbs method and the 3D-Gaussian method are illustrated in figure 3-15(a) and figure 3-15(b) which is the magnified trend of the 3D-Gaussian method.



(a) Time performance of the Gibbs method and the 3D-Gaussian method.



(b) Time performance of the of 3D-Gaussian method.

Figure 3-15: Time performance

3.4. Adding constraints on skin model

According to the GPS paradigms, the tolerance specifications encode the allowable variation in product geometry, and it should be reflected by the skin model. The probabilistic interpretation of the clouds of points introduced in section 3.2doesn't take into considerations functional requirements or manufacturing considerations. Figure 3-16 highlights the tolerance types considered in our work, respectively, form specification, orientation specification and position specification. They are denominated here as constraints.

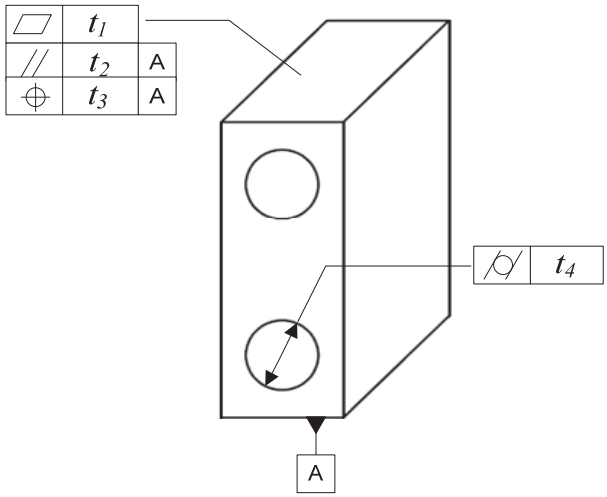


Figure 3-16: Constraints considered for skin model simulation

3.4.1 Principle component analysis

Principal Component Analysis (PCA) also known as Karhunen-Loeve transform was invented in 1901 by Karl Pearson [Pearson 01]. It is a mathematical method used to convert a set of correlated variables into a set of values of uncorrelated variables called principal components. It is usually used to evaluate the main element or structure of a set of data. Based on the covariance matrix, the PCA method proceeds in such a way that the first principal components has the highest variance (that is, accounts for all the variability in a data set), and each succeeding component in turn has the highest possible variance within the constraint.

Most of the applications of PCA are statistical in their nature. There are also some geometric applications. For a given point cloud a point p on the surface, we can

estimate the undirected normal to the surface at p as follows: find all the points in a certain neighborhood of p and compute the principal components of those points. The first principal component is an estimate of the undirected normal at p [Dimitrov 08]. A frequently used heuristic for computing a bounding box of a set of points is also based on PCA. The principal components of the point set define the axes of the bounding box. Once the directions of the axes are given, the dimension of the bounding box is easily found by the extreme values of the projection of the points on the corresponding axis [Lahanas 00] [Dimitrov 09].

Consider a discrete shape P_N represented by an arbitrary set of points $P_i = [x_i, y_i, z_i]^T$. The PCA method computes the principal axes of the discrete shape using the following three steps.

(a) The origin of the principal coordinates system is determined as the centroid of P_N which is calculated by formula 3-20

$$O_{pca} = \frac{1}{N}\sum_{i=1}^{N} p_i \quad (p_i \in P_N) \tag{3-20}$$

(b) The covariance matrix is defined by formula 3-21

$$M_{\text{cov}} = \sum_{i=1}^{N} (p_i - o_{pca})(p_i - o_{pca})^T \quad (p_i \in P_N) \tag{3-21}$$

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

After creating the random point set to simulate the skin model, we also intent to add constraints on it to satisfy the specification requirements.

3.4.2 Form specification

Based on ISO [ISO 2005] standard, the flatness is the condition of a surface having all elements in one plane A flatness tolerance specifies a tolerance zone defined by two parallel planes within which the surface must lie. In discrete geometry view, the flatness tolerance zone limits the points set of the surface within a given

bound (see figure3-17).

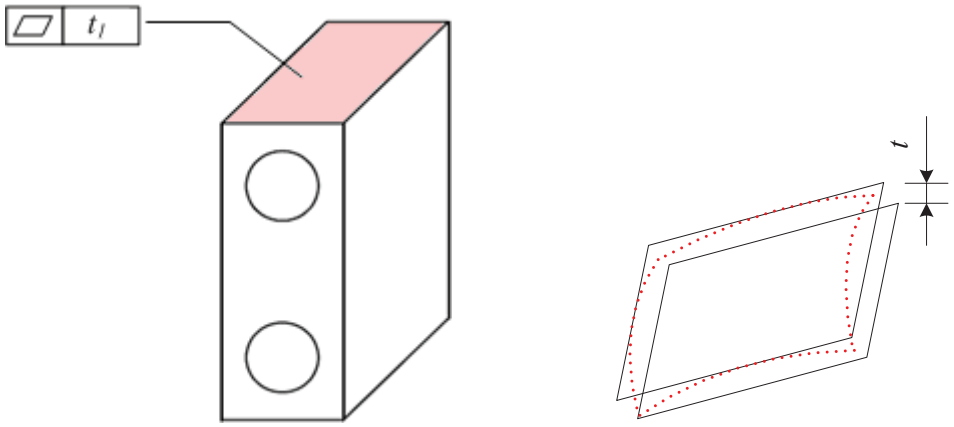


Figure 3-17: Flatness tolerance zone

To estimate the form specification of a skin model, the first step is to determine the tolerance zone direction. Many studies have been done to calculate the tolerance zone direction, and here we adopt the PCA method (see section 3.3.1) to calculate it directly. The second step of estimating the form specification is adding constrained conditions on the point set. The principle of this approach is described by figure 3-18.

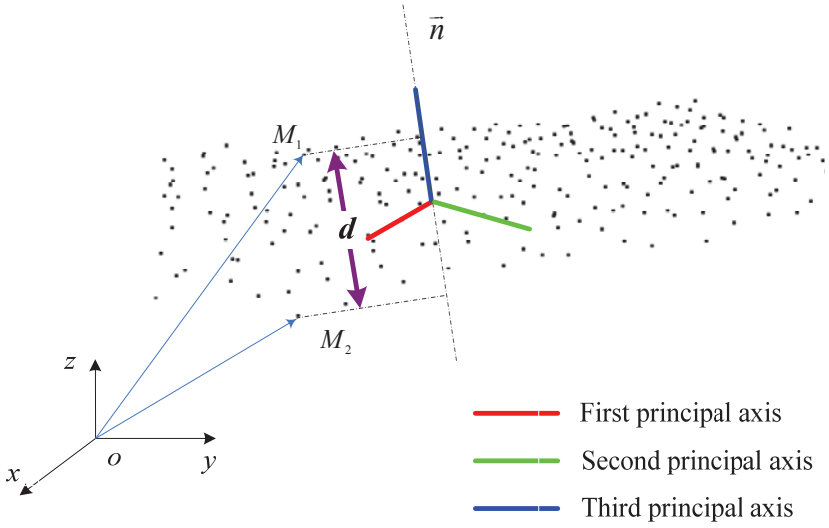


Figure 3-18: Flatness constraint

In figure 3-18, the point set is a skin model of a plane which follows the Gaussian distribution. Using PCA method the principal axis in three dimension space is evaluated. Since the workpiece size is much bigger than the deviation of the point set, the direction of the third principal axis $\vec{n}$ is deemed as the tolerance zone direction. The processing flow is illustrated in figure 3-19.

Where M_1, M_2 are the upper and the lower bound points of the skin model, and d is the enclosing box dimension that approximates the minimum zone dimension in the direction of the flatness tolerance direction.

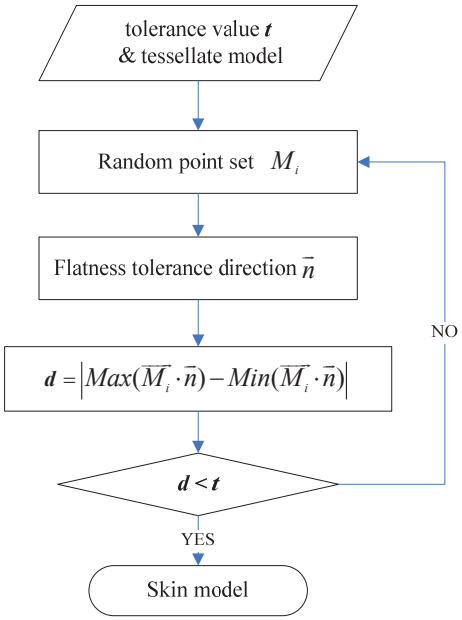


Figure 3-19: Process flow of flatness specification

3.4.3 Orientation specification

Besides form specification, the orientation specification is also taken into account in our work, and here we consider parallelism tolerance. Based on the ISO standard [ISO 2005], the parallelism is the condition of a surface or center plane, equidistant at all points from a datum plane: or an axis, equidistant along its length from one or more datum planes or a datum axis (see figure 3-20).

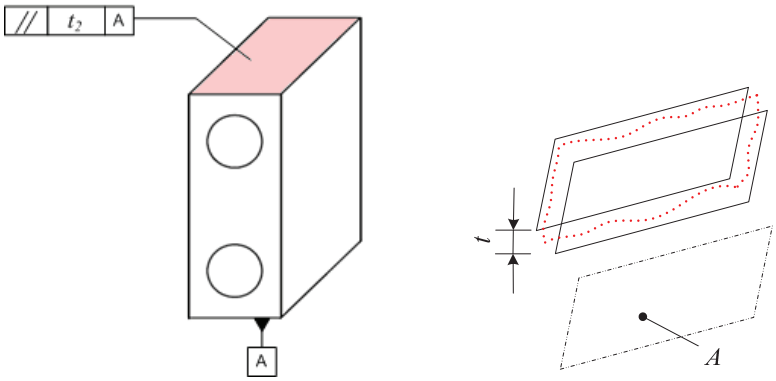


Figure 3-20: Parallelism tolerance zone

For parallelism specification, the tolerance direction is the consistent as the

normal direction of the datum plane and all the points should be within the tolerance zone. Figure 3-21 illustrates the principle of this method.

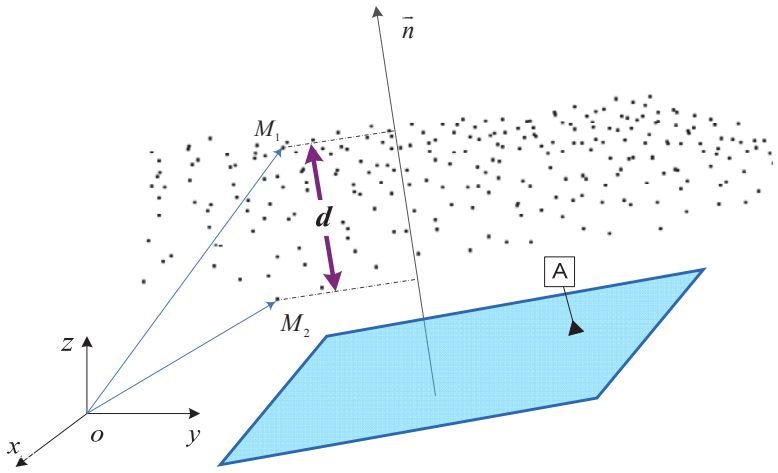


Figure 3-21: Parallelism constraint

The parallelism specification satisfies the constraints in formula 3-22.

$$\left|Max(\overrightarrow{M_i}\cdot\vec{n})-Min(\overrightarrow{M_i}\cdot\vec{n})\right|\le t_{parallelism} \tag{3-22}$$

Where $\vec{n}$ is the normal direction of the datum plane, and M_i is an arbitrary point.

The process flow of parallelism tolerance is illustrated in figure 3-22.

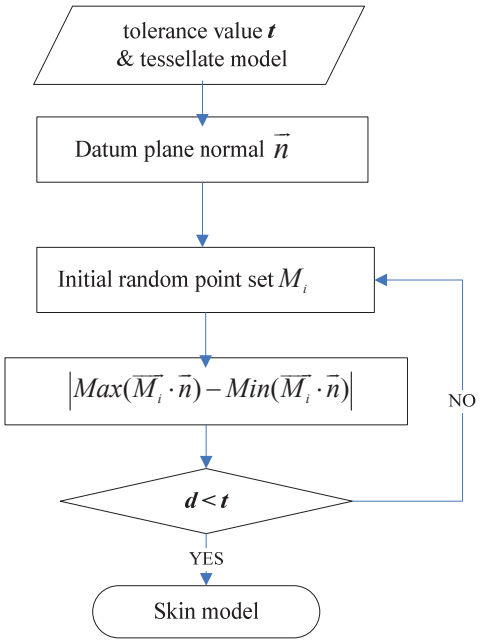


Figure 3-22: process flow of parallelism specification

3.4.4 Position specification

Based on the ISO standard [ISO 2005], the position tolerancing is used for locating features. It defines a zone within which the feature is permitted to vary from a true (theoretical exact) position. Basic dimensions establish the true position from the specified datum features as well as the relationships between the features (see figure 3-23). The principle of this method is described in figure3-24.

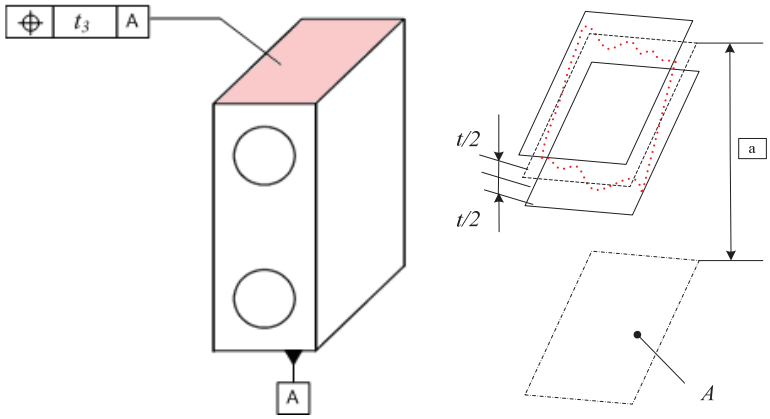


Figure 3-23: Position tolerance zone

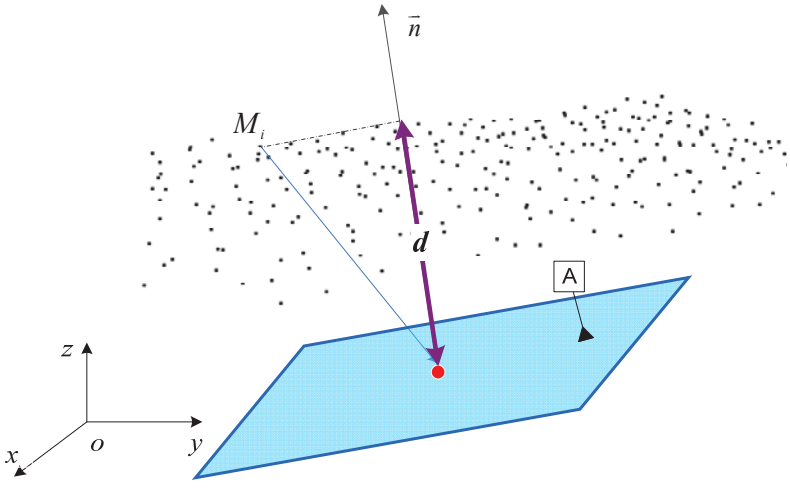


Figure 3-24: Position constraint

For position specification case, the tolerance direction follows the normal direction of the datum plane. The related constraint is described by formula 3-23.

$$d \in [a - \frac{t_{position}}{2}, a + \frac{t_{position}}{2}] \tag{3-23}$$

Where $\vec{n}$ is the normal direction of the datum plane, M_i is an arbitrary point, and d is the distance between the point M_i to datum plane P_A in the direction of vector $\vec{n}$, and a is the nominal distance value of position tolerance (true position). The process flow is then illustrated in figure 3-25.

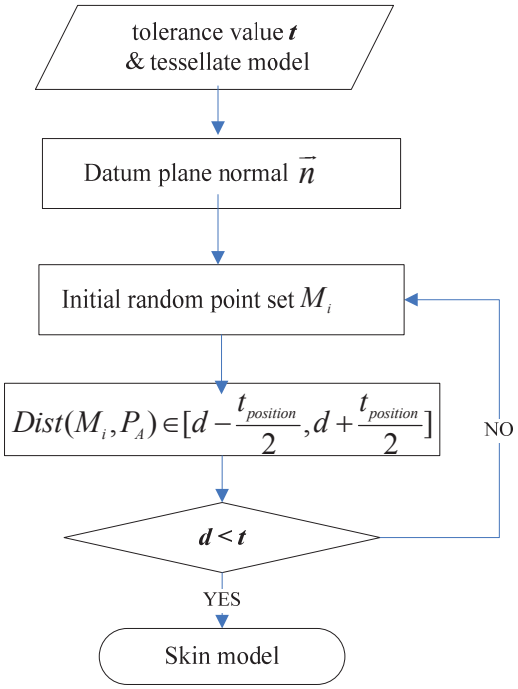


Figure 3-25: Process flow of position specification

3.4.5 Other constraints

(1). Combining constraints

The constraints on the skin model discussed above are based on the simulation of pure random errors. In general, the realistic part not only contains the random geometrical errors (uncorrelated errors) but also the systematic geometrical errors (correlated errors). In order to ensure that the skin model satisfies both correlated and uncorrelated errors, we propose to improve the constraints discussed before when combined.

Considering that the form specification is independent on any other conditions, it can be constrained first when there are also other tolerance specifications. Since the PCA method can identify the principle axis, the bounding box of the point set is then

determined, and the principal directions uncorrelate the data. Systematic geometrical errors thereby can be well described by this bounding box thanks to the position of the centroid and the principal directions orientations through a rigid transformation.

The constraints added on the point set which combine the systematic geometrical errors are the orientation and the position specification requirements. To sum up, the constraints with systematic deviation method can evaluate the constraints combined by form, orientation and position specifications.

The input of this method is a tessellated model obtained from CAD software, and based on the tolerance specification requirement the user input the constraint values, such as form specification, orientation specification, position specification, datum plane, even any combination of these specifications, and also a rotation value that the maximum value can be easily derived from the dimensions and the tolerance zone of the toleranced surface. The rotation value is used to rotate the bounding box of the point set to consider some systematic errors. Based on the input information, the initial random point set M_i can be generated by the skin model simulation method, which was introduced in section 3.3. Considering the form specification is independent on datum plane and has always a smaller constraint value than orientation and position specification, we test if the point set is within the form tolerance zone first. If the point set does not satisfy the limit value, we iterate the skin model simulation procedure until it's satisfied, else we continue the PCA evaluation step. We apply then PCA method to the point set M_i to evaluate the principal axis, which indicates the main variation direction of the surface variation. Then we rotate the point set M_i with the input angle value α, and then we evaluate if the point set M_i satisfy other constraints such as orientation and position specification requirements, which depend on the input information. If the point set M_i can satisfy the orientation and position constrains, then we can get the final skin model, otherwise we go back to the step of initial random point set generation and follow the procedures until all the conditions are satisfied.

To rotate the point set M_i in a certain angle following a principal axis, we adopt here rigid transformation calculation of coordinate systems. Assuming that

$C_1(o_1, \vec{x}_1, \vec{y}_1, \vec{z}_1)$ and $C_2(o_2, \vec{x}_2, \vec{y}_2, \vec{z}_2)$ denote the global coordinate system and the local coordinate system of the point set M_i. Where, o_i are the centroids of the global coordinate system ($i = 1$) and local coordinate system ($i = 2$) respectively, and $\vec{z}_i$ are the three normalized principal axes of coordinate systems. Let p represents an arbitrary point in point set M_i, and $\overrightarrow{o_1 p_2}$ represents the coordinates of point p in local system C_2. All the notations are described in figure 3-26. According to the relationship is shown in figure 3-26, the transformation of points from the global coordinate system to local system can be represented as formula 3-24

$$\overrightarrow{o_2 p_2} = \overrightarrow{o_2 o_1} + R_{12} \bullet \overrightarrow{o_1 p_2} \tag{3-24}$$

Where, $\overrightarrow{o_i o_j}$ $(i, j = 0,1)$ denotes the translation vectors from the origin o_i to origin o_j. R_{ij} $(i, j = 0,1)$ denotes the rotation matrix from the coordinate system C_i to the coordinate system C_j. In our method, the rotation matrix R_{ij} is the direction cosine matrix, which is specified as formula 3-25

$$R_{ij} = \begin{bmatrix} \cos(\vec{x}_j, \vec{x}_i) & \cos(\vec{x}_j, \vec{y}_i) & \cos(\vec{x}_j, \vec{z}_i) \\ \cos(\vec{y}_j, \vec{x}_i) & \cos(\vec{y}_j, \vec{y}_i) & \cos(\vec{y}_j, \vec{z}_i) \\ \cos(\vec{z}_j, \vec{x}_i) & \cos(\vec{z}_j, \vec{y}_i) & \cos(\vec{z}_j, \vec{z}_i) \end{bmatrix} \tag{3-25}$$

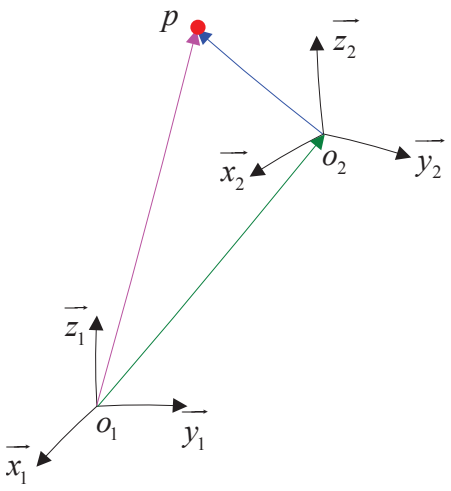


Figure 3-26: Translation between two coordinate systems

When the point set M_i is rotated with a certain angle, we need to translate its local coordinates to global coordinates back. The solution is adopting the inverse matrix, and the process can be explained by formula 3-26

$$\overrightarrow{o_1p_1} = \overrightarrow{o_1o_2} + R_{21}^{-1} \bullet \overrightarrow{o_2p_1} \tag{3-26}$$

Where, $R_{21}^{\wedge}(-1)$ is the inverse matrix of R_{21}.

The flow chat of this method is illustrated by figure 3-27.

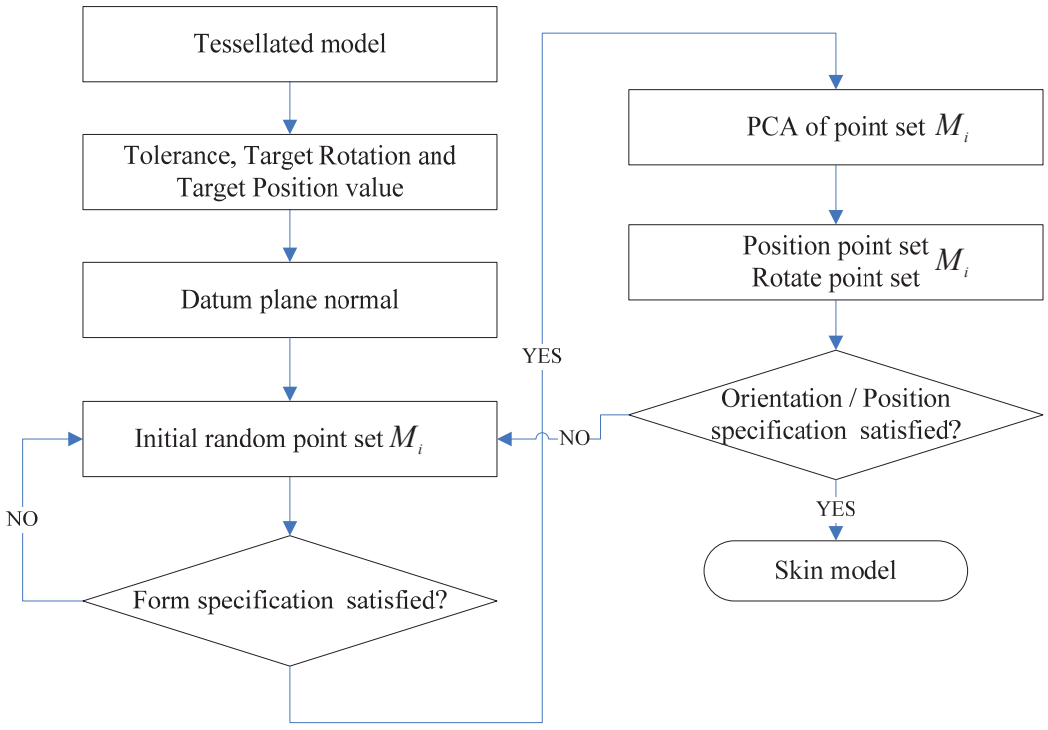


Figure 3-27: Process flow for combining constraints

(2). Cylindricity

Based on the ISO standard [ISO 2005], the cylindricity is the condition of a surface of revolution in which all points of the surface are equidistant from a common axis. The tolerance zone is two concentric cylinders within which the surface must lie (see figure 3-28).

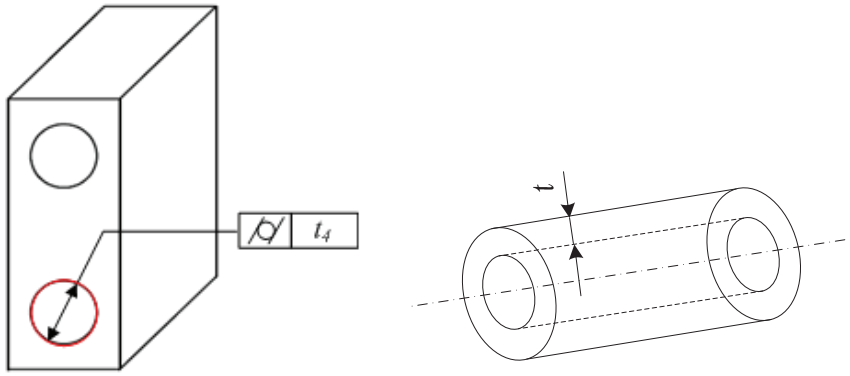


Figure 3-28: Cylindricity tolerance zone

The constraint of cylidricity of skin model is illustrated in figure 3-29. Figure 3-29 reflects the cylidricity constraint in a given 2D view. In figure 3-29, the black dot line is the nominal point set, and the red dot line illustrates the skin model of the cylinder. Considering the cylindricity specification, the skin model point set is restricted in the tolerance zone, which is illustrated by a zone of two coaxial cylinders . The distance d between the axis of the cylinder and the random point M_i is constrained by formula 3-27.

$$d \in [-t/2, +t/2] \tag{3-27}$$

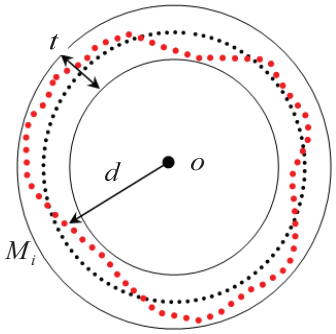


Figure 3-29: Cylindricity tolerance zone

3.5. Comparison and statistical analysis of simulation methods

3.5.1 Vertex normal direction deviation test

To compare skin model simulation methods described above, we create several

sets of skin model of the tolerance plane described in section 3.3. The characteristic we evaluate here is the deviation between random (simulated) and nominal (Tessellated CAD) points. This test is a very common technique used by inspection software to compare a set of measurement data to a reference CAD model or to some specific target points described by their coordinates on a given reference frame. .

In figure 3-30, the blue points are the nominal points, and the black points are the random points that compose the skin model. The vertex normal of nominal point is $\vec{n}_i$. The deviation evaluated in our method is the signed distance d_i, which is obtained through the projection of the vector linking the nominal point M_i to the random point R_i.

The test is based on sets of skin models using three different skin model generation methods.

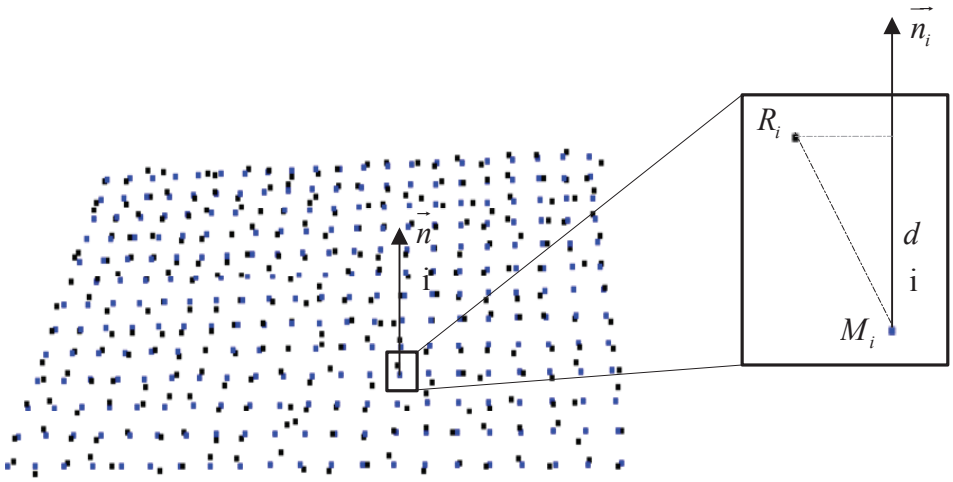


Figure 3-30: Principle of vertex normal direction deviation test

In this testing, the skin model is composed of 273 points. The flatness tolerance considered here is equal to 0.02mm, the parallelism tolerance is equal to 0.1mm, and the position tolerance is equal to 0.3mm. The number of skin model generated varies from 100 to 1000.

Simulation results are illustrated in figure 3-31, the horizontal ordinate is the number of skin models and the vertical ordinate reflects the average mean value of vertices normal direction of skin models. The Gibbs method as reported in the literature shows a slow convergence capability with the increase of the samples or the number of skin models.

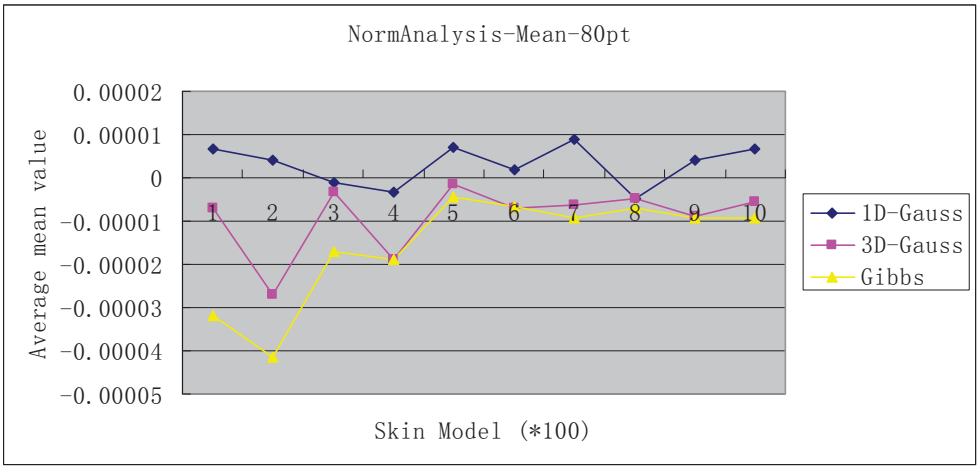


Figure 3-31: Mean value of vertex normal direction deviation

The average standard deviation between random and nominal points is also considered in our tests. It is also based on different number of skin models using the three different skin model creation methods. From figure 3-32 we can see that all of the three methods have a steady average standard deviation value, and Gibbs method has a smaller one in comparison to the other two methods.

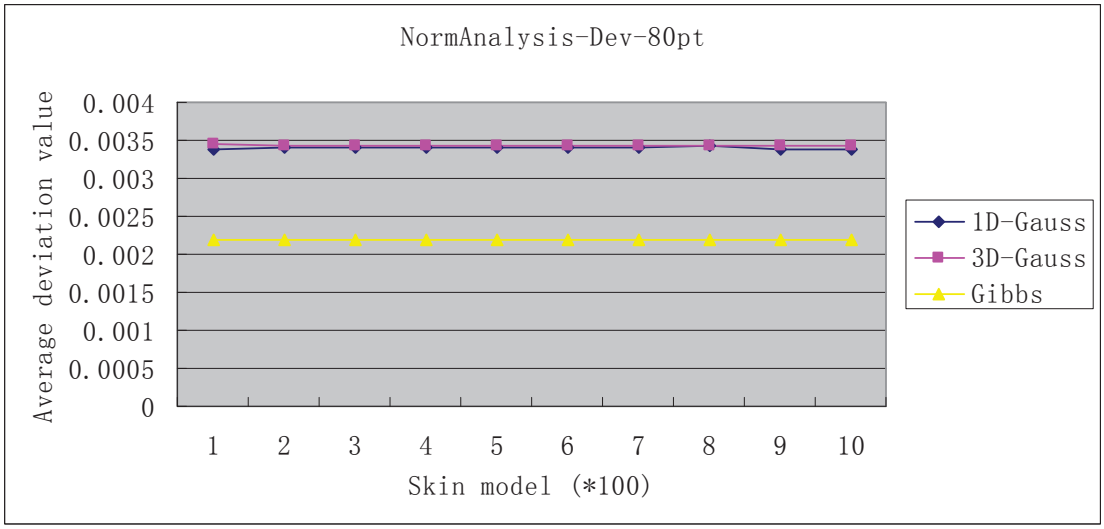


Figure 3-32: Mean standard deviation value of vertex normal direction deviation

More detailed analysis of standard deviation comparison is illustrated in figure 3-33 and figure 3-34. In figure 3-33, C1, C2 and C3 respectively represent 1D-Gaussian, 3D-Gaussian and Gibbs method. The spread of the samples of Gibbs method is smaller than the other methods, and some outliers of skin model generated by Gibbs method are even fewer. From figure 3-33 we can see that there are differences among the standard deviation at the 0.05 level of significance using

Minitab® statistical testing on standard deviations. Minitab® uses here the usual F-test [Brandt 83].

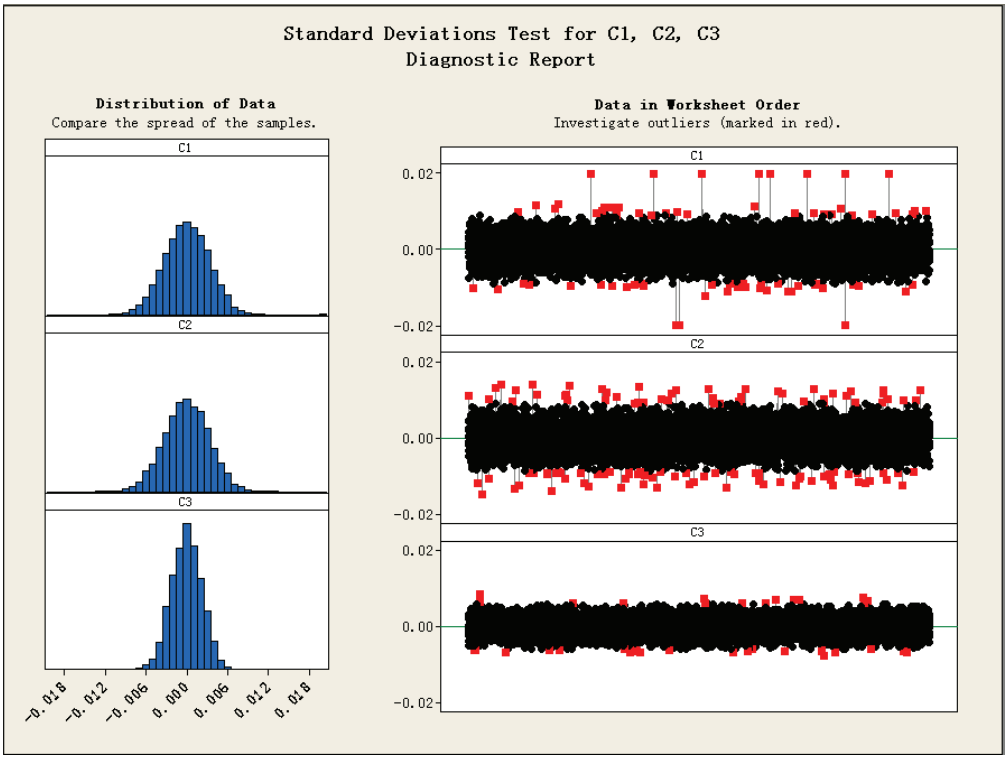


Figure 3-33: Distribution of the data of vertex normal direction deviation

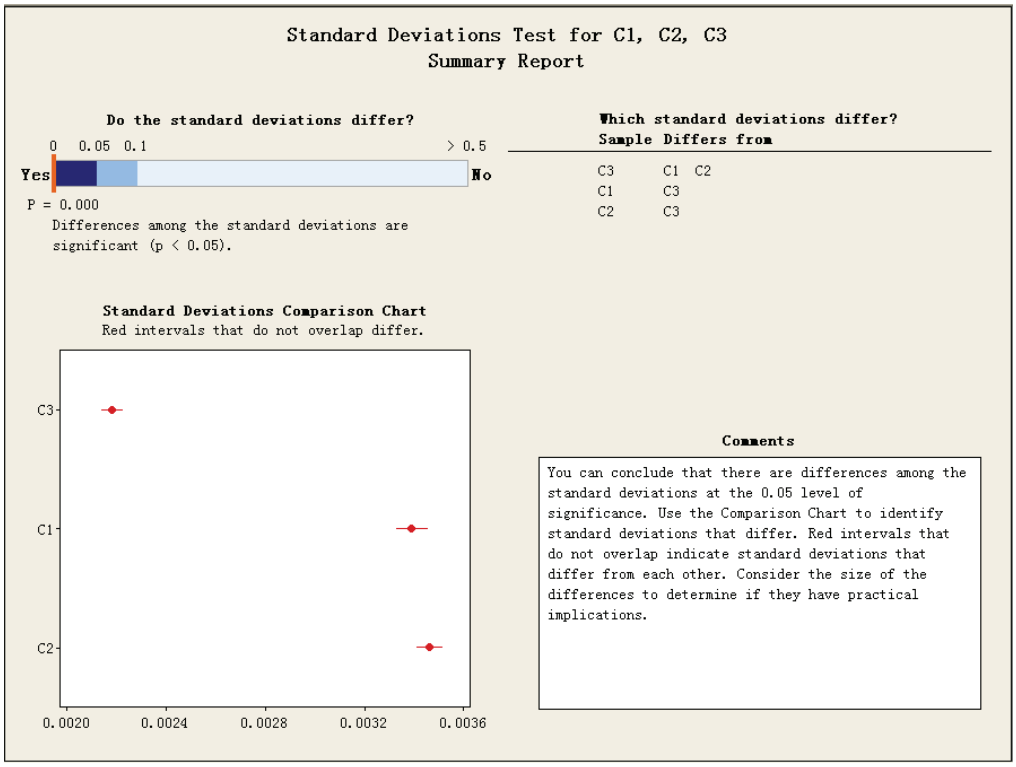


Figure 3-34: Standard deviations of vertex normal direction deviation

The statistical hypothesis test on the difference of standard deviations is a robust method of making decisions using our simulation data. The Gibbs method even if it is limited as depicted in previous sections for slow absolute time processing and weak convergence has a lower spread of data around the mean. Standard deviation is the main statistical parameter to describe the uncertainty and the variability of simulated or measured data.

Monte Carlo Markov Chain methods through the Gibbs sampler used in our work show an interesting result on standard deviations related to our simulations. However, the statistical analysis performed here in the case of vertex normal deviation is still limited in consideration of the possible local "morphing" of the shape.

Deviation analysis here considers only the position of the simulated random point without taking into consideration the local effect of the shape in its neighborhood which can be considered through the normal computation. The following section introduces a new characteristic called volume deviation that considers the small unsigned volume around the nominal point and the simulated point.

3.5.2 Volume deviation test

The principle of this test is to evaluate the deviation of the small positive volume, which is composed by the vertex normal of random and nominal points. In figure 3-35, the point M_i is the nominal point which vertex normal is $\overrightarrow{n_{M_i}}$, and R_i is the random point generated form M_i and its normal is $\overrightarrow{n_{R_i}}$. The vector $\overrightarrow{M_iR_i}$ is obtained from the points M_i and R_i. Based on the scalar triple product which can be defined also as the as the determinant of the 3 × 3 matrix having the three vectors as its rows, the obtained scalar form is also the geometric interpretation of the (signed) volume of the parallelepiped defined by the given vectors as formula 3-28

$$Vol_i = \left| \left(\overrightarrow{M_iR_i} \wedge \overrightarrow{N_{R_i}} \right) \cdot \left(\overrightarrow{N_{M_i}} \wedge \overrightarrow{N_{R_i}} \right) \right| \tag{3-28}$$

The test performed here evaluates the differences among the volume of the parallelepiped on sets of skin models using the three different skin model generation methods described in previous sections.

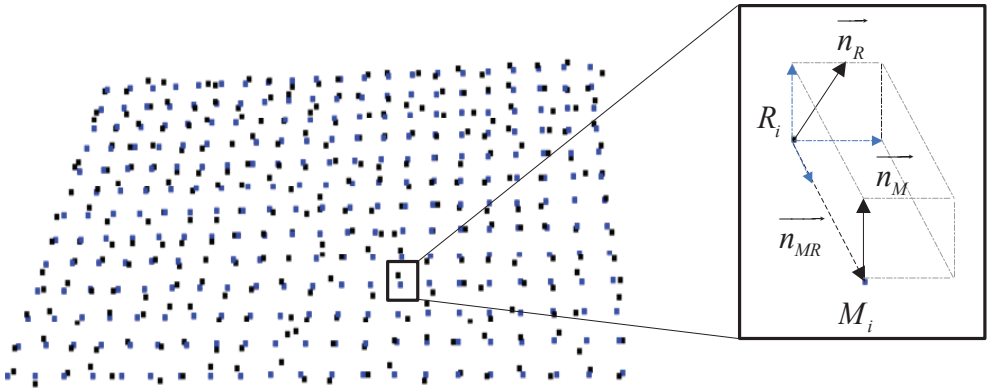


Figure 3-35: Principle of volume deviation test

In this testing, the skin model is composed by of 273 points. The flatness tolerance considered here is equal to 0.02mm, the parallelism tolerance is equal to 0.1mm, and the position tolerance is equal to 0.3mm. The number of skin model generated varies from 100 to 1000.

Simulation results are illustrated in figure 3-36, the horizontal ordinate is the number of skin model and the vertical ordinate reflect the average mean value of the volume deviation of skin models. The average volume deviation is quite steady among each skin model creation method, and the Gibbs method offers a smaller average value in comparison to the two other methods.

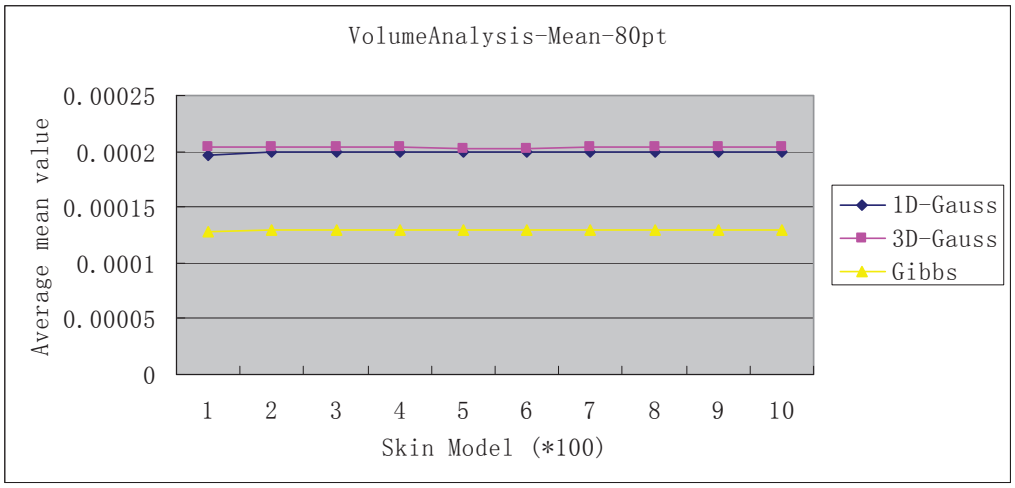


Figure 3-36: Mean value of volume deviation

The average volume standard deviation is also considered in this test .The results are illustrated in figure 3-37. We can see that all of these three methods have a steady standard deviation value, and the Gibbs method provides again the smallest value.

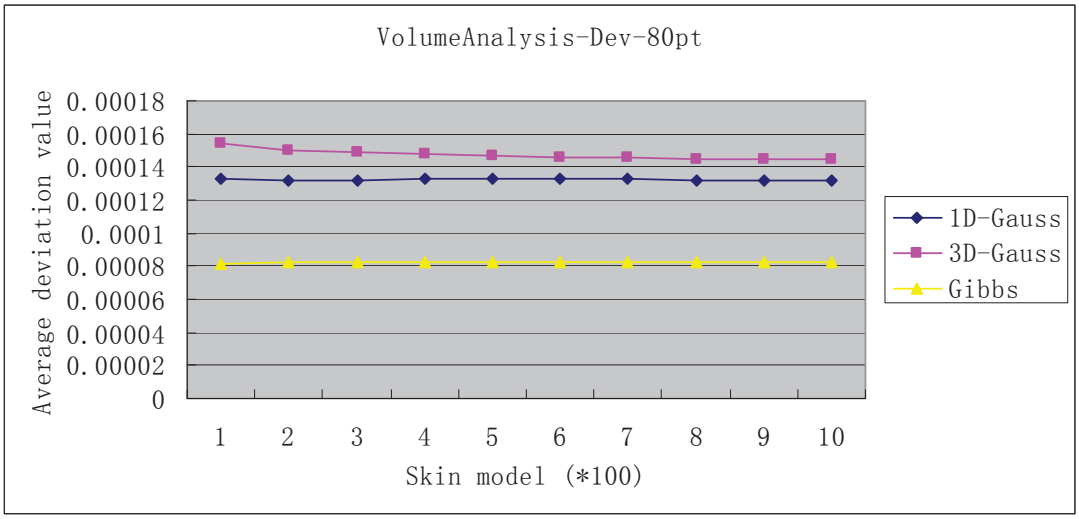


Figure 3-37: Mean standard deviation value of volume deviation

More detailed analysis of standard deviation comparison is illustrated in figure 3-38 and figure 3-39. In figure 3-38, C1, C2 and C3 respectively represent 1D-Gaussian, 3D-Gaussian and Gibbs method. . The spread of the samples of Gibbs method is smaller than the other methods, and the outliers of skin model generated by Gibbs method are fewer.

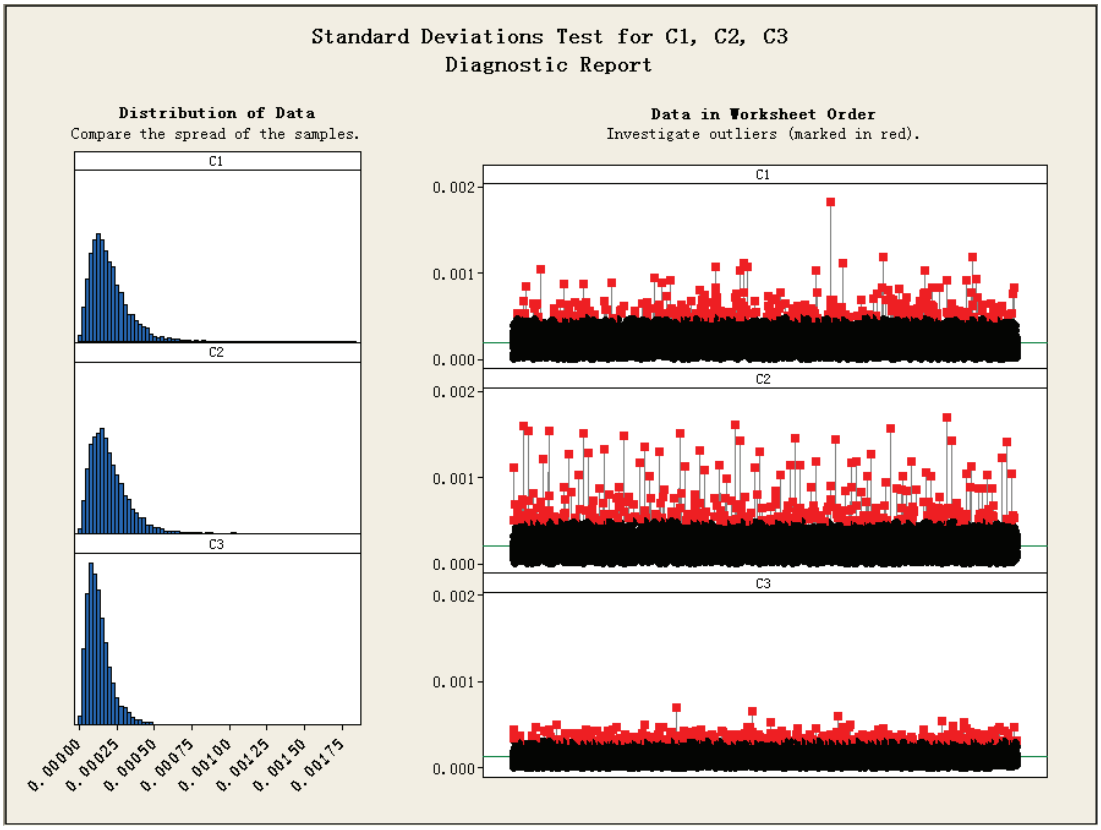


Figure 3-38: Distribution of the data of volume deviation

From figure 3-39 we can see that there are differences among the standard deviation at the 0.05 level of significance using Minitab® statistical testing on standard deviations. Minitab® uses an alternative to the usual F-test, the Levene/Brown-Forsythe test that does not require that the data are normally distributed [Lim 96]. This test performs well for both normal data and non-normal data, even when the sample sizes are small.

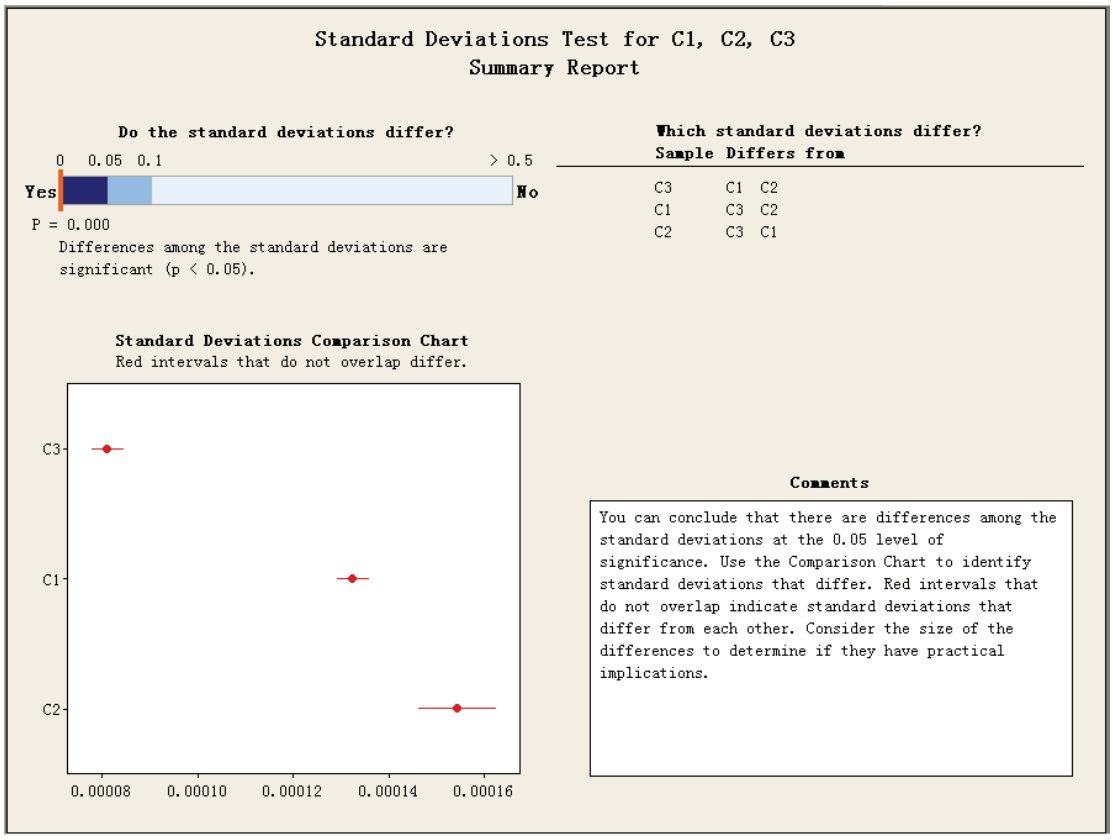


Figure 3-39: Standard deviations of volume deviation

. From the statistical analysis perspective, the Gibbs method scores well as mentioned in the previous section. It's also important to notice that this method provides lower standard deviations in the case of non-normal distributions.

The unsigned volume deviation distribution shows some similarities to the Rayleigh distribution or the one-sided distribution for form deviations [Sombstay 58]. MCMC methods are well known for their robustness even for non-normal and unknown probability density functions.

The volume deviation test considered here takes into consideration both initial CAD/nominal point and the random generated point as well as normal vectors. This new characteristic highlights also the "quality" of normal estimation for simulation.

3.5.4 Improving the visualization of the skin model

Many techniques have been proposed to show non-ideal geometry effects and uncertainties in data visualizations [Forslund 09] [Sanyal 09]. However, their effectiveness in conveying meaningful information is not well investigated. A large variety of techniques are currently used in visualization applications, selecting the most appropriate visualization technique for a given data set is a non-trivial task. Different approaches have been designed for different types of data such as vector fields and particle systems [McLouglin 10].

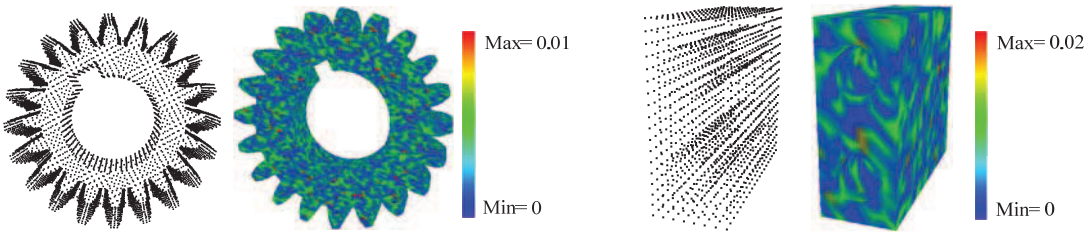
The deviations to a CAD or nominal model can be visualized on a skin model. Those deviations are much smaller than feature size and it remains difficult to visualize them within a multi-scale geometry by human eyes directly. Therefore, helping the user observes and analyzes the differences between the nominal model and the simulated skin model is an important task. The mesh or facet representation is well used for scientific visualization but cannot solve this problem directly. We propose here to improve the mesh visualization by adding a color scale technique to reflect the computed deviations.

Considering the two previous deviation tests, we adopt here two visualization possibilities: visualization based on vertex normal direction and visualization based on volumes.

(1). Visualization based on vertex normal

This section proposes to use RGB color scale mapping technique to visualize the geometrical deviations on the vertex normal direction. The basic idea of this method is that the extreme of deviations of the skin model are parallelized to a continuous color strip, which varies from the red color red to the blue.

The geometrical deviations between the simulated skin model and the initial point set are computed by the deviation in the direction of the vertices normal. The vertex normal estimation method was introduced in section 3.2.4. A continuous RGB color scale is then used to reflect the deviations on a skin model. Figure 3-40 illustrates two examples. The first one is a gear model and the second one is a, comparing the skin model represented by point set and color scale, we can see that the later is more convenient for user to identify the deviations.

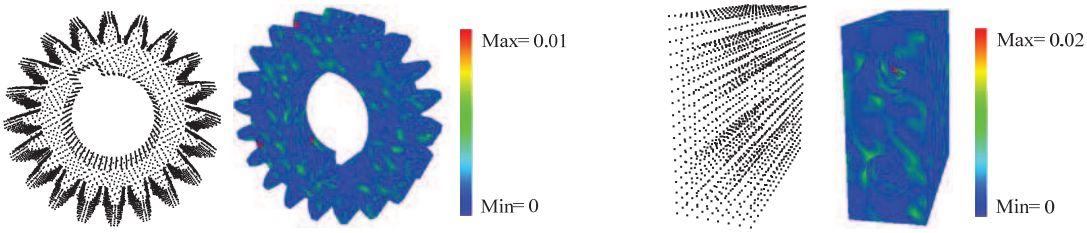


(a) Gear skin model (b) Pedestal skin model

Figure 3-40: RGB visualization based on vertex normal

(2). Visualization based on volume

The deviation in the direction of the vertex normal of a skin model can be reflected by a continuous color strip; moreover, we propose here to use the volume deviations as defined in section 3.5.2 that can also be reflected by continuous color strip. The basic idea is that since we have the knowledge of the volume deviation on each vertex of a skin model, the different colors can correspond to different volume values. In our method, the blue color reflects the smallest volume value while the red color reflects the biggest value, and from blue to red there is a continuous color scale (figure 3-41).



(a) Gear skin model (b) Pedestal skin model

Figure 3-41: RGB visualization based on volume

3.6. Conclusion

Skin model construction is one of the critical issues in GeoSpelling. This chapter proposed the developed methods to shape the skin models with random deviations. Based on the hypothesis that the random deviations follow the normal distribution, three different methods are developed to generate the skin models with discrete representations: 1-D Gaussian-based method, multi-Gaussian-based method and Gibbs-based method.

At the beginning, the nominal shape is tessellated as discrete representations (point-based and/or mesh-based). All the developed methods use this discrete nominal shape as reference but adding random deviations by different ways. For each given reference point, 1-D Gaussian method adds the random deviation along its normal direction based on 1 dimensional Gaussian possibility distribution; Multi-Gaussian method considers the random deviations in three dimensional space; Gibbs method calculates the new position of the given point by Markov Chain Monte Carlo simulation.

The nominal shapes are usually specified with geometrical tolerances to indicate the acceptable geometrical variations from the nominal shapes. The specified tolerances can acts as the constraints to fix the generated skin model within the tolerance zones. The mentioned three methods (1-D Gaussian based method, Multi-Gaussian based method and Gibbs based methods) are modified to generate appropriated skin models with the specified tolerance constraints.

Finally, a comparative analysis of these methods is discussed. The random error is considered in this chapter and the systematic error-based skin model simulation will be discussed in the next chapter

Chapter 4

Statistical Shape Analysis for Skin Model

4.1. Introduction

Because of the limited accuracy of applied manufacturing machines and tools, the skills of operators, the environments, etc., No product can be manufactured ideally without any errors. The error, indicated the variance of a real measured from its theoretical value, typically composes of random error and systematic error. The previous chapter, several methods are developed to simulate the skin model by adding random error to the nominal shape. As mentioned above, the skin model only considered random error is incomplete. The systematic error should also be simulated during the skin model shaping process.

Different from random errors which are statistical fluctuations (in either direction), systematic errors are reproducible inaccuracies that are consistently in the same directions. They usually keep the same value or follow the recognizable laws which can be calculated or simulated. In skin model shaping, it is thus reasonable to adding errors followed by some known basic shape models (e.g. second order shapes) to simulate the systematic errors. In this chapter, a set of methods are developed to simulate the systematic errors based on a group of second order shapes. In practice, the systematic errors can be simulated by adding deviations followed by a unique shape or by a combination of several shapes within the basic shapes' group.

With consideration of random and systematic errors, the shaped skin models are complete to represent the real geometrical variations. Given a nominal shape associated with specific geometrical tolerances, various skin models which satisfy the tolerance requirements can be generated to simulate the actual shape. A method based on SSM (Statistical Shape Model) is developed for skin model analysis. This is useful to evaluate the skin models and predict the trend of new skin models (actual shapes' trend prediction).

The chapter is organized as follows:

Section 4.2 reviews the literature related to variation based shape modeling methods. Section 4.3 presents the developed skin model shaping methods to simulate the systematic error. Section 4.4 gives the developed method based on SSM for skin model analysis and evaluation. Section 4.5 is the conclusion.

4.2. Shape deviations

It's well understood that shape deviations are the fundamentals of geometric product specification and verification. The two main axioms described in previous chapters highlight the combined effects of manufacturing variability and measurement uncertainty when considering a global view and consistent modeling of the skin model.

In order to enrich the skin model when considering the deviations from the nominal or CAD model, we need to assess the geometrical deviations at many different scales and from the manufacturing and the measurement perspectives.

In an early work, Dowling et al. [Dowling 97] pointed-out the lack of systematic manufacturing error considerations when assessing form deviations. Caskey et al. [Caskey 90] discussed the need for a catalog of typically occurring form errors for basic features produced by various manufacturing processes.

Manufacturing and measurement geometric deviations or errors can be classified into two main categories, which are systematic and random [Henke 99] [Desta 03]. The systematic components called also bias are deterministic, predictable and reproducible. They are caused by imperfect manufacturing devices or measurement procedures and by the ambiguity on the use of actual rules and standards. The random components are caused by non-constant sources of error either in time or in space. Random errors are expressed by statistical methods.

Research and industry reported a number of sources of manufacturing error and deviation such as locating error, clamping error, datum feature (surface) error, workpiece elastic deformations, tool wear, vibration, misalignment and machine guidance errors as well as thermal disturbances and so on. The systematic errors are generally easier to identify, isolate and correct than random errors.

Manufacturing processes leave very specific patterns or signatures on the part. In some cases, it is possible to quantify the manufacturing errors through mathematical models. In general cases, the manufacturing errors can be captured using experimental analysis. Colosimo et al. [Colosimo 07] define the manufacturing signature as the systematic pattern which characterizes the features realized by a given process in given conditions. This systematic pattern is "masked" by additional random noise caused by the natural variability of the process. A statistical analysis is then required

to obtain the manufacturing signature estimate.

Many efforts have been made to simulate the systematic deviations of shapes. All of these works can be roughly classified into three categories: analytical approaches, statistical analysis and other methods.

Henke et al. [Henke 99] introduced two models to describe form errors. One is an analytical model based on Chebyshev polynomials to model the axial errors and the other is Fourier series to model angular dependencies. Based on the Fourier series, Cho and To [Cho 01] proposed a harmonic roundness model to represent form deviations using Fourier series expansion. Mesay et al. [Mesay 03] represented systematic form error using a combination of Fourier components, and proved that this method can characterize a wide variety of the deviations.

Huang and Ceglarek [Huang 02] proposed a discrete-cosine-transformation (DCT) method which is based on decomposing the error field into a series of independent error modes. Fourier series is the first used decomposition method used to define periods on circular [Cho 02].

Samper [Samper 07] proposed a new way to define for error parameters based on the eigen-shapes of natural vibrations of surfaces. The originality of this method is that the set of form parameters can be computed for any kind of shape. The principle of this method is that the geometry variation is defined by using a geometrical basis built on the model shape, which have vectorial space properties and are defined by using a discretization of the ideal surface. Based on the value of the associated natural frequency, the model shapes are sorted with the different complexity orders, and this can help to analyze the form, undulation and roughness of a surface by the level of period of a flexure behavior. Based on this idea, Samper et al. [Samper 10] developed software based on (Finite Element Analysis) FEA to determine the model shapes.

Kurfess and Bands [Kurfess 95] modeled the manufacturing error using a sequence of models based on statistical hypothesis testing of the fitted residuals. Yang and Jackman [Yang 00] evaluated form error using statistical method without independently analyzing the statistical properties of the associated residuals.

Yang and Menq [Yang 95] reviewed form error as a deterministic component and random component. The random error was assumed to be spatially independent with a normal distribution for uncertainty parameters representing coordinate transformation elements. The deterministic component was supposed as the orthogonal deviation

between nominal surface and the artificial surface, which obtained by the measuring machine.

4.3. Systematic Deviations simulation method

As discussed before, the systematic deviations are caused by many sources such as the inherent properties of manufacturing and inspection machine, the environment condition and even the operators. Thereby the possibilities of the systematic form errors are various. Dowling [Dowling 97] argued that incorporating the systematic manufacturing errors into a modeling and assessment procedure is an important issue.

There have been already some efforts attempted on expression of form deviations based on shapes decomposition concept, and these work can classified into two categories [Samper 07]: explicit and implicit parameterizations. The explicit description of geometries is based on CAD modeling [Cubeles-Valade 98], which can be modified by limited parameters. The implicit geometries are defined by a parametric model, which can be discretized with elementary geometries [Gouskov 98]. Both these two kinds of methods are using the modification of parameters to obtain the form deviations.

In our method, we proposed using second order form deviations to simulate the systematic errors, since the second order form deviations can reflect the principle curvature and the anisotropy of complex shapes better than the first order and higher order deviations [Kurokawa 05]. Kurokawa [Kurokawa 05] proved mathematically that an arbitrary second order surface can be transformed into a fundamental form of the second order by the combination rotation, translation and scaling transformation. Based on this argument, we proposed the systematic errors defined in our method are one or a combination of some basic second order surfaces, and the details are explained below.

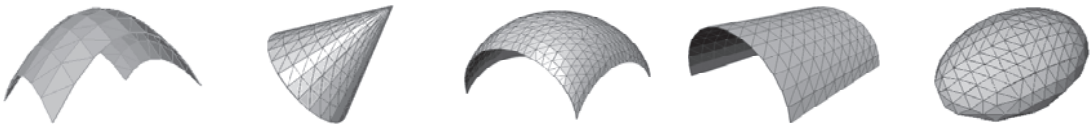
4.3.1 Plane deviation simulation

The systematic distortions of planar parts are based on the following hypothesis.

(a) The systematic error of a planar part can be described by one or more basic geometric shapes.

(b) These basic geometric shapes belong to quadric surfaces and are classified into four families: paraboloid, cone, sphere, cylinder and ellipsoid (see figure 4-1), and the morphing operation is a combination of translation and rotation of the basic shapes.

(c) And each simple basic shape is a result of the deviations on the vertex normal direction of a discrete plane.



(a) Paraboloid (b) Cone (c) Sphere (d) Cylinder (e) Ellipsoid

Figure 4-1: Types of systematic error of planar shapes

The principle of morphing to the basic geometric shape is that, the deviation can be assumed following the vertex normal direction. For each case, the morphing details are explained as follow.

(1). Paraboloid

In mathematics, a paraboloid is a quadric surface and there are two kinds of paraboloid: elliptic and hyperbolic. In our work we focus on the elliptic paraboloid, which can be represented using formula 4-1

$$z = \frac{x^2}{a^2} + \frac{y^2}{b^2} \tag{4-1}$$

Where a and b are constants that dictate the level of curvature in the x-z and y-z planes respectively.

In our method, the paraboloid surface is simulated from the discrete model of a plane. The principle of our method can be explained by figure 4-2, in which S_p denotes the paraboloid point set and I_p denotes the initial planar point set. We use PCA algorithm (see section 3.4.1) to evaluate the principal axis to obtain the local coordinate system ofc. Suppose p_i is a random point of I_p, and q_i is its corresponding point of S_p. The distance between these two points in p_i's vertex normal direction is reflected using h_i, which fulfils the formula 4-1.

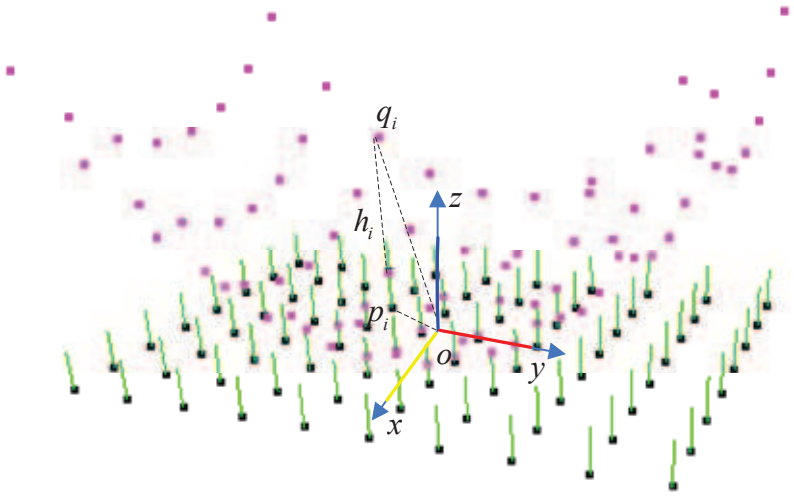


Figure 4-2: Paraboloid simulation method

During this process, we employ the third principal axis (the z axis in figure 4-2) as the z value to calculate the deviation. According to the PCA algorithm (introduced in section 3.4.1), the third principal axis represents the smallest variety of the planar shape, which can be calculated by the discrete curvature method introduced in section 3.2.5. The simulation result is illustrated in figure 4-3.

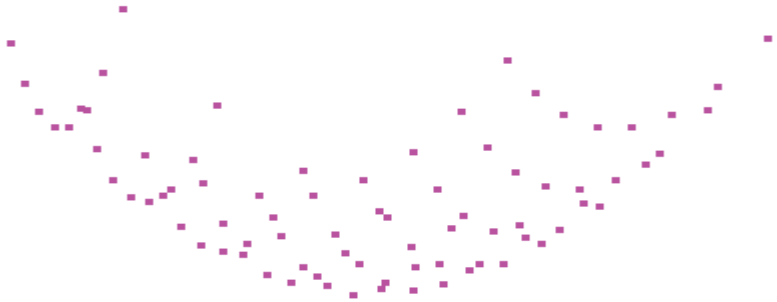


Figure 4-3: Paraboloid simulation result

(2). Cone

A cone is a quadratic surface. In figure 4-4 the height h and the base radius r oriented along the z-axis and with the base located at $z = 0$ (denoted as z_0), the cone shape can be described by formula 4-3

$$\frac{x^2 + y^2}{c^2} = (z - z_0)^2 \tag{4-3}$$

Where, (x, y, z) is the Cartesian coordinate value, and c is the ratio of radius to

height at some distance from the peak $c = \frac{r}{h}$.

In our method, we simulate the cone shape form a discrete plane, which is tessellated from a CAD model of plane surface. The principle of the method can be explained by figure 4-4.

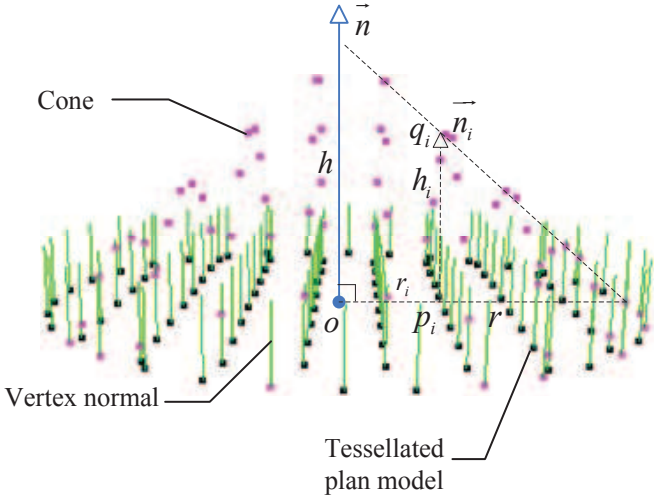


Figure 4-4: Cone simulation method

Let I_p denotes the initial planar point set and S_c denote the cone point set. The cone simulation method is based on the hypothesis that the shape can be reflected by the deviation of point set in the direction of each vertex normal. Considering the cone shape displayed in figure 4-4, the peak of the cone is located in the center of I_p with the height h and radius r. The deviation h_i of a random point p_i in its vertex normal direction $\vec{n_i}$ can be estimated by formula 4-4

$$h_i = h \cdot (1 - \frac{r_i}{r}) \tag{4-4}$$

Where r_i denotes the distance between the center of I_p and the point p_i.

During this process, the deviation of each point p_i is calculated based on the height h, which is in the normal direction $\vec{n}$. To simulate the cone shape (see figure 4-4) we construct the triangle by the peak of cone, the radius of I_p, and the center of I_p. The value h_i is obtained from the proportional calculation of this triangle. This estimation is based on the assumption that I_p and $\vec{n_i}$ follow the same direction in the space. Considering I_p is a normal tessellated plan model, the normal $\vec{n}$ and $\vec{n_i}$ has a small difference error ε, which coming from the tessellated techniques. We compared

these two normal values, and the value ε is approximately equal to 10^{-6}. The simulation result is illustrated in figure 4-5.

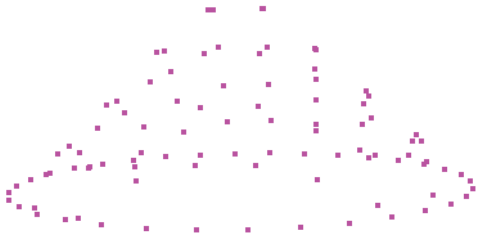


Figure 4-5: Cone simulation result

(3). Sphere

In mathematics, a sphere with center (x_0, y_0, z_0) and radius r is the locus of all point (x, y, z) satisfying the formula 4-5

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \tag{4-5}$$

In our method, we simulate the sphere point set from the tessellated plan model. The principle of this method is explained by figure 4-6. As discussed above, I_p denotes the initial planar point set and S_s denotes the sphere point set. The sphere simulation method is based on the hypothesis that the shape can be reflected by the deviation of point set in the direction of each vertex normal. In figure 4-6, the deviation h_i of a random point p_i of I_p is determined by curvature radius and the distance from the center o of I_p as formula 4-6

$$h_i = \sqrt{(R^2 - d_i^2)} \tag{4-6}$$

Where, R is the curvature radius and d_i is the distance between the random point p_i and the center of I_p denoted as o. Let (x_i, y_i, z_i) and (x_o, y_o, z_o) denote the coordinates of p_i and o, and then $d_i = \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2 + (z_i - z_o)^2}$.

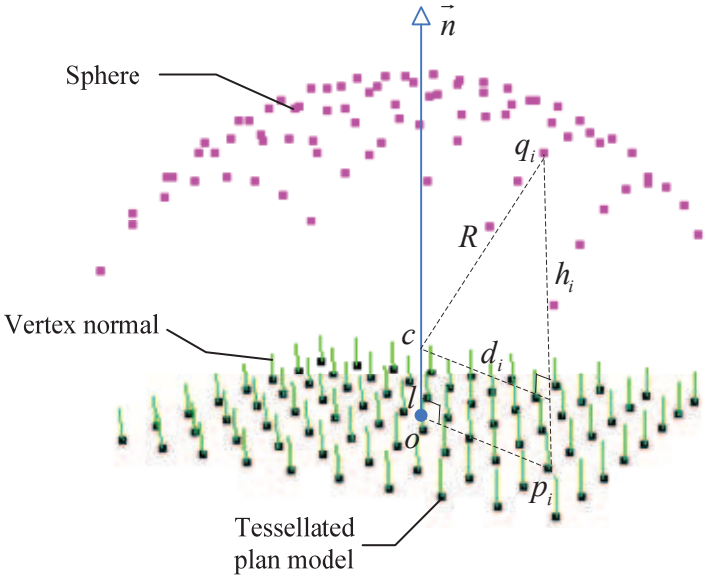


Figure 4-6: Sphere simulation method

Based on the input information including: tessellated plan, type of systematic error (sphere) and the curvature radius R, the offset of the center of sphere l, and then the sphere point set can be obtained. The simulation result is illustrated in figure 4-7.

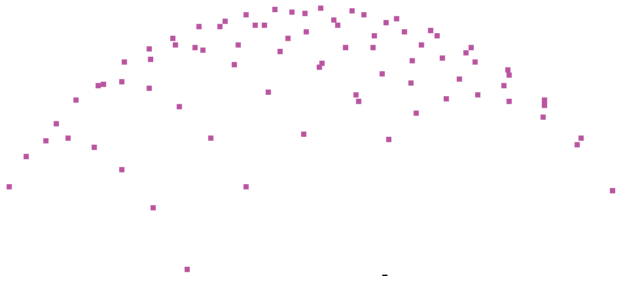


Figure 4-7: Sphere simulation result

(4). Cylinder

In mathematics, a cylinder is a quadric surface, which is formed by the points at a fixed distance called radius from a given line segment, the axis of the cylinder. With the radius r, a cylinder can be illustrated as in formula 4-7

$$(\frac{x}{r})^2 + (\frac{y}{r})^2 = 1 \qquad (4\text{-}7)$$

In our method, we simulate the cylinder point set from a planar tessellated model, and the principal of this method can be explained by figure 4-8.

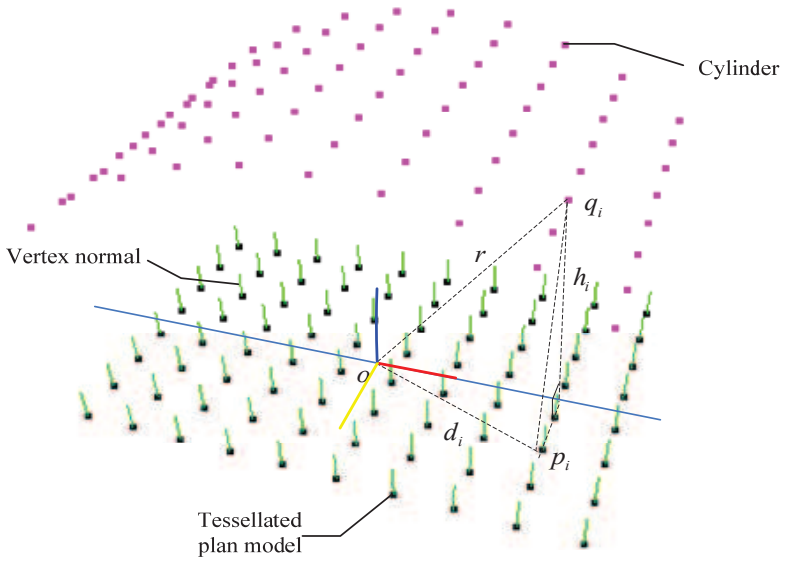


Figure 4-8: Cylinder simulation method

As defined before, I_p denotes the tessellated plane model and S_y denotes the cylinder point set. The PCA method is employed to obtain the center of tessellated plane denoted as o and the first principle axis x. Suppose p_i is a random point of I_p and d_i is the distance between p_i and o, based on the curvature radius r and the projection of d_i in the direction of x, the deviation of point p_i in its vertex normal direction h_i can be calculated. For each point in I_p, we calculate the corresponding deviation in its vertex normal direction, and then the S_y can be obtained. The simulation result is illustrated in figure 4-9.

Figure 4-9: Cylinder simulation result

(5). Ellipsoid

In mathematics, an ellipsoid is a type of quadric surface, and the equation of a standard axis-aligned ellipsoid boy is illustrated as formula 4-8

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \tag{4-8}$$

Where a and b are the equatorial radii (along x and y axes) and c is the polar radius (along the z -axis).

In our method, we simulate the ellipsoid point set from a planar tessellated model, and the principal of this method can be explained by figure 4-10. As defined before, I_p denotes the tessellated plane model and S_y denotes the cylinder point set. We use PCA algorithm (see section 3.4.1) to evaluate the principal axis to obtain the local coordinate system of I_p. Suppose p_i is a random point of I_p, dx_p and dy_p are the distance along the x -axis and y -axis. Based on the equatorial radii a,b,c, the deviation of point p_i in its vertex normal direction h_i can be calculated. For each point in I_p, we calculate the corresponding deviation in its vertex normal direction, and then the S_y can be obtained. The simulation result is illustrated in figure 4-11.

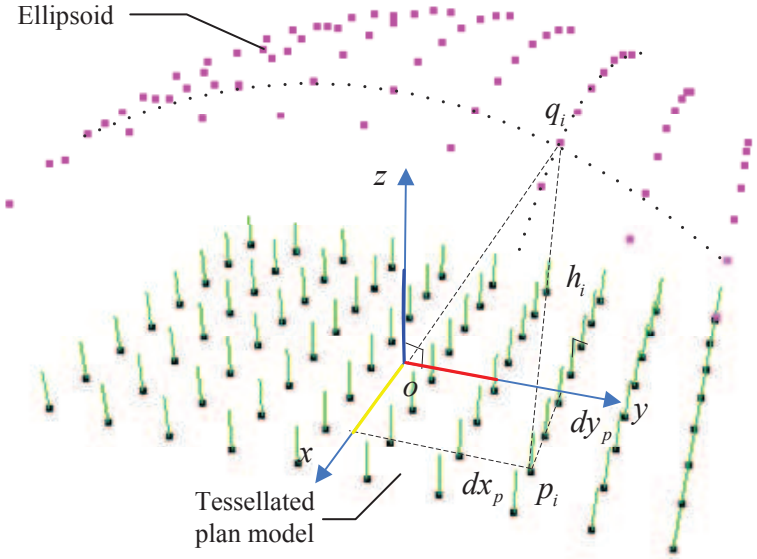


Figure 4-10: Ellipsoid simulation method

Figure 4-11: Ellipsoid simulation result

(6). Rotation

Rotation defined in our method is a rigid body movement of the point set following an axis. In our method, firstly we define a rotation axis, and then adopt a rotation matrix to obtain the rotation point set. In three-dimension space, a rotation matrix has a unit real eigenvalue, and based on the Euler's rotation theorem the rotation matrix can specify the rotation of the corresponding eigenvector. Suppose the rotation follows x axis, y axis, and z axis respectively in Cartesian coordinate system, and each of rotation is named as roll, pitch and yaw, and then the rotation matrices as formula 9 to formula 11:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{4-9}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{4-10}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-11}$$

Where, θ_x is the angle of roll, θ_y is the angle of pitch, θ_z is the angle of yaw.

The principle of our method can be explained by figure 4-12. I_p denotes the tessellated plan point set, and S_r denotes the rotation point set. Firstly using PCA method to estimate the center and the three principal axis of I_p, and then establish the local coordinate system as illustrated in figure 4-12. Suppose the rotation angle of y axis is θ_y, and a random point p_i of I_p has the distance d_i from the center of I_p, which is denoted o. q_i denotes the point of S_r, which is in the direction of vertex normal of p_i. Based on the right triangle relation among o, p_i and q_i, the deviation of each point of I_p can be calculated, and the point set S_r is obtained. The simulation result is illustrated in figure 4-13.

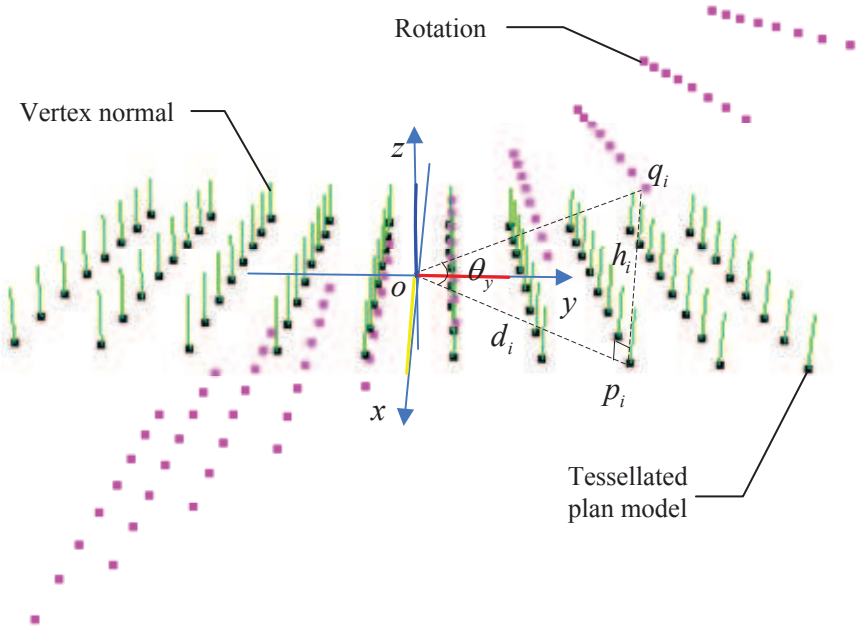


Figure 4-12: Rotation simulation method

Figure 4-13: Rotation simulation result

4.3.2 Cylinder deviation simulation

Based on [Zhang 05], in manufacturing the geometric deviation of a cylindrical part can be described by three independent terms: cross-section form error, axial form error and cross-section size error. The cross-section form errors are defined as the error of out of roundness of the part, and they are usually caused by spindle defects and fixturing distortion. The axial errors occur in a section plane that passes through the axis of the manufactured part. The sources of the axial errors include machine tools (e.g. spindle misalignment), workpieces (e.g. workpiece deflection and heat expansion), and machine fixtures (e.g. work centre misalignment), etc. The

cross-section size errors are equal to the difference between the actual and the predicted size (diameter or radius) of the manufactured part, and the most common raisons are set-up error and the tool wear. According to the classification of the geometric deviation of a cylindrical part, there are four typical manufactured errors: taper, concave, convex, and banana (see table 4-1).

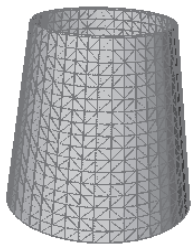
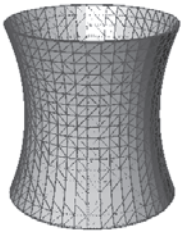
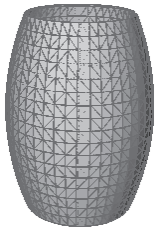
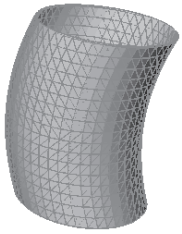
Taper	Concave	Convex	Banana
			

Table 4-1: Types of systematic error of cylinders

(1). Taper

In geometry, a taper is an object with a gradual thinning or narrowing towards one end. In our method, the taper is simulated form a tessellated cylinder model. The principle of this method is based on the hypothesis that the taper can be obtained by offsetting each point of tessellated cylinder model in the direction of its vertex normal, see figure 4-14. Let I_c denotes the initial tessellated plan and S_t denotes the taper point set. The point p_i is a random point of I_c, and q_i is its corresponding point of S_t with distance d_i in the direction of vertex normal of p_i. Considering the curvature of a taper is a linear variable, the deviation of point p_i can be calculated by the proportional relationship as in formula 4-12

$$d_i = (R - r) \cdot \frac{h_i}{l} \tag{4-12}$$

Where d_i is the deviation of p_i in the direction of its vertex normal, R and r are the limited curvature radiuses of the taper, l is the height of cylinder and h_i is the vertical distance between point p_i and one end of the cylinder. The simulation result is illustrated in figure 4-15.

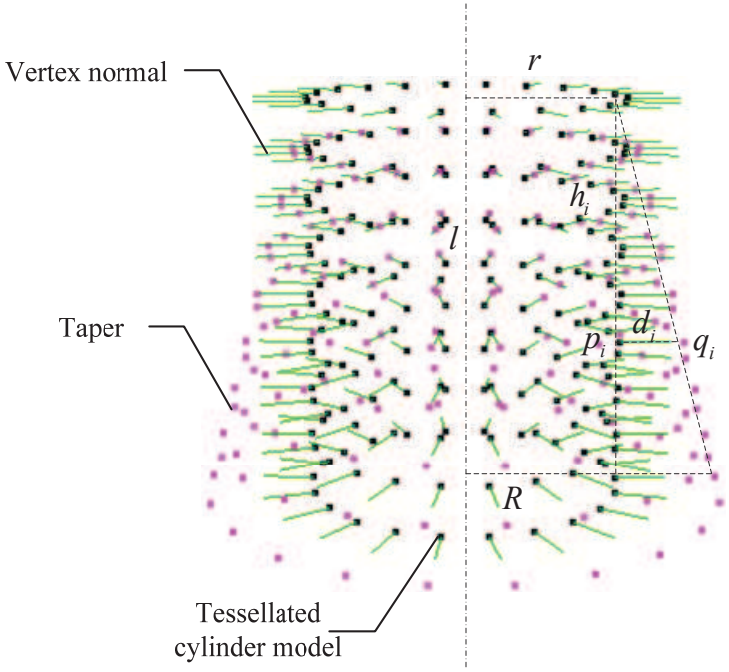


Figure 4-14: Taper simulation method

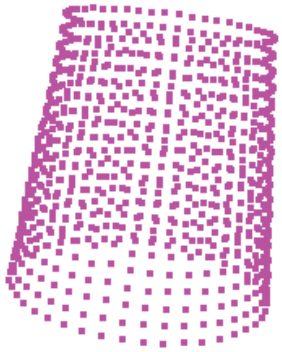


Figure 4-15: Taper simulation result

(2). Convex

In Euclidean space, an object is convex if for every pair of points within the object, every point on the straight line segment that joins them is also within the object. According to this definition, there are various types of convex models, such as solid cubes. While in our method, we simulate the convex shape following the ellipsoid model.

An ellipsoid is a closed type of quadric surface, the standard axis-aligned ellipsoid in a Cartesian coordinate system is interpreted by formula 4-13:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \tag{4-13}$$

Where a and b are the equatorial radii along the x axes and y axes and c is the polar radius along z axis.

In our method, we simulate the ellipsoid from a tessellated cylinder model. Let I_c denotes the initial tessellated cylinder point set and S_{cv} denotes the convex point set, and the principal of our method can be explained by figure 4-16.

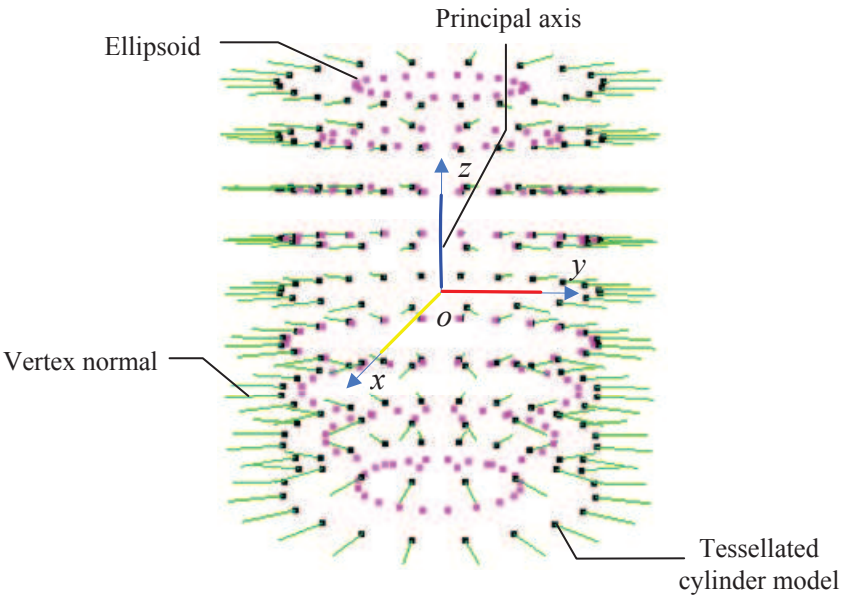


Figure 4-16: Ellipsoid simulation method

The PCA method is used to evaluate the three principal axes of I_c, which construct the local coordinate system of I_c. The local coordinate system is located as illustrated in figure 4-16. Since our method is based on the hypothesis that the ellipsoid can be obtained by offsetting the points of I_c in the direction of its vertex normal, we assign the deviation values for each point of I_c using the formula 4-13. The simulation result is illustrated in figure 4-17.

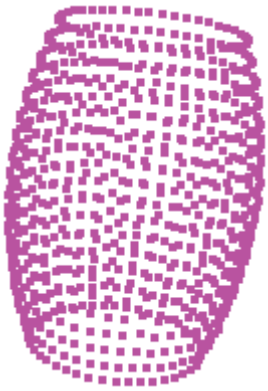


Figure 4-17: Ellipsoid simulation result

(3). Concave

The concept concave means curving in or hollowed inward, as opposed to convex. There are many kinds of concave shapes, while in our method we simulate the hyperboloid surface as the concave systematic error.

In mathematics, a hyperboloid is a quadric surface in three dimensions space, and can be described by formula 4-14:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1 \tag{4-14}$$

Where a and b are the equatorial radii along the x axes and y axes and c is the polar radius along z axis.

In our method, we simulate the hyperboloid from a tessellated cylinder model. Let I_c denotes the initial tessellated cylinder point set and S_{cc} denotes the convex point set, and the principal of our method can be explained by figure 4-18. The PCA method is used to evaluate the three principal axes of I_c, which construct the local coordinate system of I_c. The local coordinate system is located as illustrated in figure 4-18. Since our method is based on the hypothesis that the hyperboloid can be obtained by offsetting the points of I_c in the direction of its vertex normal, we assign the deviation values for each point of I_c using the formula 4-14. The simulation result is illustrated in figure 4-19.

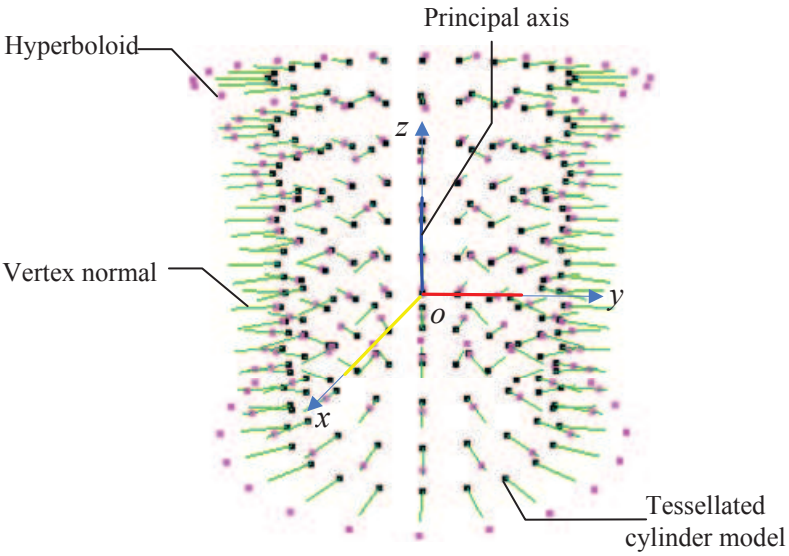


Figure 4-18: Hyperboloid simulation method

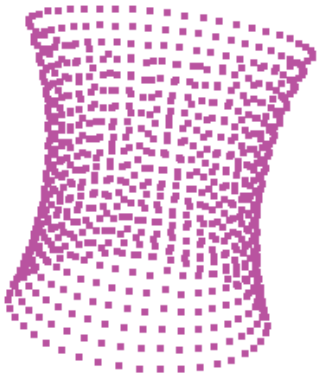


Figure 4-19: Hyperboloid simulation result

(4). Banana

The banana systematic error is named by the appearance of the form error of a cylinder. There is no simple analytical definition of a banana shape, while we propose using quadric surface to simulate it, such as, ellipsoid and hyperboloid etc. The banana simulation approach we implemented here is based on the hyperboloid surface, which can be described by formula 4-15:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1 \tag{4-15}$$

Where p is the distance from the focus and the directrix of the hyperboloid

surface.

We simulate the banana form error from a cylinder point set, and the principle can be explained by figure 4-20. For convenience, let I_c denotes the initial cylinder point set and S_b denotes the banana point set. Since the simulation should be independent on the spatial position of the cylinder, the local coordinate system of I_c should be considered. The PCA method is employed to estimate the three principle axes, which are deemed as the local coordinate axes. In figure 4-20, the S_b can be obtained by displacing the points of I_c in y axis direction following formula 4-15. The simulation result is illustrated in figure 4-21.

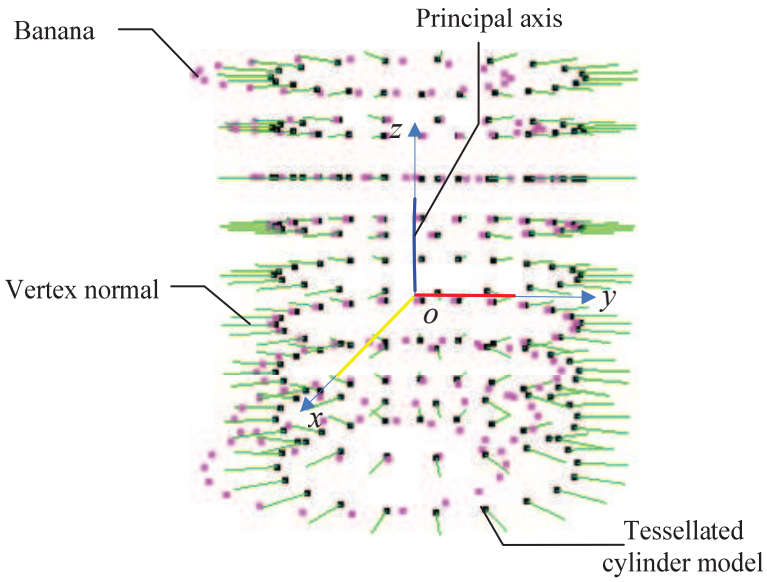


Figure 4-20: Banana-shape simulation method

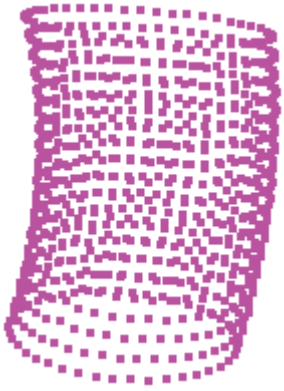


Figure 4-21: Banana-shape simulation result

4.3.3 Skin Model Simulation considering both systematic and random deviations

An actual geometric feature can be decomposed into three portions, which follows the shape of the feature as defined by its designed form; a spatially correlated portion (when the systematic error exists, the measurements in close proximity on a surface show strong correlation); and a spatially uncorrelated portion, i.e., the random error portion [Xia 08]. As such, we model the form error as arising from systematic errors and random errors added to an ideal geometric form, and there are two examples illustrated in table 4-2.

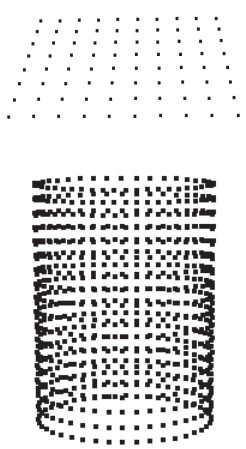
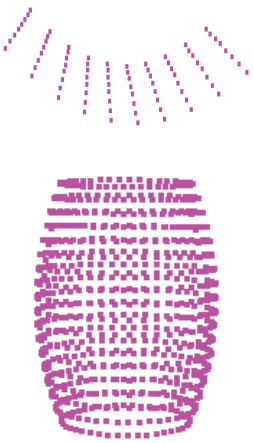
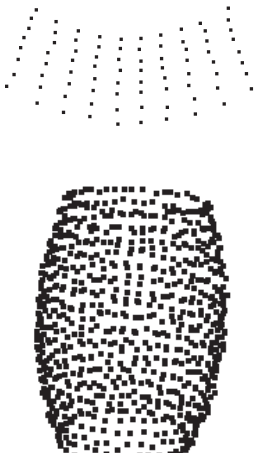
Nominal model	Systematic deviation	Random deviation	Skin model
			

Table 4-2: Skin model with form errors

4.4. Statistical shape modeling

Statistical analysis of shape variation or Statistical shape modeling (SSM) is commonly used for variability considerations in computer graphics, image processing and bioinformatics domains [Cootes 01]. The basic idea of this method is to establish, form a training set, the pattern of "legal" variation in the shapes and spatial relationships of structures in a given class of shapes. Statistical analysis is used to give an efficient parameterization of this variability, providing a compact representation of shapes [Rhodri 08]. The key ideas underlying statistical shape analysis were developed by Kendall [Kendall 77] Bookstein [Bookstein 78] and Cootes [Cootes 01].

4.4.1 Mean vector and covariance matrix

The computation of the mean vector and the covariance matrix is the common solution to analyze the multivariate data. The mean vector is often referred to as the centroid and the covariance matrix as the dispersion or matrix. The definition of mean vector and the covariance matrix can be explained by the following formula 4-16

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \tag{4-16}$$

Where $\boldsymbol{X}$ is used to refer to random vectors, and X_i, $i = (1, 2, \ldots N)$ is used refer to random scalars. In formula 4-16, if the items in the column vector are random variables, then the covariance $\sum_{ij}$ is the difference whose (i, j) item is the covariance as formula 4-17

$$\sum_{ij} = \mathrm{cov}(X_i, X_j) = E[(X_i - \overline{X}_i)(X_j - \overline{X}_j)] \tag{4-17}$$

Where $\overline{X}$ is the mean vector and $\overline{X} = \sum_{i=1}^{n} X_i / N$. By definition, the covariance is symmetric with $\sum_{i,j} = \sum_{j,i}$. The covariance of any component X_i with itself is that component's variance as formula 4-18

$$\sum_{ii} = \mathrm{cov}(X_i, X_i) = E[(X_i - \overline{X})^2] \tag{4-18}$$

To summarize all the covariance of the vector $\boldsymbol{X}$, the covariance matrix can be obtained as formula 4-19

$$\sum = \begin{bmatrix} \sum_{1,1} & \sum_{1,2} & \cdots & \sum_{1,n} \\ \sum_{2,1} & \sum_{2,2} & & \\ \vdots & & \ddots & \vdots \\ \sum_{n,1} & & \cdots & \sum_{n,n} \end{bmatrix} \tag{4-19}$$

If $\sum_{i,j} > 0$, it means that the two variables vary together in the same direction relative to their expected values, and otherwise $\sum_{i,j} > 0$ it means these two variables

are have the different tends to the expected value. When $\sum_{i,j} = 0$, it means that there is no linear dependency between these two variables.

4.4.2 Principle of Statistical shape model method

To establish a statistical shape model, four steps are needed [Stegmann 02] [Liu 08a] [Durrleman 09]:

(a) Acquiring a training set of observation shapes;

(b) Determining the correspondence among the observation shapes, in order to establish the relationship of the training set;

(c) Aligning the training set through registration operations;

(d) Analyzing the principal components and establishing the statistical shape model.

(1). Acquiring a training set

Statistical shape modeling is well developed in image processing and anatomical analysis [Fleute 99]. In our case, a training set of 3D statistical shape model is built from the point set of geometric shape model, which aligned to the Cartesian coordinate system. A mathematical representation of an n-point shape model in a spatial space is described as formula 4-20

$$X = [x_1, x_2, ...x_n, y_1, y_2, ...y_n, z_1, z_2, ...z_n]^T \quad (4\text{-}20)$$

Where, X is the n-point shape model referred also as Point Distribution Model or PDM [Cootes 92] and $(x_i, y_i, z_i)^T$ is the coordinates of the shape model in spatial space. In practice the number n of points is fixed to a value which permits sufficient details of the shape.

The requirements for the training set for our skin models are as follow.

(a) The samples show strong similarities and encapsulate enough variability informations;

(b) The size of the sample should be choosen to avoid large data for time processing and small data for the reliability of the results;

(c) The observed shape models can be provided from the different activities and phases of the product life cycle, such as engineering simulation, manufacturing and inspection, etc.

(2). Determining the correspondence

Since a training set contains a number of samples, establishing the relationship among them is a critical problem. There are many efforts that have been done to solve this problem. According to the study of Niu [Niu 07], these solutions are roughly separated into two categories, landmark-based method and landmark-free method. A landmark as defined in [Stegmann 02] is a point of correspondence on each object that matches between and within populations. Dryden and Mardia [Dryden 98] discriminates landmarks in to three subgroups:

(a) Anatomical landmarks: points assigned by an expert that corresponds between organisms in some biologically meaningful way.

(b) Mathematical landmarks: points located on an object according to some mathematical or geometrical property, i.e. high curvature or an extremum point.

(c) Pseudo-landmarks: constructed points on an object either on the outline or between landmarks.

In general, there are some principles that should be followed to determine the landmarks. Firstly, the points that contain the important geometrical or mechanical information should be demarked. Secondly, the number of landmarks should be proper. If the density of landmarks is too small the model can not reflect enough geometrical information, and in contrary, the calculation process will be time consuming. At last, the landmarks in a training set should have the corresponding relationship, which means that the related landmarks can present the same geometrical character of a shape.

(3). Alignment

Considering that the models in the training set are different from each other by scale, position and rotational effects, an alignment operation is necessary to make the shape prior independent of translation, rotation and. In our case, we suppose that the scale effects are non-significant and we will focus mainly on rotation and translation effects.

Since the shape model is represented by point set, registration techniques as described in chapter 2 can be employed here. First, using PCA method we estimate the principal components and the centroid position of the point set, the local coordinate system of each sample can be determined (Figure 4-22).

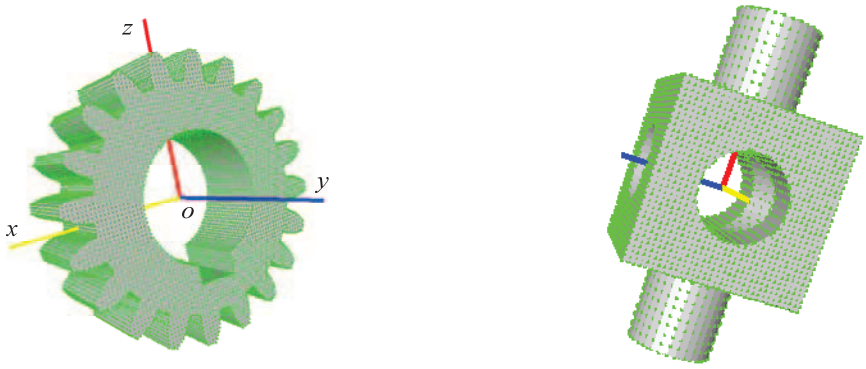


Figure 4-22: Principal axis estimation by PCA method

Secondly, we use coordinates transformation techniques to align the point sets. Figure 4-23 illustrates the principle of coordinate system transformation calculation. $C_0(o_0, \vec{x}_0, \vec{y}_0, \vec{z}_0)$ denotes the global coordinate system; and $C_1(o_1, \vec{x}_1, \vec{y}_1, \vec{z}_1)$, $C_2(o_2, \vec{x}_2, \vec{y}_2, \vec{z}_2)$ and $C_n(o_n, \vec{x}_n, \vec{y}_n, \vec{z}_n)$ represent the local coordinate systems that locate the training samples. The transformation translates all the local coordinate systems to the global coordinate system to obtain the consistent reference systems. The details of the transformation technique are explained in section 3.4.5.

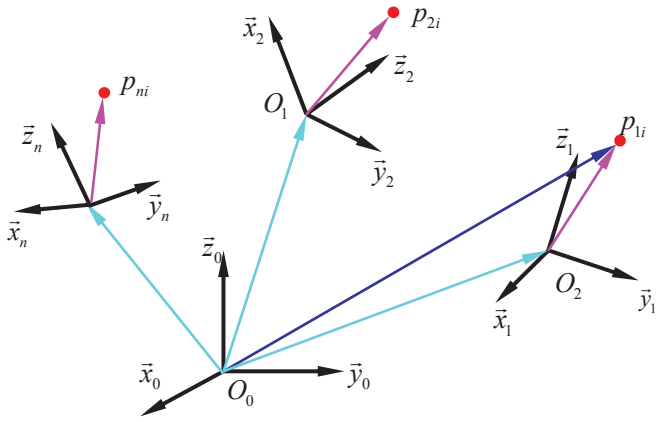
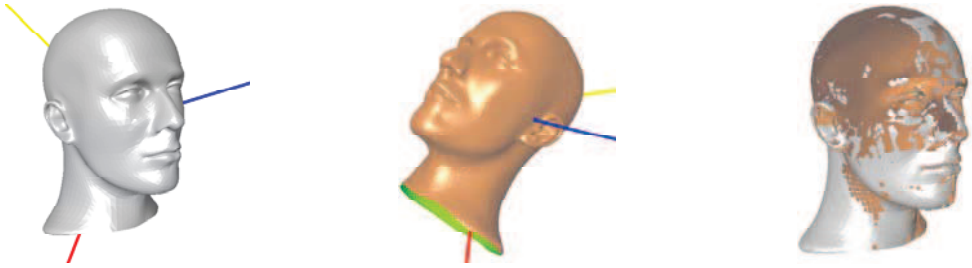


Figure 4-23: Coordinate system transformation

An example of transformation technique explained above is illustrated in figure 4-24. In figure 4-24(a) and (b) are two models located in their local coordinate systems, and figure 4-24(c) is the transformation result in global coordinate system.



(a) (b) (c)

Figure 4-24: Alignment based on principal poses [Zhao 10]

Since the samples are in the same reference system, the common alignment solution is to find out the nearest point pairs between two point sets. In image processing domain, the Procrustes Analysis method [Gerber 09] is used. The procrustes distance is a least squares type shape metric that require two aligned shapes with one-to-one point correspondence [Gerber 09] [Stegmann 02]. In discrete geometry domain, the ICP (Iterative Closest Point) algorithm is popular [Zhao 10]. ICP is an algorithm for searching the corresponding point pairs between two discrete shapes. Many efforts have been attempted to improve the ICP method, and we adopt here CFR (Curvature based Fine Registration) [Zhao 10] method, which combines the Euclidean distance and the principal curvature ratio distance.

Let p_i and q_i are two arbitrary points in different samples of a training set, and let d_g denotes the geometric distance between points p_i and q_i. The geometric distance is defined as formula 4-21

$$d_g(p_i, q_i) = \lambda \cdot d_e + (1-\lambda) \cdot k \cdot d_c \tag{4-21}$$

Where, d_e is the Euclidean distance, d_c is the curvature ratio distance, $\lambda \in [0,1]$ is a user predefined coefficient to balance the contributions of d_e and d_c.

Given a point p, its corresponding point pair q_k in the n point sample S is the point that has the closest geometric distance to p, and it can be calculated as formula 4-22

$$d_g(p, S) = d_g(p, q_k) = \min_{i \in \{1,2..,n\}} d_g(p, q_i) \tag{4-22}$$

We construct the corresponding point pairs for all the samples, and then we can calculate the mean model for the training set and establish its statistical model.

(4). Statistical shape model

The principle of statistical shape model method is that based on statistical analysis, we extract the common characteristics of a training set, which is composed by different geometrical shape models, and establish the statistical model for the samples.

The mean model concept is proposed in the statistical shape model domain. Suppose we have a collection of n discrete shapes (skin model): $X_1, X_2, ..., X_n$, and for each skin model the X_1 is a d dimension vector, where d is equal to the number of points composed a skin model. Since the skin model is calculated in the vector space, one may compute directly the empirical mean model using the method presented in [Durrleman 09], as in formula 4-23

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{4-23}$$

The covariance of this collection of skin models can be calculated through formula 4-24

$$\Sigma_X = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T \tag{4-24}$$

The covariance matrix Σ_X is a $d \times d$ dimension vector, and then the eigenvalue $(\lambda_0, \lambda_1, ..., \lambda_{n-1})$ and eigenvector $(\omega_0, \omega_1, ..., \omega_{n-1})$ of covariance matrix Σ_X can be calculated. The principle components can be reflected by the eigenvector of covariance matrix Σ_X. Let $W = [\omega_0, \omega_1, ..., \omega_{n-1}]$, then we can get $\Sigma_X W = W \Lambda_X$ and Λ_X is the diagonal matrix which can be described as formula 4-25

$$\Lambda_X = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_{d-1} \end{pmatrix} \tag{4-25}$$

The eigenvalues reflect the variance of the principal components. When the principal component has the bigger eigenvalue, it means that this principal component reserve much more information of the initial sample. In our method, we select the principal components having bigger eigenvalues to simulate the initial sample vector using the first t $(t < d)$ maximal eigenvalues that fulfill the formula 4-26

$$\sum_{k=0}^{t}\lambda_k \Big/ \sum_{k=0}^{d-1}\lambda_k \le \varepsilon \tag{4-26}$$

Let $W_X = [\omega_0, \omega_1, ..., \omega_{t-1}]$, and each vector of the sample data can be described by the formula 4-27

$$X \approx \overline{X} + W_x b_x \tag{4-27}$$

Where $\overline{X}$ is the mean; the t dimension vector b_x is the shape coefficient that controls the models of variation.

The main function of statistical shape model is to determine the mean model among numerous sampling shape models and to predict new shape model belonging to the same shape family. Since $\overline{X}$ and W_x are invariants in a training set, the various shape models are obtained and determined by b_x. Under Gaussian distribution assumptions as in [Hufnagel 09] [Ma 10], the variation range of each component of $b_x = \pm 3\sqrt{\lambda_i}$ $(i = 0,1,...,t-1)$.

4.4.3 Application

The skin model as a specification surface model can be imagined by different designers using different approaches. Considering a specific specification requirement, different skin models can be obtained determining a common representative class of skin models and generating new models which has the same characteristics as initial samples is a critical problem toward a representative “mean” skin model.

We propose here to use technique statistical shape analysis and modeling for skin model representation. Here we consider a discrete plane tessellated from a CAD model for test. The tessellated plane is composed of 80 points, and we use this point set to generate the training set. The training set is made of nine samples, which is generated by the three skin model simulation methods introduced in chapter3. Each sample of the skin model is illustrated in table 4-3.

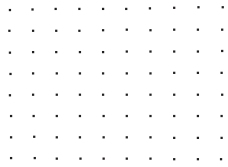
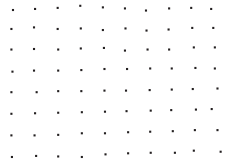
1D-Gauss	Multi-Gauss	Gibbs
 (a)	 (b)	 (c)
 (d)	 (e)	 (f)
 (g)	 (h)	 (i)

Table 4-3: Training set

In table 4-3, each sample is composed of 80 points, which are deemed as the landmarks in our method. The distribution of landmarks is described as follows:

(a) 32 type pseudo-landmarks: 1, 2,…, 32.

(b) 48 type mathematical landmarks: 33, 34,…,80.

The 32 pseudo-landmarks are placed at the boundary of the plane. The 48 mathematical landmarks are placed inside the plane (figure 4-25).

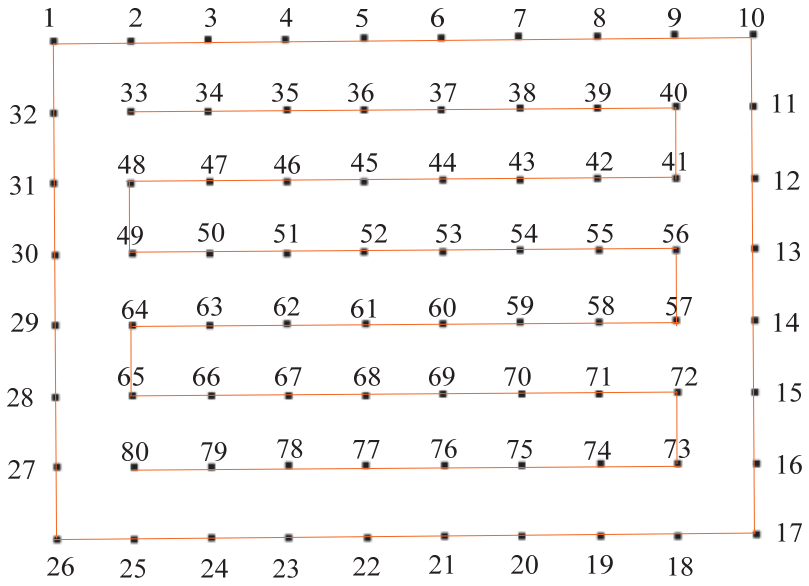


Figure 4-25: Landmarks description

To find out the mean shape, we create the point distribution model of each shape using the coordinates of the landmarks. Then we align the nine samples based on the eighty landmarks. The samples are imported in our software platform, and then the PCA method is employed to calculate the local coordinate system for each sample.

We transform all the samples represented in their local system to the global coordinate system, and then an initial registration of the samples is obtained. The CFP registration algorithm [Zhang 04b] is used to refine the initial registration result, and the correspondence of each point of a sample is determined. The scatter of the training set in the global coordinate system is illustrated in figure 4-26.

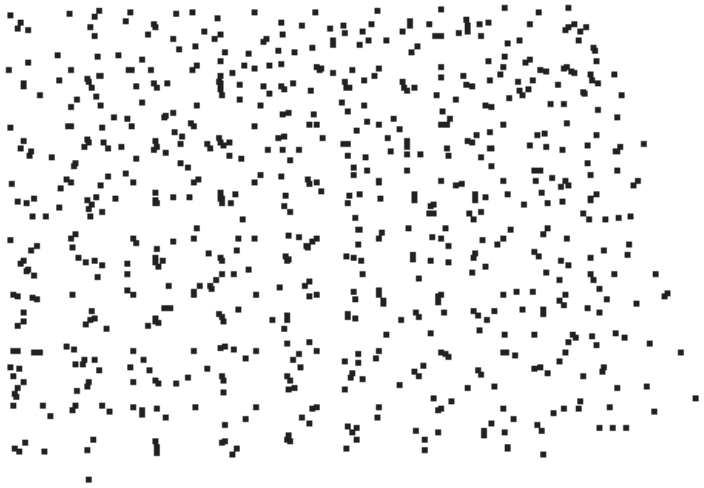
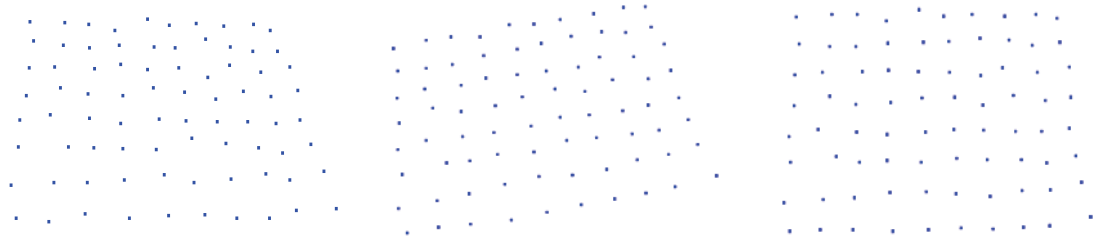


Figure 4-26: Scatter of the nine samples

Based on the aligned samples, the formula 4-23 is used to calculate the mean model of the training set. This result reflects the average distribution of the models (see figure 4-27).

Figure 4-27: Mean model

The deviation between the mean model and the samples can be calculated by the covariance (see formula 4-24), which reflect the influence of each element. Based on formula 4-25, the statistical shape models are then characterized. The models displayed in figure 4-28 are obtained for different values of b_x .



(a) $b_x = -3\sqrt{\lambda}$ (b) $b_x = 0$ (c) $b_x = +3\sqrt{\lambda}$

Figure 4-28: Predicting models

4.5 Conclusion

This chapter presents the skin model simulation methods considering the systematic errors and the skin model analysis method based on statistic shape models.

The first part of this chapter discussed the developed methods for skin model simulation considering the systematic errors. A group of second order shape models are implemented as the constraints of adding systematic errors on skin models. Six basic shapes in this group are generalized to simulate the shapes with systematic errors: Taper, spherical, cylindrical, hyperboloid, ellipsoid, banana-based shape. These six shape types cover the general shapes induced by systematic errors. The proposed methods provide the reasonable simulation results.

The complete skin models including both random errors and systematic errors can be simulated based on the developed methods in this chapter and the previous one. A new method based on Statistic Shape Models is then developed for skin model analysis. With a set of training skin models acquired from different sources, the method first aligns them together first. The statistic shape model including mean model and covariance are computed. The statistic shape model predicts the trend of the skin models, which can be used to evaluate the real shapes.

Chapter 5

A Case Study

5.1. Introduction

The aim of this chapter is to further illustrate the methods described in previous chapters by presenting a detailed case study of a cross-shaped workpiece.

A software framework has been developed as the testing platform for the developed methods of discrete geometry processing (normal estimation, random and systematic errors simulation, mean model evaluation and statistical shape model prediction) described above.

Based on the CAD model and the measurement data, the form errors of the workpiece are simulated by our software framework. Since our software framework is an open system, the mean model and statistical models are also evaluated based on the training set of shape models simulated by other simulation methods.

This chapter is organized as follow:

In section 5.2, we describe the platform and the development tools of our software. Section 5.3 presents a case study including skin model simulation and statistical shape modeling. Section 5.4 is the conclusion of this chapter.

5.2. Platform overview

5.2.1 Development tools

Our software system is developed using C++ language based on Microsoft Visual Studio C++ 2005 platform. The Computational Geometry Algorithms Library (CGAL) is employed to construct the polyhydric data structure of the discrete point data. For the visualization of the shape model, Advanced Graphics Programming Techniques Using OpenGL is adopted here. The details of the development tools are briefly discussed below.

(1). Microsoft Visual Studio

Microsoft Visual Studios [Visual 11] an integrated development environment mainly used to develop console and graphical user interface applications. Visual Studio supports different programming languages, such as Visual C++, Visual Basic.NET, and Visual C#, etc., by language services that allow the code editor and

debugger in the same platform.

Microsoft Visual C++ is a compiler for integration with the Visual Studio IDE, and it supports the C++/CLI specification to write managed code and mixed mode code. It also supports COM (Component Object Model) and MFC (Microsoft Foundation Class) libraries to design Graphical User Interfaces.

Microsoft Visual C++ is adopted in our work to design the user interface based on MFC frameworks and to integrate our developed algorithms.

(2). Computational Geometry Algorithms Library

Computational Geometry Algorithms Library (CGAL) [Cgal 11] is an open source project used to provide efficient and reliable geometric algorithms for computer graphics, computer aided design and modeling, mesh generation, numerical methods, etc.

The CGAL also provides the data structures and algorithms on discrete geometric objects like points and meshes. These data structures include Voronoi diagrams, polygons, polyhedra, mesh generation, geometry processing, etc.

The functions of CGAL employed in our work mainly focus on the geometric algorithms and the polyhedra data structure. The 3D polyhedral surfaces are composed of vertices, edges, facets and incidence relationships on them. The representation of polyhedra surface is a halfedge data structure, which restricts the class of representable surfaces to orientable 2-manifolds with boundaries. The details of the halfedge data structure are introduced in section 3.2.3.

(3). Open Graphics Library

The Open Graphics Library called also OpenGL [OpenGL 11] developed by Silicon Graphics Inc., is a developing environment, which provides thousands of interfaces for 2D and 3D computer graphics applications, and it is widely used in CAD, virtual reality, scientific visualization, etc.

The well-specified OpenGL standard has language bindings for C, C++, Java, etc. Considering the high visual quality and performance of OpenGL to support CAD visual display, OpenGL is adopted in our work for the graphics processing.

5.2.2 Main interface

Figure 5-1 shows a screen shot of the main user interface, and the most important interaction tools are annotated:

Point tool allows the user to show/hide the vertices of the discrete model.

Mesh tool allows the user to show/hide the polygon mesh of the discrete model.

Facet tool allows to user to show/hide the facet rendering of the discrete model.

1D-Gaussian tool is used to generate skin models with the random deviations of 1D-Gaussian distribution.

3D-Gaussian tool is used to generate skin models with the random deviations of multi-Gaussian distribution.

Gibbs tool is used to generate skin models with the random deviations of MCMC Gibbs sampling.

Form tool is used to simulate the form specification of a planar model.

Orientation tool is used to simulate the orientation specification of a planar model.

Position tool is used to simulate the position specification of a planar model.

Systematic error-plane tool is used to simulate the systematic deviations of a planar model and the basic systematic shapes including: parabola, cone, sphere, cylinder and transformation.

Systematic error-cylinder tool is used to simulate the systematic deviations of a cylindrical model and the basic systematic shapes including: taper, convex, concave and banana.

Form error tool is used to simulate the form error of discrete models, which integrate both random deviations and systematic deviations.

Mean model tool is used to calculate the mean model form a training set, which composed by numerous skin models.

Statistical model tool is use to predict new models from a training set, and the new models inherit the primary properties of the training set.

Status bar at the bottom of the screen is used to show the time consuming of a

specific process and the information of input discrete model, such the number of vertices, the number of edges and the number of facets.

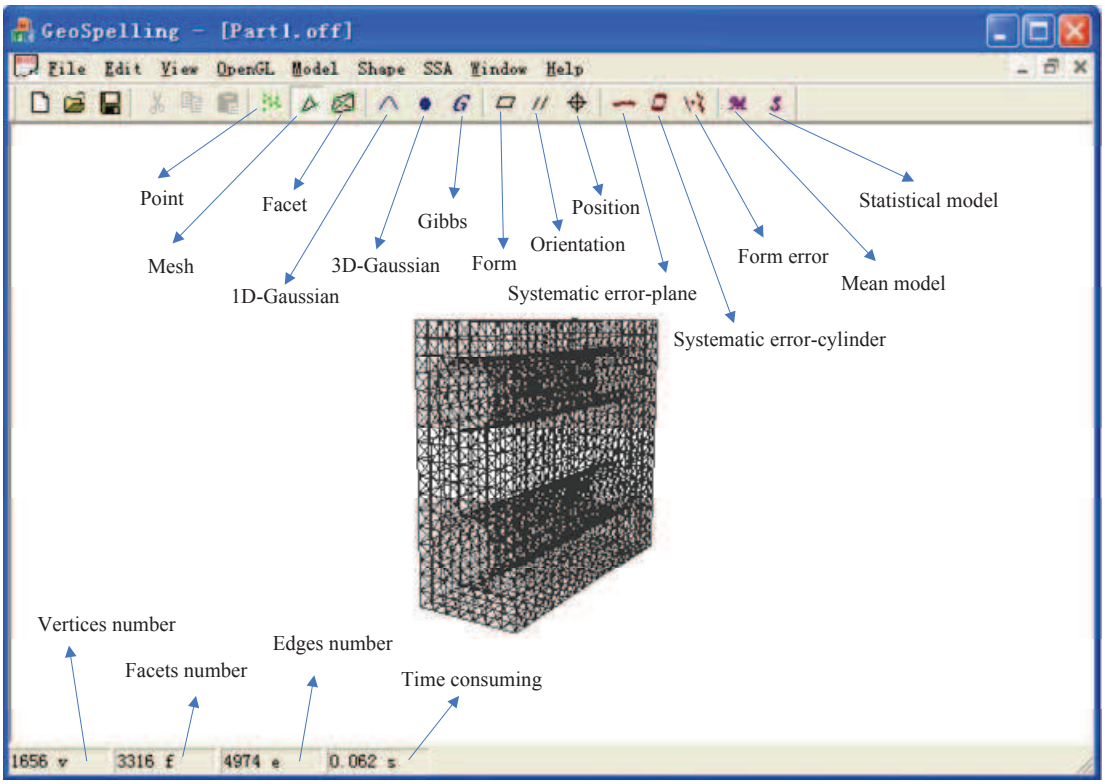


Figure 5-1: The main interface of the software

5.2.3 Menu specification

The menu specifications displayed in figure 5-2 describe detailed functions of our developed system.

(a) Model. The "Model" menu is used to reflect the basic two kinds of information of the input tessellated shapes. One is the mode of display such as point mode, mesh mode and facet mode, and the other one is the normal information, which includes different vertex normal estimation methods and facet normal estimation method.

(b) Random. The "Random" menu is used to simulate the skin models with random errors especially Gaussian distribution deviations. In this menu there are three options: 1D-Gaussian method, Multi-Gaussian method and Gibbs method.

(c) Systematic. The "Systematic" menu is used to simulate the skin model with the systematic deviations.

The “Curvature” option is used to estimate the surface curvature including Gaussian curvature and mean curvature.

The “Systematic Plan” option is used to simulate the systematic deviations of planar shapes, and the basic shapes that can be simulated.

The “Systematic Cylinder” option is used to simulate the systematic errors of cylindrical shapes, and the basic shapes that can be simulated.

The “Form error” option is used to simulate the skin models with both random deviations and systematic deviations.

(d) SSA. The “SSA” command is used to evaluate the information of statistical shape models, which are skin models in our case.

The “Mean model” option is used to estimate the mean model of a training set of skin models.

The “SSM” is the abbreviation of the statistical shape model, and this option is used to predict new models of the training set of skin models.

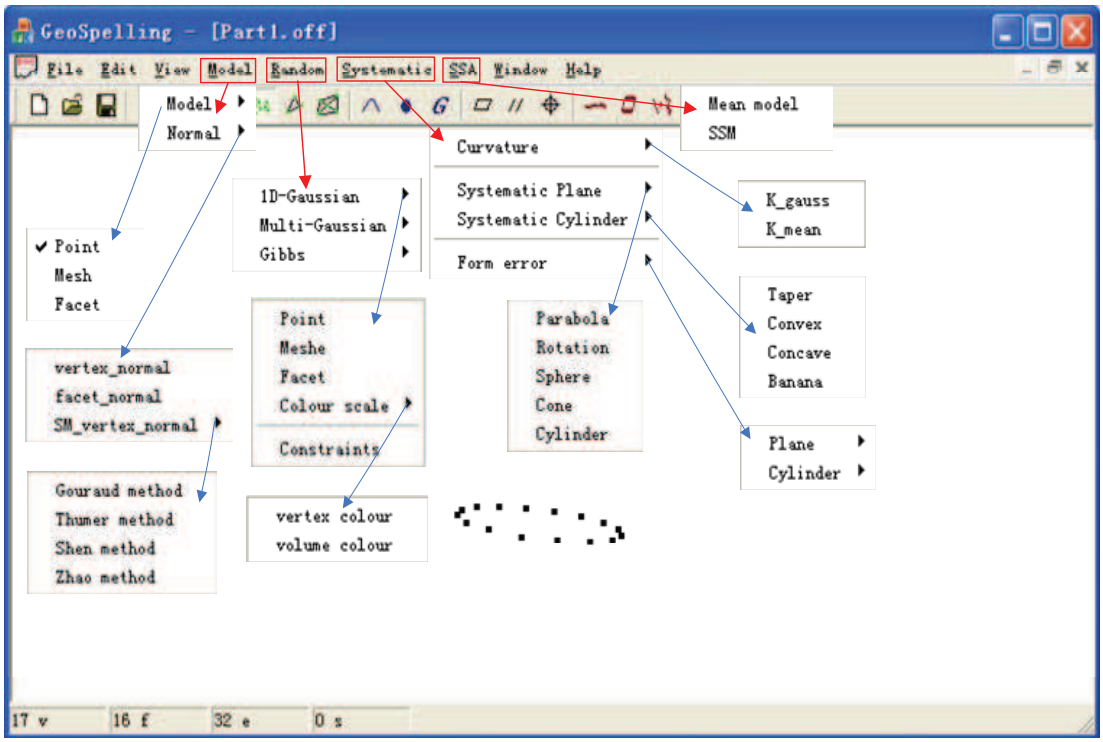


Figure 5-2: Menu specifications of the software

5.3. Application

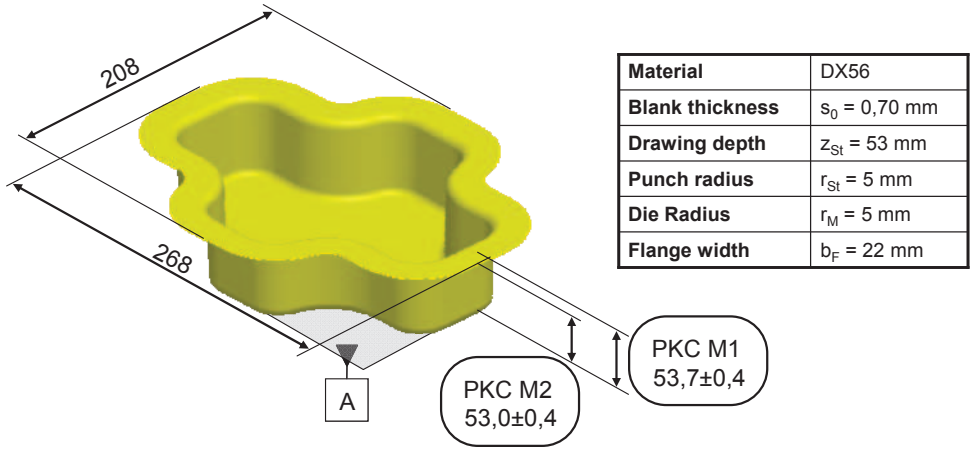
5.3.1 Skin model simulation

The case study is based on a sheet metal part manufactured in a one-stage sheet metal forming process [Stockinger 10a]. The process of sheet metal forming is selected here because of the existing simulation tools dealing with accurate material models and taking multistage process steps into account (gravity, stamping, springback, etc.) [Stockinger 09].

The manufacturing process is simulated using stochastic FE techniques. For the discretisation of the blank, shell elements are used. The tooling is modeled as rigid parts and a process macro is used to define the process including stamping velocity, blank holder force and friction. An initial model of the stamping process serves as a basis for the variation of process parameters. The selected variables (described in figure 5.3) are computed using Latin Hypercube Sampling under the assumption of the independence of the variables and normal distributions [Stockinger 10b]. Ten samples are considered in this work.

Measured data are also available. For the purpose of this case study, 29 manufactured cross-shaped parts where measured using ATOS I fringe projection system.

Figure 5-3 is the CAD model of the cross-shaped part, and its geometric specification is described in Appendix 1. In our work, we only focus on the skin model simulation of the bottom plane with the flatness specification constraint.



Material	DX56
Blank thickness	s_0 = 0,70 mm
Drawing depth	z_{St} = 53 mm
Punch radius	r_{St} = 5 mm
Die Radius	r_M = 5 mm
Flange width	b_F = 22 mm

Figure 5- 3: The cross-shaped part [Stockinger 10a]

Based on the CAD model (see figure 5-4), the bottom plane can be extracted using CATIA V5 GSD utilities as illustrated in figure 5-5. In order to discretize the bottom plane, the tessellation operation is implemented using CATIA V5 software. Figure 5-6 illustrates the tessellation result composed of 2392 points and 4550 facets.

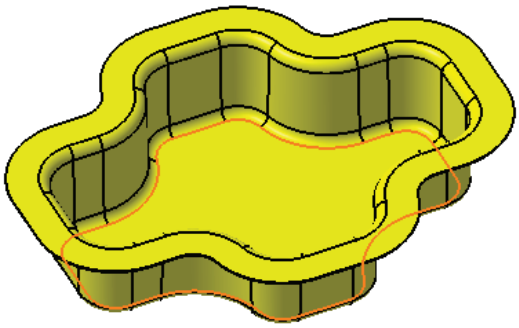


Figure 5-4: Segmentation of CAD model

Figure 5-5: Bottom plane extraction

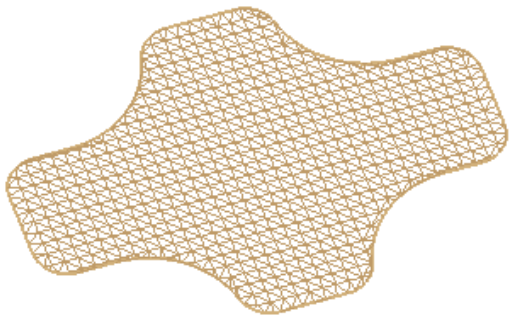


Figure 5-6: Tessellation of the bottom plane

Based on the tessellated CAD model and the geometrical specification of flatness, we can generate different skin models using the methods discussed in section 3.3. Figure 5-7 illustrates three skin models of the bottom plane with a tolerance of flatness equals to 0.3mm.

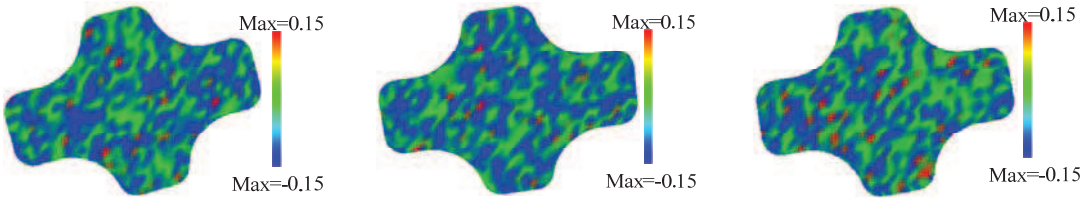


Figure 5-7: Skin models of bottom plane with flatness constraint

On the other hand based on the information from measurement data, we can improve the skin model considering both systematic and random errors. Figure 5-8 shows the measured point set of the bottom plane of the cross-shaped part .The analysis of this point set is illustrated in figure 5-9 and Appendix 2.

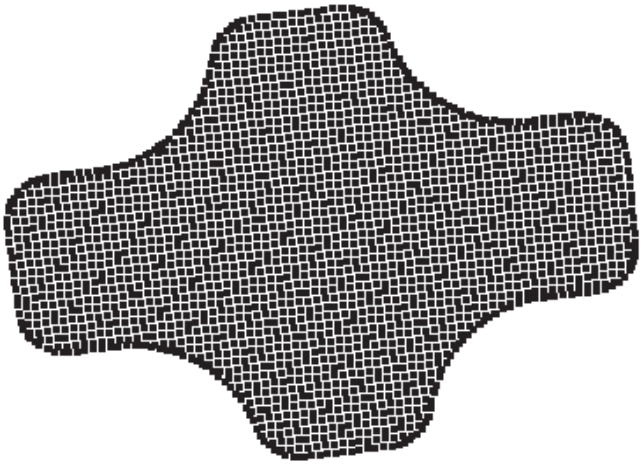


Figure 5-8: Measured point set of cross shaped part

After filtration operation, the measured point number is equal to7845, and the point set is associated to different shapes respectively, as a least-square plane, a least-square ellipsoid, a least-square cone, a least-square cylinder and a least-square sphere. The detail performances of each association result are illustrated in table 5-1. In table 5-?, the standard deviation of the point set to the associated ellipsoid and the associated cylinder has the smallest value that equal to 0.072 mm. Considering the limited deviation from point set to the associated plane is smaller than the cylinder shape, so we simulate the systematic errors using ellipsoid shape here.

Items / Shapes	Standard deviation	Maximum deviation	Minimum deviation	Mean deviation
Plane	0.081	0.271	-0.195	0
Ellipsoid	0.072	0.243	-0.162	0
Cone	0.138	0.531	-0.193	0
Cylinder	0.072	0.254	-0.175	0
Sphere	0.120	0.507	-0.135	0

Table 5-1: Performances of each systematic error simulation

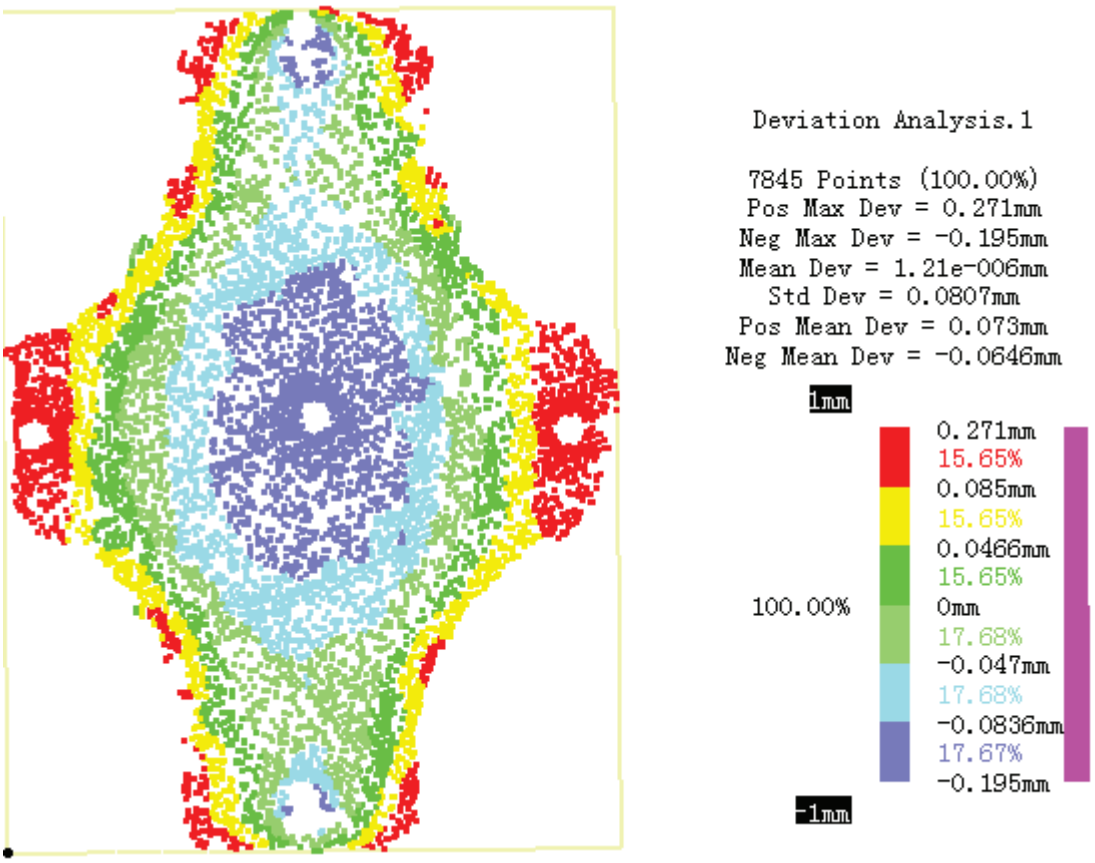


Figure 5-9: Analysis of measured point data of cross shaped part

Based on the nominal discrete bottom plane model (as figure 5-6), the skin model with ellipsoid systematic deviations are simulated as in figure 5-10. To visualize the deviations clearly, the skin model with color scale representation is illustrated in figure 5-11, and the limited deviations are reflected by red and blue color.

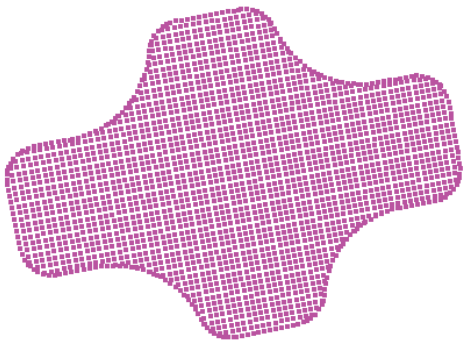


Figure 5-10: Skin model with systematic deviations

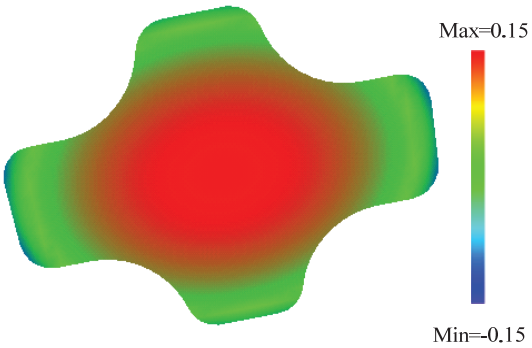


Figure 5-11: visualization of systematic deviations

To verify that the skin model obtained from the simulation of ellipsoid systematic errors, the deviation analysis is adopted (see figure 5-12). We associate the

point set as a planar shape, and the analysis result is as follow. The point number is equal to 2392, and the limit values of the deviation are equal to 0.258 and -0.119 as respectively. The mean deviation is equal to -9.6e-005mm, and the standard deviation is equal to 0.0904mm. Comparing the deviation between the skin model and the measured point set (see figure 5-9), the random errors are absent.

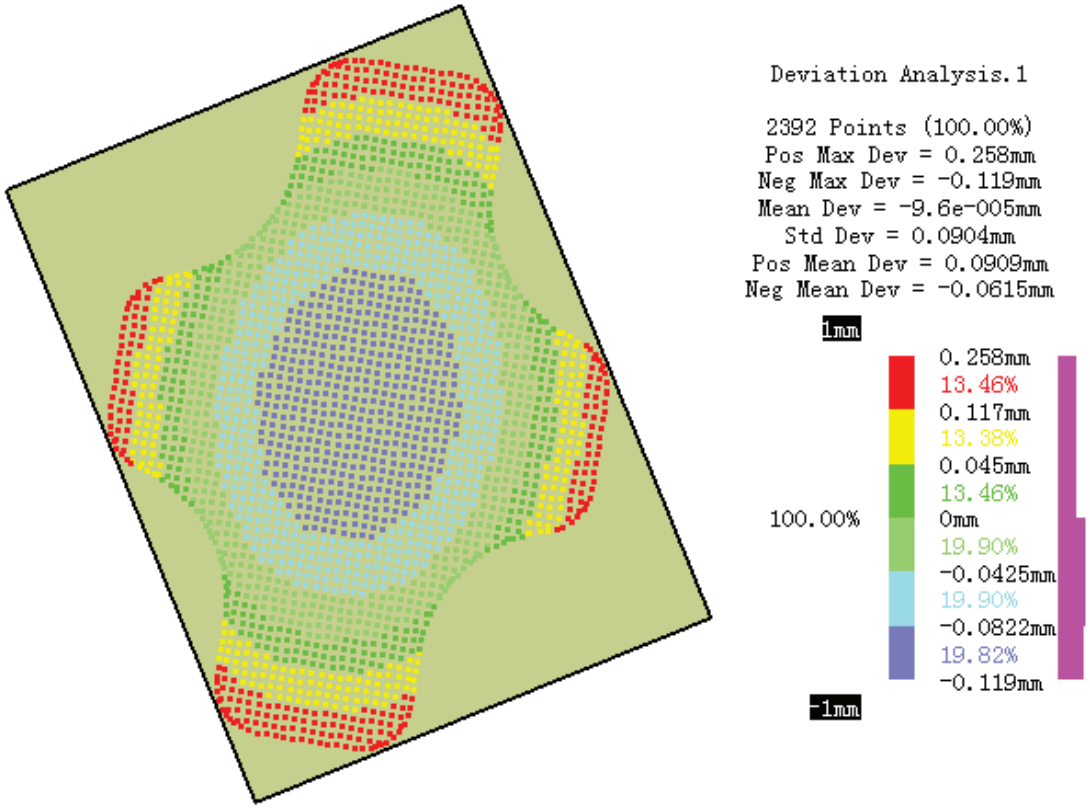


Figure 5-12: Analysis of skin model with systematic errors

The random errors are added to the skin model with ellipsoid systematic errors. The Gibbs method is adopted to simulate the random errors following the Gaussian distribution. The skin model with both systematic errors (ellipsoid) and random errors (Gaussian distribution) is illustrated as figure 5-13. Figure 5-14 is the color scale representation of the skin model, and the color changed is continuous from red to blue but with some noise. This illuminate that the deviations follow the ellipsoid shape variety, but it also with random noises.

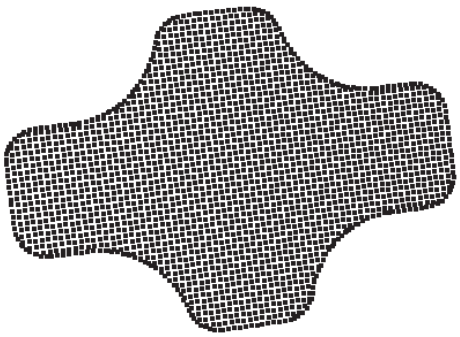


Figure 5-13: Skin model with both systematic and random errors

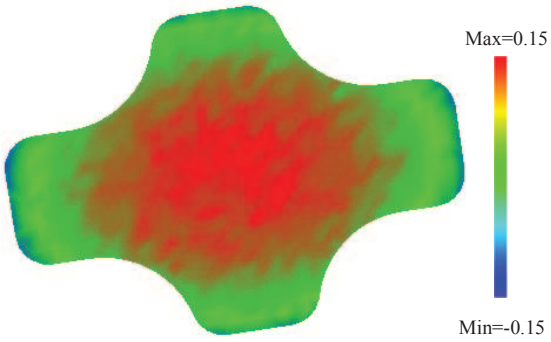


Figure 5-14: Color scale of skin model with both systematic and random errors

To compare the simulated model and the measured point set, the analysis of the skin model is illustrated in figure 5-15. As the process of measured point set, we associate the skin model as a planar shape, and take the deviations into account. The point number is equal to 2392, and the limit values of the deviations are equal to 0.282mm and -0.153mm. The mean deviation is equal to -0.001mm and the standard deviation is equal to 0.092mm.

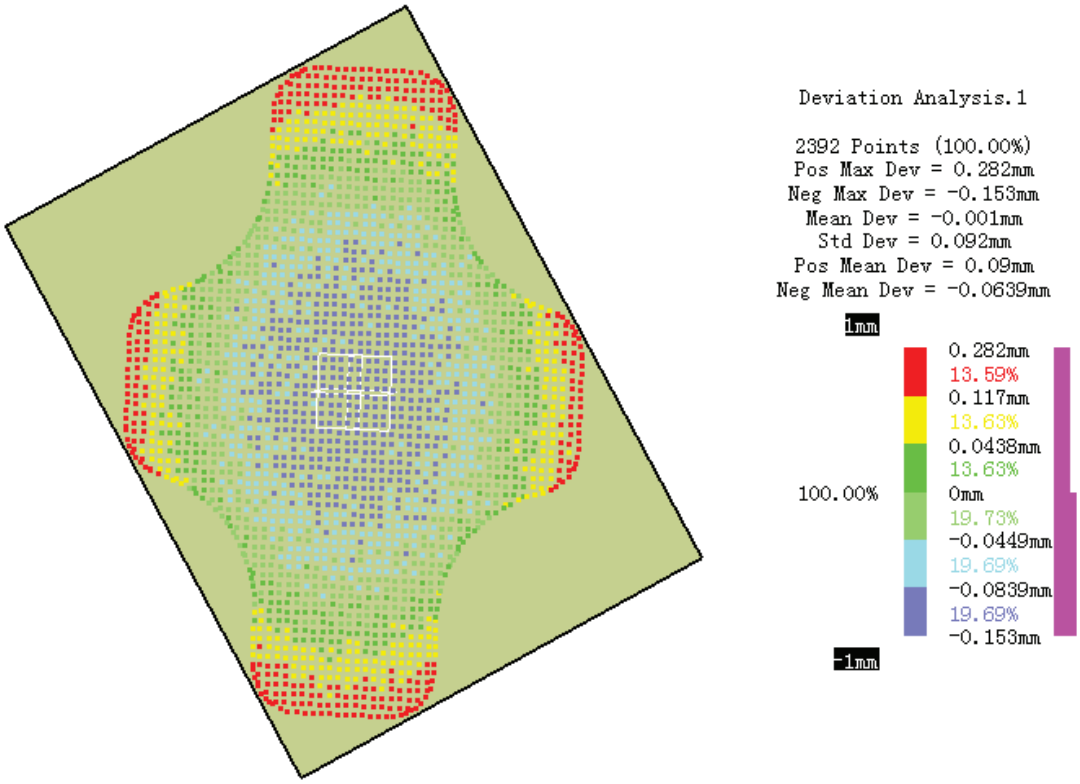


Figure 5-15: Analysis of skin model with both systematic and random errors

Based on the compare between the simulated skin model and the measured model, the limited deviation values are both within the geometric specification value, which is the flatness constraints value equal to 0.3mm. The difference of standard

deviations is equal to 0.0016, and this reflects these two models have quite similar distributions. The simulation methods of our method are satisfied in this study case.

5.3.2 Statistical shape modeling

(1). Mean model

The training set considered here is composed of ten cross-shaped models obtained using stochastic FEA thanks to Lehrstuhl für Konstruktionstechnik - Friedrich-Alexander Universität Erlangen-Nürnberg [Stockinger 11] (Figure 5-16 and Appendix 3).

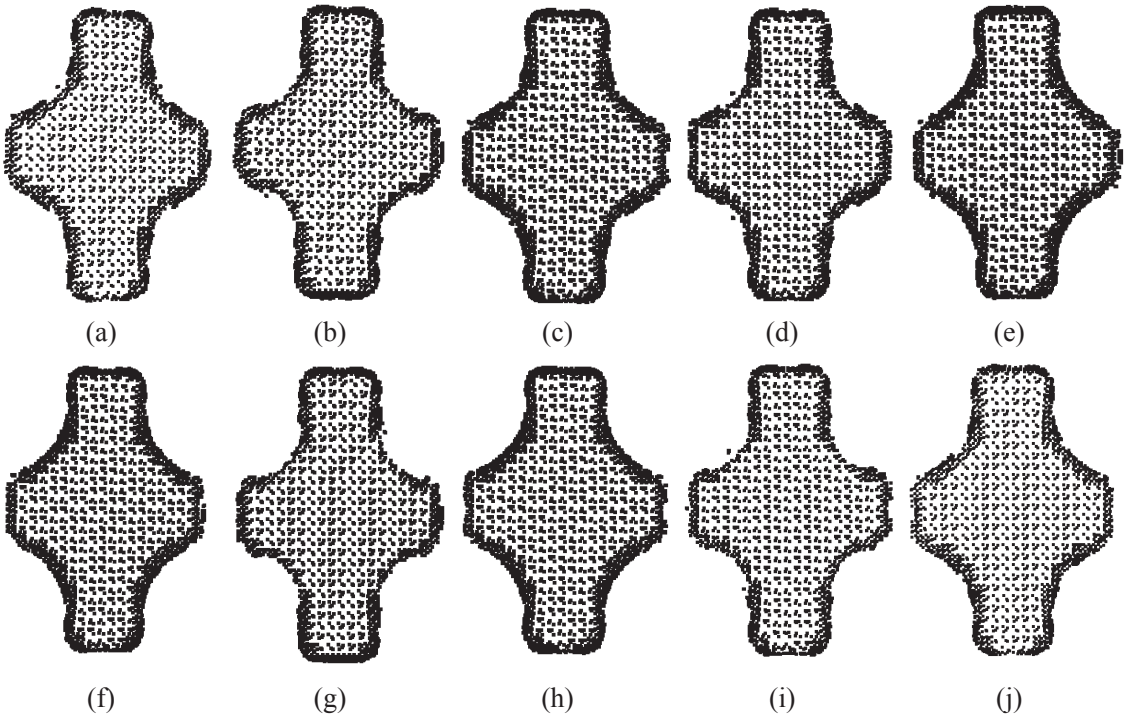


Figure 5-16: Training Set

To establish the relationship among ten simulated models of the training set, we adopt landmark technique to obtain the correspondences. Considering the models are planar shapes without high curvature variation, the mathematical landmark technique is not suitable here. The pseudo-landmark technique is then adopted in our method.

For convenience, we use a matrix arrangement technique instead of integers to tag the landmarks as illustrated in figure 5-17. Each landmark corresponds to a unit grid, marked here as (i, j) which represents also a kind of pixel coordinates. Instead of using all the points, the landmark (i, j) refers to the mean point p of all the points inside the corresponding grid.

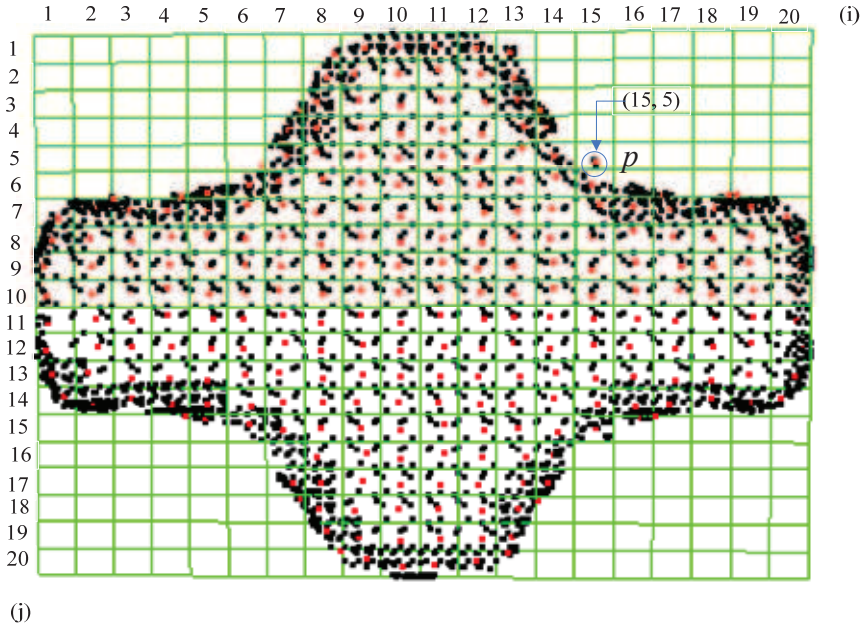


Figure 5-17: Landmarks arrangement

The ten models of the training set are all enumerated here by a 20×20 grid (Figure 5-17). The ten models of the training set are then imported into our platform. Considering that the FEA simulation models are in a uniform scale and have the same coordinate reference system, the simulated models are aligned using the registration algorithm described in chapter 4. Figure 5-18 illustrates the alignment of the samples, which are denoted using different colors.

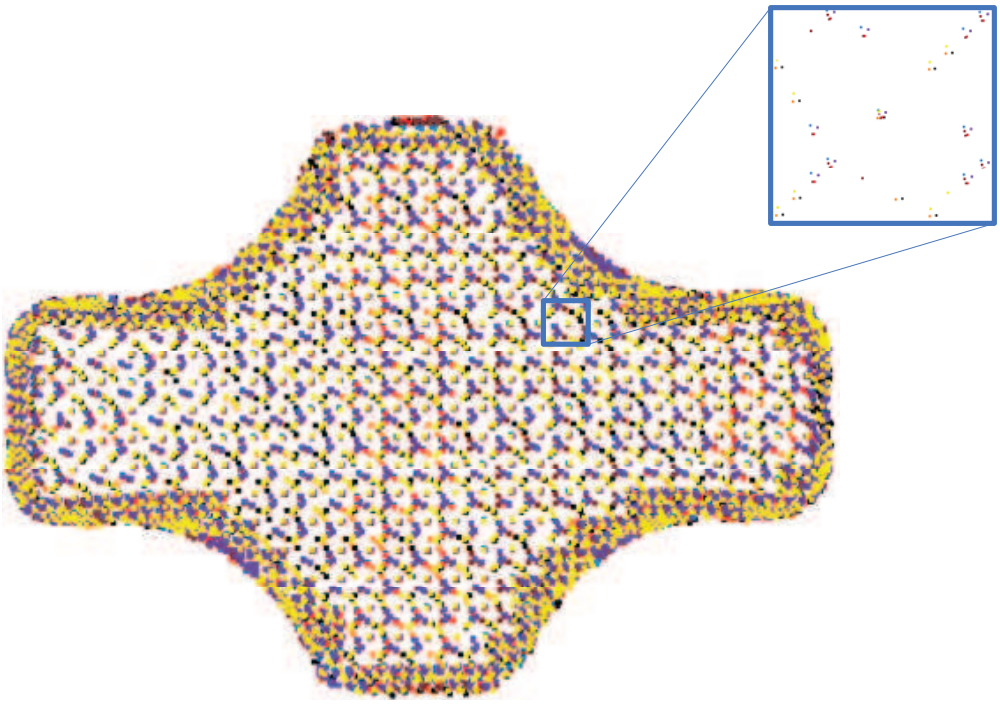


Figure 5-18: Alignment of the training set

After aligning the ten simulation models, the mean model is then computed. The points of the mean model are the centroids of its neighborhood points. In figure 5-19, the training set is reflected by black points and the mean model is displayed by red points (figure 5-19).

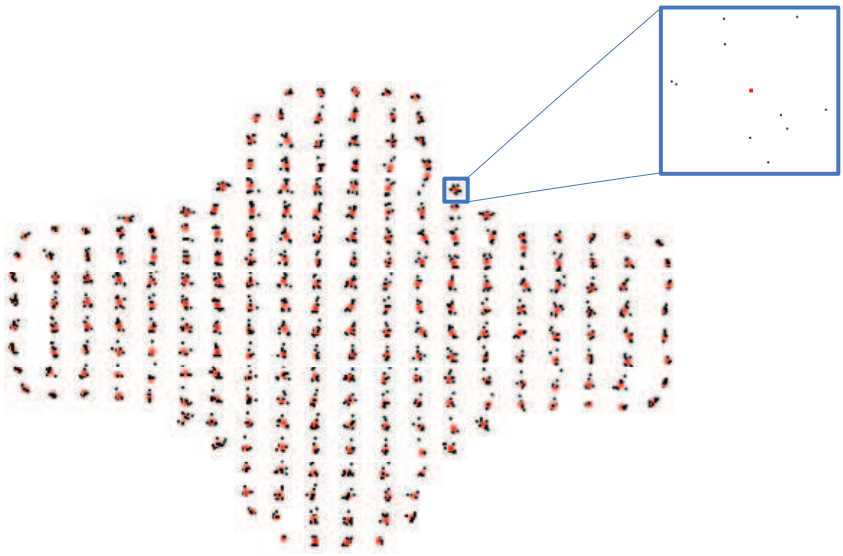


Figure 5-19: The mean model of the training set

To analyze the deviations of the mean model, we associate the point set to a least-square plane. The mean model is composed of 273 points, the maximum deviation to the least-square plane is equal to 0.191mm and the minimum deviation is equal to -0.186mm. The standard deviation of the mean model is 0.103mm. The details of the deviation analysis are illustrated in figure 5-20.

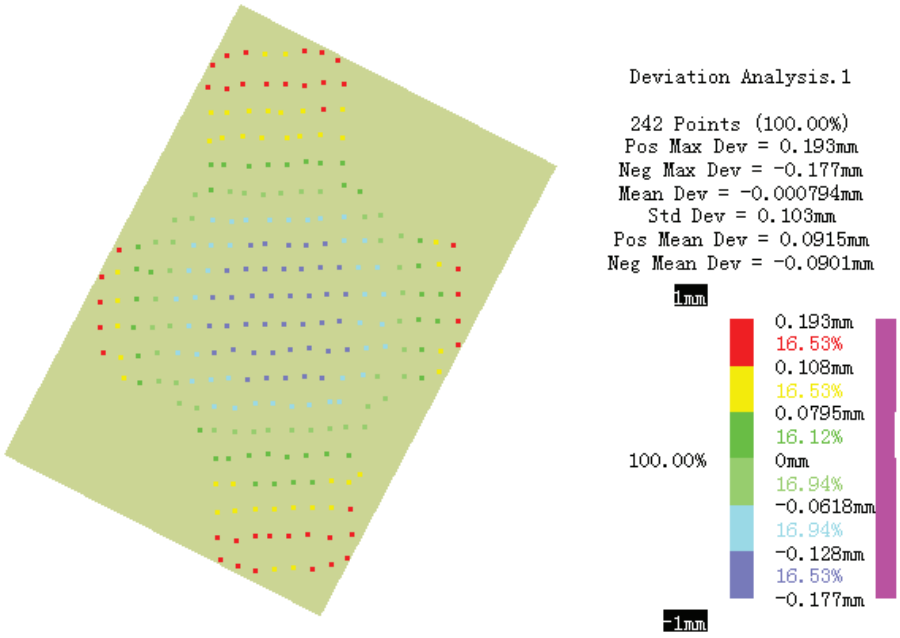


Figure 5-20: Analysis of the mean model

The training set considered here is anther ten cross-shaped skin models, which are obtained using the method introduced in section 5.3.1. The ten samples are illustrated in figure 5-21 and Appendix 4.

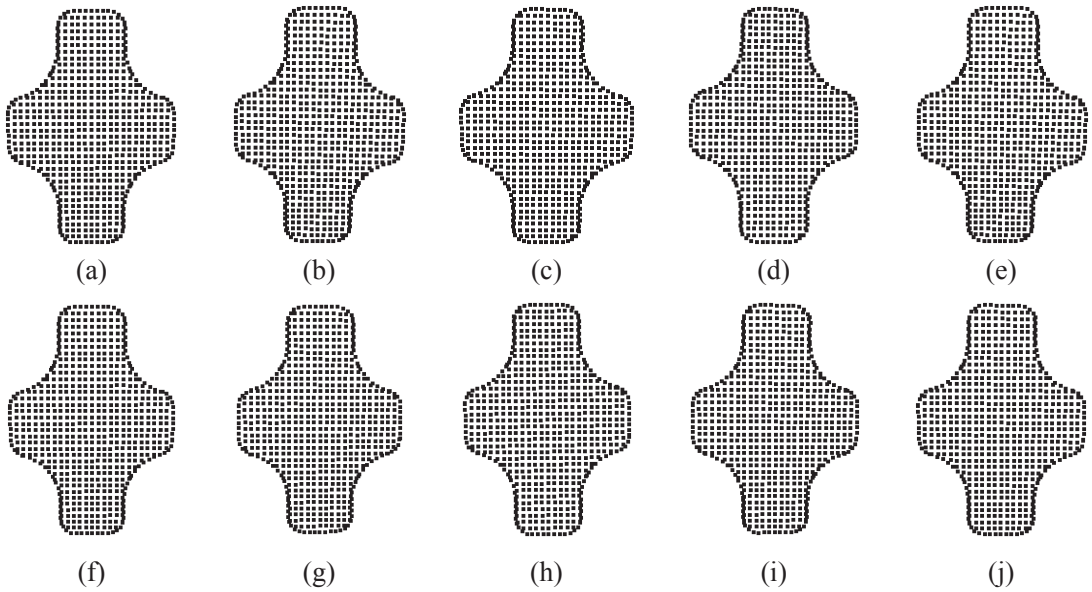


Figure 5-21: Training Set of skin models

Following the same processing procedure of mean model simulation approach, the mean model can obtained. To analyze the deviations of the mean model, we associate the point set to a least-square plane. The mean model is composed of 242 points, the maximum deviation to the least-square plane is equal to 0.239 mm and the minimum deviation is equal to -0.133mm. The standard deviation of the mean model is 0.0917 mm. The details of the deviation analysis are illustrated in figure 5-22.

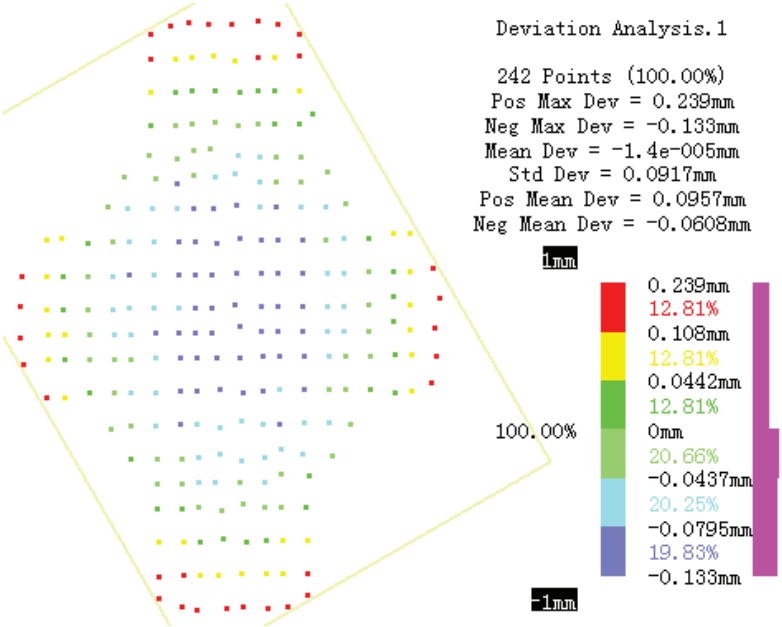


Figure 5-22: Analysis of the mean model

Based on the analysis results of two mean models that simulated by our method (introduced in section 5.3.1) and FEA method, the deviations compared to the measurement data are compared as table 5-2.

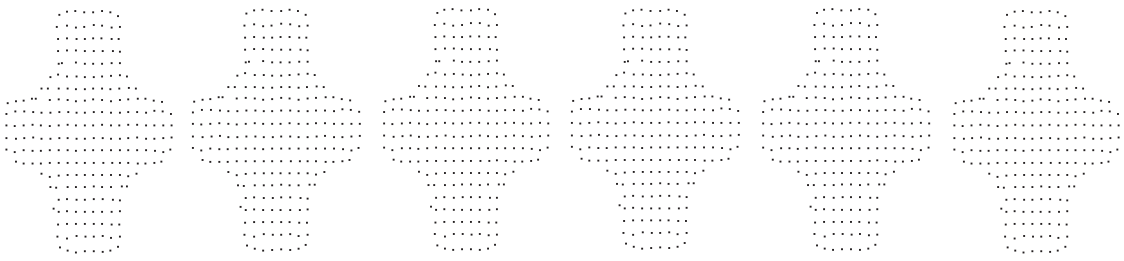
Items / Data	Standard deviation	Maximum deviation	Minimum deviation	Mean deviation
Measured data	0.0807	0.271	-0.195	1.21e-006
Our method	0.0917	0.239	-0.133	1.4e-005
FEA method	0.103	0.193	-0.177	-0.000794

Table 5-2: Comparison of two kinds of mean models

According to table 5-2, our method provides a smaller standard deviation and mean deviation mean model than the one simulated by FEA method. It can be deduced that our method fit the form errors better than the FEA method in this case.

(2). Statistical shape model

Based on the PCA technique, the deviation between the mean model and each model of the training set can be calculated by the covariance matrix and then the influence of each component can be detected. According to the two mean models simulated above and the statistical shape analysis method introduced in the previous chapter, new models can be predicted (see figure 5-23).



(a) b_1: $-2\sqrt{\lambda_1}$ (b) b_1: $+2\sqrt{\lambda_1}$ (c) b_2: $-2\sqrt{\lambda_2}$ (d) b_2: $+2\sqrt{\lambda_2}$ (e) b_3: $-2\sqrt{\lambda_3}$ (f) b_3: $-2\sqrt{\lambda_3}$

Figure 5-23: The predicted models

Figure 5-23 shows the first three modes of the principal component analysis. The modes of the model are sorted in decreasing magnitude of their corresponding eigenvalues. In the first line, the mode corresponding to the largest eigenvalue λ_0 is varied between $-2\sqrt{\lambda_0}$ and $+2\sqrt{\lambda_0}$, in the second column the same is done for the second model and so on. The result show the large variability included in the

cross-shaped model. The full details of the predicted models are shown in figure 5-24.

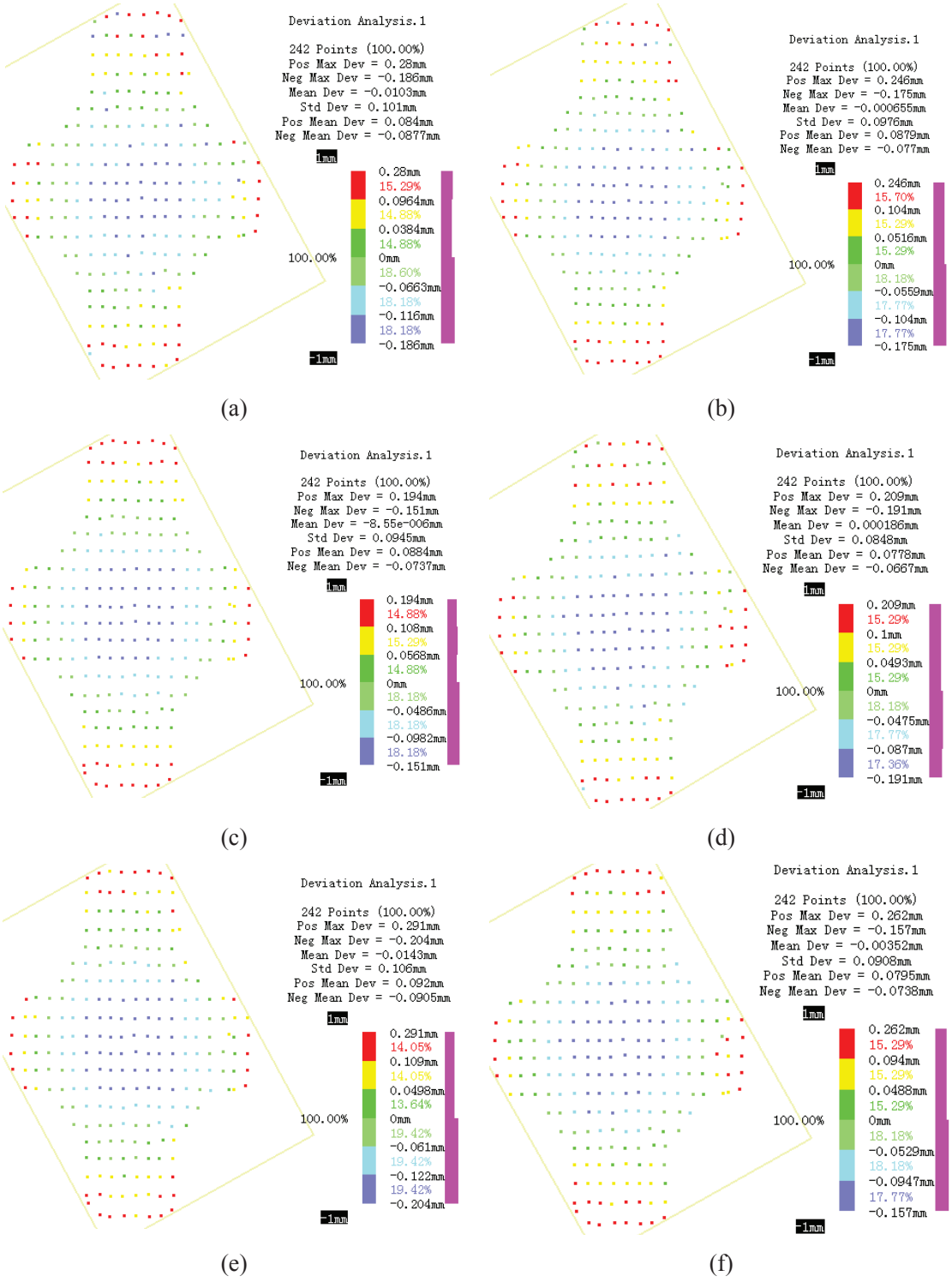


Figure 5-24: Analysis of the predicted models

The statistical shape models predicted above are based on the training set of samples simulated by our method (introduced in section 5.3.1) and FEA method. Hence they have a combined distribution performance of these two kinds of samples.

5.4. Conclusion

This chapter presents a detailed case study of a cross-shaped sheet metal workpiece. An open modular framework software has been developed to test and support the methods and algorithms developed in this thesis.

The detailed processing results of the studied case are presented here. The main results concern the simulation of skin model that includes both systematic and random deviations. A new systematic errors simulation method based on shape decomposition technique is presented.

Based on the measurement data, a comparison between the two kinds of simulated skin models (by our method and FEA method) is presented, and our method is proved has a better approximative simulation result experimentally.

Based on the training set of simulated models using stochastic FEA, we create a mean model and derive the main characteristics of the training set. We have described a method for building 3D statistical shape models for mechanical models considering the training set composed by the samples simulated using different method. Based on the principle components analysis of the predicted models, the large variability included in the tested models is discussed.

Conclusion

A coherent and complete tolerancing process is necessary in mechanical applications to manage geometrical variations along the product lifecycle. GeoSpelling is proven to be an appropriated model for this purpose, which is standardized as ISO 17450. This thesis presents several significant contributions to achieve GeoSpelling for geometrical product specification from two aspects: discrete geometry based framework for geometrical operations in GeoSpelling and skin model simulation.

A set of methods based on discrete geometry processing techniques are associated to provide solutions for the six geometrical operations defined in GeoSpelling: partition, extraction, filtration, association, collection and construction. This discrete or computational geometry based methods solve the operationalization of GeoSpelling in a convenient and executable way.

Skin model is one of the foundation and kernel concepts of GeoSpelling. The simulation of the skin model is based on the nominal model of the shape, on random deviations and systematic deviations. Considering the point-based data are widely used in applications, the discrete skin model representations are applied in our research. Different discrete shape modeling methods are developed for skin model.

Three different methods are developed to generate the skin models by adding random deviations to the discrete nominal model: 1-D Gaussian-based method, multi-Gaussian-based method and Gibbs based method. Firstly, the nominal shape is tessellated as discrete representations (point-based and/or mesh-based) from CAD model. The normal direction at each point on nominal shape is then estimated. Finally, 1-D Gaussian method adds the random deviation along a given point's normal direction based on 1 dimensional Gaussian possibility distribution; Multi-Gaussian method considers the random deviations in three dimensional space; while Gibbs method calculates the new position of the given point by Markov Chain Monte Carlo simulation. During the skin model simulation, the specified tolerances are used as constrains to locate the simulated skin models within the specified tolerance zone.

Systematic errors are reproducible inaccuracies that are consistently in the same directions. Considering systematic errors usually follow some specified functions which can be traced and simulated, a group of predefined second order shapes are implemented as the constraints to simulate the systematic deviations. The shape group contains six basic shapes which are proven as the most likely shape the systematic

error can follow. A set of methods are developed to adding systematic deviations based on the predefined shape group and the testing shows the reasonable results.

Based on the complete skin model which includes random deviations and systematic deviations, a new method based on Statistic Shape Models is developed for skin model analysis and trend prediction. Firstly, a set of training skin models acquired from different sources. The corresponding landmarks are then searched and all the skin models are aligned together. The statistic shape model of the set of training skin models including mean model and covariance are computed. The statistic shape model provides a prediction to the trend of the skin models, which can be used to evaluate the real shapes.

A modular software framework is developed to test and support the methods and algorithms developed in this thesis. The software framework is developed based on MS VC 8.0 platform, and it is also supported by OpenGL and GCAL libraries [CGAL].

A detailed case study of a cross shaped workpiece was presented. The skin models of the studied workpiece are simulated and generated by the developed software framework. The results are compared to real measured data and the simulated results of another FEA-based method. The comparative results provide that the proposed methods can provide satisfying performance.

Contributions:

The main contributions of the dissertation to GeoSpelling for geometrical product specification are listed as follows:

- A new framework for geometrical operationalization of GeoSpelling is defined based on discrete geometry processing techniques. The corresponding methods used in discrete geometry for geometry representation and processing are associated to the feature (including ideal and non-ideal) representation and geometrical operations in GeoSpelling. The developed framework provides new convenient and executable solutions for feature representations and operations in GeoSpelling.
- A survey of the common used methods for discrete normal estimation is researched. In order to generate reliable estimation results, a comparative result is analyzed to specify the suitable applications of each method.
- Three different methods (1-D Gaussian method, Multi Gaussian method

and Gibbs method) are developed to simulate the skin models by adding the random deviations. The three methods provide three different ways to add random deviations. All the three methods use the specified tolerances as constraints to determine the location and shape of the simulated skin model. The developed methods provide new ways and solutions for skin model simulation based on the discrete shape model techniques.

- New methods are developed to simulate the skin model considering systematic errors. A group of parametric second order shapes are implemented to simulate the shapes of systematic deviations. The shape elements in the group can be added and/or deleted according to different applications. The developed methods are convenient to implement and the testing results are proven satisfying.

- Based on complete skin model that includes both random and systematic errors, a new method based on statistic shape model is developed for skin model analysis and prediction. The developed method requires a set of training skin models from different sources as input. It can compute the statistic shape model considering the mean model and covariance, which provides a trend prediction of the new skin models.

- A software framework is developed to support the skin model simulation methods presented in this thesis. The framework is modular and user-friendly. It is easy to integrate new functions related with discrete geometry processing.

Future works:

The discrete geometry processing for improve GeoSpelling contains many other topics, which have not been addressed in this thesis. Some promising works for the future are presented as follow.

- Random errors simulation.

The random errors of skin models are simulated by Gaussian distribution, and only three kinds of methods are considered in our work. It should be employing other noise simulation methods to simulate the random errors, such as Perlin noise, Poission noise, etc.

- Tolerance specification.

The tolerance specifications only consider the flatness, parallelism and position constraints, and it should be take into account other possible specifications.

The tolerance specifications only satisfy the planar and cylindrical shapes, and it should be extant to other shape models.

- Systematic errors simulation.

The systematic errors simulation methods only adapt the planar and cylindrical shape models, and it should be satisfy other kind of shape models. The regulation of basic shapes combination to simulate complex systematic errors should be considered.

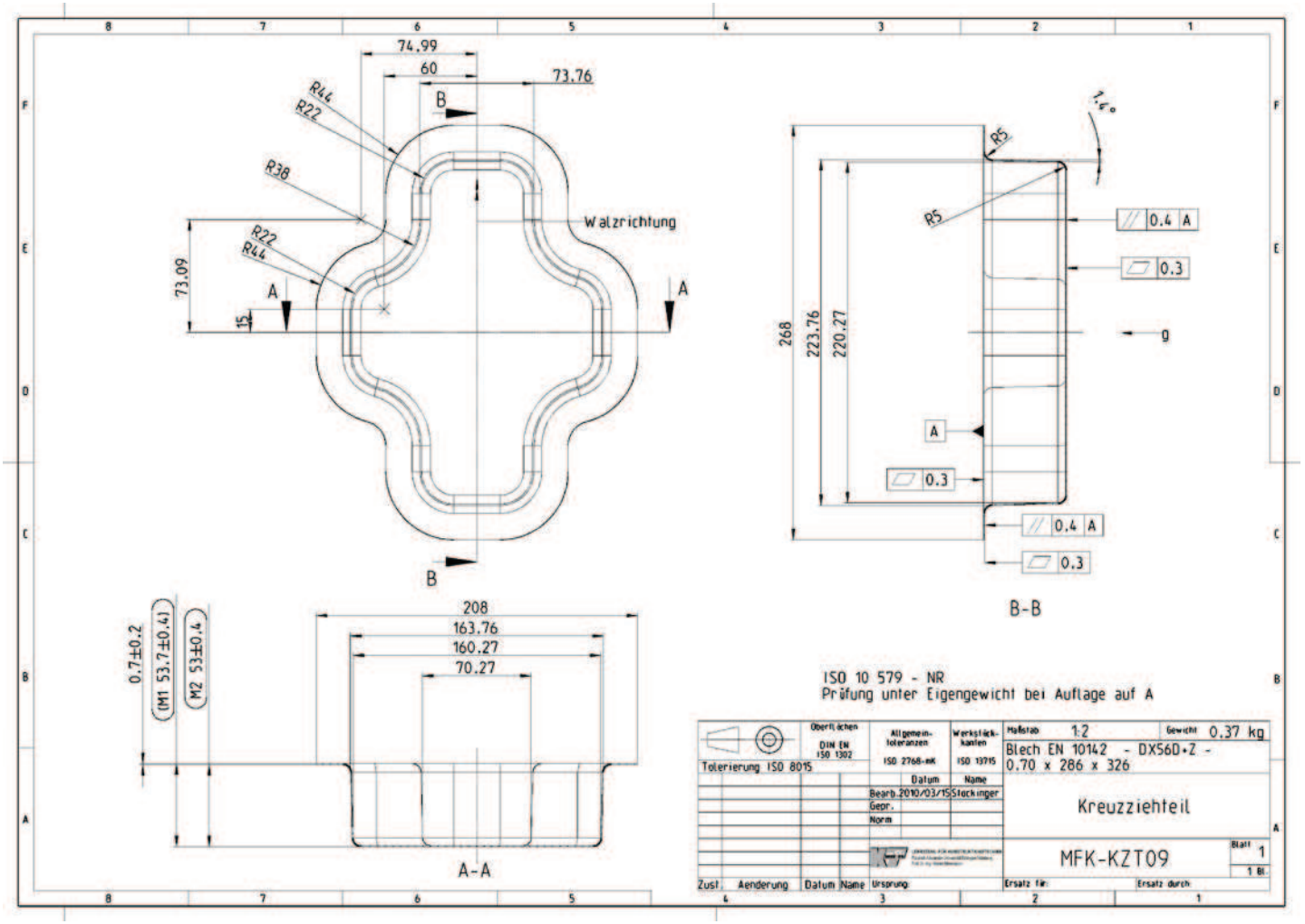
- Edges connection of skin models.

The skin model simulation methods are based on the local surface of a geometrical model. It should be considered how to connect each pieces of skin model to an entire skin model, since there are distortions on the connected edges.

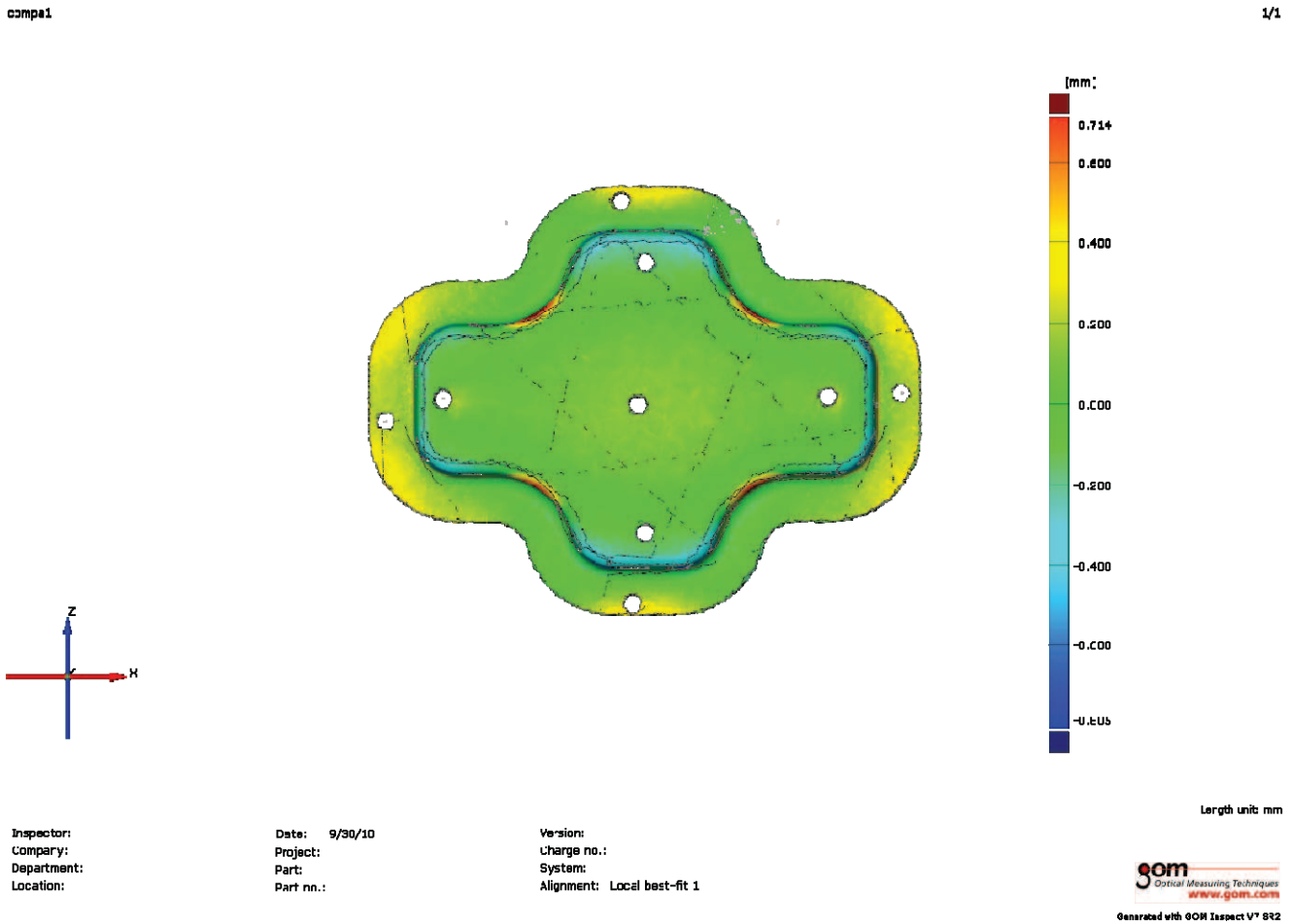
- Statistical model.

To predict reliable shape models that can reflect the principal characteristics of a training set, it should be investigate an efficient scope of the model coefficients.

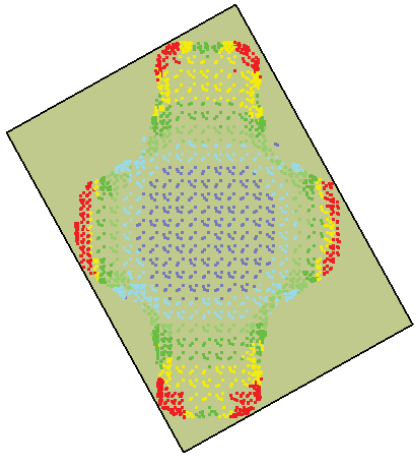
Appendix 1:

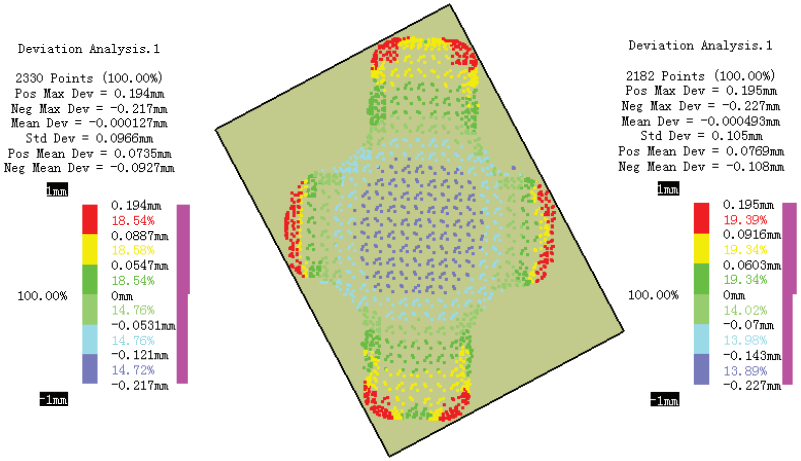

74.99
60
73.76
R44
R22
R38
Walzrichtung
73.09
15
A
B
B-B
R5
14°
0.4 A
0.3
268
223.76
220.27
g
208
163.76
160.27
70.27
0.7±0.2
(M1 53.7±0.4)
(M2 53±0.4
A-A
ISO 10 579 - NR
Prüfung unter Eigengewicht bei Auflage auf A
Tolerierung ISO 8015
Oberflächen DIN EN ISO 1302
Allgemeintoleranzen ISO 2768-mK
Werkstückkanten ISO 13715
Maßstab 1:2
Gewicht 0,37 kg
Blech EN 10142 - DX56D+Z -
0,70 x 286 x 326
Datum Name
Bearb. 2010/03/15 Stockinger
Gepr.
Norm
Kreuzziehteil
MFK-KZT09
Blatt 1
1 Bl.
Zust. Aenderung Datum Name Ursprung
Ersatz für:
Ersatz durch:

Appendix 2:

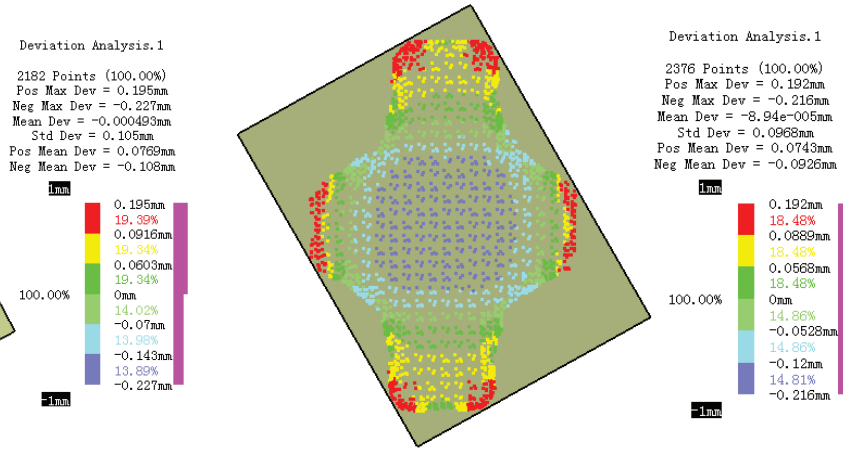

compa1
1/1
[mm]
0.714
0.600
0.400
0.200
0.000
-0.200
-0.400
-0.600
-0.685
Z
X
Inspector:
Company:
Department:
Location:
Date: 9/30/10
Project:
Part:
Part no.:
Version:
Charge no.:
System:
Alignment: Local best-fit 1
Length unit: mm
gom
Optical Measuring Techniques
www.gom.com
Generated with GOM Inspect V7 SR2

Appendix 3(1):



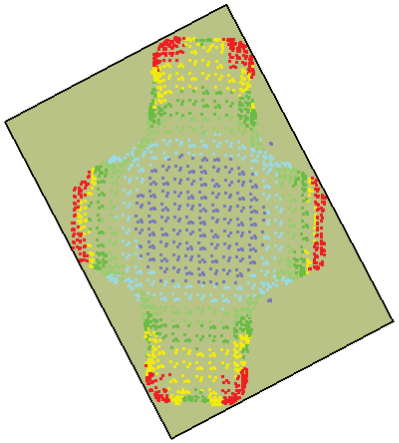


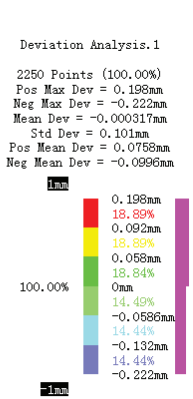
(a)



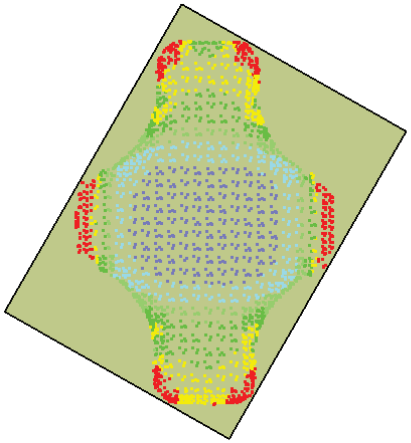
(b)

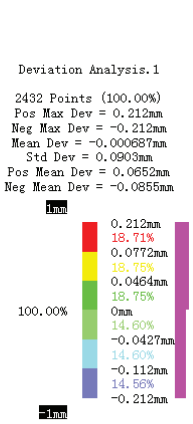
(c)



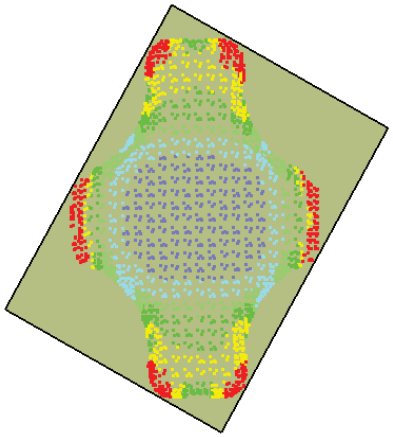


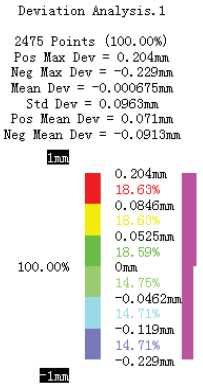
(d)





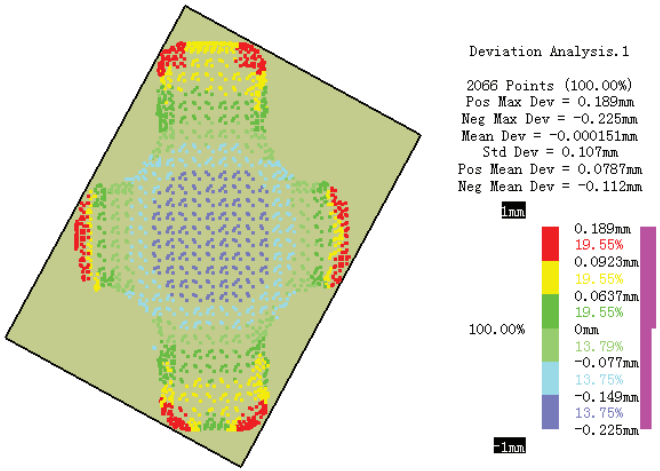
(e)



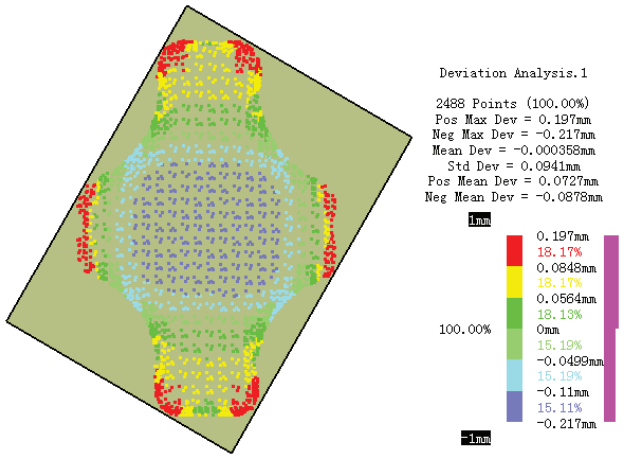


(f)

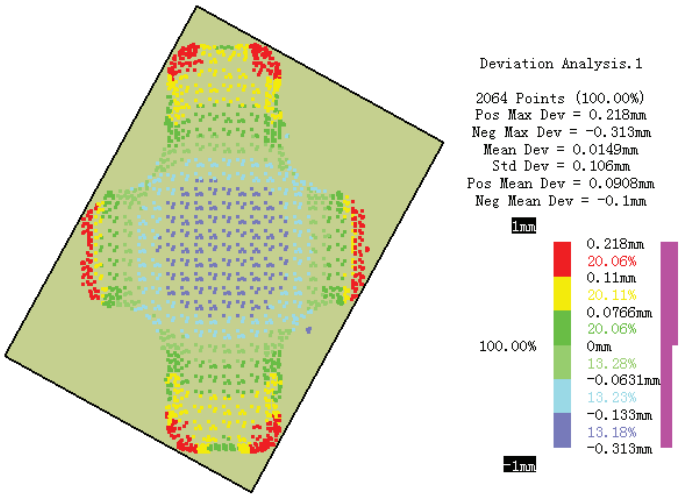
Appendix 3-(2):



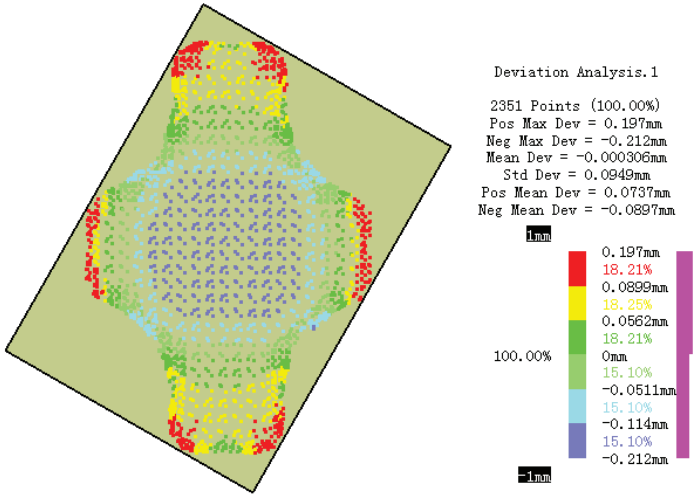
(g)



(h)

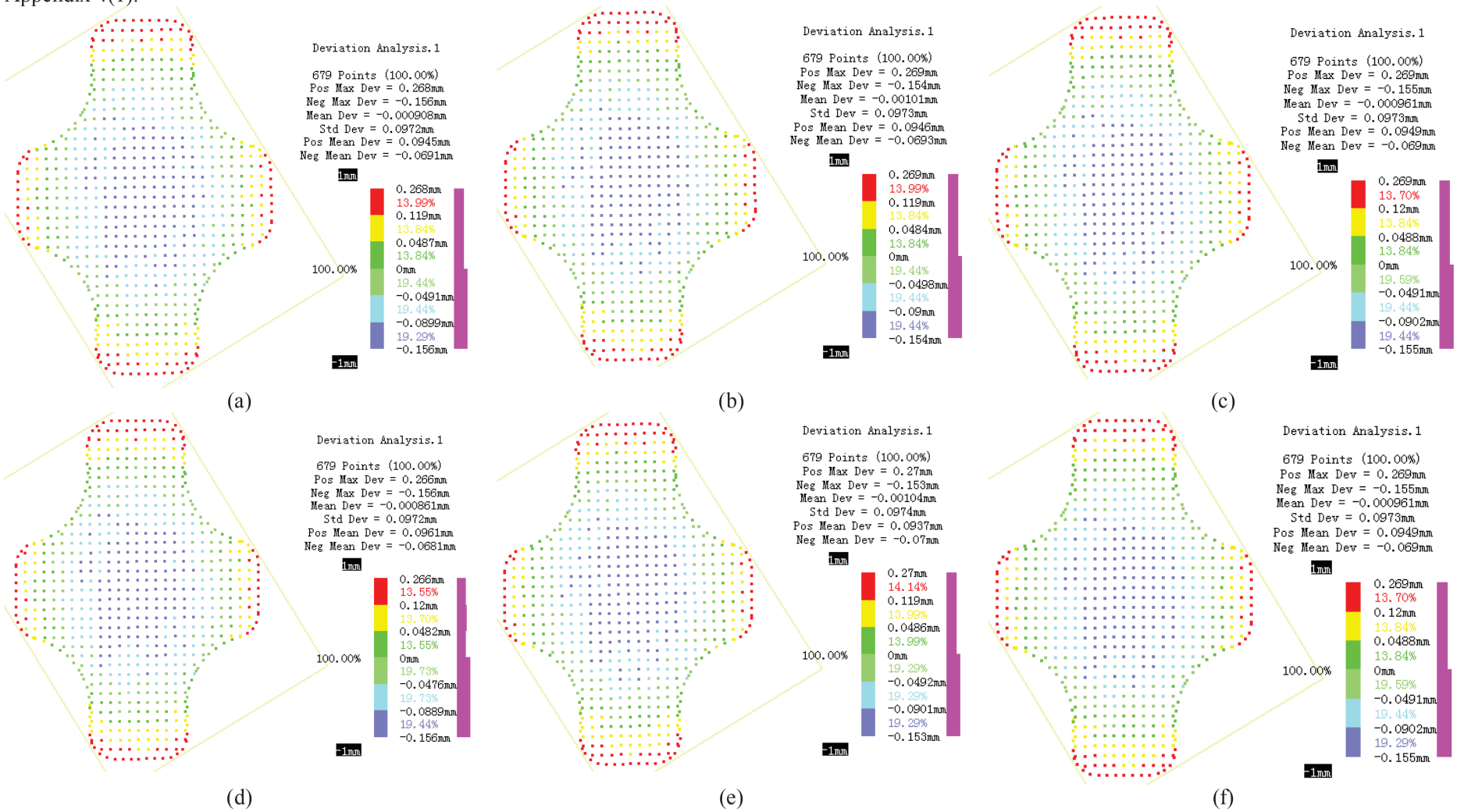


(i)

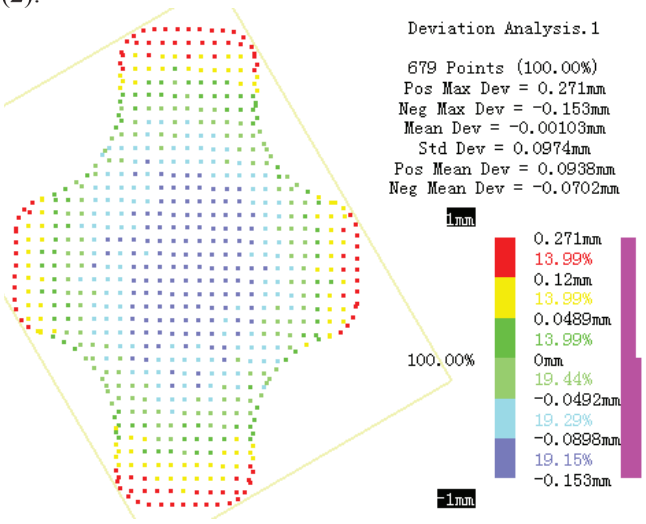


(j)

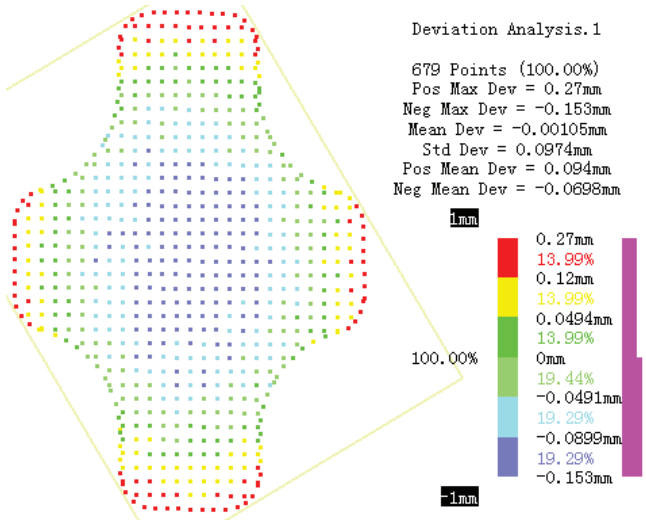
Appendix 4(1):


Deviation Analysis.1
679 Points (100.00%)
Pos Max Dev = 0.268mm
Neg Max Dev = -0.156mm
Mean Dev = -0.000908mm
Std Dev = 0.0972mm
Pos Mean Dev = 0.0945mm
Neg Mean Dev = -0.0691mm
1mm
0.268mm
13.99%
0.119mm
13.84%
0.0487mm
13.84%
0mm
19.44%
-0.0491mm
19.44%
-0.0899mm
19.29%
-0.156mm
100.00%
-1mm
(a)
Deviation Analysis.1
679 Points (100.00%)
Pos Max Dev = 0.269mm
Neg Max Dev = -0.154mm
Mean Dev = -0.00101mm
Std Dev = 0.0973mm
Pos Mean Dev = 0.0946mm
Neg Mean Dev = -0.0693mm
1mm
0.269mm
13.99%
0.119mm
13.84%
0.0484mm
13.84%
0mm
19.44%
-0.0498mm
19.44%
-0.09mm
19.44%
-0.154mm
100.00%
-1mm
(b)
Deviation Analysis.1
679 Points (100.00%)
Pos Max Dev = 0.269mm
Neg Max Dev = -0.155mm
Mean Dev = -0.000961mm
Std Dev = 0.0973mm
Pos Mean Dev = 0.0949mm
Neg Mean Dev = -0.069mm
1mm
0.269mm
13.70%
0.12mm
13.84%
0.0488mm
13.84%
0mm
19.59%
-0.0491mm
19.44%
-0.0902mm
19.44%
-0.155mm
100.00%
-1mm
(c)
Deviation Analysis.1
679 Points (100.00%)
Pos Max Dev = 0.266mm
Neg Max Dev = -0.156mm
Mean Dev = -0.000861mm
Std Dev = 0.0972mm
Pos Mean Dev = 0.0961mm
Neg Mean Dev = -0.0681mm
1mm
0.266mm
13.55%
0.12mm
13.70%
0.0482mm
13.55%
0mm
19.73%
-0.0476mm
19.73%
-0.0889mm
19.44%
-0.156mm
100.00%
-1mm
(d)
Deviation Analysis.1
679 Points (100.00%)
Pos Max Dev = 0.27mm
Neg Max Dev = -0.153mm
Mean Dev = -0.00104mm
Std Dev = 0.0974mm
Pos Mean Dev = 0.0937mm
Neg Mean Dev = -0.07mm
1mm
0.27mm
14.14%
0.119mm
13.99%
0.0486mm
13.99%
0mm
19.29%
-0.0492mm
19.29%
-0.0901mm
19.29%
-0.153mm
100.00%
-1mm
(e)
Deviation Analysis.1
679 Points (100.00%)
Pos Max Dev = 0.269mm
Neg Max Dev = -0.155mm
Mean Dev = -0.000961mm
Std Dev = 0.0973mm
Pos Mean Dev = 0.0949mm
Neg Mean Dev = -0.069mm
1mm
0.269mm
13.70%
0.12mm
13.84%
0.0488mm
13.84%
0mm
19.59%
-0.0491mm
19.44%
-0.0902mm
19.29%
-0.155mm
100.00%
-1mm
(f)

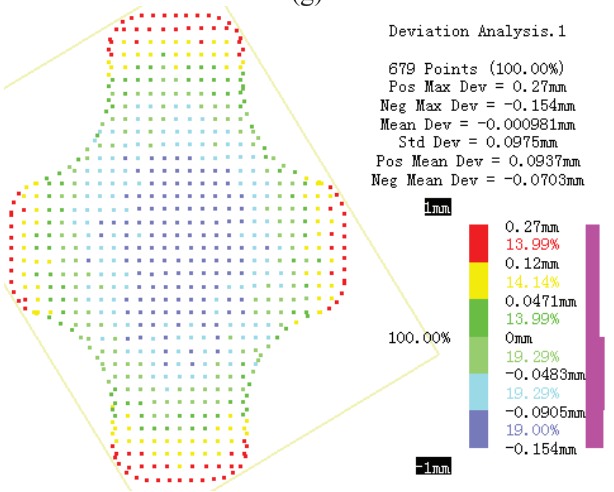
Appendix 4(2):



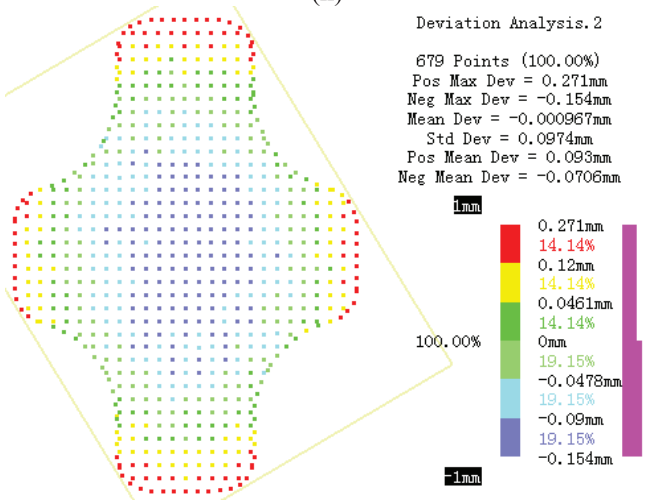
(g)



(h)



(i)



(j)

References

[Ahn 04] S. J. Ahn, Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space, LNCS, vol. 3151. Springer, Heidelberg, 2004

[Alexa 01] M. ALEXA, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, Point set surfaces, In VIS ’01: Proceedings of the conference on Visualization ’01, IEEE Computer Society, Washington, DC, USA, 21–28

[Alexa 03] M. ALEXA, J. Behr, D. COHEN-OR, S. Fleishman, D. Levin, C. T. Silva, Computing and rendering point set surfaces, IEEE Transactions on Visualization and Computer Graphics, 2003, 9(1): 3–15[Alliez 05] P. Alliez, C. Gotsman, Recent Advances in Compression of 3D Meshes (in Advances in Multiresolution for Geometric Modelling), Springer-Verlag, 2005

[Alliez 09] P. Alliez, Variational Approaches to Digital Geometry Processing, Habilitation à diriger des recherches, Université de Nice Sophia Antipolis, France, 2009

[Almhdie 07] A. Almhdie, C. Leger, M.Deriche, et al. 3D registration using a new implementation of the ICP algorithm baed on a comprehensive lookup matrix: application to medical. Pattern Recognition Letters, 28: 1523-153

[Anselmetti 93] B. Anselmetti, P. Bourdet, Optimization of a workpiece considering production requirements. Computers in Industry, 1993, 21(1):23-34

[Anselmetti 06] B. Anselmetti, Generation of functional tolerancing based on positioning features. Computer-Aided Desgn, 2006, 38(8): 902-919

[Anselmetti 10] Bernard Anselmetti, Robin Chavanne, Yang Jianxin, Nabil Anwer. Quick GPS: A new CAT system for single part tolerancing. Computer-Aided Design, 2010, 42(9): 768-780. (EI: 20103213137406)

[Anwer 03] N. ANWER, G. CID, L. MATHIEU, XML bsed tolerance mangemement for integrated design, 2003 Internation CIRP Design Seminar, COROM paper, 12p. Grenoble, France, May 12-14, 2003

[Aranda 10] S. Aranda, J. M. Linares, J. M. Sprauel, Best-fit criterion within the context of likelihood maximization estimation, Measurement 43 (2010): 538–548

[Aristides 80] G. Aristides, R. Requicha, Representations for rigid solids: Theory, methods, and systems, ACM Comput. Surv., 12(4):437-464, 1980.

[Bakolas 03] V. Bakolas, Numerical Generation of Arbitrarily Oriented Non-Gaussian Three-Dimensional Rough Surfaces, Wear, 2003, 254(5–6): 546–554.

[Ballu 95] A. Ballu, L. Mathieu, Univocal expression of functional and geometrical tolerances for design, manufacturing and inspection, Computer Aided Tolerancing, 4th CIRP Seminar, 1995, Tokyo, JAPAN, 31-46

[Baumgart 75] B. G.. Baumgart, Winged-Edge Polyhedron Representation for Computer Vision National Computer Conference, May 1 9 7 5, Stanford University

[Benkö 04] Pál Benkö, Tamás Várady, Segmentation methods for smooth point regions of conventional engineering objects, Computer-Aided Design, 36(2004), 511-523

[Besl 92] P. Besl, N. McKay, A Method for Registration of 3D Shapes, Trans. PAMI, Vol. 14, No. 2, 1992

[Bi 08] L. Bi, L. G. Wang, J. H. Chen, X. L. Feng, Spacial Boolean operatons of 3D mesh model, Huazhong Univ. of Sci. & Tech. (Natural Science Edition), 36(5):82-85, 2008.

[Biasotti 08] S. Biasotti, L. De Floriani, B, Falcidieno, et al. Description shapes by geometrical-topological properties of real functions, ACM, V (N): 1-102

[Bjorke 89] O. Bjorke, Computer-Aided Tolerancing (2nd), New York: ASME press, 1989

[Blackmore 97] D. Blackmore, G. Zhou, A new fractal model for anisotropic surfaces, in: Transactions of the Seventh International Conference on Metrology and Properties of Engineering Surfaces, Gothenburg, Sweden, 1997, 147–153.

[Bloomenthal 91] J. Bloomenthal, K. Shoemake, Convolution surfaces, Computer Graphics, 25 (4): 251-256, 1991

[Bolle 91] R. M. Bolle, B. C. Vemuri, On three-dimensional surface reconstruction methods. IEEE Trans. Pattern Anal. Mach. Intell, 1991, 13(1): 1–13

[Bookstein 78] F. L. Bookstein, The Measurement of Biological Shape and Shape Change. Lecture Notes in Biomathematics, Springer‐Verlag Berlin‐Heidelberg‐New York 24(1978): 199-219

[Botsch 05] M. Botsch, High Quality Surface Generation and efficient

Multiresolution Editing Based on Triangle Meshes, PhD thesis, RWTH Aachen, 2005. Ph.D. Thesis

[Botsch 10] M. Botsch, L. Kobbelt, M. Pauly, Pierre Alliez and Bruno Levy, Polygon Mesh Processing , AK Peters, ISBN 978-1-56881-426-1, 2010

[Bourdet 87] P. Bourdet, Contribution a la mesure tridimensionnelle: Modele d'identification geometrique des surfaces, metrologie fonctionnelle des pieces mecaniques, correction geometrique des machines a mesurer tridimensionlelles. Docteur ES-SCIENCES thesis, Universite de Nancy, Vol.1, 1987.

[Bourdet 96] P. Bourdet, L. Mathieu, C.Lartigue, A. Ballu, The concept of the small displacement torsor in metrology, advanced mathematical tools in metrology II, 40(1996): 110-122

[Brandt 83] S. Brandt, Statistical and Computational Methods in Data Analysis, North Holland, 1983

[Bronnimann 01] H. Brönnimann, Designing and Implementing a General Purpose Halfedge Data Structure, Algorithm Engineering 2001: 51-66

[Campatelli 11] G. Campatelli, Tolerance synthesis using axiomatic design, Proceedings of ICAD 2011, The Sixth International Conference on Axiomatic Design, Daejeon, March 30-31, 2011, 152-157

[Carmo 76] M. P. do Carmo. Differential Geometry of Curves and Surfaces. Englewood Cliffs, NJ: Prentice Hall, 1976.

[Caskey 90] G. Caskey, Y. Hari, R. Hocken, D. Palanivelu, J. Raja, R. Wilson, K. Chen, J. Yang, Sampling Techniques for Coordinate Measuring Machines, Design and Manufacturing Systems Conference, 1990, 779-786

[Cayre 04] F. Cayre, O. Devillers, F. Schmitt, H. Maitre, Watermarking 3d triangle meshes for authentication and integrity, Technical Report 5223, INRIA, 2004.

[Cgal 11] http://www.cgal.org/

[Chang 07] W. Chang, Surface Reconstruction from Points, UCSD CSE Technical Report CS2008-0922, 2007

[Chiabert 01] P. Chiabert, M. Costa, Probabilistic evaluation of invariant surfaces through the Parzen's method, 7th Cirp International Seminar on Computer-Aided Tolerancing, Ecole Normale, france

[Cho 01] N. Cho, J. Tu, Roundness modeling of machined parts for tolerance analysis, Precision Engineering, Journal of the International Societies for Precision Engineering and Nanotechnology, 25(2001):35-47

[Cho 02] N. Cho, J. F. Tu, Quantitative circularity tolerance analysis and design for2Dpercision assemblies, International Journal of Machine Tools & Manufactuer, 42(2002): 1391-1401

[Clement 91] A. Clement, A. Desrochers, A. Riviere, Theory and pratice of 3D tolerancing for assembly, CIRP seminar on Computer aided tolerancing, Penn State University, USA, May 1991

[Clement 93] A. Clement, A. Riviere, Tolerancing versus nominal modeling in next generation CAD/CAM system, CIRP Seminar on Computer Aided Tolerancing, ENS Cachan, Paris, April 1993, 97-113

[Clement 94] A. Clement, A. Riviere, M. Temmerman, Cotation tridimensionelle des systemes mecaniques: theorie and pratique, PYC Edition, France, ISBN 2-85330-132-X, 1994

[Clement 97] A. Clement, The TTRSs: 13 Oriented constraints for dimensioning, tolerancing & inspection, Advanced Mathematical Tools in Metrology III, 1997 World Scinentific Publishing Company, 24-42

[Cohen-Steiner 97] D. Cohen-Steiner, J.-M. Morvan, Restricted Delaunay triangulation and normal cycle, In 19th Annual ACM Symposium on Computational Geometry'03, 312–321, 2003

[Cohen-Steiner 03] D. Cohen-Steiner, J.M. Morvan, Restricted Delaunay Triangulation and Normal Cycle, Proceedings of 19th Annual ACM Symposium on Computational Geometry, 312-21, 2003

[Colosimo 07] B. M. Colosimo, M. Pacella, On the use of principal component analysis to identify systematic patterns in roundness profiles, Quality and reliability engineering international, 23(2007): 707 -725.

[Cootes 01] T.F. Cootes and C. J. Taylor. Statistical models of appearance for medical image analysis and computer vision. In Proc. SPIE Medical Imaging, 2001, 236-248

[Cootes 92] T.F. Cootes, C. J. Taylor. Active Shape Models - 'Smart Snakes'. In

Proceedings of British Machine Vision Conference. Springer-Verlag, 1992, pp.266-275.

[Cressie 93] N. A. C. Cressie, Statistics for Spatial Data, Wiley, New York, 1993.

[Cubeles-Valade 98] C. Cubeles-Valade, A. Riviere, Proceedings, Nominal and Actual Geometry Explicit Declaration, Proceedings Integrated Design and Manufacturing in Mechanical Engineering, 1998, Compiegne, France, Batoz, Chedmail, and F. Cognet, eds., Kluwer Academic, Dordercht, 341-348

[Dantan 05] J. Y. Dantan, L. Mathieu, A. Ballu, Tolerance synthesis : quantifier notion and virtual boudary, Computer-Aided Design, 2005, 37(2) : 231-240

[Davenport 92] Davenport, J. H.. "Primality testing revisited". Proceeding ISSAC '92 Papers from the international symposium on Symbolic and algebraic computation: 123 129. doi:10.1145/143242.143290.

[Davidson 02] J. K. Davidson, A. Mujezinovic, J. J. Shah, A new mathematical model for geometric tolerance as applied to round faces, ASME Journal of Mechanical Design, 2002(124):609-622

[Desta 03] M. T. Desta, H. Y. Feng, D. O. Yang, Characterization of general systematic for errors for circular features, International Journal of Machine Tools & Manufacture, 43(2003): 1069-1078

[Devadoss 11] S.L. Devadoss, J. O'Rourke, Discrete and Computational Geometry, Princeton University Press, ISBN 978-0691145532, 2011

[Dimitrov 08]D. Dimitrov, Geometric Applications of Principal Component Analysis, PhD Thesis, Freie Universitat Berlin, 2008

[Dimitrov 09] D. Dimitrov, C. Knauer, K. Kriegel, G. Rote, Bounds on the quality of the PCA bounding boxes, Computational Geometry, 42(2009): 772–789

[Dowling 97] M. Dowling, P. Griffin, K. Tsui, C. Zhou, Statistical issues in geometric feature inspection using coordinate measuring machines, Technometrics, (39):3-17

[Dryden 98] I. L. Dryden, K. V. Mardia. Sttatistical shape analysis. John Wiley and Sons, 1998

[Durrleman 09] S. Durrleman, X. Pennec, A. Trouvé, N. Ayache, Statistical Models on Sets of Curves and Surfaces based on Currents, Medical Image Analysis,

13(5):793-808, October 2009

[Eastman 82] C. M. Eastman, S. F. Weiss, Tree structures for high dimensionality nearest neighbor searching, Information Systems, vol. 7, No. 2 1982

[ElMaraghy 93] H. A. ElMaraghy, W. H. ElMaraghy, A system for modeling geometric tolerances for mechanical design, Proc 3rd CIRP seminar on Computer Aided Tolerancing, ENS Cachan, 1993, 11-24

[ElMaraghy 05] W. ElMaraghy, P. Bourdet, H. ElMaraghy, Small displacement torsor theory and various compensation schemes for complex form realization, CPI'2005

[Etesmi 93] F. A. Etesmi, Mathematical model for geometric tolerancing, Journal of Mechanical Design, 1993, 115(3):81-86

[Fleute 99] M. Fleute, S. Lavallée, R. Julliard, Incorporating a Statistically Based Shape Model into a System for Computer-Assisted Anterior Cruciate Ligament Surgery, Medical Image Analysis, 3(3): 209-222, 1999.

[Fog 01] www.agner.org/random/theory

[Forslund 09] K. Forslund, R. Söderberg, The Effects of Geometrical Variation on Perceived Quality - A Close Look at the Concept of Visual Robustness, Proceedings of the 11th International Conference on Computer Aided Tolerancing, March 26-27, Annecy, France, 2009.

[Funkhouser 05] T. Funkhouser, M. Kazhdan, P. Min, P. Shilane, Shape-based retrieval and analysis of 3d models, Communications of the ACM, 48(6):58-64, 2005.

[Gerber 09]] K. Gerber. Statistical shape analysis with SAS. Behavior Research Methods.2009

[Giordano 01] M. Giordano, B. Kataya, E. Pairel. Tolerance analysis and synthesis by means of clearance and deviation spaces. Proc. 6th CIRP International Seminar on Computer-aided Tolerancing, Enschede, Netherlands, 2001, 145-154

[Giordano 07] M. Giordano, S. Samper, J.P. Petit, Tolerance analysis and synthesis by means of deviation domains, axi-symmetric cases, Journal of Models for Computer Aided Tolerancing in Design and Manufacturing, Springer, 85-94, 2007

[Gossard 88] D. C. Gossard, R. P. Zuffante, H. Sakurai, Representing dimensions, tolerances and features in MCAE systems, IEEE Computer Graphics & Applications, 1988, 22(1): 51-59

[Gouraud 71] H. Gouraud, Continuous shading of curved surfaces, IEEE Transactions on Computers C-20 (6): 623–629, 1971

[Gouskov 98] A. Gouskov, S. Tichkiewitch, 1999, Proceedings, Influence of the Machine-Tool Defects on the Formation of the Residual Defects on a Surface: Application to Turning, Integrated Design and Manufacturing in Mechanical Engineering ’98, Compiegne, France, Batoz, Chedmail, and F. Cognet, eds., Kluwer Academic, Dordrecht, 341-348

[Gross 07] M. Gross, H. Pfister (Eds.), Point Based Graphics, Morgan Kaufmann, 2007.

[Gupta 91] S. Gupta, J. U. Turner, Variational solid modeling for tolerance analysis. Proceedings of the ASME International Computers in Engineering Conference, 1991(1): 487-494

[Haimes 98] R. Haimes, G. Follen, Computational Analysis Programming Interface, Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulaitons, University of Greenwich, June 1998

[Henke 99] R. P. Henke, K. D. Summerhays, J. M. Baldwin, R. M. Cassou, C. W. Brown, Methods for evaluation of systematic geometric deviations in machined parts and their relationships to process variables, Precision Engineering, 23(1999): 273-292

[Hildebrandt 04] K. Hildebrandt, K. Polthier. Anisotropic filtering of non-linear surface features, Symposium of Eurographics on Computer Graphics Forum, 23 (3): 391-400, 2004

[Hildebrandt 06] K. Hildebrandt, K. Polthier, and M. Wardetzky. On the Convergence of Metric and Geometric Properties of Polyhedral Surfaces. In Geometriae Dedicata, pp. 89-112. Aire-la-Ville, Switzerland: Eurographics Association, 2006.

[Hillyard 78] R. C. Hillyard, I. C. Braid, Analysis of dimensions and tolerances in computer-aided mechanical design, Computer-Aided Design, 1978, 10(3):161-166

[Hoffmann 82] P. Hoffmann, Analysis of tolerances and process inaccuracies in discrete part manufacturing, Computer-Aided Design, 1982, 14(2): 83-88

[Hoppe 92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, ACM SIGGRAPH 1992 Proceedings, 71-78

[Hufnagel 09] H. Hufnagel, X. Pennec, J. Ehrhardt etc. Computation of a probabilistic statistical shape model in a maximum-a-posteriori framework. Methods Inf Med. 2009;48(4):314-9. Epub 2009 Jun 19

[ISO 2005] ISO 17450-1, 2005, Geometrical Product Specifications (GPS)—General concepts—Part 1: Model for geometrical specification and verification.

[Jagannathan 07] Anupama Jagannathan, Eric L. Miller, Three-Dimensional Surface Mesh Segmentation Using Curvedness-Based Region Growing Approach, IEEE Transactions on Pattern Analysis and Machine Intelligence. 29(12): 2195-2204, 2007.

[Jayaraman 89] R. Jayaraman, V. Srinivasan, Geometric tolerancing I: Virtual boundary requiment. IBM Journal of Research Development, 1989, 33(2):90-104

[Johnson 85] R. H. Johnson, Dimensioning and Tolerancing Final Report, R84-GM-02-2, CAM-I, Arlington, Texas, USA, 1985

[Kalos 09] M. H. Kalos, P. A. Whitlock, Monte Carlo Methods, Second Edition, Wiley-VCH Verlag GmbH & Co, 2009

[Ke 04] Yinglin Ke, Shuqian Fan, Segmentation Technique Based on Boundary Features of Point Cloud, Chinese Journal of Mechanical Engineering. 40(9): 116-120, 2004.

[Ke 06] Y. L. Ke, X. Chen, The analysis of geometrical property and segmentation on cloud of point, Chinese Journal of Mechanical Engineering. 42(8): 7-15, 2006.

[Kendall 77] D. G. Kendall, The diffusion of shape, Advances in Applied Probability, 1977(9): 428-430.

[Kendall 84] D.G. Kendall, Shape manifolds, procrustean metrics, and complex projective spaces, Bulletin of the London Mathematical Society, 16 (2): 81-121

[Kettner 99] L. Kettner. Using generic programming for designing a data structure for polyhedral surfaces, Journal of Computational Geometry: Theory and

Applications, 13: 65-90, 1999

[Kim 00] W. S. Kim, S. Raman, On the selection of flatness measurement points in coordinate measuring machine inspection, International Journal of Machine Tools & Manufacture, 40(2000),:427-443

[Koenderink 92] J.J. Koenderink, R.J. van Doorn. Surface shape and curvature scales, Journal of Imaging and Vision Computing, 10 (8): 557-65, 1992

[Kroese 11] D.P. Kroese, T. Taimre, Z.I. Botev, Handbook of Monte Carlo Methods, Wiley Series in Probability and Statistics, John Wiley & Sons, New York, 2011.

[Kurokawa 05] S. Kurokawa, Y. Ariura, Evaluation of shot peened surfaces using characterization technique of three-dimensional surface topography, Journal of Physics: Conference Series, 13(2005): 9-12

[Lahanas 00] M. Lahanas, T. Kemmerer, N. Milickovic, D. B. K. Karouzakis, N. Zamboglou, Optimized bounding boxes for three-dimensional treatment planning in brachytherapy, Med. Phys. 27(2000):2333–2342

[Lancaster 81] P. Lancaster, K. Salkauskas, Surfaces Generated by Moving Least Squares Methods, Mathematics of Computation. 37(1):141-158, 1981.

[Leonidas 85] L. J. Guibas, J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Transactions on Graphics, 1985, 4(2): 75–123

[Li 98] H. L. Li, Study on sampling strategy in coordinate measuring machines, University of Shanghai for Science and Technology. 20(3): 251-253, 1998.

[Li 08] H. Li, R. W. Sumner, M. Pauly, Global correspondence optimization for non-rigid registration of depth scans, Computer Graphics Forum (SGP), 2008

[Lim 96] T. S. Lim, W. Y. Loh, A comparison of tests of equality of variances, Computational Statistical & Data Analysis 22(1996): 287-301

[Liu 08a] 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, 677-682, 2008

[Liu 08b] Yu Liu, Youlun Xiong, Automatic segmentation of unorganized noisy point clouds based on the Gaussian map, Computer-Aided Design, 40(2008), 576-594

[Lomonosov 04] 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

[Luby 96] M. Luby, Pseudo randomness and Cryptographic Applications, Princeton Univ Press, 1996.

[Ma 10] J. Ma, M.I. Miller, L. Younes. A Bayesian Generative Model for Surface Template Estimation. International Journal of Biomedical Imaging. 2011, 1-14

[Mathieu 03] L. Mathieu, A. Ballu. GEOSPELLING: a common language for Geometrical Product Specification and Verification to express method uncertainty. Proceedings of the 8th CIRP Seminar on Computer Aided Tolerancing, Charlotte, USA, April, 2003

[Mathieu 05] L. Mathieu, A. Ballu, A model for a coherent and complete tolerancing process, Proceedings of the 9th CIRP Seminar on Computer Aided Tolerancing, Tempe (Arizona, USA), April 2005, 1-10

[Mathieu 07] L. Mathieu, A. Ballu, “GPS card”: a tool for univocal expression of geometrical specifications, Proceedings of the 10th CIRP Seminar on Computer Aided Tolerancing, 1-10, 2007

[McLouglin 10] Tony McLouglin, Robert S. Laramee, Ronald Peikert, Frits H. Post and Min Chen, Over Two Decades of Integration-Based Geometric Flow Visualization, in Computer Graphics Forum, 29(6):1807-1829, 2010

[Meek 00] 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

[MeshLab] http://meshlab.sourceforge.net/

[Meyer 02] 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

[Michael 82] W. Michael, J. N. Siddall, The optimizaiton problem with optimal tolerance assignment and full acceptance, ASME Journal of Mechanical Design, 1982, 104(4): 855-860

[Mokhtarian 01] F. Mokhtarian, N. Khalili, P. Yuen, Multi-scale free-form 3D object recongnition using 3D models, Journal of Image and Vision Computing, 19(5): 271-281, 2001

[Mujezinovic 04] A. Mujezinovic, J. K. Davidson, J. J. Shah, A new mathematical model for geometric tolerance as applied to polygonal faces, Journal of Mechanical Design, 2004, 126(3):504-518

[Nackman 84] L. Nackman, Two-dimensional critical point configuration graphs, IEEE Transactions on Pattern Analysis and machine Intelligence, 6 (4): 442-450

[Nassef 93]A. Nassef, H. A. ElMaraghy, Allocaiton of tolerance types and values using genetic algorithms, proc, 3rd CIRP seminar on Computer Aided Tolerancing, ENS Cachan, 1993, 147-156

[Naylor 96] B. Naylor, Computational representations of geometry, SIGGRAPH'96 Course Notes, 2996

[Nealen 92] A. Nealen, An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Date Approximation and Interpolation, Discrete Geometric Modeling Group TU Darmstadt. 1992.

[Niu 07] J. Niu, Z. Li, G. Salvendy. Mathematical methods for shape analysis and fom comparison in 3D anthropometry : a literature review. Digital Human Modeling. HCII 2007, LNCS 4561, 161-170

[O'neill 97] B. O'neill. Elementary Differential Geometry, Academic Press, 1997

[OpenMesh]http://www.openmesh.org/

[Pasko 04] A. Pasko, V. Adzhiev, Function-based shape modeling: mathematical framework and specialized language, Automated Deduction in Geometry, Lecture Notes in Artificial Intelligence 2930, Ed. F. Winkler, Springer-Verlag, 132-160, 2004

[Patir 78] N. Patir, A Numerical Procedure for Random Generation of Rough Surfaces, Wear, 47(2),1978, 263–277.

[Pearson 01] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine, 6(2):559–572, 1901

[Peridier 11] V. J. Peridier, FacetSurface Representation, The Mathematica Journal,

13(2011)

[Peterson 98] I. Peterson. The Jungles of Randomness: A Mathematical Safari. Wiley, NY, 1998. (pp. 178) ISBN 0-471-16449-6

[Petitjean 02]S. Petitjean. A Survey of Methods for Recovering Quadrics in Triangle Meshes. ACM Computing Surveys 34:2 (2002), 211-262.

[Pramanik] N. Pramanik, A generic framework for design evolution: Form function to form mapping to synthesis of tolerances [D]. Syracuse: Syracuse University, 2003

[Reizer 11] R. Reizera, L. Galdab, A. Dzierwab, P. Pawlus, Simulation of textured surface topography during a low wear process, Tribol Int, 2011

[Requicha 86] A. A. G. Requicha, S. Chan, Representation of geometric features, tolerances and attributes in solid modelers based on constructive solid geometry. IEEE Journal of Robert & Automation, 1986, 2(3): 156-165

[Robbiano 10] F. Robbiano, Descriptions of 3D objects based on concepts, content, and context, PhD Thesis, Universita degli Studi di Genova, 2010

[Ronald 88] J. Ronald, J. Evans, N. Boersma etc. The Entropy of a Poisson Distribution. SIAM Review 30 (2): 314–317. doi:10.1137/1030059. 1988

[Rossignac 86] J. Rossignac, A. Requicha, Offsetting operations in solid modeling, Computer Aided Geometric Design, 3(2): 129-148, 1986

[Rouchka 97] E.C. Rouchka. A Brief Overview of Gibbs Sampling. IBC Statistics Study Group, Institute for Biomedical Computing, May 20, 1997

[Roy 91] U. Roy, C R Liu, T. C. Woo. Review of dimensioning and tolerancing: representation and processing. Computer-Aided Design, 23(7):466-483

[Roy 97] U. Roy, Y. C. Fang, Optimal tolerance re-allocation for the generative process sequence, IIE Transactions, 1997, 29: 37-44

[Roy 98a] U. Roy, B. Li, Representation and interpretation of geometric tolerances for polyhedral objects I: Form tolerances, Computer-Aided Design, 1998, 30(2):151-161

[Roy 98b] U. Roy, B. Li, Representation and interpretation of geometric tolerances for polyhedral objects II: Size, orientation and position tolerances, Computer-Aided

Design, 1999, 31(4):273-285

[Rubinstein 81] R. Y. Rubinstein, Simulation and the Monte Carlo method, John Wiley & Sons, ISBN 0-471-08917-6, 1981

[Rusinkiewicz 01] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, In 3D Digital Imaging and Modeling, 2001

[Stockinger 09] A. Stockinger,; H. Meerkamm, Concept for the Integration of Manufacturing Simulations into Tolerance Analysis, 11th CIRP International Conference on Computer Aided Tolerancing, Annecy. 2009

[Stockinger 10] A. Stockinger, S. Wittmann, M. Martinek, S. Wartzack, H. Meerkamm, Virtual Assembly Analysis: Standard Tolerance Analysis Compared to Manufacturing Simulation and Relative Positioning, In: Marjanovic D., Štorga M., Pavkovic N., Bojcetic N. (Hrsg.): Proceedings of the 11th International Design Conference DESIGN 2010, Dubrovnik: 2010, S. 1421-1430.

[Stockinger 11] A. Stockinger, Computer Aided Robust Design – Verknüpfung rechnerunterstützter Entwicklung und virtueller Fertigung als Baustein des Toleranzmanagements, [D], Düsseldorf : VDI-Verlag, 2011. - Fortschritt-Bericht VDI Reihe 1; 409

[Salomons 96] O. W. Salomons, H. J. Jonge Poerink, F. J. Haalboom, etc., A computer aided tolerancing tool I: Tolerance specification, Computers in Industry 31(1996): 161-174

[Samper 07] S. Samper, F. Formosa, Form Defects Tolerancing by Natural Modes Analysis, Journal of Computing and Information Science in Engineering, 7(2007):44-51

[Sanyal 09] J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, Robert Moorhead, A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets, IEEE Transactions on Visualization and Computer Graphics,15(6): 1209-1218, 2009

[Shah 98] J.J. Shah, Y. Yan, B. C. Zhang, Dimension and tolerance modeling and transformations in feature based design and manufacturing, Journal of Intelligent Manufacturing, 1998, 9(5):475-488

[Shapird 01] V. Shapird, Solid modeling, Technical report, University of Wisconsin

Madison, 2001

[Shen 05] H. C. Shen, J. H. Li, L. S. Zhou, Estimation of Triangular Mesh Vertex Normal Vector and Discrete Curvature, Computer Engineering and Application, 2005, (6): 12-15

[Sombstay] H. Sombstay,T. Nguyen Huu, Variabilité d'une fabrication en tenant compte des défauts de forme, Revue de Statistique Appliquée, 1958, 6(1): 17-36

[Sourin 96] A. Sourin, A. Pasko, Function representation for sweeping by a moving solid, IEEE Transactions on Visualization and Computer Graphics, 2(1): 11-18, 1996

[Speckhart 72] F. H. Speckhart, Calculation of tolerance based on a minimum cost approach, ASME Journal of Engineering for Industry, 1972, 94(2): 447-453

[Srinark 08] T. Srinark, Lecture note on differential geometry, 2008

[Srinivasan 89] V. Srinivasan, Geometric tolerancing II: Conditional tolerance, IBM Journal of Research Development, 1989, 33(2): 105-122

[Stegmann 02] M.B. Stegmann, D.D. Gomez. A brief introduction to statistical shape analysis. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2002

[Steiner 03] D. Cohen-Steiner, J.M. Morvan, Restricted Delaunay Triangulation and Normal Cycle, Proceedings of 19th Annual ACM Symposium on Computational Geometry, 312-21, 2003

[Stout 10] K.J. Stout, L. Blunt, Three Dimensional Surface Topography, Butterworth Heinemann, 2010

[Tamal 06] K. D. Tamal, Curve and Surface Reconstruction: Algorithms with Mathematical Analysis, Cambridge Monographs on Applied and Computational Mathematics, 2006

[Taron 07] M. Taron, Registration & Modeling of Shapes with Uncertainties: Contributions and Applications to Knowledge Based Segmentation, PhD Thesis, Ecole Des Ponts Paristech, 2007

[Taubin 95] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1995, 902-907

[Taubin 95] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 902-907

[Taygeta 95] http://www.taygeta.com/about.html

[Tech 01] T. B. Tech, A. Senthil-Kumar, V. Subramanian, A CAD integrated analysis of flatness in a form tolerance zone, Computer-Aided-Design, 2001(33):853-865

[Thurmer 98] G. Thürmer, C. A. Wüthrich. Varying Neighborhood Parameters for the Computation of Normals on Surfaces in Discrete Space. Computer Graphics International 1998: 616-625

[Tsoi 07] K.H. Tsoi, K.H. Leung, P.H.W. Leong. High performance physical random number generator. IET Comput. Digit. Tech., 2007, 1, (4): 349-352

[Turner 87] J. U. Turner, Tolerances in computer-aided geometric design, [D], Rensselaer Polytechnic Insititute, Troy, 1987

[Turner 88] J. U. Turner, M. J. Wozny, A mathematical theory of tolerances, Geometric Modeling for CAD Applications, 1988, 163-187

[Turner 90] J. U. Turener, M. J. Wozny, The M-space theory of tolerances, Proceedings of the ASME 16th Design Automation Conference, 1990 (1): 217-225

[Visual 11] http://msdn.microsoft.com/en-us/visualc/aa336395

[Wang 06] H. Wang, N. Pramanik, U. Roy, etc., A Scheme for Mapping Tolerance Specifications to Generalized Deviation Space for Use in Tolerance Synthesis and Analysis, IEEE Transactions on Automation Science and Engineering, 2006(3):81-91

[Wijgerse 07] S. Wijgerse. Generating realistic city boundaries using two-dimensional Perlin noise. PhD thesis. University of Twente. EEMCS, 2007

[Wirtz 93] A. Wirtz, Vectorial tolerancing a basic element for quality control, 3rd CIRP seminar on Computer Aided Tolerancing, ENS Cachan, 1993, 115-128

[Woo 02] H. WOO, E. KANG, S. WANG, et al, A new segmentation method for point cloud data, International Journal of Machine Tools & Manufacture. 42(2):167-178, 2002

[Wu 07] WU Shixiong, WANG Chengyong, Segmentation on random point cloud,

Chinese Journal of Mechanical Engineering. 43(2): 230-233, 2007.

[Xia 08]H. F. Xia, Y. Ding, J. Wang, Gaussian process method for form error assessment using coordinate measurements, IIE Transactions, 2008, 40(10):931-946

[Xu 08] X. L. Xu, Study on the Theories and Methods of Functional Tolerancing Based on New GPS, [D], Department of Mechanical Engineering, Zhejiang University, 2008

[Yang 95] Z. Yang, C. Menq. Uncertainty Analysis for Coordinate Estimation Using Discrete Measurement Data. Proceedings of Manufacturing Science and Engineering. ASME International Mechanical Engineering Congress and Exposition, November 12-17, 1995

[Yang 99] M. YANG, E. LEE, Segmentation of measured point data using a parametric quadric surface approximation, Computer-Aided Design. 31(7): 449-457, 1999.

[Yang 00] T. H. Yang, J. Jackman, Form error estimation using spatial statistics, ASME Journal of Manufacturing Science and Engineering 122(2000): 262-272

[Zhang 96] Y. F. Zhang, A. Y. C. Nee, J. Y. H. Fuh, K. S. Neo, H. K. Loy, A neutral network approach to determining optimal inspection sampling size for CMM, Computer-integrated manufacturing systems. 9(3): 161-169, 1996.

[Zhang 00] S. G. Zhang, A. Ajmal, J. Wootton, A. Chisholm, A feature based inspection process planning system for co-ordinate measuring machine (CMM), Journal of Materials Processing Technology. 107(1-3): 111-118, 2000.

[Zhang 04a] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognition, 37, 1-19, 2004

[Zhang 04b] 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, 205-209, 2004

[Zhang 05] Zhang X. D., Zhang C., Wang B., Feng S. C., Unified functional tolerancing approach for precision cylindrical components, International Journal of Production Research, Vol.43, No.1,1 January, 2005, 25-47

[Zhang 07] L. N. Zhang, Y. H. Zhen, P. Zhen, Model of Extraction Operation and the

Research of its Application Specification Based on GPS, Journal of Mechanical Strength, 2007, 29(4):632-636

[Zhao 10] H. B. Zhao, Multisensor integration and discrete geometry processing for coordinate metrology. PhD thesis, 2001, ENS Cachan, Jan.

[Zhang 11] X. Zhang, X. Jiang, P. J. Scott, Minimum Zone Evaluation of the Form Errors of Quadric Surfaces. Precision Engineering, 2011, 35 (2): 383-389. ISSN 0141-6359

[Zou 07] M. Zou, B. Yu, Y. Feng, X. Peng, A Monte Carlo Method for Simulating Fractal Surfaces, Physica A, 2007, 386(1): 176–186