2008

Manufacturability analysis for non-feature-based objects

Ye Li
Iowa State University

Recommended Citation

Li, Ye, "Manufacturability analysis for non-feature-based objects" (2008). *Retrospective Theses and Dissertations*. 15690.
https://lib.dr.iastate.edu/rtd/15690

Manufacturability analysis for non-feature-based objects

by

Ye Li

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Matthew C. Frank, Major Professor
Frank E. Peters
Douglas D. Gemmill
Palaniappa A. Molian
Eliot H. Winer

Iowa State University

Ames, Iowa

2008

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

ABSTRACT

This dissertation presents a general methodology for evaluating key manufacturability indicators using an approach that does not require feature recognition, or feature-based design input. The contributions involve methods for computing three manufacturability indicators that can be applied in a hierarchical manner. The analysis begins with the computation of *visibility*, which determines the potential manufacturability of a part using material removal processes such as CNC machining. This manufacturability indicator is purely based on accessibility, without considering the actual machine setup and tooling. Then, the analysis becomes more specific by analyzing the complexity in setup planning for the part; i.e. how the part geometry can be oriented to a cutting tool in an accessible manner. This indicator establishes if the part geometry is accessible about an *axis of rotation*, namely, whether it can be manufactured on a 4th-axis indexed machining system. The third indicator is geometric *machinability*, which is computed for each machining operation to indicate the actual manufacturability when employing a cutting tool with specific shape and size. The three manufacturability indicators presented in this dissertation are usable as steps in a process; however they can be executed alone or hierarchically in order to render manufacturability information. At the end of this dissertation, a *Multi-Layered Visibility Map* is proposed, which would serve as a re-design mechanism that can guide a part design toward increased manufacturability.

CHAPTER 1. GENERAL INTRODUCTION

Introduction

Manufacturability refers to the relative difficulty in the process of physically converting raw material into a finished product on machines according to a design specification. Manufacturability of a product design has become an important issue due to the historic barriers between the design and manufacturing departments in a company. When a designer creates a design model, the major focus is to make the design optimal from functional or aesthetics perspectives that may not be agreeable to manufacturing engineers. This could lead to difficulty in manufacturing practices and even cause rework effort in the modification of designs to improve manufacturability. This creates waste in modern industry; both extending time to market for new products, and/or slowing the creation of finished products from raw materials if the design changes are not completed correctly, if at all.

As an important issue in product design, manufacturability significantly influences production cost and lead time, which are characteristics of manufacturing industry competitiveness. Research has shown that around 70% of the cost of a product has been determined at the design stage; improper design features may become irreversible limitations when it comes to the manufacturing stage. Therefore the importance of manufacturability analysis has been widely recognized by both the academic community and industry. Extensive effort has been dedicated to manufacturability analysis, where analytic methods were applied to investigate the manufacturability of a product design and the results are then fed back to the designer (Fig. 1.1). Manufacturability analysis has become a key component in implementing concurrent engineering as it simultaneously takes into account manufacturing issues in the design stage. Manufacturability analysis has a broad scope of research topics, such as accessibility, fixturability, tooling, quality, cost, assembly etc. Depending on the specific application, the objective in manufacturability analysis could be one or a combination of these measures.

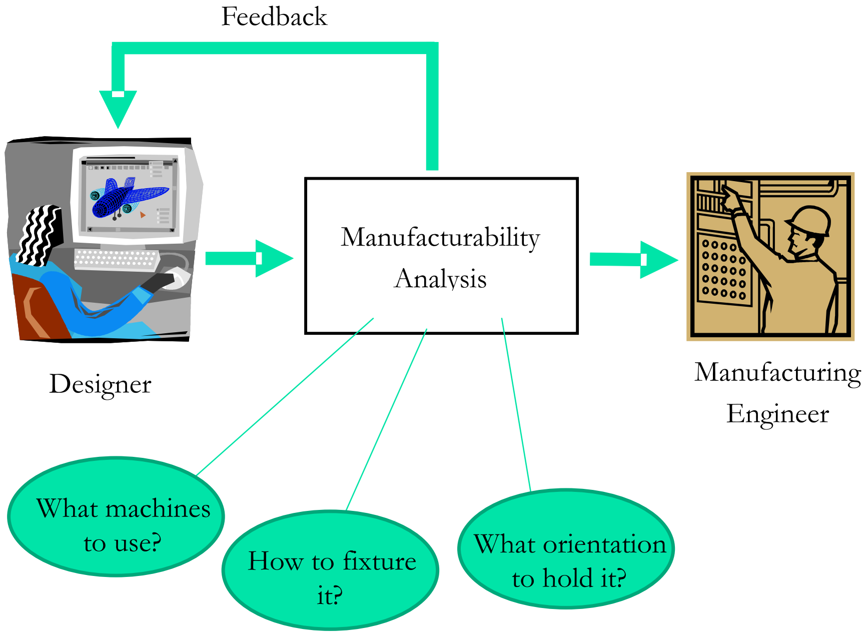


Fig.1.1 Manufacturability analysis

Manufacturability analysis has experienced three stages of development, coinciding with the advancement of computational techniques and the growth of industry need. These three stages are: (1) general design for manufacturability guidelines; (2) feature-based manufacturability analysis, and most recently (3) Non feature-based analysis.

Design for manufacturability guidelines came about after it was realized that traditional "over-the-wall" design could create downstream issues. Within a manufacturing company, the design and manufacturing departments have sometimes been two stand-alone departments. To address this challenge, guidelines that enabled design for manufacturability had been proposed in the form of general rules of thumb that were gained from the experiences of manufacturing practitioners. These guidelines suggested easy and economical design features, and were used as references by designers so that manufacturability is considered early in product design. With these guidelines, a designer had a better understanding of manufacturing; therefore difficulty of manufacturing could be significantly reduced. The implementation of design for manufacturability guidelines relies extensively on the experience of the design engineers. Unfortunately, these guidelines rely heavily on the skill of the designer using them; hence, improvements are highly variable.

The advent of computer technology offered the opportunity of automating the process of analyzing the manufacturability of a product design. This is primarily realized through a process called *feature recognition*. A feature is an aggregate of geometric entities that together convey a geometric meaning in both product design and manufacturing. Through feature recognition, the relation between design features and manufacturing processes are identified. The example shown in Fig. 1.2 illustrates the process of manufacturability of a prismatic model, where there is one pocket feature and four hole features. Through recognition, these features are identified and mapped to one pocket milling operation and four drilling operations. Since this manufacturability analysis relies on feature recognition, it is obviously imperative that the design model is described using features. This imposes a constraint on the capability of feature-based manufacturability analysis whereby non feature-based models cannot be processed. On a non feature-based model, almost no traditional recognizable feature exists; therefore the mapping mechanism from design

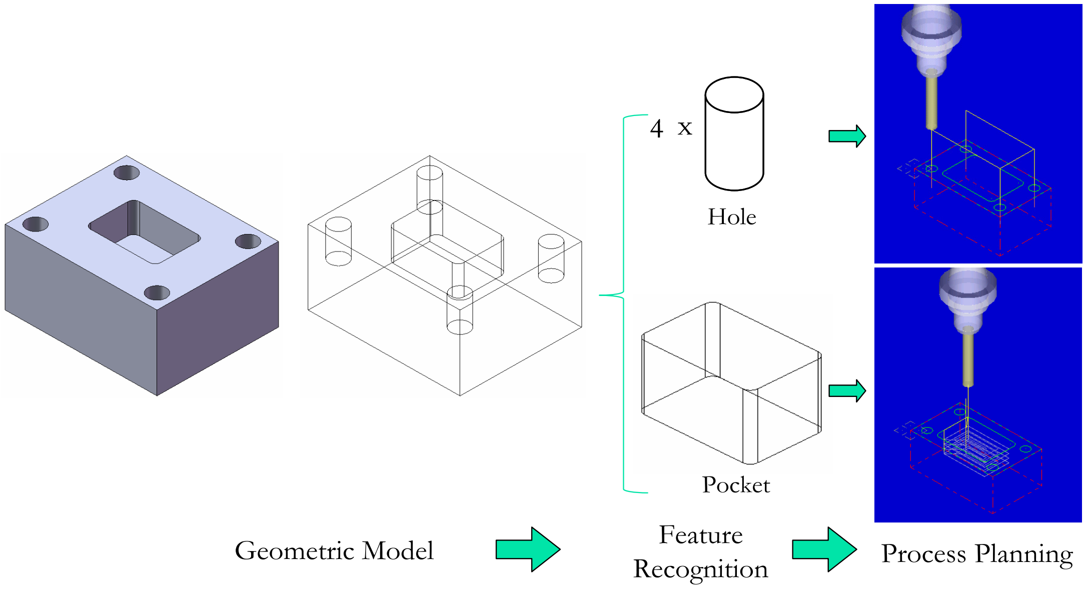


Fig.1.2 Manufacturability analysis for feature-based models

features to manufacturing features does not work. To process non feature-based models, a new manufacturability analysis approach is needed that can do away the feature-recognition process.

Non feature-based manufacturability analysis stems from the need to handle non feature-based geometric models, which are becoming increasingly prevalent. The most prominent application is in biomedical engineering where the surfaces studied are from bones, organs and other tissue (Fig. 1.3). These CAD models are typically re-constructed from medical image data such as computed tomography (CT) and magnetic resonance imaging (MRI). As bio-manufacturing is currently becoming a new frontier in manufacturing research sought by both industry and the academic community, methodologies for manufacturability analysis should be able to accommodate these new challenges.

A second application is in the bridge between art and manufacturing where an artist manually creates a sculptured model (Fig. 1.4) and then uses reverse engineering techniques to scan the surface to construct a CAD model.

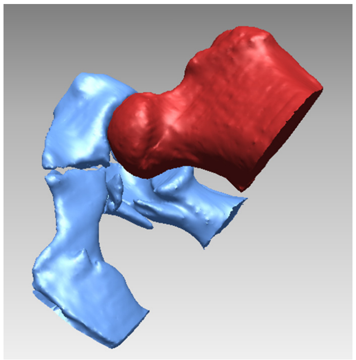


Fig.1.3 Bio-based models (hip joint)

Fig.1.4 Artistic models (sculpture)

One viable way to make non feature-based objects is to use *Rapid Prototyping* processes. *Rapid Prototyping,* also known as *Layered Manufacturing*, generally refers to techniques that can create a 3-D object through the process of successively building and stacking 2-D layers of material. By doing so, the complexity of constructing 3-D objects is made simpler; thus human effort of process planning, tooling preparation and fabrication are almost eliminated. The layered nature of *Rapid Prototyping* also simplifies complexity of manufacturability analysis because 3-D manufacturability is reduced into 2-D manufacturability on each layer. However rapid prototyping processes are limited in both materials and accuracy; therefore, they mostly limited to prototype models rather than functional components.

Computer Numerical Controlled (CNC) machining has received some attention as a possible method for realizing the *Rapid Prototyping* of functional parts. They offer high precision, low cost, and widely available functional materials [Hassold 1995, Schmidt 1997, Wang *et al.* 1999, Frank *et al.* 2002, Frank *et al.* 2003]. Currently many analysis tools exist for evaluating the *machinability* of a feature-based design. Representative research related to feature-based machining are reviewd in the work of Gupta *et al*. [Gupta *et al*. 1997]. However, little or no research specifically addresses manufacturability analysis without the direct use of features or feature recognition. Current Computer Aided Manufacturing (CAM) software is readily capable of generating toolpaths given a set of surfaces of a part and a cutting orientation (3-axis machining); however, it is not designed to calculate geometric machinability. If non-machinable regions could be identified, the actual machined surfaces can be computed and presented as feedback to a designer.

Research Motivation

Traditional Design for Manufacture (DFM) methodologies rely heavily on feature recognition, which lends them incapable of processing increasingly prevalent non-feature based objects. The motivation behind this research is to extend design for manufacturability beyond the limitations of feature recognition, and develop a methodology to analyze the manufacturability of non-feature based objects, in particular, for material removal processes such as CNC machining. With this non feature-based manufacturability analysis, a designer can receive feedback that consists of three types of manufacturability: potential manufacturability, manufacturability specific to a machine setup, and manufacturability specific to a cutting tool. The *potential* manufacturability does not rely on a machine setup, but is purely the manufacturability of the design from the standpoint of *accessibility*. This result will inform a designer whether or not the geometric model can be potential produced by a material removal processes, regardless of the actual tooling used. For the second type, the manufacturability specific to a machine setup will present the actual manufacturability on a chosen machine fixture. Each machine has an inherent constraint on accessibility due to its kinematic structure. A 5-axis machine has more accessibility then a 4-axis machine, and a 4 axis machine better than a 3-axis machine. Mathematically, manufacturability specific to a machine setup is a subset of the potential manufacturability. Moreover, the third type of manufacturability pertains to the shape and size of a specific cutting tool. This manufacturability is also called geometric *machinability*.

These three types of manufacturability hierarchically analyze the actual manufacturability of a non feature-based model, and inform the designer with three different levels of knowledge. The designer can then perform re-design with respect to those feedbacks on each level, accordingly.

Research Objectives

The primary objective of this dissertation is to develop a methodology to, in a hierarchical manner, analyze the manufacturability of non-feature based objects, based on material removal processes such as CNC machining. To achieve this major objective, the following sub-objectives are presented:

1) The first sub-objective is to develop a method of computing visibility for polyhedral CAD models. The visibility computed is global visibility, and will be used to represent *potential manufacturability*. Moreover the input will accommodate arbitrary convex polygonal facetted models.

2) The second sub-objective is to develop a method of computing feasible workpiece set-up orientations for machining non feature-based models using a 4-axis indexed rapid machining set-up. This corresponds to the *manufacturability specific to a machine setup*.

3) The third sub-objective is to develop a geometric machinability method that can analyze a non feature-based model and precisely predict the coordinate locations of un-machinable regions, taking into account the shape and size of a flat-end milling tool. This corresponds to *manufacturability specific to a cutting tool*.

Dissertation Organization

The remainder of this dissertation is organized as follows: Chapter two presents a discussion of the relevant literature. Chapters three to five are comprised of three Journal publications or submissions, where each paper presents one of the 3 sub-objectives of this work. Finally, chapter six presents future research opportunities and conclusions from this research.

CHAPTER 2. LITERATURE REVIEW

Feature-Based Manufacturing and Non-Feature-Based Manufacturing

Feature-based technology is widely used in manufacturing process planning. Boosted by the computational power from the advancement of computer technology, feature-based technology is automated and has established its important role in integrated product development over the past two decades. A *feature* consists of a collection of geometric entities that together pass on meaningful content from upstream to downstream in a production realization environment. The use of a feature is intended to link product designs created in a CAD system to manufacturing process planning activities. For example a hole created in a product model may be mapped as a drilling process, or a pocket mapped to a milling process. This enables a high level of communication between a designer and a process planner and also among designers in a collaborative design environment. As such, information transferred is therefore not based on basic surface patches but on those meaningful unit *features*. In addition to the information provided by a feature, the representation of a feature uses parametric formation, which allows users to create or edit a feature by changing its parameters. Most commercial CAD/CAM software, such as Solidworks, Pro-E, MasterCam, etc., contains functional modules that can assist users to create or edit feature-based models.

Given a feature-based model, manufacturability analysis can be carried out provided that those features are identified. The process of extracting features from a feature-based model is called *feature recognition*. Feature recognition is the first step in performing any feature-based analysis. A number of reviews on feature recognition can be found in [Allada and Anand 1995, 1996; Miao et al, 2002; Salomons et al, 1993; Zulkifli and Meeran 1999]. With features successfully recognized, design information becomes interpretable in the manufacturing stage, which facilitates concurrent engineering and thus speeds up the product development cycle.

The recognized features on a design model and their interrelation make the design description ready for tasks such as machining setup planning. Ferreira and Liu developed a rule-based system to generate setup orientations for workpieces described with features [Ferreira and Liu 1988]. Demey et al, determined the minimal number of setups, considering both the physical conditions and the economical and quality issues [Demey et al, 1996]. Chu and Gadh classified features into single approach direction features and multi approach direction features, and then determined the minimized number of setups along with knowledge-based rules [Chu and Gadh 1996]. Built upon the interpretable information carried by feature-based models, researchers have enriched rule-based approaches with user-intended objectives or techniques, and therefore provided the integration of human knowledge into feature-based manufacturing process planning. In determining setup orientations, feature-based workpieces also receive consideration of other physical constraints coming from machine configurations and fixturing devices [Cevdet 2004]. Wu and Chang developed an automated setup selection method based on tolerance analysis [Wu and Chang 1998]. The setups are ranked and then released for fixture selection. Yen et al, integrated setup planning with geometric positioning and tolerancing for fixture planning [Yen et al, 2001]. To meet the constraints imposed on setup planning from machine capabilities and design information, a number of techniques such as fuzzy-set [Ong and Nee 1996] and GA and SA [Ong et al, 2002] have been used in searching for optimal solutions.

The primary reason for using feature-based approaches owes somewhat to the fact that most designs, particularly those for mechanical parts, are geometrically composed of *features*. However the increasing need to handle freeform shapes such as those reconstructed from reverse engineering techniques (e.g. laser scanning, CT and MRI scanning) pose new challenges to manufacturing process planning. On those freeform shapes, there may be no definable "features", which lends feature-based approaches incapable. Since the primary concern of setup planning for material removal processes is to provide accessibility for the cutting tool, researchers identified accessibility as the necessary condition of manufacturability and have used it to extract setup planning through geometric operations. For feature-based models, feature accessibility can be checked by calculating the feature accessibility volume and testing the intersection of the feature accessibility volume with the part [Regli 1995, Regli et al 1995, and Gupta and Nau 1995]. For non-feature-based geometries, visibility is generally used as the approximation of accessibility because the method of computing feature accessibility volume is not applicable to non-feature-based models. Therefore it is important to determine the visibility for non-feature-based models. Visibility can be categorized into two classes: local visibility and global visibility. Gan et al, constructed a visibility map from a Gaussian map [Gan et al, 1994]. The visibility map constructed from a Gaussian map was used to compute setup orientations for 4- and 5- axis machining [Tang et al, 1992; Chen et al, 1993; Haghpassand and Oliver 1995]. For the example of 4-axis milling, a feasible setup orientation should be one that allows a great circle orthogonal to it to intersect all spherical visibility polygons. However the visibility constructed from Gaussian maps is local visibility that cannot guarantee global accessibility. In manufacturing process planning, particularly for machining processes, global visibility is required to avoid collision of the cutting tool with the workpiece. Suh and Kang used a discretized model to construct global visibility and a similar method to find a feasible setup orientation for 4-axis milling [Suh and Kang 1995]. Dhaliwal et al, computed accurate non-visibility for an object represented by triangular facets by projection and convex hull operations [Dhaliwal et al, 2003]. Balasubramaniam et al, used graphic techniques to obtain visibility information [Balasubramaniam et al, 2000].

A second limitation of feature-based approaches lies in the constraint imposed by the concept a *feature* itself. In feature-based approaches, a manufacturable feature requires that all surfaces constituting the feature be completely machined from one cutting orientation; otherwise that feature will be considered non-manufacturable if such a direction does not exist. Though feature-based approaches are effective in determining manufacturability, the constraint they impose actually rules out the possibility of completely machining a feature from a combination of two or more directions, which would provide flexibility and increased solution space in planning machining process. Non-feature based process planning intends to more extensively explore the solution space by breaking each feature into finer constituting *elements* and performing the analysis on those elements. Mukerjee and Jain [Mukerjee and Jain 1997] proposed a featureless Computer Aided Process Planning (CAPP) model where process planning is done before feature recognition. The authors intended to generate more choice of process sequences by implementing the featureless CAPP model. Frank et. al. [Frank et. al. 2006] analyzed 2D global visibility on Stereolithography (STL) slices where the basic elements are those line segments comprising of the sliced chain. The necessary machining orientations are searched for 4th axis indexable machining by executing a Greedy search algorithm.

In addition to exploring in an increased solution space, non-feature-based approaches are fundamentally capable of processing non-feature-based geometric models since all analysis are performed on the basic elements constituting the models, not requiring feature recognition. Since there are few definable features existing on non-feature-based models, feature representation is not able to describe non-feature-based models. One feasible way of describing non-feature-based geometries is to use polygonal models. One common form of polygonal model is an STL model, which is created through tessellating a geometric model into a triangular-facetted model. The non-feature-based analysis can then be performed on the triangular facets that constitute the surfaces of a non-feature-based model. Dhaliwal et al, computed non-visibility for the triangular facets of an STL model [Dhaliwal et al, 2003]. Li and Frank [Li and Frank 2007] extended the non-visibility computation for geometric models comprised of arbitrary polygonal facets. Frank et. al. [Frank et. al. 2006] computed the visibility of a sliced STL model and searched the necessary machining orientations for a 4-axis indexed rapid machining setup. Li and Frank [Li and Frank 2008] used the visibility computed from each facet to search the feasible axes of rotation for the 4-axis indexed rapid machining setup.

There are certainly advantages if a feature based model can be used, or if features can be recognized and utilized in process planning. However, feature based design and manufacturing planning can be not only difficult, but can also pose unintended restrictions on process planning solutions. Non feature based process planning cannot take advantage of the parametric forms of, and increased information available when features are used; however, research has shown that they can provide improved flexibility and variety in process planning solutions.

Rapid Prototyping and Manufacturing

The advent and growth of *rapid prototyping* come from a substantial demand of human-effort-less rapid creation of 3-D parts directly from CAD design to speedup product development process. Ever since 1980's, numerous processes have emerged as feasible approaches for executing the task of *rapid prototyping*. One major reason behind the innovation of *rapid prototyping* processes is due to the lack of a true integration of CAD/CAM related to CNC machining [Yan and Gu 1996]. Inspired by the simplified manner of building 3-D objects through stacking 2.5 D layers, people are desiring a similar way of producing functional parts, a process called *rapid manufacturing*.

Although *rapid prototyping* processes free designers from manufacturing constraints by doing away with the need of preparing tooling, the staircase inherent with these layer-based processes generates undesired surface finish and is strongly dependent on the building direction. The error from the staircase can be represented by either cusp height, the maximum distance of manufactured part surface perpendicular to the CAD model surface [Alexander et al. 1998], or volumetric error, the difference of the volume of the material used to create the part with the volume determined by the CAD model [Masood and Rattanawong 2002].

Rapid prototyping processes are broadly divided into two categories from a process standpoint: material addition and material removal [Pham and Gault 1998]. Representative material additive processes include Stereolithography (SLA), Fused Deposition Modeling (FDM), Laminated Object Manufacturing (LOM), Selective Laser Sintering (SLS), Ballistic Particle Manufacturing (BPM), Three Dimensional Printing (3DP), and laser-engineered net shaping (LENS). *Rapid prototyping* by material removal refers to the application of CNC machining in a rapid manner. Shape deposition manufacturing (SDM) is a hybrid process combining both material addition and removal [Merz et al 1994]. According to the material used, these processes can also be classified into metallic fabrication process and non-metallic fabrication process. It has been found that materials used for *rapid prototyping* are limited and are not suited for making functional parts for intended physical environments in most cases. In addition to that, material properties of some RP parts are not well understood yet. SLA and FDM produce parts from plastic materials. Hague et al. [Hague et al. 2004] found that SLA process produces isotropic parts and material properties do not demonstrate a clear relation with the variation of building direction. Montero et. al.[Montero et. al. 2001] discovered the anisotropic properties of parts made by FDM. Hague et al. [Hague et al. 2004] also reported that the material property of SLS parts is found to be anisotropic through experiment study. Such an experiment result is complying with the works of Gibson and Shi [Gibson and Shi 1997] and Hur et al. [Hur et al. 2001]. The materials used for BPM are limited and should be easily melt, like wax, thermoplastics and aluminum [Yan and Gu 1996]. Although some *rapid prototyping* processes can use metallic materials, the parts produced are inferior compared with conventional processes in surface roughness and dimensional accuracy, and therefore need post processing. LOM can produce parts for the application of *Rapid Tooling* from metal rolls in addition to the materials like paper, plastic and composite. However a post machining process is needed to smooth the staircases. SLS parts demonstrate undesired surface roughness, and the shrinkage and distortion are the factors keeping SLS from making accuracy parts. LENS also suffers from its limited spatial resolution or accuracy, rendering it a near net shape process [Levy et. al. 2003]. 3DP is now available to make metal parts, however the accuracy is impaired by its second operation [Radstok 1999].

Rapid Tooling (RT) is a technical extension of *rapid prototyping* intended to innovate tool-making in order to shorten time-to-market from a production stand point. *Rapid Tooling* takes the tool-less advantage of *rapid prototyping* and aims to apply the prototypes produced by *rapid prototyping* in making tool to reduce lead-time of production. *Rapid Tooling* produces tool, the critical means of production, and therefore is actually a production-oriented pre-manufacturing stage. *Rapid Tooling* can be devided into two classes: non-direct tooling, by which prototypes are not used as production tools but need other processes to produce tools, and direct tooling whereby *rapid prototyping* provides prototypes as tools directly [Karapatis et. al. 1998]. Although *Rapid Tooling* provides profits for production, it is not economically justifiable for *rapid manufacturing* where production quantity is very low. Since rapid tools are produced from materials and processes significantly different from those traditional processes, *Rapid Tooling* are faced with such challenges as accuracy, durability, and quantity and quality of the parts [Segal and Campbell, 2001].

Recently, CNC machining have been paid increasing attention to as widely available processes to rapidly manufacture prototype or even functional parts, for the high precision, low cost, and widely available materials [Hassold 1995, Schmidt 1997, Wang *et al.* 1999, Frank *et al.* 2002, Frank *et al.* 2003]. Relvas and Simoes‘s study [Relvas and Simoes 2004] shows the competency of CNC machining in dimensional deviations, time and expenses, compared with material additive processes including SLS, SLA and LOM. However the biggest challenge for using CNC machining for *rapid manufacturing* lies in fixturing [Wang *et al.* 1999, Frank *et al.* 2002]**.** Sarma and Wright [Sarma and Wright 1997] presented reference free part encapsulation (RFPE) as a universal fixturing approach. Choi et al. [Choi et al. 2001] implemented RFPE, and developed a feature-based CAD/CAM system for an encapsulation system. As a new Phase-change fixturing method, RFPE solved the problem of losing location datum when more than one setup is necessary in machining the part. Bandyopadhyay *et al*. [Bandyopadhyay *et al*. 1993] developed a fixture-free machining center for machining block-like component, whereby a specially designed workpiece-holding device allows bar-form material to be fed automatically and positioned for unattended machining of all the six faces of the block-like components. Frank et. al. [Frank et. al. 2004] fixtured workpiece automatically for a 4 axis indexer rapid machining set-up through the use of *sacrificial supports*. Such *sacrificial supports* added to the ends of CAD model are created in-process during machining and are finally removed after all operations are completed.

Design for Manufacturing and Re-Design

Design for Manufacturing (DFM) is a philosophy that ushers a designer to a good design from a manufacturing perspective. The concept of DFM breaks the isolation between design and manufacturing by simultaneously considering manufacturing constraints in the design stage. DFM is conceptually an alternative solution, conducted earlier in design stage, to the problems that may occur later in manufacturing stages. Implementation of DFM has been shown to lead to a great amount of economic benefits, and has received considerable attention since 1970’s [Boothroyd, 1993]. DFM was also contended as a competitive alternative choice to avoid heavily investing in automated equipments [Fabricius 1994].

The research scope of DFM is not quite clear, like the word *manufacturing*. Narrowly speaking, it concerns manufacturing processes for creating a design; broadly it covers nearly all aspects of product development cycle [Polo 2001, Stoll 1986]. Taylor [Taylor1997] discussed the strategies for DFM in multi-facility and global manufacturing context. The principles of DFM are very process specific. Usually principles of one process cannot be transplanted to another one. Polo 2001 [Polo 2001] and Bralla [Bralla 1986] gave DFM guidelines of nearly all traditional manufacturing processes. CNC machining as a major manufacturing process was also provided with DFM guidelines by Hodgson and Pitts [Hodgson and Pitts 1991]. The guidelines centers on minimizing the number of set-up, tool used and transfers between machines.

One approach for implement DFM in design activities is to visually inspect a design drawing against a checklist as the one provided by Bralla [Bralla 1986]. Such a manual inspection is very experience-dependent and may have problem when a design is getting complex. Stauffer et al**.** [Stauffer et al. 2003] presented a useful template for organizing and presenting DFM guidelines. Fabricius [Fabricius 1994] reported a seven-step DFM procedure that could reduce 25-30 percent manufacturing cost. Barton et al. [Barton et al. 1996] developed database architecture and a statistical modeling methodology that can incorporate manufacturing experience and update DFM design rules. The advance of computer technology allows the description, reasoning and manipulation of a design on computer. Assisted by computer technology, DFM is no longer a manual and experience-dependent practice, but can be made automated and integrated into CAD/CAM systems. A feature as a set of correlated geometric elements is of considerable significance to the integration of design and manufacturing. The concept of feature also benefits the implementation of DFM. Mill et al. [Mill et al. 1994] developed a simultaneous workstation to discover problems that may rise from machining operations in a feature-based design. Dumitrescu and Szecsi [Dumitrescu and Szecsi 2002] described a DFM implementation system that can analyze design features, map those features to manufacturing features, and check manufacturability. This DFM system takes into account materials, production types, surface finishes, tolerances and manufacturing processes.

The computational resources provided by computers allow the application of many techniques to promote the practice of DFM. S.Wesley and Li [S.Wesley and Li 1996] discussed a design critique system for rotational parts using knowledge-based language. Fauvel [Fauvel 1994] described an information model that allows concurrent manufacturability analysis. Ong et al. [Ong et al. 2003] used fuzzy set to evaluate the relative ease or difficulty of machining features in design; an analytic hierarchy process (AHP) method is employed to assign weighting factors. Liu et al [Liu et al.1995] proposed a fuzzy logic expert system, SMARTDFM, to address machining aspects in the early stage of product design. Huang and Mak [Huang and Mak 1999] demonstrated a web-based DFMA technique.

Since manufacturing actually addresses many issues, each of which targets different objectives, a DFM methodology may have its objective shifted to one specific aspect of manufacturing. Design for economic manufacture [Corbett 1986] pay attention to the cost issue of manufacturing and looks for the most economic way to produce a design. Design for life cycle manufacture [Stoll 1997] studies DFM philosophy at all stages of the engineering design process. Lenau [Lenau 1996] studied early selection of processes and materials in DFM and describe a computer-based tool called Designers Manufacturing Inspirator for a designer to examine materials/processes. Ji and Lau [Ji and Lau 1999] discussed dimensional aspect in DFM/Concurrent Engineering environment. Taylor [Taylor 1997] introduced design for global manufacturing and assembly (DFGMA) as a tool for assisting a designer to make decisions holistically.

Redesign is an important constitutional component of DFM. Redesign is a process aiming at improving or correcting an existing design that violates DFM rules. The output of redesigning process is either a better or remedied design complying with DFM rules. Artificial Intelligence (AI) has been applied in generating redesign suggestions. Desa et al. [Desa et al. 1987] described methods for system redesign of an engineering product and created a simple expert system to implement it. Adalier and Tsatsoulis [Adalier and Tsatsoulis 1992] developed an intelligent system called REINRED based on the concept of case based reasoning to realize redesign for manufacturability. So far most redesign approaches depend on feature-based analysis. Hayes and Gaines [Hayes and Gaines 1996] generated design alternatives and looked for one of them with more manufacturability. The design alternatives are generated through feature recognition process. Das et al. [Das et al. 1996] generated redesign suggestions to reduce set-up time by offering alternative manufacturing features while satisfying a designer's intent. Hayes [Hayes 1996] described a plan-based design advisor that takes a feature-based design, generates manufacturing plan and provides suggestions to redesign shapes to reduce manufacturing cost. Zhou and Gaines [Zhou and Gaines 2003] developed an automated redesign for machined parts (Arm) that can identify non-machinable shapes of a feature-based design and can transform them into machinable features, whereby manufacturability of a design is increased.

Manufacturability can also be improved from other design aspects. Lin et al. [Lin et al.2003] substituted the dimension tolerance of a design with an alternative one to obtain improved manufacturability of machining process, with design functionality not compromised.

Machinability and Visibility

Machinability analysis is taking an increasingly important role as complex surfaces are used in the design of a wide variety of parts. Current Computer Aided Manufacturing (CAM) software is readily capable of generating toolpaths given a set of surfaces of a part and a cutting orientation (3-axis machining). However, determining the setup orientation can be difficult and moreover, it may be very challenging to determine if the part can be created using machining at all. An appropriate setup orientation can guarantee an effective cutting of the surface, while an inappropriate one will leave too much material in certain regions.

Many researchers have studied machinability analysis and its closely related workpiece setup problem. Most of the approaches are based on visibility, which is essentially line of light accessibility. Su and Mukerjee [Su and Mukerjee 1991] presented a method to determine machinability of polyhedral objects. A convex enclosing object is constructed to make each face of the part orthogonally visible to the planes of the enclosing object. The part is then considered to be machinable from the normal-vector directions of the enclosing object planes. Later, computational geometry on the sphere was utilized to analyze visibility by Chen and Woo [Chen and Woo 1992] who performed pioneering work on computational geometry algorithms that could be used for determining workpiece set-up and machine selection. Tang et al. [Tang et al. 1992] formulated the problem of workpiece orientation as finding the maximum intersection of spherical polygons. Gan et al. [Gan et al. 1994] discussed the properties and construction of spherical maps and presented an efficient way to compute a visibility map from a Gaussian map. Chen et al. [Chen et al. 1993] partitioned the sphere by spherically convex polygons to solve the geometric problem of determining an optimal workpiece orientation for 3-, 4- and 5-axis ball end milling. A visibility map is generated by using the normal vectors of a specified portion of the surface of a part, therefore it cannot guarantee global accessibility. Yang et al. [Yang et al. 1999] computed visibility cones based on convex hull analysis, instead of relying on visibility maps. Yin et al. [Yin et al. 2000] defined complete visibility and partial visibility, and presented a C-space based method for computing visibility cones. A sculptured surface is approximated by its convex hull [Yang et al. 1999] and the spherical algorithms [Chen and Woo 1992, Gan 1990] are used in the approach of Yin [Yin et al. 2000]. The convex hull may in some cases have a significant deviation from the true surface and therefore cannot guarantee a precise computation. Suh and Kang [Suh and Kang 1995] constructed a binary spherical map to compute the point visibility cone in order to algebraically solve machining configuration problems, including workpiece setup orientation. The part surface is decomposed into triangular patches. An occupancy test of the patches is conducted on a triangular-represented unit sphere to generate global visibility. Dhaliwal et. al. [Dhaliwal et. al.2003] presented a similar approach for computing global accessibility cones for polyhedral objects, but with exact mathematical conditions and algorithms. Balasubramaniam et al. [Balasubramaniam et al. 2000] analyzed visibility by using computer hardware (graphics cards). Frank et al. [Frank et al. 2006] analyzed 2D global visibility on STL slices and searched the necessary machining orientations for 4th axis indexable machining by executing a Greedy search algorithm. All these visibility-based approaches determine the necessary condition for machinability; however, they ignore tool geometry and therefore true accessibility (machinability) is not guaranteed.

Su and Mukerjee [Su and Mukerjee 1991] took into account the cutter information by constructing a new part model through offsetting the original part surface by the amount of the cutter radius. Machinability was further guaranteed by checking the topology of this offset part surface. This method is effective for the machinability analysis of a ball end cutter, but not for that of a flat end cutter, because the effective radius of a flat end cutter is variable with the change of tool tilting angle. Haghpassand and Oliver [Haghpassand and Oliver 1995] and Radzevich and Goodman [Radzevich and Goodman 2002] considered both part surface and tool geometry. However, tool size was not taken into account due to the fact that Gaussian mapping does not convey any size information of the part surface and/or the tool. Balasubramaniam et al. [Balasubramaniam et al. 2000, Balasubramaniam et al. 2003] verified tool posture from visibility results by collision detection before interpolating the toolpath for 5-axis machining.

Over the past years, feature-based technologies have been an active field among the manufacturing research community. Regli[Regli 1995], Regli et. al. [Regli et al 1995], and Gupta and Nau [Gupta and Nau 1995] discussed feature accessibility and checked it by calculating the feature accessibility volume and testing the intersection of the feature accessibility volume with the part. Gupta and Nau [Gupta and Nau 1995] recognized all machining operations that could machine the part, generated operation plans, and checked and rated different plans according to design needs. A comprehensive survey paper on manufacturability by Gupta et al. [Gupta et al. 1997] reviewed representative feature-based manufacturability evaluation systems. Recently, Shen and Shah [Shen and Shah 1998] checked feature accessibility by classifying the feature faces and analyzing the degree of freedom between the removal volume and the workpiece. The MEDIATOR system reported by Gaines et al. [Gaines et al. 1999] used the knowledge of manufacturing equipment to identify manufacturing features on a part model. Accessibility is examined by testing the intersection of removal volumes with the part. Faraj [Faraj 2003] discussed the accessibility of both 2.5 D positive and negative features. Other researchers presented featured-based approaches to determine workpiece setups [Ferreira and Liu 1988, Demey et. al. 1996, Wu and Chang 1998, Ong et. al. 2002]. Although feature-based approaches are capable tools to handle feature-based design, they cannot lend themselves to free-form surfaces where definable features may not exist. In addition, feature-based approaches suggest that all the geometric elements comprising of a feature are treated together as an entity. This actually imposes a constraint to the analysis of a part model. For example, it might be feasible to machine a portion of a part feature in one orientation and then finish the remaining surfaces of the feature in one or more successive orientations.

CHAPTER 3. COMPUTING NON-VISIBILITY OF CONVEX POLYGONAL FACETS ON THE SURFACE OF A POLYHEDRAL CAD MODEL

A paper published in *Computer Aided Design,* v 39, n 9, 2007

Ye Li and Matthew C. Frank

Abstract

Visibility has found wide applications in manufacturing operations planning and computer vision and graphics. The motivation of this paper is to accurately calculate visibility for objects whose surface is represented by polygonal facets. In this paper, the authors focus on determining *non-visibility cones*, which are the complementary sets of *visibility*. This is accomplished by determining sliding planes that comprise the boundaries of a non-visibility cone. The approach presented in this paper directly evaluates the boundaries of the non-visibility cone of an arbitrary convex planar polygon due to the visibility blocked by obstacle polygons. The method is capable of calculating visibility for convex polygons with any number of sides, not limited to triangular facetted models. Implementation is demonstrated in this paper for three to six sided polygonal models.

1. Introduction

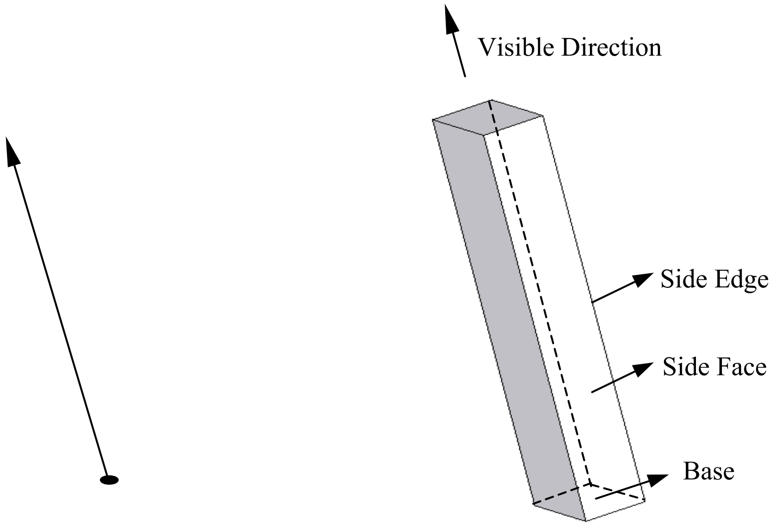
The concept of visibility is based on line of sight accessibility. Visibility describes the *reachability* of light rays to the surface of an object, and has become an important characteristic for analyzing a surface. Visibility analysis of an object has found applications in a wide variety of manufacturing activities and computer graphics. In manufacturing, the concept of visibility is used to analyze the necessary condition of physical accessibility, such as that of a cutting tool in CNC machining [Chen and Woo 1992, Suh and Kang 1995, Balasubramaniam et. al. 2000, Frank et. al. 2006] or a contact probe in CMM measurement [Spyridi and Requicha 1990, Lim and Menq 1994, Kweon and Medeiros 1998]. Visibility is also applied for assembly fixture calibration [Chen et. al. 2001 and Kong et. al. 2005] and mold parting determination [Chen et. al. 1993, Yin et. al. 2000, Fu et. al. 2002, Elber et. al. 2004, Khardekar et. al. 2006]. In computer graphics, visibility can be used for surface shading, occlusion culling and view-dependent mesh simplification [Stewart 1999].

2. Related Work

Given the important role visibility plays in those aforementioned applications, many researchers have proposed approaches to computing visibility. Spyridi and Requicha [Spyridi and Requicha 1990] studied the probe accessibility of a CMM by abstracting the probe as a half line, which is actually equivalent to visibility. Accessibility cones are divided into local accessibility cones (LAC) and global accessibility cones (GAC). The LAC of a feature can be obtained from the intersection of all half spaces corresponding to the normals of the Gaussian image of the feature. The GAC of a feature is found by computing Minkowski sums for each point on that feature. Chen and Woo [Chen and Woo 1992] studied visibility on a unit sphere and introduced the idea of a Visibility Map. Similar to a LAC, a visibility map is constructed from a Gaussian map, and therefore only provides local visibility, not global visibility. Both of these approaches imply the utilization of a “feature” in the analysis. However, for a free-form object where no definable features can be recognized, the surface visibility will turn out to be the null set. This (feature-less models) can occur, for example, when CAD models are created from reverse engineering methods such as laser scanning or from Computer Tomography (CT) or Magnetic Resonance Imaging (MRI).

In order to compute global visibility for complex objects, Yang et. al. [Yang et. al. 1999] used the convex hull of surface patches to compute non-visibility. However, a convex hull based approach is prone to exaggerate the non-visibility, because a convex hull encloses the true surface and the non-visibility of the convex hull may include directions that are visible to the true surface. Suh and Kang [Suh and Kang 1995] decomposed a part surface into triangular patches and constructed a binary spherical map for visibility. The computed visibility is actually the visibility of only the centroid of each triangle patch; therefore it is an approximate solution. Dhaliwal et. al. [Dhaliwal et. al. 2003] also presented an approach of computing accurate non-visibility for an object represented by triangular facets. An obstacle triangular facet is projected onto the unit sphere centered on the three vertices of the triangular facet under study. The non-visibility cone is obtained by finding the spherical convex hull of these three projected images. This approach has been used for designing sacrificial multi-piece molds [Huang et. al. 2003] and permanent multi-piece molds [Priyadarshi and Gupta 2004]. Other researchers used graphics techniques to analyze visibility used for CMM measurement [Lim and Menq 1994, Spitz et. al. 1999], for CNC machining [Balasubramaniam et. al. 2000], and for testing mold parting direction [Khardekar et. al. 2006]. Graphics based approaches are limited to a certain resolution inherent to computer hardware and provide a sampling of a visibility cone; therefore they cannot offer exact solutions.

The motivation of this paper is to find visibility for polyhedral objects with arbitrary convex planar polygon geometry, as most solid objects can be represented by a polyhedron. Our goal is to determine the non-visibility of a polygonal facet due to another, and the approach is applicable to any polygonal representation, not just triangles as in previous research. An approach is presented in this paper to directly compute the boundary of the non-visibility cone for one polygonal facet due to another.



(a) Point Visibility (b) Facet Visibility

Fig. 1 Point visibility and facet visibility

3. Non-Visibility of a Polygonal Facet

3. 1 Polygonal facet visibility

The visibility considered in this research is based on a planar facet. A planar facet here specifically refers to a convex planar polygon. A concave planar polygon would be first decomposed into convex polygons. Methods for decomposing concave polygons into convex polygons can be found in [O'Rourke 1998]. Similar to the concept of *point visibility*, *polygonal facet visibility* means the reachability of light rays along a direction towards every point comprising the facet with no interference. Geometrically, *polygonal facet visibility* is represented as a 3-D volumetric light beam. The polygonal facet as the base is imagined to be fully occupied with light sources with parallel light rays that emit along a direction **(**Fig. 1**)**. Such a direction is said to be a visible direction if no object obstructs the 3-D light beam. In this paper we will use *base facet* to refer to the polygonal facet being analyzed for visibility, while all other polygonal facets are called *obstacle facets*. The maximum visibility of a polygonal facet in this paper is limited to a hemisphere, the pole of which is the normal vector of the facet (Fig. 2). That says that the visible directions of a facet are restricted to those from the external side while directions from the internal side are considered non-visible.

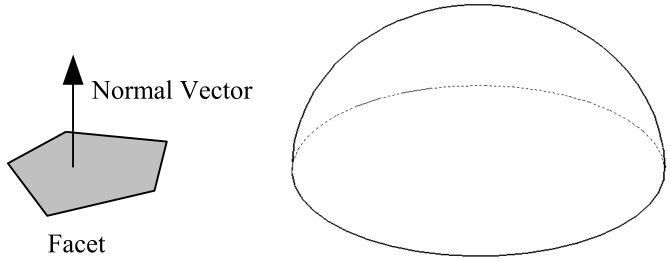


Fig. 2 Maximum visibility of a facet

3. 2 Tracing the boundary of an obstacle polygonal facet

The obstacle objects are also represented using convex polygonal facets in this paper. Any polygonal facets above the plane where the *base facet* under study lies are treated as potential *obstacle facets*. An *obstacle facet* will block part of the visibility of the *base facet* and therefore causes a region of non-visibility on the hemisphere. Such a region of non-visibility can be obtained by tracing the boundary of the *obstacle facet* with the 3-D light beam emitted from the base facet. The 3-D light beam stays tangent to the boundary of the *obstacle facet* during the tracing process. Since the boundary of the *obstacle facet* is a closedloop poly-line chain, the light beam will come back to the starting point once it finishes tracing the entire boundary of the *obstacle facet*, thereby tracing a closed region on the unit hemisphere. It should be noted that the side edges of the 3-D light beam change orientation simultaneously when the orientation of the 3-D light beam varies. Consequently all of these edges describe the same traced region on the hemisphere as the light beam traces over the boundary of an *obstacle facet*. Each of these edges can be used to analyze visibility on the unit hemisphere at any stage during the tracing process. This simultaneousness makes it possible to study visibility using different side edges that belong to the current contact pair of geometric elements from both the *obstacle facet* and 3-D light beam. This will be expanded on in Section 3.3 below. Figure 3 illustrates the traced region obtained from tracing the light beam along the boundary of an obstacle polygon (Fig.3*a*-Fig.3*h*). Figure 3*i* illustrates the traced region on a unit hemisphere.

3.3 Sliding planes

The boundary tracing process between the 3-D light beam and an *obstacle facet* actually consists of a set of sliding movements, as can be seen in Fig. 3. Each sliding movement occurs between a contact pair, composed of one geometric component from the boundary of the *obstacle facet* and one from the boundary of the 3-D light beam extruded from the *base facet*. These contact pairs can comprise of a side edge of the 3-D light beam and an edge of the *obstacle facet* (Fig. 4*a*), a side facet of the 3-D light beam and a vertex of the *obstacle facet* (Fig. 4*b*), or a side facet of the 3-D light beam and an edge of the *obstacle facet* (Fig. 4*c*). The orientation change of the 3-D light beam over one sliding step between a contact pair is illustrated in Fig.4. A hemisphere is constructed on the *base facet* with the origin to be the vertex point denoted by the symbol "■"in Fig.4. As the 3-D light beam slides between the contact pair, a side edge of the light beam, the end point of which is the origin of the hemisphere denoted by "■", is actually moving on a plane called the *sliding plane* in this paper. A *sliding plane* is an imaginary plane and is defined by the pair of static elements

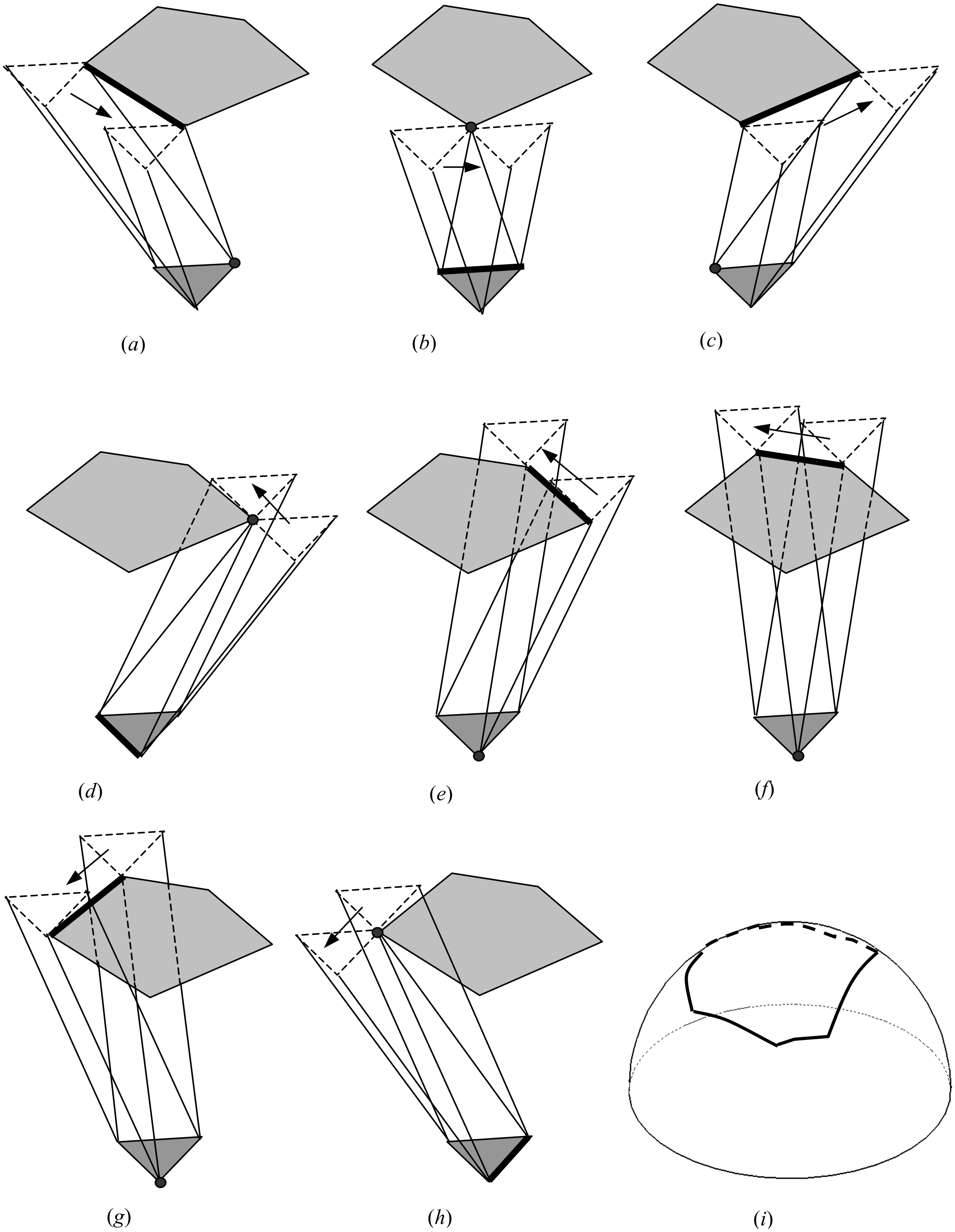


Fig. 3 Illustration of non-visibility by boundary tracing

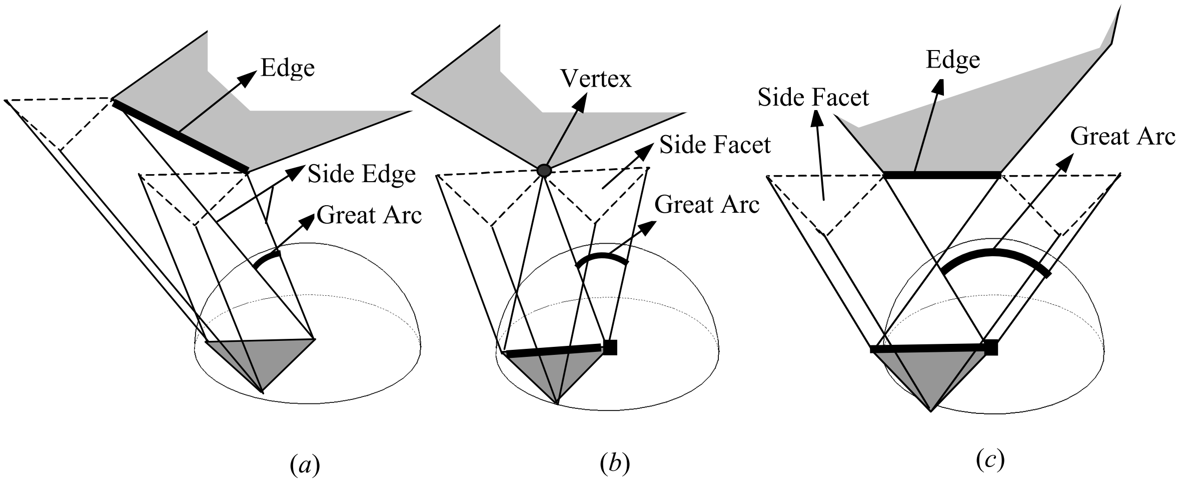


Fig. 4 A sliding step corresponding to a Great Arc

from a contact pair. The pair of static elements of a contact pair includes one component from the *obstacle facet* (an edge or a vertex) and one stationary element of the 3-D light beam component (an edge or a vertex of its *base facet*). A *sliding plane* is tangent to both the *base facet* and the *obstacle facet* on the static elements pair. In Fig. 4*a*, the *sliding plane* is defined by the edge in bold of the *obstacle facet* and the vertex point of the *base facet* denoted by symbol "■". In Fig. 4*b*, the *sliding plane* is determined by the vertex of the *obstacle facet* denoted by symbol "●" and an edge of the *base facet* in bold. In Fig. 4*c*, the *sliding plane* is determined by the edge in bold of the *obstacle facet* and the edge in bold of the *base facet*. It should be noted that the *sliding plane* described in Fig. 4*c* only exists when the two edges in bold are parallel. It can be seen that all the *sliding planes* shown in Fig.4 pass through the origin of the unit hemisphere, and therefore, by the definition of a great arc, the orientation change of the 3-D light beam resulting from the sliding movement corresponds to a great arc on the unit hemisphere (shown as a bold great arc (Fig. 4*a*-4*c*). Such *sliding planes* are also denoted in the example shown in Fig. 3. They are determined by the edge in bold and vertex denoted by "●" (Fig. 3*a*-3*h*). The traced region in Fig. 3*i* is therefore a spherical polygon, enclosed by a closed chain of spherical great arcs.

We use Polygon-O to denote the *obstacle facet* with edges (E_{Oi}, i=1…m) and vertices (V_{Oi}, i=1…m), and Polygon-B to express the *base facet* with edges (E_{Bi}, i=1…n) and vertices (V_{Bi}, i=1…n). The 3-D light beam constructed from Polygon-B is denoted by Beam-B. The foregoing discussion indicates that each great arc edge of the traced spherical polygon on the unit hemisphere corresponds to a sliding movement of the 3-D light beam on a *sliding plane* along the boundary of an *obstacle facet*. All the great arc edges constituting the entire boundary of the spherical polygon are obtained when the sliding movements are completed on all *sliding planes*. Therefore the traced region of the Polygon-B due to the obstruction of Polygon-O can be computed once all the *sliding planes* are determined and constructed. To determine the *sliding planes*, it is important to understand the geometric concept and property of a *sliding plane*. A *sliding plane* is defined by the stationary element (an edge or a vertex of the *base facet*) of the 3-D light beam and part of the boundary of the *obstacle facet*. The 3-D light beam slides along the boundary of an *obstacle facet* on a *sliding plane*; however, no intersection further than tangency will occur between the 3-D light beam and the *obstacle facet*. A *sliding plane* can be extended to infinity, bisecting the space into two subspaces. Both the *obstacle facet* and 3-D light beam extruded from the *base facet* are tangent to a *sliding plane* and therefore belong to different subspaces. This is summarized below as **Observation 1** and **Properties 1** and **2**.

Observation 1: A *sliding plane* (*SP*) is a plane on which the 3-D light beam extruded from a *base facet* slides one step along the boundary of an *obstacle facet*.

Property 1: A *sliding plane* (*SP*) divides a 3-D space into two half-spaces denoted by Space-O and Space-B. Both the *base facet* and the *obstacle facet* belong to one subspace exclusively, e.g. Polygon-O $\subset$ Space-O and Polygon-O $\not\subset$ Space-B, and Polygon-B $\subset$ Space-B and Polygon-B $\not\subset$ Space-O. A *sliding plane* (*SP*) is tangent to the boundaries of both facets, e.g. $SP \cap$ Polygon-O= E_{Oi} or V_{Oi} (i=1…m), and $SP \cap$ Polygon-B= E_{Bi} or V_{Bi} (i=1…n).

Property 2: A *sliding plane* (*SP*) corresponds to a half great circle on a unit hemisphere, bisecting the hemisphere into two regions; one containing non-visible directions, and the other containing visible directions for the *base facet*.

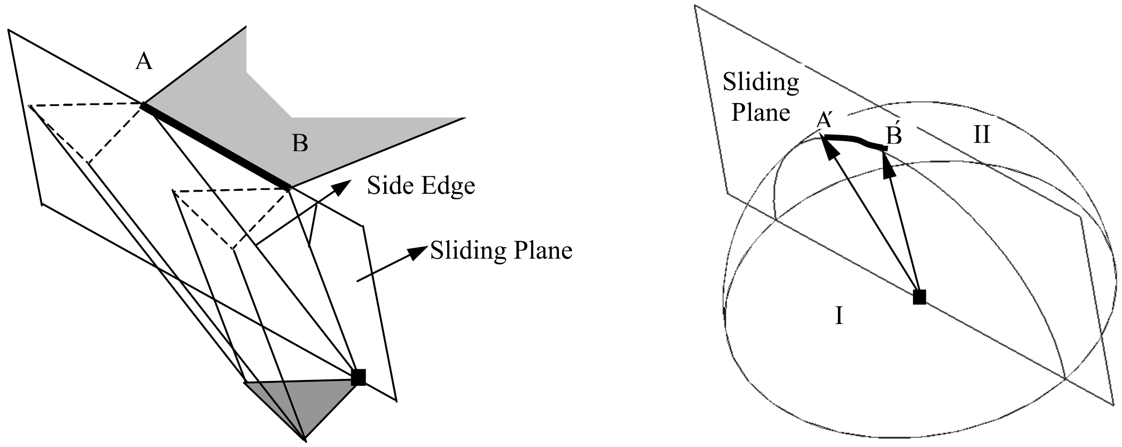
Property 2 of a *sliding plane* is illustrated by the example in Fig. 5. The *sliding plane* is determined by bold edge AB of the *obstacle facet* and the vertex denoted by "■" of the *base facet* (Fig. 5*a*). The 3-D light beam extruded from the *base facet* can slide in tangency to the *sliding plane* in the subspace containing the *base facet* at any orientation and will have no collision with the *obstacle facet*. The range of orientation variation of the 3-D light beam on the *sliding plane* is described as a half circle on the unit hemisphere (Fig. 5*b*). This half circle bisects the hemisphere into two regions *I* and *II* (Fig. 5*b*). Region *I* corresponds to the orientation domain where the 3-D light beam is in the subspace containing only the *base facet* and represents the visibility of the *base facet*. The 3-D light beam will have no intersection with the *obstacle facet* if the orientation is contained in region *I*. Region *II* contains the non-visible orientations of the *base facet* due to the interference from the *obstacle facet*. The 3-D light beam will have intersection with both subspaces containing the *base facet* and the *obstacle facet* respectively, if the orientation is in region *II*. Therefore the 3-D light beam may intersect the *obstacle facet*. The great arc ÁB́ on the half circle in Fig. 5*b* is part of the boundary of the traced spherical region.

3.4 Non-visibility cone of polygonal facet

In this section, we will show that the traced region of a 3-D light beam around an *obstacle facet* is the non-visibility cone of the *base facet* due to the *obstacle facet*.

Lemma 1: the spherical polygon obtained from tracing the 3-D light beam around the boundary of an *obstacle facet* is a convex spherical polygon.

Proof: Refer to Fig.6. Suppose there are *m sliding planes* for the 3-D light beam to trace around the boundary of the *obstacle facet*, and SP(*i*) and SP(*i+1*) are two consecutive *sliding planes*. Corresponding to the two *sliding planes* are two great circles GC(*i*) and



(*a*) (*b*)

Fig 5. A sliding plane bisecting unit hemisphere

GC($i+1$) on the hemisphere. Great circle GC(i) divides the hemisphere (H) into two regions I_i and II_i, written as $H=I_i+II_i$. Great circle GC($i+1$) divides the hemisphere into two regions I_{i+1} and II_{i+1}. Great arcs AB and BC are part of the boundary of the traced spherical region on great circles GC(i) and GC($i+1$) respectively. Great arcs AB and BC are mapped on to plane π through central projection as line segment A'B' and B'C'. Since spherical regions I_i and I_{i+1} both guarantee visibility of the *base facet*, then the traced spherical region belongs to the region shared commonly by regions II_i and II_{i+1} on the hemisphere (the region contained by half planes ABO and CBO). The angle between half planes ABO and CBO is less than 180°, therefore the planar angle A'B'C' projected on plane π through central projection is less than 180°. Since SP(i) and SP($i+1$) are two consecutive *sliding planes*, the inner angles of the planar polygon projected on plane π through central projection are all less than 180°, making the polygon convex. By the property of central projection, the corresponding traced spherical polygon is convex. □

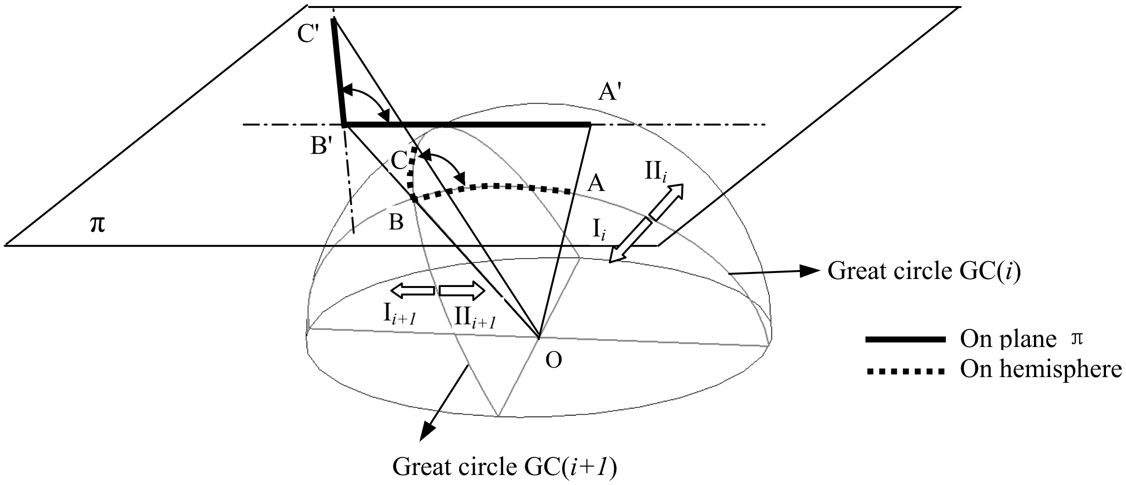


Fig. 6 Convexity of the spherical polygon

Lemma 2: the spherical polygon obtained from tracing the 3-D light beam around the boundary of an *obstacle facet* describes the non-visibility of the *base facet* due to the *obstacle facet*.

Proof: each *sliding plane* corresponds to a great half circle bisecting the hemisphere. The traced region is a closed spherical region shared commonly by regions II of all the great half circles, represented by $\bigcap II_i$, where *i=1,2…m* and *m* is the number of *sliding planes*. Since the complete set of orientations is represented as a hemisphere (*H*), any point *P* on the hemisphere outside of the boundary of the traced region belongs to the complementary set of the traced region, i.e.

$$\begin{aligned} P \in H - \bigcap II_i &= \overline{\bigcap II_i} \\ &= \overline{II_1} \cup \overline{II_2} \cup \ldots \cup \overline{II_m} \\ &= I_1 \cup I_2 \cup \ldots \cup I_m \end{aligned}$$

That means point *P* belongs to region *I* of a *sliding plane*, i.e. $\exists$ *i* between 1 and *m* so that $P \in I_i$. Therefore *P* represents a visible orientation.

Lemma 1 states convexity of the traced spherical polygon on the hemisphere and the planar polygon mapped on plane π through central projection. For a point *Q* inside the spherical polygon, the image of *Q* through central projection, *Q*' on plane π, is inside the planar polygon. A line *E*'*F*' through *Q*' on plane π will cross two different edges of the planar polygon at *E*' and *F*' (Fig. 7). Correspondingly, a great half circle *EF* through *Q* will cross the two different edges of the spherical polygon on the hemisphere at *E* and *F*. Since the two edges describe the sliding of the 3-D light beam on two different edges of the boundary of the *obstacle facet*, the 3-D light beam will cross the *obstacle facet* if moving along *EF*. The sliding of the 3-D light beam along the great circle of *EF* is a radial sliding process (Fig. 8*a*), by which each edge of the 3-D light beam slides radially on planes corresponding to exactly the same great circle *EF* on the hemisphere. The swept volume is considered to be constructed by parallel layers (Fig. 8*b*), each of which is formed by translating a convex polygon identical to the *base facet* over a linearly changed distance with regard to its height. Since each layer is convex, these layers stacked on top of one another build a convex volume. Therefore the swept volume of the 3-D light beam along great circle *EF* is convex. The intersection of such a convex swept volume with the convex *obstacle facet* is also convex. The point *Q* on the hemisphere therefore cannot be a visible direction, which otherwise is contradictory to the convexity of the intersection. □

So far we have proved that the traced spherical region contains all the non-visible orientations of the *base facet* due to the existence of the *obstacle facet*. In the following section, we describe how to determine the *sliding planes*.

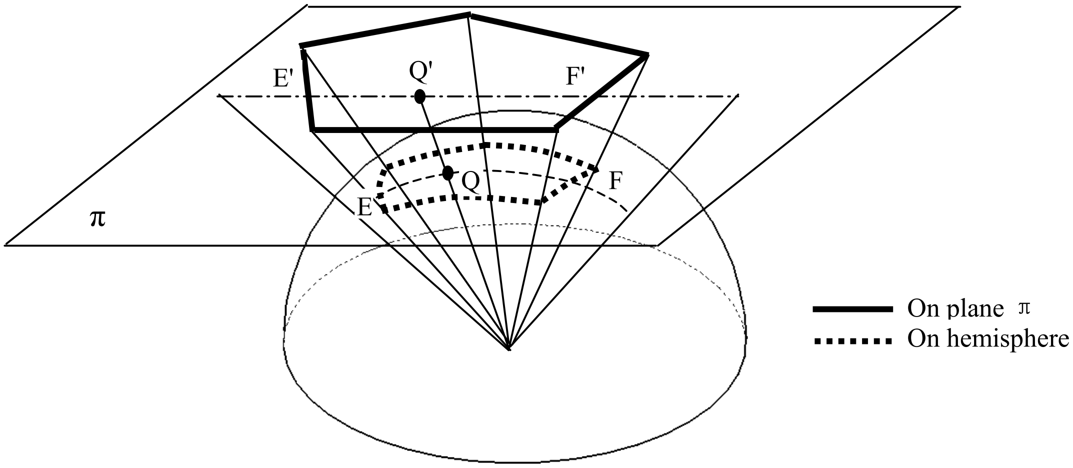


Fig. 7 An interior point of spherical polygon

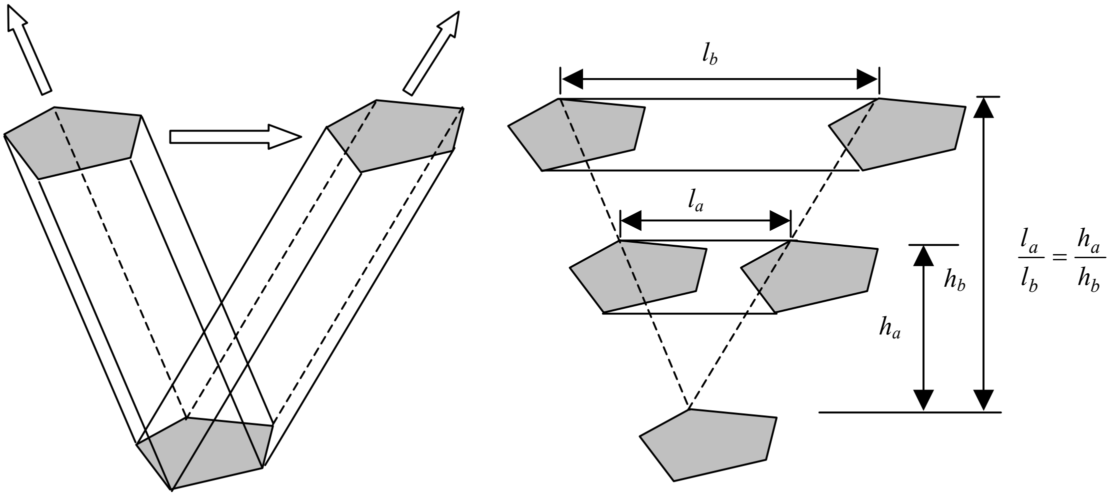


Fig. 8 Swept volume of a 3-D light beam

4. Determination of Sliding Planes

A *sliding plane* is tangent to both the *base facet* and the *obstacle facet*; tangent elements of both facets with the *sliding plane* could be a vertex or edge. A pair of tangent geometric elements from the *base facet* and the *obstacle facet* could be a combination of a vertex and a vertex, a vertex and an edge, or an edge and an edge. Such a pair of tangent geometric elements is actually a pair of static elements that defines a *sliding plane*. Although a pair composed of a vertex and a vertex is feasible for defining a contact sliding movement, the sliding movement is degenerated to be on a single line, and hence, the orientation change of the 3-D light beam maps to a point, a form of degeneracy of a great arc, on a unit hemisphere. Such a pair of elements for tangency is neglected in our analysis because it does not contribute to the construction of the spherical polygon that represents non-visibility. Other combinations, like a vertex and an edge, and an edge and an edge, contribute to the construction of a spherical polygon by mapping onto the unit hemisphere as great arcs (Fig. 4).

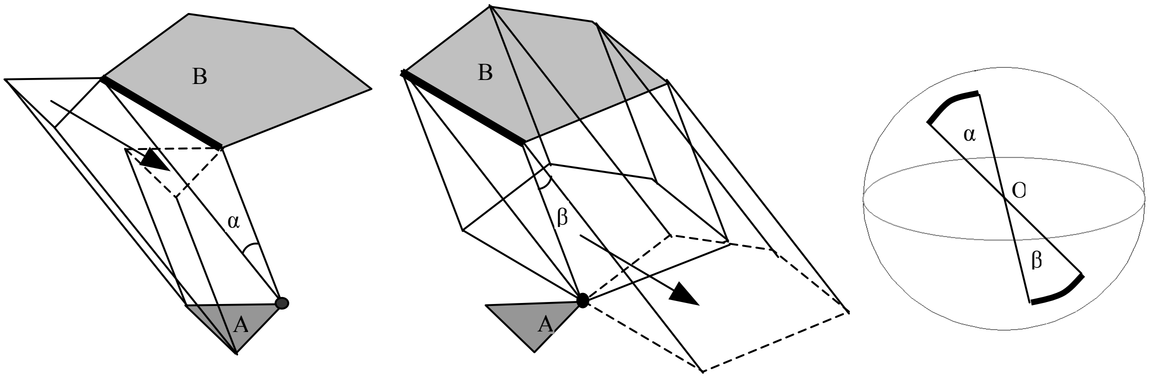
Since the 3-D light beam traces the boundary of an *obstacle facet* by sliding movements on a set of *sliding planes*, the non-visibility due to that *obstacle facet* can be determined as long as all these *sliding planes* are found. The set of *sliding planes* we are investigating are restricted to those determined by tangent pairs that include a vertex from one facet and an edge from the other facet (Figs. 4*a, b*). The vertex and vertex pair is neglected because the geometric degeneration, and the edge and edge pair (Fig. 4*c*) can be treated as a special case that is a combination of the two cases described in Fig. 4*a* and Fig. 4*b*. The construction of *sliding planes* depends on the tangent pairs between two convex polygons, regardless of *base facet* or *obstacle facet*; hence, we have **property 3**.

Property 3: Two convex polygonal facets share the same set of *sliding planes.*

Given **property 3**, **lemma 3** is provided below.

Lemma 3: given two convex polygonal facets, the non-visibility of each facet due to the other facet is symmetric about the origin of a unit sphere, if each facet is above the other facet along its normal.

Proof: Fig. 9. illustrates a *sliding plane* determined by the edge (in bold) of facet B and the vertex of facet A, denoted by "●". Fig. 9*a* shows the sliding of the 3-D light beam extruded from facet A along facet B on the *sliding plane*; the orientation change is represented as angle α. Correspondingly, a 3-D light beam extruded from facet B slides along facet A on the *sliding plane*; the orientation change is angle β (Fig. 9*b*). The two edges of angle α are parallel to those of angle β. They are expressed spherically as great arcs on a unit sphere and are symmetric about the origin of the unit sphere (Fig. 9*c*). Fig.9 shows a tangent pair that includes a vertex from facet A and an edge from facet B, the case previously shown in Fig. 4*a*. The same conclusion can be made for the tangent pair that includes an edge from facet A and a vertex from facet B, the case previously shown in Fig. 4*b* and the tangent pair that includes an edge from both facet A and B, the case previously shown in Fig. 4*c*. Since the two corresponding great arcs on one *sliding plane* are symmetric about the origin of the unit sphere, the non-visibility cones traced on all *sliding planes* are symmetric as well. □



(*a*) (*b*) (*c*)

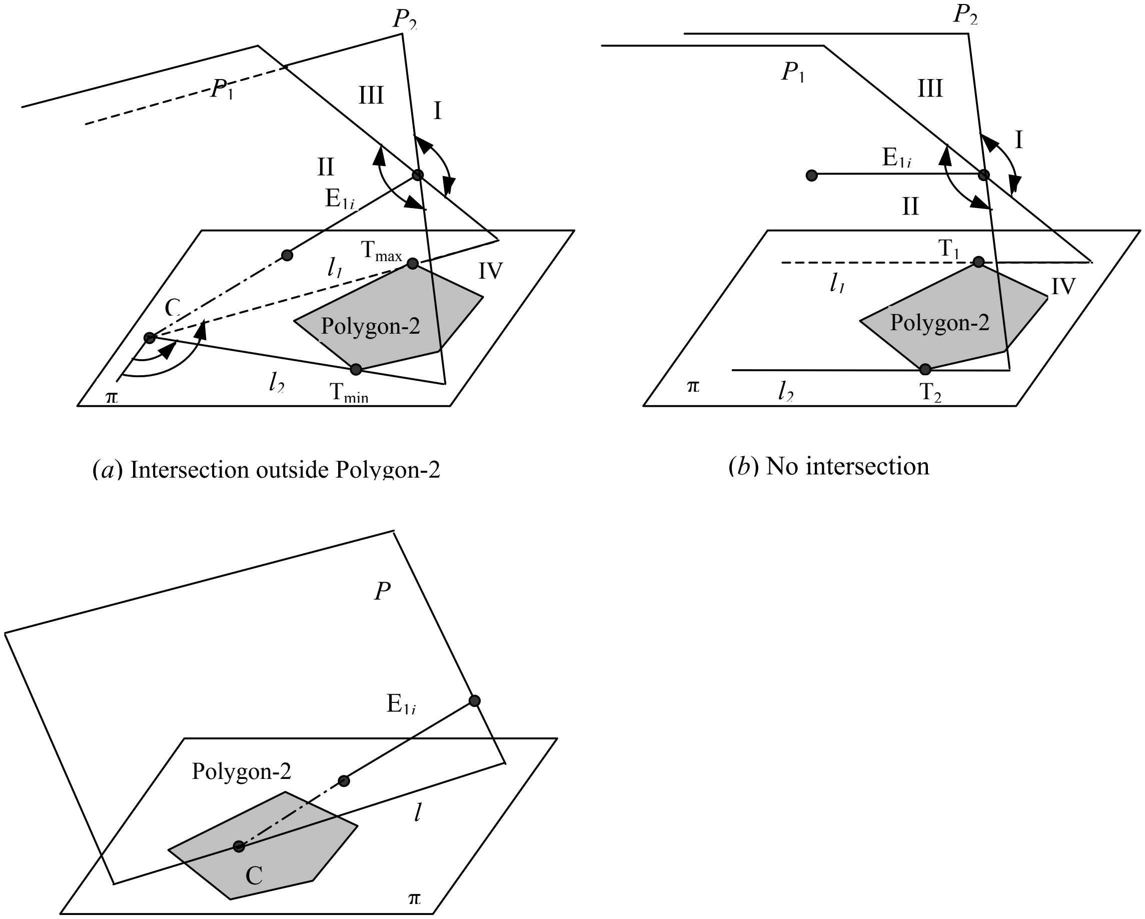
Fig. 9 Reflectivity of non-visibility of two convex polygonal facets

4.1 Determining sliding planes from polygon edges

To study the construction of *sliding planes* between two convex polygonal facets, we denote them simply by polygon-1 (*m* edges) and polygon-2 (*n* edges), instead of distinguishing them as a *base facet* or an *obstacle facet*. Consider an edge E_{1i} of polygon-1. E_{1i} or its extension can intersect with the plane (π) on which polygon-2 resides. The intersection point can be either inside polygon-2 (Fig. 10*a*) or outside polygon-2 (Fig. 10*c*) (note: polygon-1 is not shown in Fig.10). E_{1i} or its extension can even have no intersection with and is parallel to plane π (Fig. 10*b*). These three cases are discussed separately below.

1). E_{1i} or its extension intersects outside Polygon-2

If E_{1i} or its extension intersects outside Polygon-2 at point C (Fig. 10*a*), then any plane containing E_{1i} must have point C on its intersection line with plane π. All planes that contain E_{1i} are like a plane cluster pivoting on E_{1i}. To find those that are tangent to Polygon-2, a set of polar lines can be constructed from point C by connecting point C to the vertices of



(*a*) Intersection outside Polygon-2

(*b*) No intersection

(*c*) Intersection inside Polygon-2

Fig. 10 Determination of sliding planes

Polygon-2. The polar angles are compared, and the vertex of polygon-2 with maximum polar angle (T_{max}) and the vertex with minimum polar angle (T_{min}) are located. Polar lines l_1 and l_2 containing T_{max} and T_{min}, respectively, can establish two planes, P_1 and P_2, with E_{1i}. These two planes are potentially feasible *sliding planes,* because they both contain E_{1i} and at the same time are tangent to polygon-2. Planes P_1 and P_2 divide the 3-D space into four sub-spaces, I, II, III, and IV. In the situation where Polygon-1 falls within sub-spaces I or II, then there is one *sliding plane*. In Fig. 10*a*, if Polygon-1 is in sub-space I, then P_1 is the *sliding plane* to which Polygon-1 and Polygon-2 are tangent on different sides, however P_2 is an invalid *sliding plane* to which Polygon-1 and Polygon-2 are tangent on the same side. Likewise, Polygon-1 in sub-space II makes P_2 a *sliding plane* while P_1 is an invalid *sliding* *plane*, to which Polygon-1 and Polygon-2 are tangent on the same side. If Polygon-1 is in sub-space III, then both planes P_1 and P_2 are valid *sliding planes*, because they are both tangent to Polygon-1 and Polygon-2, and Polygon-1 and Polygon-2 are on different sides. Therefore, there are two great arcs that are mapped on to the unit hemisphere by sliding the 3-D light beam along E_{1i}. However, when Polygon-1 is contained in sub-space IV, there is no valid *sliding plane*, because both Polygon-1 and Polygon-2 are tangent to P_1 and P_2 on the same side.

2). E_{1i} parallel to Polygon-2

This is a case slightly different from the first case where E_{1i} or its extension intersects outside Polygon-2 at a point C. When E_{1i} is parallel to Polygon-2 (Fig. 10*b*), the intersection point could be considered to be at infinity. Two lines l_1 and l_2 parallel to E_{1i} can be constructed to be bounding Polygon-2 on plane π where Polygon-2 lies. Two tangent planes P_1 and P_2 are created as potential *sliding planes* from l_1 and E_{1i}, and l_2 and E_{1i}, respectively. As in the first case, four sub-spaces, I, II, III, and IV, are generated by planes P_1 and P_2. The same conclusions as to valid *sliding planes* can be drawn with respect to each of the sub-spaces I, II, III, and IV.

3). E_{1i} intersects inside Polygon-2

If E_{1i} or its extension intersects inside Polygon-2 at a point C (Fig. 10*c*), then any plane containing E_{1i} will also intersect with plane π inside Polygon-2. No plane tangent to Polygon-2 can be made through E_{1i}. Therefore no *sliding plane* exists.

The above discussion can be summarized into **Observation 2 and 3**.

Observation 2: Given two convex polygons, Polygon-1 and Polygon-2. If an edge of Polygon-1, E_{1i} or its extension, has no intersection point with Polygon-2, then two tangent planes through E_{1i}, P_1 and P_2, can be found to be tangent to Polygon-2. P_1 and P_2 divide the space into four sub-spaces I, II, III, and IV (Fig. 10*a* and Fig. 10*b*). If Polygon-1 belongs to either sub-space I or II, then there is one *sliding plane* with E_{1i}. If Polygon-1 belongs to sub-space III, then there are two *sliding planes* with E_{1i}. If Polygon-1 belongs to sub-space IV, then there is no *sliding plane* with E_{1i}.

Observation 3: Given two convex polygons, Polygon-1 and Polygon-2. If an edge of Polygon-1, E_{1i} or its extension, has an intersection point within Polygon-2, then there is no *sliding plane* with E_{1i}.

Once edges of Polygon-1 are all processed with vertices of Polygon-2 for *sliding planes*, edges of Polygon-2 will go through the same procedures with vertices of Polygon-1 to determine feasible *sliding planes*. All *sliding planes* can be found by doing so.

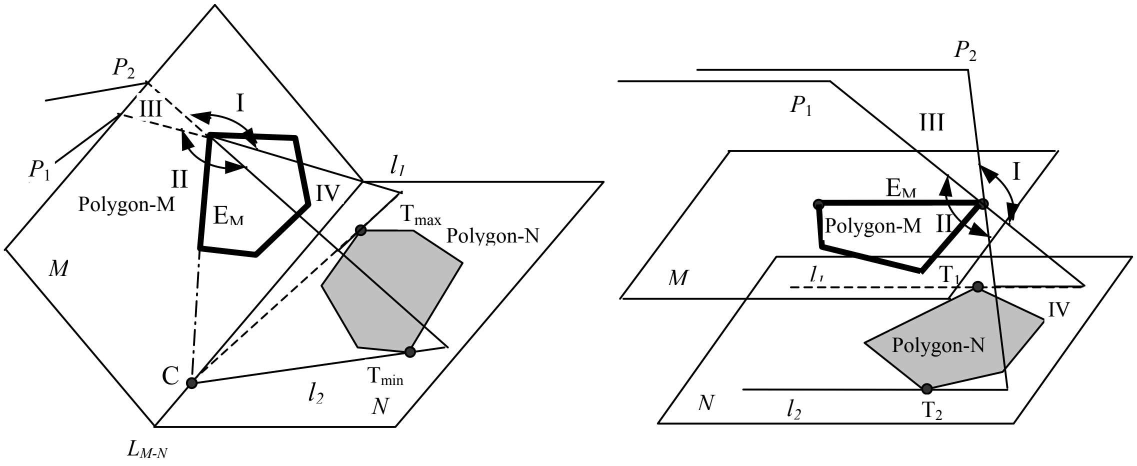
4.2 Edge number of a non-visibility cone

Based on **Observation 2 and 3**, the following lemmas are proposed and proved.

Lemma 4: Given two convex polygons, Polygon-M (with *m* edges, $\mathrm{E}_{\mathrm{M}i}$, i=1…m) on Plane-M and Polygon-N (with *n* edges, $\mathrm{E}_{\mathrm{N}i}$, i=1…n) on Plane-N. If the intersection line of Plane-M and Plane-N does not intersect Polygon-M and Polygon-N, then the spherical polygon mapped on the unit hemisphere has (m+n) edges.

Proof: Since the intersection of Plane-M and Plane-N does not intersect Polygon-M or Polygon-N, the intersection of Plane-M and Plane-N, line $l_{M\text{-}N}$ in Fig. 11*a* is outside of both Polygon-M and Polygon-N. For an edge $\mathrm{E}_{\mathrm{M}i}$ of Polygon-M, its extension intersects line $l_{M\text{-}N}$ at point C. Two tangent planes are constructed using the process described in section 4.1. It is shown in Fig. 11*a* that Polygon-M belongs to sub-space I. Actually Polygon-M can only lie in sub-space I or II. Suppose Polygon-M is in sub-space III or IV, Plane-M will have intersection with Polygon-N; that is, line $l_{M\text{-}N}$ will cross through Polygon-N. However this is contradictory to the fact that $l_{M\text{-}N}$ is outside of both Polygon-M and Polygon-N. Thus Polygon-M can only belong to sub-space I or II. From **Observation 2**, we know that once an edge is in either sub-space I or II, there is one valid *sliding plane* associated with that edge that corresponds to a great arc on the unit hemisphere. So for Polygon-M with *m* edges, there are *m* spherical edges mapped on the unit hemisphere. The same proof can be performed on Polygon-N. Therefore there are altogether m+n edges for the spherical polygon.

Figure 11b shows that Plane-M is parallel to Plane-N. In this case, line $l_{M\text{-}N}$ is infinitely away. Similarly Polygon-M can only belong to sub-space I or II. Therefore there are also m+n edges for the spherical polygon on the unit hemisphere. □



(*a*) Intersect outside

(*b*) Parallel

Fig. 11 No intersection on both polygons

Lemma 5: Given two convex polygons, Polygon-M (with *m* edges, E_{Mi}, i=1…m) on Plane-M and Polygon-N (with *n* edges, E_{Ni}, i=1…n) on Plane-N. If the intersection line of Plane-M and Plane-N does not intersect Polygon-M, but intersect Polygon-N, then the spherical polygon mapped on the unit hemisphere has $((m-m_0)*2+n)$ edges. m_0 is the number of a subset of Polygon-M edges,{E_{Mi}: E_{Mi} makes Polygon-M in sub-space IV, or, E_{Mi} or its extension intersects the inside of Polygon-N}.

Proof: Given that the intersection line of Plane-M and Plane-N does not intersect Polygon-M, but intersects Polygon-N, then Polygon-M is either in sub-space III or sub-space IV if we are analyzing one edge of Polygon-M and neither that edge, E_{Mi}, nor its extension intersects the inside of Polygon-N. Referring to Fig. 12*a* and Fig. 12*b* and based on **Observation 2**, if Polygon-M falls within sub-space III, two *sliding planes* can be obtained through E_{Mi}; if within sub-space IV, then no *sliding plane* exists. Also, if the edge E_{Mi} or its extension instersects the inside of Polygon-N, there is no *sliding plane* associated with E_{Mi} from **Observation 3**. Therefore there are $(m-m_0)*2$ *sliding planes* that can be constructed through the edges of Polygon-M.

Since the intersection line of Plane-M and Plane-N does not intersect Polygon-M, every edge of Polygon-N, E_{Ni}, or its extension intersects outside of Polygon-M **(**Fig. 12*c*).

Similar to the proof of **Lemma 4**, for each edge E_{Ni}, Polygon-N always exists in sub-space I or II. Therefore, there are *n sliding planes* that can be constructed.

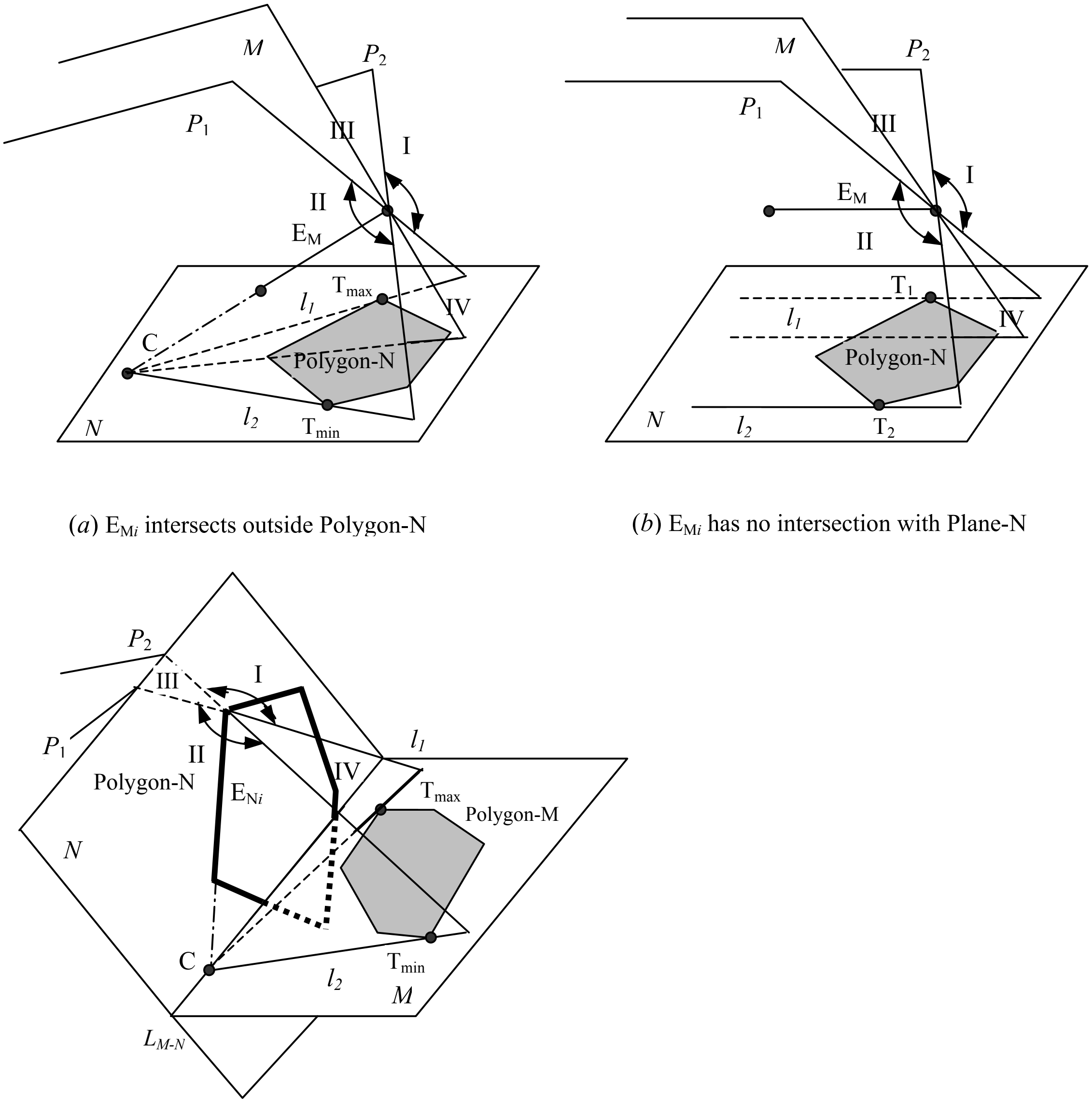
Altogether there are $((m\text{-}m_0)*2+n)$ *sliding planes*, and hence, $((m\text{-}m_0)*2+n)$ edges for the corresponding spherical polygon on a unit hemisphere. □

5. Implementation

In this section, two examples are given to show the effectiveness of the proposed method in computing the non-visibility of convex polygons. Example 1 demonstrates the case of non-visibility of two convex polygons described in **Lemma 4**. Figure 13 shows two triangles, T_1 and T_2. Triangle T_1 has vertices V_{11}, $V_{12,}$ and $V_{13,}$ and edges e_{11}, e_{12} and $e_{13.}$ Triangle T_2 has vertices V_{21}, V_{22} and V_{23}, and edges e_{21}, e_{22} and $e_{23.}$ The vertex coordinates as well as normal vectors are listed in Table 1. The intersection line of the two planes containing triangles T_1 and T_2 crosses neither of these two triangles. Therefore **Lemma 3** holds for example 1. Six *sliding planes* are found and their corresponding defining elements are shown in Table 2. Figure 14 shows non-visibility cones $IV_{1/2}$ (the non-visibility of T_1 due to the obstacle T_2) and $IV_{2/1}$ (the non-visibility of T_2 due to the obstacle T_1) on a unit sphere. The symbol "*" represents vertices of non-visibility cones in Fig. 14. Both $IV_{1/2}$ and $IV_{2/1}$ are closed spherical polygons and have six vertices and edges, the summation of the number of edges in T_1 and the number of edges in T_2. The vertices of non-visibility cones $IV_{1/2}$ and $IV_{2/1}$ are listed in Table 3. It can be seen that non-visibility cones $IV_{1/2}$ and $IV_{2/1}$ are symmetrical with respect to the origin of the unit sphere.

Table 1 Coordinates and normal vectors of two triangles (unit: cm)

Polygon	**Normal Vector**	**Vertex 1 (V_{11}/V_{21})**	**Vertex 2 (V_{12}/V_{22})**	**Vertex 3 (V_{13}/V_{23})**
Triangle 1(T_1)	(0, 0, 1)	(0.85, 0.92, 0)	(5.73, 1.97, 0)	(2.51, 4.36, 0)
Triangle 2(T_2)	(-0.46, -0.56, -0.69)	(-1.88, 3.95, 4.97)	(3.33, 2.06, 3.06)	(1.40, 5.25, 1.77)



(*a*) E_{Mi} intersects outside Polygon-N

(*b*) E_{Mi} has no intersection with Plane-N

(*c*) E_{Ni} always intersects outside Polygon-M

Fig. 12 Intersection with one polygon

Table 2 Sliding planes of example 1

Sliding Plane Number	**Defining Elements**	**Sliding Plane Number**	**Defining Elements**
1	e_{11} and V_{23}	4	e_{21} and V_{13}
2	e_{12} and V_{21}	5	e_{22} and V_{11}
3	e_{13} and V_{22}	6	e_{23} and V_{12}

Table 3 Vertices of spherical non-visibility cones $IV_{1/2}$ and $IV_{2/1}$

Non-visibility cones	**Vertex 1** **($V_{IV\,11}/V_{IV\,21}$)**	**Vertex2** **($V_{IV\,12}/V_{IV\,22}$)**	**Vertex 3** **($V_{IV\,13}/V_{IV\,23}$)**	**Vertex 4** **($V_{IV\,14}/V_{IV\,24}$)**	**Vertex 5** **($V_{IV\,15}/V_{IV\,25}$)**	**Vertex 6** **($V_{IV\,16}/V_{IV\,26}$)**
$IV_{1/2}$	(0.12, 0.92, 0.38)	(-0.76, 0.57, 0.31)	(-0.82, 0.21, 0.53)	(-0.66, -0.06, 0.75)	(0.60, 0.28, 0.75)	(0.21, -0.59, 0.78)
$IV_{2/1}$	(-0.12, -0.92, -0.38)	(0.76, -0.57, -0.31)	(0.82, -0.21, -0.53)	(0.66, 0.06, -0.75)	(-0.60, -0.28, -0.75)	(-0.21, 0.59, -0.78)

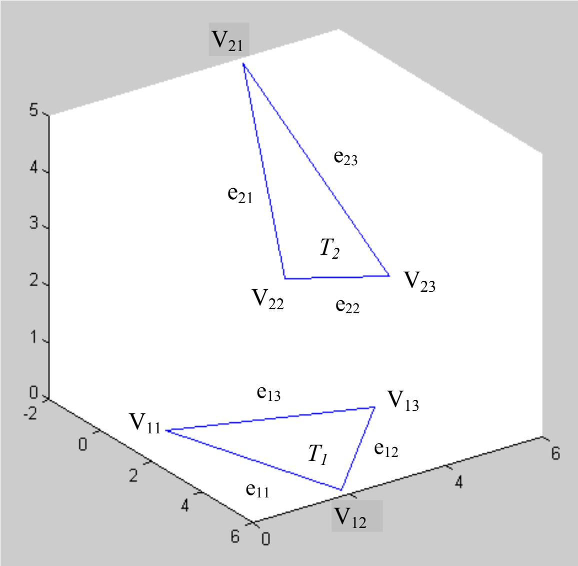


Fig. 13 Two triangles

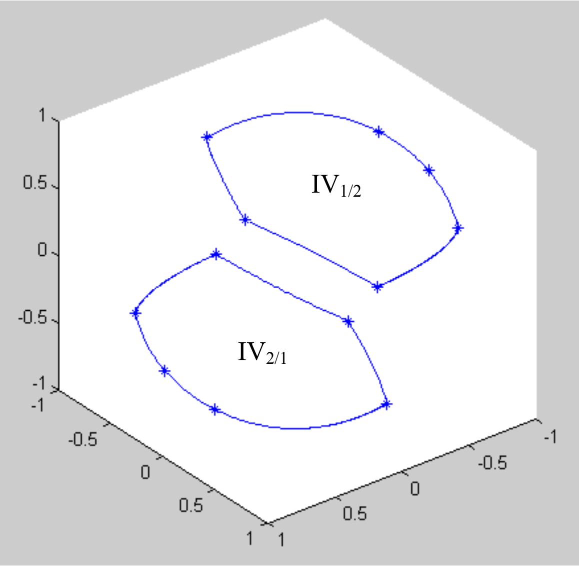


Fig.14 Mutual non-visibility cones on unit sphere

Example 2 illustrates the case as described in **Lemma 5**. Two convex polygons, one with 5 edges (a 5-gon, denoted as P_5) and the other with 6 edges (a 6-gon denoted as P_6), are displayed in Fig. 15. Polygon P_5 has vertices V_{51}, $V_{52,}$ ... $V_{55,}$ and edges e_{51}, e_{52}, …e_{55}. Polygon P_6 has vertices V_{61}, $V_{62,}$ ... $V_{66,}$ and edges e_{61}, e_{62}, …e_{66}. The vertex coordinates as well as normal vectors are listed in Table 4. The intersection line of the two planes containing polygon P_5 and P_6 crosses polygon P_5, but not P_6 (Fig. 15). In addition, extensions of edges e_{62} and e_{66} intersect inside of P_5, as is represented by the dash lines in Fig.15. Therefore example 2 does not belong to the situation described in **Lemma 4**. According to **Lemma 5**, m_0 is the number of a subset of the edges of a polygon such that those edges make the polygon in sub-space IV, or, those edges or their extension intersects the inside of the other Polygon. For example 2, the subset includes e_{63}, e_{64} and e_{65} (because they make polygon P_6 in sub-space IV) and e_{62} and e_{66} (because their extension intersects the interior of polygon P_5). Therefore m_0 for example 2 is 5, and m and n are the numbers of edges of polygons P_6 and P_5, respectively. Consequently the number of *sliding planes* is obtained as $(m\text{-}m_0)*2+n$=(6-5)*2+5=7. The defining elements of *sliding planes* of example 2 are listed in Table 5. The corresponding non-visibility cone of Polygon P_5 due to the interference of polygon P_6 is shown in Fig. 16. The vertices of the non-visibility cone $IV_{5/6}$ are listed in Table 6.

However, if Polygon P_6 is translated along the X-axis by the amount of 3.5(cm) to P'_6, then the intersection line of the two planes containing polygon P_5 and P'_6 will have intersection with neither of them (Fig. 17). Under this circumstance, both **Lemma 3** and **Lemma 4** hold for polygon P_5 and P'_6. Hence the non-visibility of P_5 due to P'_6 ($IV_{5/6'}$) and the non-visibility of P'_6 due to P_5 ($IV_{6'/5}$) are symmetrical about the origin of the unit sphere and both have 11(5+6) edges (Fig. 18).

Table 4 Coordinates and normal vectors of a 5-gon and a 6-gon (unit: cm)

Polygon	**Normal Vector**	**Vertex 1 (V_{51}/V_{61})**	**Vertex 2 (V_{52}/V_{62})**	**Vertex 3 (V_{53}/V_{63})**	**Vertex 4 (V_{54}/V_{64})**	**Vertex 5 (V_{55}/V_{65})**	**Vertex 6 (-/V_{66})**
Polygon P5	(0, 0, 1)	(-2.10, -2.25, 0)	(2.54, -1.89, 0)	(3.91, 0.72, 0)	(1.45, 3.55, 0)	(-1.76, 2.81, 0)	—
Polygon P6	(-0.61, -0.61, -0.51)	(0.88, -1.97, 4.31)	(-1.77, 0.68, 4.31)	(-5.19, 1.22, 7.75)	(-4.65, -1.03, 9.81)	(-2.12, -3.96, 10.30)	(1.49, -5.13, 7.37)

Table 5 Sliding planes of example 2

Sliding Plane Number	Defining Elements	Sliding Plane Number	Defining Elements
1	e_{51} and V_{62}	5	e_{55} and V_{61}
2	e_{52} and V_{62}	6	e_{61} and V_{51}
3	e_{53} and V_{62}	7	e_{61} and V_{54}
4	e_{54} and V_{61}		

Table 6 Vertices of spherical non-visibility cones $IV_{5/6}$

Vertex number	Coordinates	Vertex number	Coordinates
1	(0.06, 0.56, 0.83)	5	(-0.08, -0.79, 0.61)
2	(-0.65, 0.39, 0.65)	6	(0.38, 0.69, 0.62)
3	(-0.80, -0.01, 0.60)	7	(0.57, 0.05, 0.82)
4	(-0.53, -0.47, 0.71)		

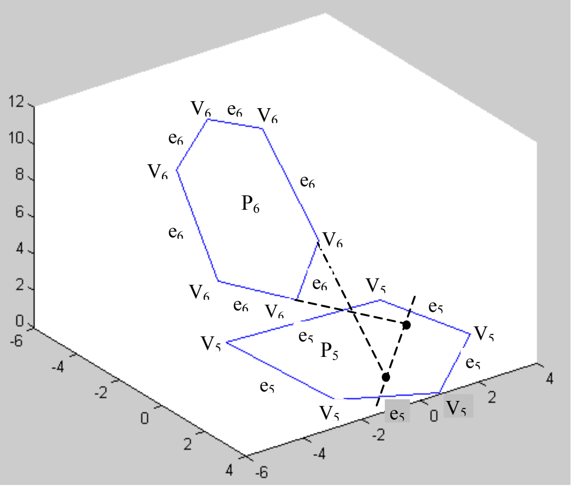


Fig. 15 A 5-gon and a 6-gon

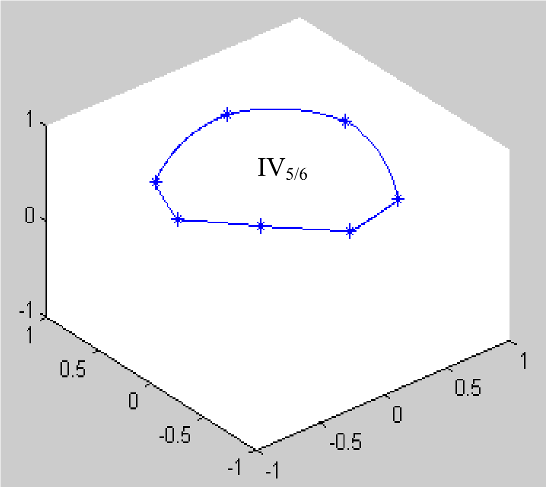
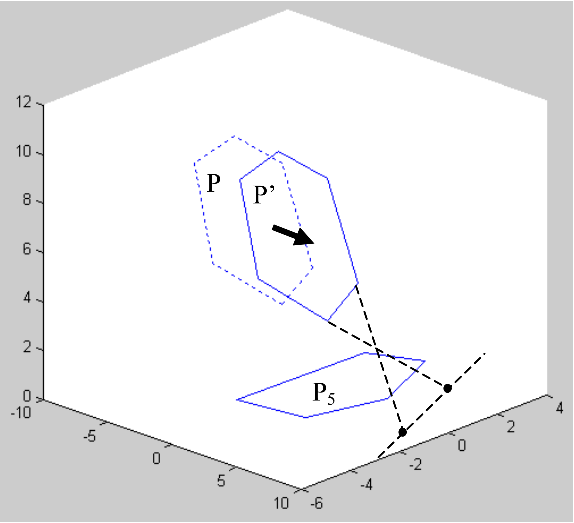


Fig. 16 Non-visibility of 5-gon due to 6-gon



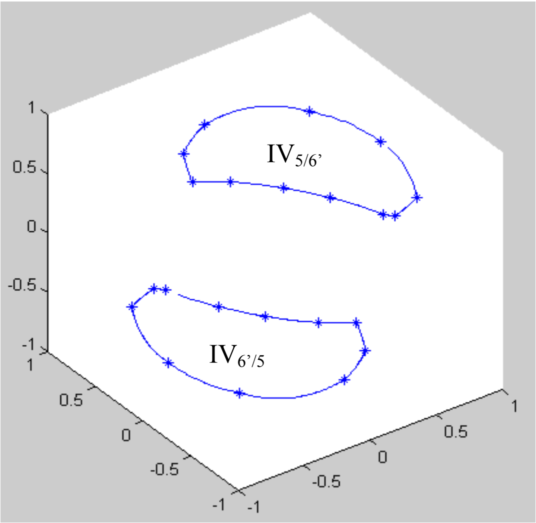


Fig. 17 Translated polygon P'_6

Fig. 18 Non-visibility cones $IV_{5/6'}$ and $IV_{6'/5}$

6. Computational Results

This section provides the examples of applying our proposed approach in computing visibility of 3-D objects. Computational time is given for the examples. We used STL files that are available from commercial software to demonstrate the computational results. The approach is implemented in the C programming language on a Pentium (R) D CPU 3.00Ghz PC running Windows XP. Table 7 shows the computational time of three part models as well as the number of tessellated facets on their surfaces. The second part in Table 7 (a block with a cylinder through hole) was tested with different facet numbers by changing surface tolerance, and the result is shown in Table 8 and Fig 19**.**

To obtain the visibility of a facet, its non-visibility cones with all other facets should be computed. Therefore for a polyhedral CAD model with n facets, the computational complexity of computing its visibility is $O(n^2)$. Currently in our program, only those facets that are either partially or completely above the plane where the facet under study resides are used to compute non-visibility cones. The non-visibility cones with those facets that are below the plane are assigned as null, because the potential visibility of a facet is limited to a hemisphere with the pole pointing to its normal vector. Since the visibility of a concave facet is only blocked by the facets in the same concave region [Chen et. al. 1993], the computational performance of the approach can be improved through identifying concave

Table 7 Computational results of three CAD models

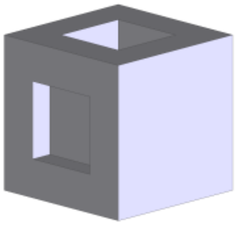
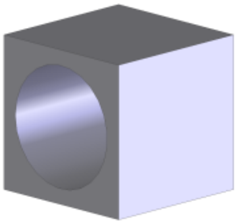
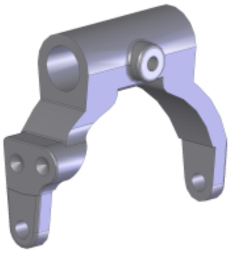
Part Number and Name	1. Block with Four Square Pockets	2. Block with a Cylinder Through Hole	3. Linkage
Part Geometry			
Number of Facets	76	140	2196
Computational Time (sec)	0.703	1.812	183.301

Table 8 Computational time of Part No. 2 with different facet number

Number of Facets	140	196	256	320	380
Computational Time (sec)	1.812	3.187	5.079	7.875	10.844

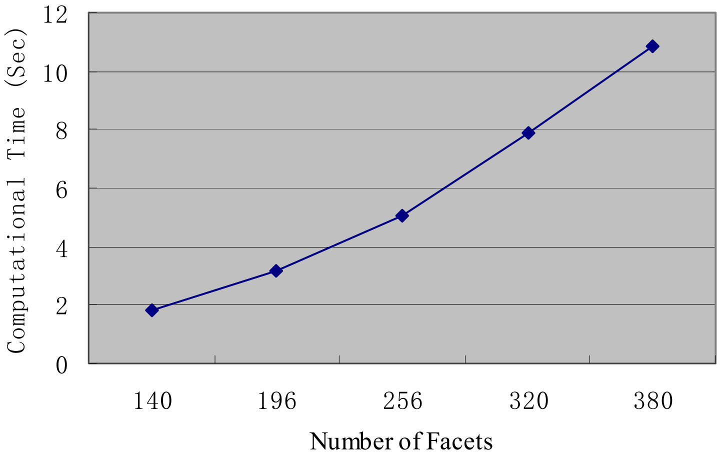


Fig. 19 Computational time v.s. facet number : Part No. 2

regions. The concave facets should also be grouped into their corresponding concave regions. It is expected that this is where our future improvement lies.

7. Conclusion

This paper presented a method of computing non-visibility, the exact complementary set of visibility, for polygonal facets on a polyhedral CAD model. The boundaries of a non-visibility cone are computed directly by determining *sliding planes*. The input can be arbitrary convex planar polygonal facets; hence the approach can be used for any polygonal model with planar facets. The visibility considered in this paper is complete visibility, requiring each point on a facet to be visible. Therefore a partial visible facet will be rated as completely non-visible. In our future work, partial visibility will be taken into account, and will be used as a guide to further subdivide a partially visible facet so that the facet may be visible from the combination of a set of directions. This can help determine workpiece set up orientations in *feature-less machining* methods, which we have previously developed in other research. Another interest of our future work is to construct an analytical model of the *sliding planes* between two facets. Since *sliding planes* determine the boundaries of a non-visibility cone, an analytical model of the *sliding planes* can be used to help adjust the location and geometry of an obstacle facet, leading to a new method of re-designing polyhedral models for increased visibility.

In conclusion, the method of computing non-visibility presented in this paper can have useful applications for manufacturing analysis, such as determination of setup orientations and axes of rotation for multi-axis machining. In addition, it can be used as an accurate analysis of the manufacturability of a design. These topics will be the focus of future research and publications.

References

[Balasubramaniam et. al. 2000] Balasubramaniam, M., Laxmiprasad, P., Sarma, S., and Shaikh, Z., 2000, “Generating 5-axis NC Roughing Paths Directly From A Tessellated Representation,” Computer Aided Design, **32**(4), pp. 261-277.

[Chen and Woo 1992] Chen, L. L., and Woo, T. C., 1992, “Computational Geometry on the Sphere with Application to Automated Machining,” ASME Journal of Mechanical Design, **114**, pp. 288-295.

[Chen et. al. 1993] Chen, L.L., Chou, S.Y., and Woo, T. C., 1993, “Parting Directions For Mould And Die Design”, Computer-Aided Design, Volume 25, Issue 12, pp. 762-768.

[Chen et. al. 2001] Chen, H.K., Hu S. J., and Woo T. C., 2001, “Visibility Analysis and Synthesis for Assembly Fixture Certification Using Theodolite Systems”, ASME Journal of Manufacturing Science and Engineering, Vol. 123, pp. 83-89.

[Dhaliwal et. al. 2003] Dhaliwal, S., Gupta, S. K., Huang, J., and Priyadarshi, A., 2003, “Algorithms for Computing Global Accessibility Cones,” ASME Journal of Computing and Information Science in Engineering., 3(3), pp. 200–209.

[Elber et. al. 2004] Elber, G., Chen, X., and Cohen, E., 2004, “Mold Accessibility Via Gauss Map Analysis”, Proceedings - Shape Modeling International SMI 2004, pp. 263-272.

[Frank et. al. 2006] Frank, M.C., Wysk, R.A., and Joshi, S.B., 2006, “Determining Setup Orientations from the Visibility of Slice Geometry for Rapid CNC Machining”, ASME Journal of Manufacturing Science and Engineering, 128, pp. 228–238.

[Fu et. al. 2002] Fu, M. W., Nee, A. Y. C., and Fuh, J. Y. H., 2002, “ The Application Of Surface Visibility And Moldability To Parting Line Generation”, Computer-Aided Design, Volume 34, Issue 6, May pp. 469-480

[Huang et. al. 2003] Huang, J., Gupta, S.K., and Stoppel, K., 2003, “Generating Sacrificial Multi-Piece Molds Using Accessibility Driven Spatial Partitioning”, Computer Aided Design, v 35, n 13, November, pp. 1147-1160

[Khardekar et. al. 2006] Khardekar R., Burton G., McMains S., 2006, “Finding Feasible Mold Parting Directions Using Graphics Hardware”, Computer-Aided Design, v 38, n 4 pp. 327–341

[Kong et. al. 2005] Kong, Z., Huang, W. and Ceglarek, D., 2005, “Visibility Analysis for Assembly Fixture Calibration Using Screen Space Transformation”, ASME Journal of Manufacturing Science and Engineering, Vol. 127, pp. 622-634.

[Kweon and Medeiros 1998] Kweon, S. and Medeiros, D. J., 1998, “Part Orientations For CMM Inspection Using Dimensioned Visibility Maps ”, Computer-Aided Design, Volume 30, Issue 9, August, pp. 741-749

[Lim and Menq 1994] Lim, C. P. and Menq, C. H., 1994, “CMM Feature Accessibility And Path Generation” International Journal of Production Research, 32(3), pp. 597–618.

[O'Rourke1998] O'Rourke, J. 1998, “Computational Geometry in C (2nd Ed.)”, Cambridge University Press.

[Priyadarshi and Gupta 2004] Priyadarshi, Alok K., and Gupta, Satyandra K, 2004, “Geometric Algorithms For Automated Design Of Multi-Piece Permanent Molds”, Computer Aided Design, v 36, n 3, March, pp. 241-260

[Spitz et. al. 1999] Spitz, S. N., Spyridi, A. J., and Requicha, A.A.G., 1999, “Accessibility Analysis for Planning of Dimensional Inspection with Coordinate Measuring Machines”, IEEE Transactions on Robotics and Automation, Vol. 15, No. 4, pp. 714-727.

[Spyridi and Requicha 1990] Spyridi, A. J. and Requicha, A. A. G., 1990, “Accessibility Analysis For The Automatic Inspection Of Mechanical Parts By Coordinate Measuring Machines,” in Proc. IEEE Int. Conf. Robot. Automat., Cincinnati, OH, May, pp. 1284–1289.

[Stewart 1999] Stewart, A.J., 1999, "Computing Visibility From Folded Surfaces", Computers & Graphics, Vol. 23, No.5, pp. 693-702.

[Suh and Kang 1995] Suh, S. H., and Kang, J. K., 1995, "Process Planning for Multi-axis NC Machining of Free Surfaces," International Journal of Production Research, 33(10), pp. 2723-2738.

[Yang et. al. 1999] Yang, W., Ding, H., and Xiong, Y., 1999, "Manufacturability Analysis For A Sculptured Surface Using Visibility Cone Computation", International Journal of Advanced Manufacturing Technology, v 15, n 5, pp. 317-321

[Yin et. al. 2000] Yin, Z.P., Ding, H., and Xiong, Y.L., 2000, "Visibility Theory And Algorithms With Application To Manufacturing Processes", International Journal of Production Research, Vol. 38, No. 13, pp. 2891-2909.

CHAPTER 4. COMPUTING AXES OF ROTATION FOR SETUP PLANNING USING VISIBILITY OF POLYHEDRAL CAD MODELS

A paper submitted to *Computer Aided Design*

Ye Li and Matthew C. Frank

Abstract

This paper presents a method for determining feasible axes of rotation for setup planning, based on the visibility of a polyhedral model. The intent of this work was to develop a feature-free approach to setup planning, with the specific focus on multi-axis machine setups. Visibility mapping can provide a quantitative evaluation of a surface, a feature or an entire part model; however, the next step is to use this information for process planning. In this paper, we present an approach of using a visibility map to evaluate axes of rotation that could be used in an indexer-type setup on a machine tool. Instead of using expensive and complicated multi-axis machining, it may be feasible to machine using multiple 3-axis toolpaths if a single axis of rotation can be used to rotate the part through the minimum set of orientations. An algorithm is presented that is capable of processing visibility information from a polyhedral model; hence the method is generic and does not require feature detection. As such, the work is applicable to a variety of applications; in particular for subtractive rapid prototyping where complex geometry may not contain recognizable features.

1. Introduction

CNC machining is widely used in the creation of complex shapes for aerospace, automobile and biomedical industries and lately for use in subtractive rapid prototyping. As a material removal process controlled by programs, CNC machining has demonstrated its capabilities in accuracy, efficiency and repeatability. It is particularly flexible in that it can use a wide range of materials unlike almost any other manufacturing or rapid prototyping processes. However, CNC machining is still limited by the complexity in process planning required to create the NC code. In particular *tool accessibility*, which is required by a material removal process, is a significant challenge in the implementation of CNC machining, especially for rapid prototyping. An accessibility analysis is required before a machining operation can be processed. The more accessibility a machining setup provides, the more complex shapes it is able to machine; therefore, accessibility is an indicator to describe the flexibility and versatility of a machine, or of a particular machine setup. Modern multi-axis CNC machines rely heavily on the simultaneous motions of several axes in order to access the cutting surfaces of the part. The complexity and cost involved in process planning and programming to generate the toolpaths increases as the number of controlled axes increases (typically 2 to 5). Similarly, the cost of a CNC machine increases significantly with the number of controllable axes. With this in mind, an economical method for machining complex geometries is sought, with the purpose of reducing the cost and difficulty related to process planning. This may be possible if a set of setup orientations about a rotary indexer can machine the part completely, and only require simpler 3-axis programming.

Accessibility of milling machines is often geometrically approximated by the concept of *visibility*, in the form of line-of-sight accessibility. Hence, the accessibility of a 3-axis milling machines is a *single point* on a unit sphere whereas the accessibility of 4- and 5-axis machines expand to a *great arc* and a *cone* on a unit sphere, respectively. Since a geometric point is the constitutive element comprising a one-dimensional arc and a two-dimensional cone, the capability of a 3-axis machine is therefore a subset of that of a 4-axis machine, and hence a subset of a 5-axis machine. In addition to regular 3-5 axis machines, 3-axis machines with a 4th axis indexer provide inclined end milling capabilities on 3-axis machines in order to create curved surfaces [Ip and Loftus 1993] and generally increased accessibility. A 4-axis indexed machine is constructed from a 3-axis machine with at least one, but sometimes two opposing chucks mounted on the work table. A 4-axis indexed machine rotates the workpiece about the axis of rotation between operations; thus its accessibility is a great circle, instead of a great arc, allowing access to the workpiece from all radial directions about the axis of rotation. This capability avoids numerous re-fixturing/re-clamping operations and can allow access to some undercut features. Recently, this method of 4-axis indexed machining has been implemented as a *Rapid Prototyping* strategy called CNC-RP and has demonstrated capability in creating complex geometries without the use of complex 4- and 5-axis NC programming [Frank et al, 2004]. This paper is motivated in part by unsolved problems in the complete implementation of the CNC-RP method in software. This work can avoid one manual operation that the user currently performs, a detrimental characteristic with regard to the expectations of process planning for an RP process. This step is the initial selection of an axis of rotation for a particular part geometry.

The selection of an axis of rotation is a critical step in the implementation of a 4-axis indexed machining strategy. A proper axis of rotation provides

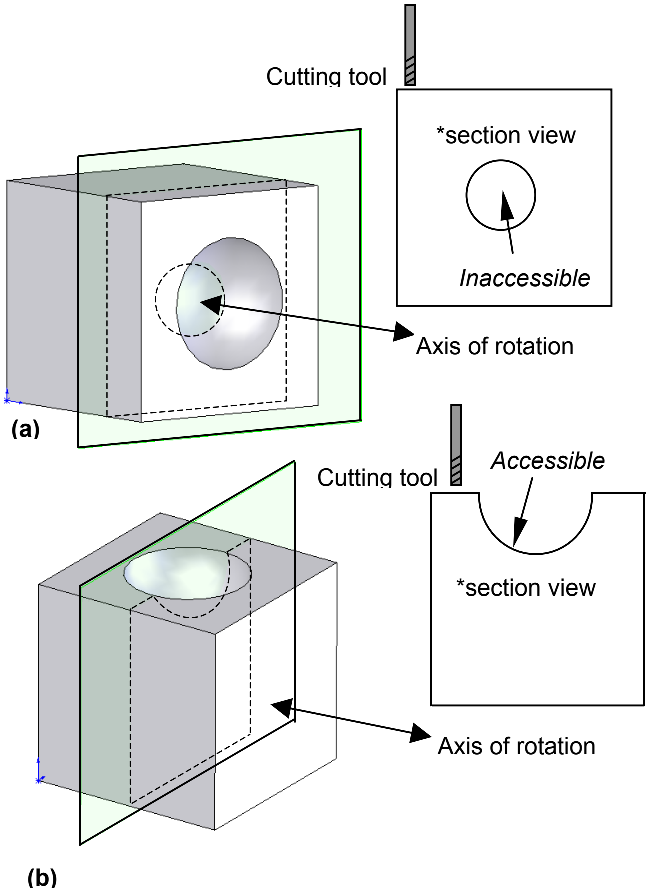


Fig.1 Axis of rotation and accessibility to a simple feature on a cube; (a) poorly chosen axis with no accessibility to hemispherical pocket and (b) proper choice with pocket accessible

better accessibility, reducing the number of setups and the need for re-fixturing. Figure 1 shows a simple cubic part with a hemisphere pocket on one side to illustrate axes selection. A poorly chosen axis of rotation produces an inaccessible surface that requires a second setup, while a better axis can obviously make the part machinable in one setup. Granted, a proper axis for this part may seem intuitive; however, for a part with complex freeform surfaces, an analytical method for searching feasible axes of rotation is necessary.

2. Literature Review

Workpiece setup in CNC machining is directly related to a part's manufacturability and the resulting quality of the machined surfaces; therefore, research on this issue has received extensive attention. Much of the literature employs the concept of *features* and related *feature recognition* methods, which can facilitate process planning by linking geometric part information to manufacturing processes. A feature is an aggregate of geometric entities that together convey important information to the downstream manufacturing activities. Feature recognition is the first step in performing feature-based analysis. A number of reviews on feature recognition can be found in [Allada and Anand 1995, 1996; Miao et al, 2002; Salomons et al, 1993; Zulkifli and Meeran 1999]. Assuming features can be recognized, design information becomes interpretable in the manufacturing stage, which facilitates concurrent engineering and thus speeds up the product development cycle.

The recognized features on a design model and their interrelation make the design description ready for machining setup planning. Ferreira and Liu developed a rule-based system to generate setup orientations for workpieces described with features [Ferreira and Liu 1988]. Demey et al, determined the minimal number of setups, considering both the physical conditions and the economical and quality issues [Demey et al, 1996]. Chu and Gadh classified features into single approach direction features and multi approach direction features, and then determined the minimized number of setups along with knowledge-based rules [Chu and Gadh 1996]. The rule-based approaches allow setups generated with user-intended objectives, therefore provide the integration of human knowledge into manufacturing process planning. In determining setup orientations, feature-based workpieces are also receiving other physical constraints from machine configurations and fixturing devices [Cevdet 2004]. Wu and Chang developed an automated setup selection method based on tolerance analysis [Wu and Chang 1998]. The setups are ranked and then released for fixture selection. Yen et al, integrated setup planning with geometric positioning and tolerancing for fixture planning [Yen et al, 2001]. To meet the constraints imposed on setup planning from machine capabilities and design information, a number of techniques such as fuzzy-set [Ong and Nee 1996] and GA and SA [Ong et al, 2002] have been used for searching for optimal solutions.

The reason for using feature-based approaches owes somewhat to the fact that most designs, particularly those for mechanical parts, are geometrically composed of *features*. However the increasing need to handle freeform shapes such as those reconstructed from reverse engineering techniques (e.g. laser scanning, CT and MRI scanning) pose new challenges to manufacturing process planning. On those freeform shapes, there may be no definable "features", which lends feature-based approaches incapable. Since the primary concern of setup planning for material removal processes is to provide accessibility for the cutting tool, researchers identified visibility as the necessary condition of providing accessibility and have used it to extract setup planning through geometric operations. Suh and Kang used a discretized model to construct global visibility and a similar method to find a feasible setup orientation for 4-axis milling [Suh and Kang 1995]. Gan et al, constructed a visibility map from a Gaussian map [Gan et al, 1994]. Other researchers used the visibility map constructed from a Gaussian map to compute setup orientations for 4- and 5- axis machining [Tang et al, 1992; Chen et al, 1993; Haghpassand and Oliver 1995]. For 4-axis milling, a feasible setup orientation should be one that allows a great circle orthogonal to it to intersect all spherical visibility polygons. However the visibility constructed from Gaussian maps is local visibility that cannot guarantee global accessibility.

3. Visibility

Visibility describes the accessibility of a line of sight. A line connecting a point ***A*** on a surface with a viewing point not obstructed by any other surfaces or objects, is said to denote a visible direction of point ***A***. Depending on the location of the viewing point relative to point ***A***, visibility is classified as one of two categories: *local* visibility and *global* visibility. The viewing point of local visibility is relatively close to the point and does not consider all surrounding surfaces/objects as obstacles to visibility. Local visibility of a point is determined only by its normal vector. However, global visibility must consider all surrounding surfaces and objects, which could be potential obstacles blocking visibility. Therefore global visibility provides a more accurate description of accessibility. It can be said that for manufacturing process planning, global visibility gains greater importance as the complexity of geometry of the part surfaces increases.

As a material removal process, CNC machining requires accessibility by the cutting tools, which makes the determination of global visibility a critical step in process planning. Suh and Kang constructed a binary spherical map to compute global visibility [Suh and Kang 1995]. However it cannot obtain the exact global visibility, because the computed visibility actually represents the visibility of the centroid of each triangular patch on the surface model. Dhaliwal et al, computed accurate non-visibility for an object represented by triangular facets by projection and convex hull operations [Dhaliwal et al, 2003]. Balasubramaniam et al, used graphic techniques to obtain visibility information [Balasubramaniam et al, 2000]. However, graphic based approaches may not render exact visibility due to the resolution of the hardware.

In this paper, an approach developed by the authors to compute exact visibility is used to generate feasible axes of rotation for 4-axis indexed milling operations. The previous work on visibility is based on determining its complementary set, *non-visibility*. A non-Visibility

Cone due to one obstacle facet is obtained by tracing a 3-D light beam extruded from the facet under analysis along the boundary of the obstacle facet (Figure 2). This approach is able to compute visibility for convex polygons with any number of sides, not limited to triangular facetted models; therefore it has flexibility for various input models [Li and Frank 2007].

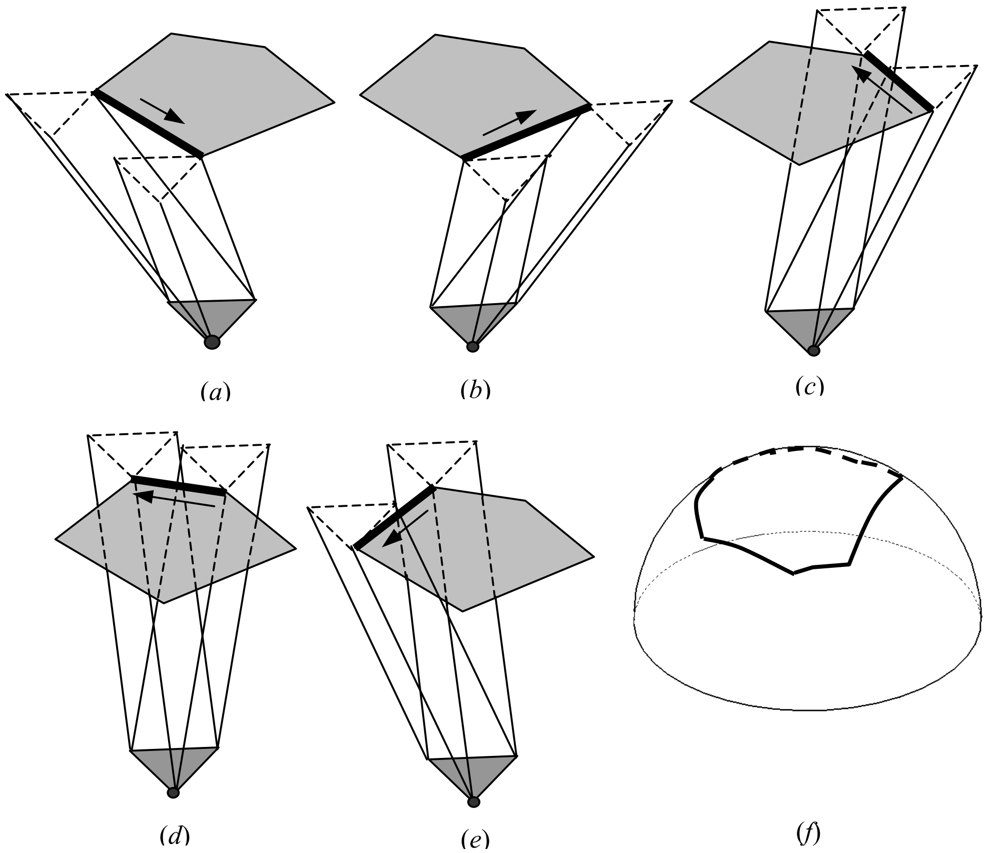


Fig. 2 Illustration of Non-visibility by Boundary Tracing [Li and Frank 2007]; (***a-e***), tracing the boundary of an obstacle polygon with respect to another polygon, and (***f***) coinciding swept arcs of the non-visibility map on a unit sphere

4. From Visibility to Axis of Rotation

Though visibility is generally represented as a cone on a unit sphere, this section will begin with analysis of the simplest form of visibility- *Point Visibility*, and then extend to two more complex forms of visibility- *Arc Visibility* and *Cone Visibility*. Point Visibility, as the basic element, constitutes Arc Visibility and Cone Visibility. In this section, the relations between these three forms of visibility and axes of rotation are identified, which will facilitate the search for axes of rotation.

4.1 Point visibility

Point visibility is the simplest form of visibility and its relation with axes of rotation will be used to determine further relations of Arc and Cone Visibility to axes of rotation. In a milling machine setup, a feasible cutting direction to a surface patch can be geometrically interpreted as a Point Visibility to that surface patch and is denoted as a point on a unit sphere. Given this cutting direction, the X axis of the machine, which aligns with the axis of rotation in a typical 4-axis indexed setup, can be oriented (in the x-y plane) anywhere as a feasible orientation as long as it is perpendicular to the cutting direction (Fig. 3a). Collectively, all feasible axes of rotation perpendicular to the cutting direction form a great circle on the unit sphere (Fig. 3b). **Therefore the first relation can be stated as:** *Each Point Visibility corresponds to one great circle of axes of rotation.*

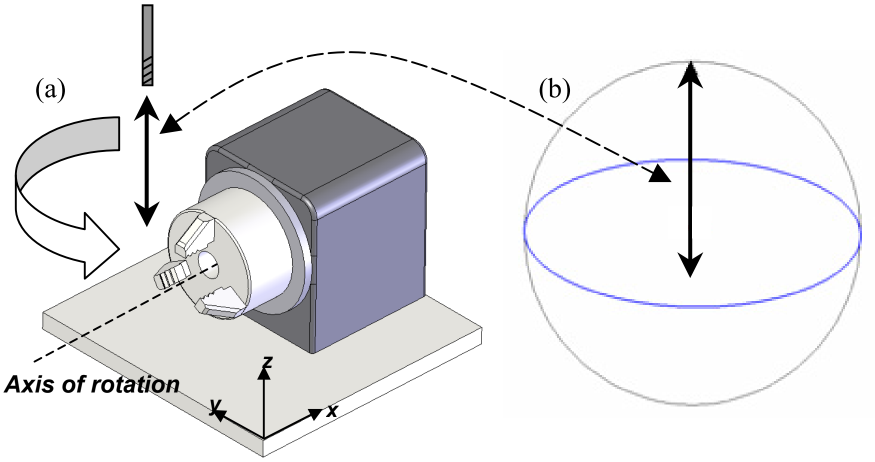


Fig. 3 Point Visibility and axes of rotation; (a) Point Visibility for one tool position and orientation on a 4th axis setup and (b) corresponding great circle

4.2 Arc visibility

The *Arc Visibility* considered in this paper is limited to great arcs on a unit sphere. Since an arc of visibility is a collection of Point Visibility, the axes of rotation of the arc are the collection of all axes of rotation corresponding to each visibility point constituting that great arc. A great arc on a unit sphere can be perceived as the trajectory of a visibility point moving from one end of the arc to the other. The great circle, representing feasible axes of rotation for each point on the arc, maintains a perpendicular relation with the Point Visibility on the unit sphere. As a visibility point moves on a great arc on the unit sphere, the great circle perpendicular to it will rotate as well. The set of all feasible axes of rotation corresponding to an Arc of Visibility is the spherical area that the great circle sweeps on the unit sphere, which is the area between two great circles corresponding to the two end points of the great arc (shaded area in Fig. 4). **Therefore the second relation can be stated as:** *Each great arc of visibility generates an area between two great circles, corresponding to the end points of the great arc for feasible axes of rotation.*

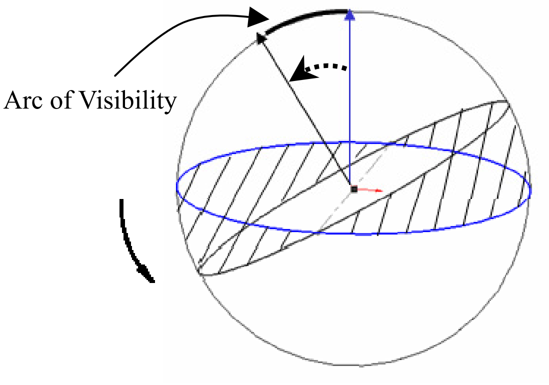


Fig.4 Arc of Visibility and shaded region between two great circles representing feasible axes of rotation

4.3 Cone visibility

In most cases, visibility is given as a Visibility Cone, which maps onto the unit sphere as a region or a number of separate regions. Unlike an Arc of Visibility that can be represented by a trajectory of Point Visibility, a Visibility Cone usually cannot be represented into a set of equal-length great arcs. Instead, a Visibility Cone can be discretized into a series of great arcs truncated from great circles by the Visibility Cone boundary (Fig. 5). If axes of rotation for each of these great arcs are obtained, then their union gives the feasible axes of rotation for the Visibility Cone. **Therefore the third relation can be stated as:** *A Visibility Cone corresponds to the union of axes of rotation of the great circles used to approximate the Visibility Cone.*

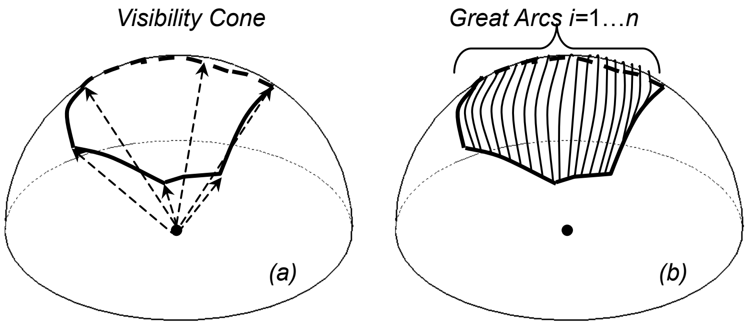


Fig.5 Visibility Cone discretized to visibility arcs; (a) Visibility Cone (b) corresponding visibility arcs used to approximate surface on the unit sphere

The three relations stated above are used to map visibility information to axes of rotation on a unit sphere. However the visibility calculation is only neccessary for facets located on concave regions, since a facet on a convex region has visibility up to a hemisphere; which includes an infinite number of half great circles. From the second relation in 4.2, a half great circle of visibility corresponds to the area between the two great circles corresponding to the two end points of the half circle, which is actually a complete unit sphere surface (Fig. 6). Therefore any axis can serve as a feasible axis of rotation for convex facets.

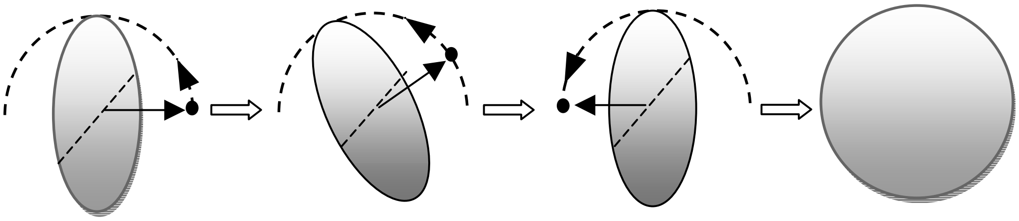


Fig. 6 Half great circle corresponds to a unit sphere for axes of rotation

5. Computing Axes of Rotation

Practically, the visibility of a polygonal facet is given as a 3-D Visibility Cone, which maps on the unit sphere as a region or a number of separate regions. In this research the 3-D visibility regions for each facet *i* (*i*=1…*m*) on the unit sphere are not exact, but are approximated by 2-D great arcs (Fig. 5). This approximation is termed a *rasterization*, by which 3-D Visibility Cones for each facet F_i (i=1…m) are approximated by a raster of great arcs (Great Arc GA_{ij},j=1…. n). Axes of rotation can then be obtained from these rasterized great arcs using the relation stated in section 4.2.

5.1 Rasterization of the Visibility Cone

From section 4.3, a Visibility Cone can be approximated by a set of great arcs contained within the boundary of the Visibility Cone. The rasterization for a Visibility Cone can be executed in an infinite number of ways, depending on the normal directions of the great arcs that cover the visibility region. The representation for a great arc GA_{ij} can be written in a five-variable format $\left[\vec{V}_{xij},\vec{V}_{yij},\vec{V}_{zij},S_{ij},E_{ij}\right]$, where $\vec{V}_{xij},\vec{V}_{yij},\vec{V}_{zij}$ are the components of the great arc GA_{ij}'s normal vector $\vec{V}$ along X,Y and Z axes; S_{ij},E_{ij} are the two ending points where the great arc GA_{ij} intersects the visibility boundary of facet F_i (Fig. 7). The two ending points S_{ij},E_{ij} of the great arc GA_{ij} depend on the normal vector $\vec{V}=(\vec{V}_{xij},\vec{V}_{yij},\vec{V}_{zij})$, as the great arc GA_{ij} at different location when the normal vector changes.

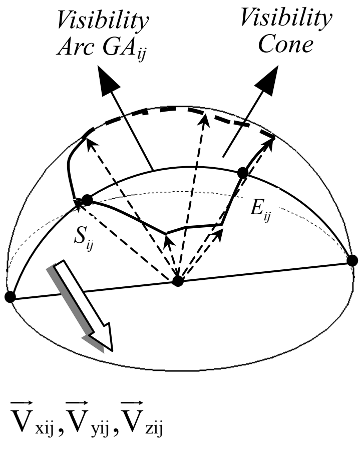


Fig. 7 A visibility arc intersecting with a Visibility Cone

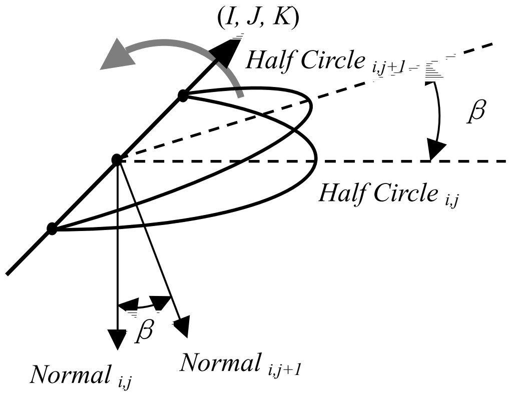


Fig. 8 Half circles pivoting along a common axis

Given a limited number of great arcs to rasterize a Visibility Cone, it is an interesting problem to find an optimal set of great arcs to cover the Visibility Cone. The challenge would be to ensure that the axes of rotation computed from these great arcs best approximate the axes of rotation of the Visibility Cone; however, it is outside the scope of this paper.

In this research the authors used a straightforward method by which all great arcs used to approximate a Visibility Cone are part of great half circles pivoting along a common axis with a fixed interval angle, β, among them (Fig. 8). The use of great arcs about a common pivoting axis provides a vertical relation between great arcs' normal vectors and the common axis, as below:

$(\vec{V}_{xij},\vec{V}_{yij},\vec{V}_{zij})\bullet(I,J,K)=\vec{V}_{xij}\times I+\vec{V}_{yij}\times J+\vec{V}_{zij}\times K=0$ ------------------------------(1)

where *I, J* and *K* are the directional components of the common pivoting axis.

In addition to this relationship, the unit sphere where $(\vec{V}_{xij},\vec{V}_{yij},\vec{V}_{zij})$ locates makes the magnitude of vector $\vec{V}$ to be 1, as below:

$\vec{V}_{xij}^{\,2}+\vec{V}_{yij}^{\,2}+\vec{V}_{zij}^{\,2}=1$ --(2)

Conditions (1) and (2) reduce the five-variable description of Great Arc$_{ij}$ $\left[\vec{V}_{xij},\vec{V}_{yij},\vec{V}_{zij},S_{ij},E_{ij}\right]$ to a three variable description, $\left[\vec{V}_{xij},F(\vec{V}_{xij}),G(\vec{V}_{xij}),S_{ij},E_{ij}\right]$, where $F(\vec{V}_{xij})$ and $G(\vec{V}_{xij})$ are functions of $\vec{V}_{xij}$. To further simplify this three variable representation, the pivoting axis is chosen to be one of the coordinate axes, (the Y axis in this case), by which $F(\vec{V}_{xij})$=0 and $G(\vec{V}_{xij})=\sqrt{1-\vec{V}_{xij}^{\,2}}$. This makes the representation of a great arc to be $\left[\vec{V}_{xij},0,\sqrt{1-\vec{V}_{xij}^{\,2}},S_{ij},E_{ij}\right]$. Through manipulation of the first three items, $\left[\vec{V}_{xij},0,\sqrt{1-\vec{V}_{xij}^{\,2}},S_{ij},E_{ij}\right]$ is reduced to $\left[\gamma_j,S_{ij},E_{ij}\right]$, where γ_j is the yawning angle of those pivoting great half circles with respect to the horizon. $\gamma_j=a\cos(\vec{V}_{xij})$, if γ_j is less than 180 degrees; and $\gamma_j=360^o-a\cos(\vec{V}_{xij})$ if γ_j is greater than 180 degrees.

With a fixed interval angle β among half circles, angle γ_j can be calculated sequentially as $\gamma_j = (j-1)\times\beta$, where *j* is the sequential number of the great half circle (Fig. 9). For example, in this study β is given as 0.5 degree, therefore the first great half circle aligning with positive X axis has *j*=1 and $\gamma_1 = 0^o$; the great half circle aligning with negative X axis has *j*=361 and $\gamma_{361} = 180^o$. Since visibility is computed for all facets comprising a polygonal model, the pivoting axis is made to be Y axis for rasterizing the Visibility Cone for each facet. Therefore a unit sphere rasterized with great half circles pivoting around Y axis will satisfy all facets of the CAD model. The Visibility Cone of each facet will be mapped upon this unit sphere by computing the two intersection points S_{ij} and E_{ij} of each great circle with the Visibility Cone boundary. Upon obtaining the two intersection points S_{ij} and E_{ij}, a great visibility arc is then ready for mapping to axes of rotation using the relation stated in section 4.2.

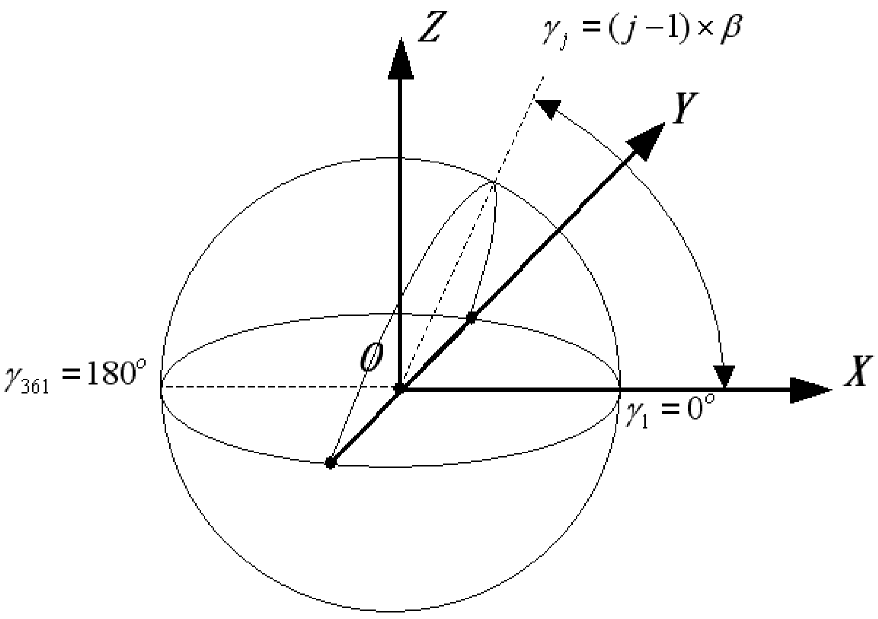


Fig. 9 Great half circle angle γ

5.2 Discretization of great half circles

In addition to discretizing a unit sphere into a raster of great half circles pivoting around the Y axis with an interval angle *β,* each of these great half circles is further discretized into a circular array of points, with an interval angle *α* on the half circle plane (Fig. 10a). In this manner, the unit sphere is discretized into a spherical grid of points (Fig. 10b). This eases both computational and implementation effort and avoids numerical complexity. Both the areas of the Visibility Cone and axes of rotation can be represented using discretized points. The number of points on a unit sphere is:

Num (points) $= M \times N$, where M=integer $(\dfrac{360}{\beta})$ and N=integer $(\dfrac{360}{\alpha})$ ------ (3)

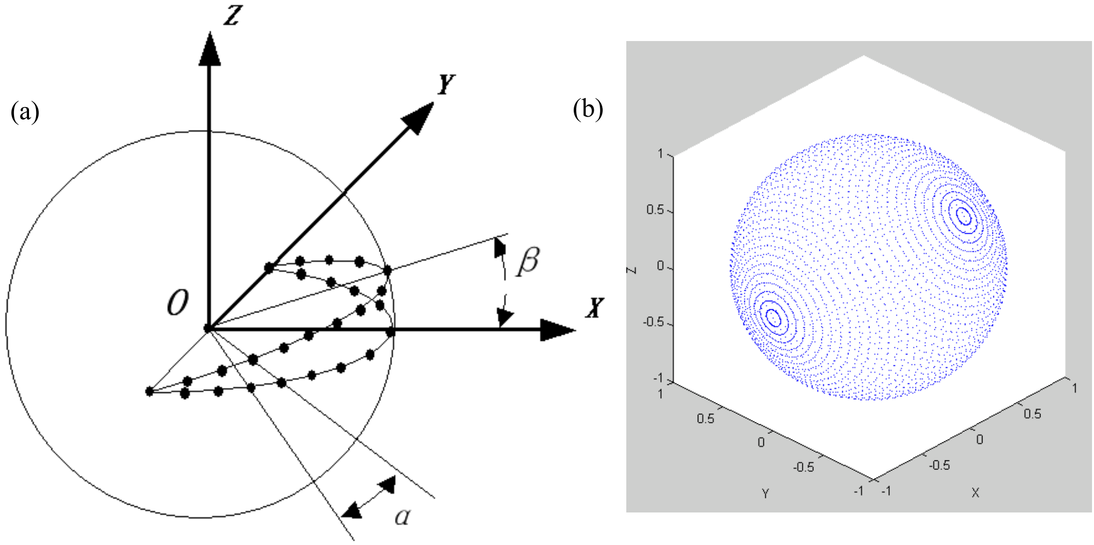


Fig. 10 Discretization of a Unit Sphere; (a) Circular point array on a great half circle, (b) Sphere represented by a grid of points

5.3 Steps in computing the axes of rotation

Using the rasterization process above and the relations in section 4, the axes of rotation for a CAD model can be computed. Each 2-D visibility great arc (GA_{ij}) corresponds to an area for feasible axes of rotation on the unit sphere (AR_{ij} is used to represent the area of axes of rotation for visibility GA_{ij}, AR_{ij}, j=1…. n). The union of these areas (AR_{ij}, j=1…. n)

of all great arcs (Great Arc GA_{ij}, i=1…. n) is the set of feasible axes of rotation for facet F_i on the unit sphere. Therefore for a polygonal facet, the process for finding feasible axes of rotation begin by computing visibility and then mapping Visibility Cones onto a unit sphere and rasterizing with great arcs. Then the corresponding axes of rotation areas of each great arc are grouped as a union to represent all feasible axes of rotation for that facet. This process repeats for all facets (i=1…m) and any axes of rotation shared commonly by all facets are the globally feasible axes of rotation for the part. This process is summarized below in the following steps.

Steps:

1: Compute Visibility Cone for facet i (i=1…m)

2: Rasterize Visibility Cones by a set of great arcs Arc $_{i,j}$ (j=1…n)

3: Find possible axes of rotation from great arcs

$$GA_{ij} \longrightarrow \mathrm{AR}_{i,j}\ (j=1\ldots..n)$$

$$\mathrm{AR}_i = \mathrm{AR}_{i,1} \cup \mathrm{AR}_{i,2} \cup \ldots \cup \mathrm{AR}_{i,n}$$

4: Continue for all facets

5: Global set of axes of rotation becomes:

$$\mathrm{AR} = \mathrm{AR}_1 \cap \mathrm{AR}_2 \cap \ldots \cap \mathrm{AR}_m$$

6. Implementation

The proposed method for computing feasible axes of rotation was implemented in C programming language on a Pentium IV, 3.06 GHz PC running Windows XP. Several example parts were tested using the software; many of which have been subsequently machined on a CNC mill in the laboratory. In this section, the authors provided three increasingly complex models, which either do or do not contain feasible axes of rotation. A more complex model is illustrated to provide a detailed comparison of the difference between using feature-based methods and those proposed in this paper.

The axes of rotation found by the software are presented as points on the unit sphere, as described above. However, since two antipodal points on a unit sphere are actually denoting one axis of rotation, only a hemisphere is rasterized with great arcs. This is due to the fact that for antipodal points, any point on one hemisphere can be diametrically mapped on to the other hemisphere. In this study, the interval angle β between any two adjacent great half circles to rasterize a hemisphere was set to 0.5 degrees and each half great circle was discretized into points with an interval angle α of 0.5 degree (Fig. 11). Using this rasterization, a hemisphere is approximated by 361 great half circles, each of which is discretized into 361 points. While the relation in section 4.2 was implemented, for any point falling into the spherical area denoting feasible axes of rotation for a facet, the value for that point will be added by one. In other words, each orientation on the sphere is given a value representing the number of facets that it would satisfy. Therefore, if a point's value is equal to the number of facets in the model, then a globally feasible axis has been found. Any given part model could have one, several, or no feasible axis that would satisfy all facets. However, if no single axis of rotation exists, one could alternately select the axis with the *most* facet coverage. Moreover, the coverage values for each axis could be used in a more elaborate decision system, for example, if the user was not only interested in visibility from an axis, but also in minimizing stock diameters required to contain the part, tool length requirements, stock length along the axis, etc.

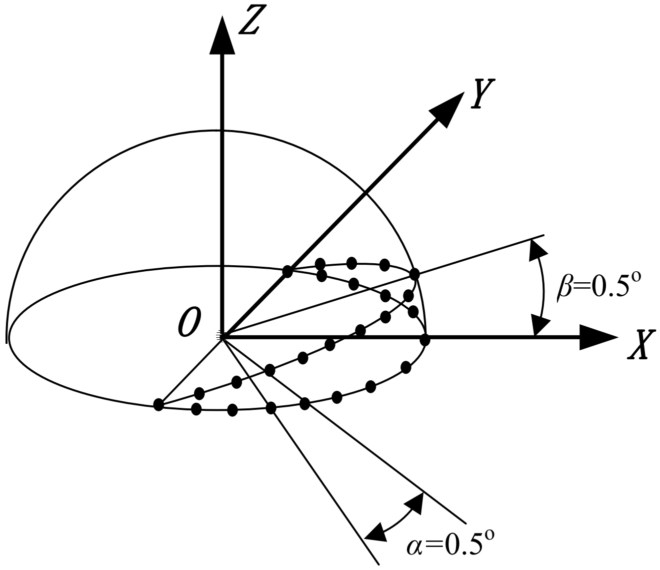
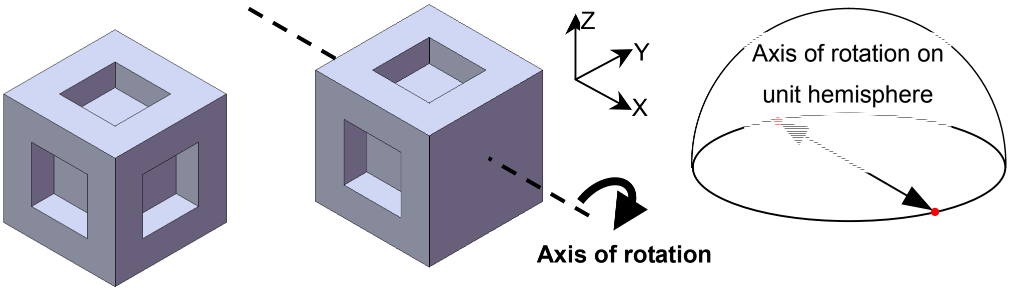


Fig. 11 Rasterization process with α=0.5° and β=0.5°



(a) Three orthogonal square pockets, with no axis of rotation

(b) Two orthogonal square pockets, with corresponding axis of rotation

Fig. 12 Example One: Orthogonal square pockets on a prismatic part

The first example illustrated in Fig. 12 yields the obvious conclusion that at most two mutually orthogonal square pockets can be machined using a 4-axis indexed setup; however, three orthogonal square pockets result in no feasible axis of rotation. The part with two orthogonal square pockets has two antipodal points on the equator of a hemisphere as one feasible axis of rotation (Fig. 12b). This result could inform a designer about the actual manufacturability of a prismatic part with pockets on its sides. If there are more than two orthogonal pockets, the part will need more than one axis of rotation and therefore at least two machining setups if employing a 4-axis indexed machine. Example 2 is a simple block with a cylindrical thru-hole, which is used to compare feature-based versus feature-free analysis (Fig. 13a). Using a feature-based approach; the cylindrical thru-hole will have only single point visibility on top of a hemisphere and therefore one great circle as feasible axes of rotation, represented by the equator of the hemisphere (Fig. 13b). To illustrate the feature-free approach, an STL model was used to compute axes of rotation for each facet, which yields the result shown in Fig. 13c. The feasible axes of rotation are more than just one great circle as in the feature-based approach; rather, they form a spherical band on the hemisphere with a width of 15.5^o. This suggests that a feature-free approach can lead to more solution sets for machining process planning. In this example, the traditional feature-based approach restricts the axis of rotation to be perpendicular to the hole's visibility direction; whereas a feature-free approach allows the part model to tilt up to 15.5^o.

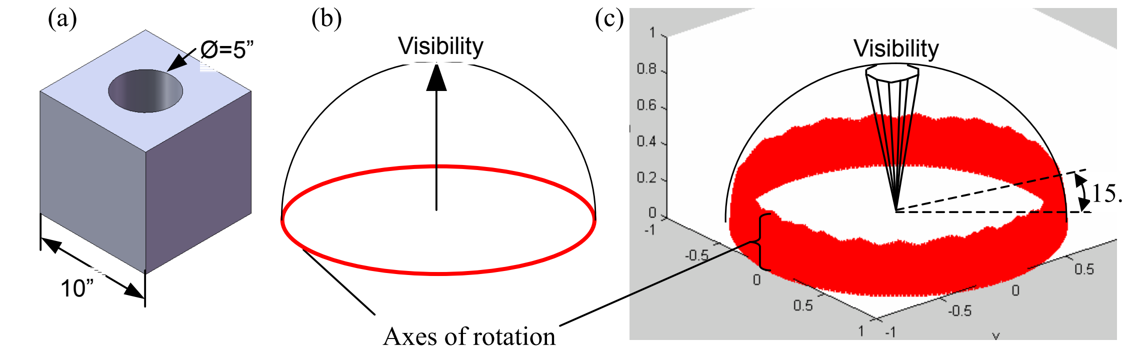


Fig. 13 Example Part (a) Square cube with thru-hole, (b) Feature-based approach to Axes yield the equator on a unit sphere, (c) Feature-free approach expands to a band about the equator

This simple example can be illustrated as a part setup on an indexed milling machine using opposing indexer and tailstock chucks, as shown in Fig. 14. In the feature-based approach, the part model would need to be machined such that the hole is perpendicular to the rotary axis (Fig. 14a); however, the feature-free analysis indicates that the part could be tipped from vertical (Fig. 14b).

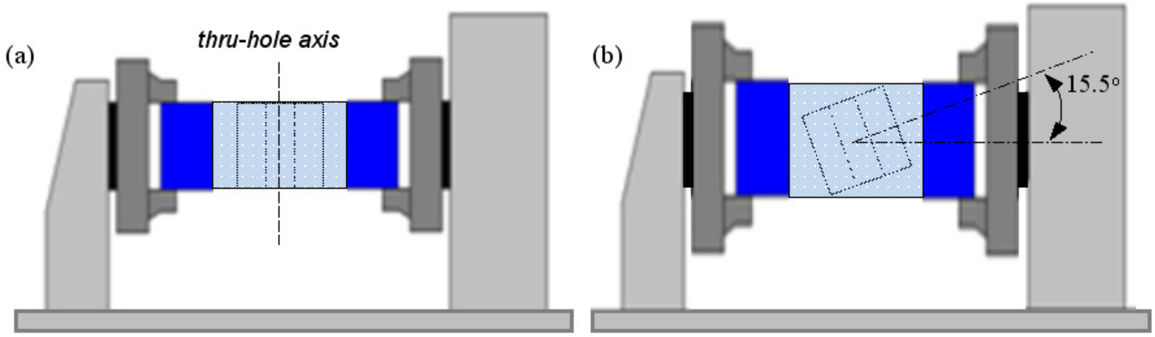


Fig. 14 Setup orientations on a 4-axis indexed machine, (a) A single feature-based solution, (b) One of several feature-free solution results

Of course, a practical machinist would perhaps *never* set up a part in the manner shown in Fig. 14b. Moreover, the hole may not even be manufacturable in this setup unless very small diameter tools or a different cutting process was available (e.g.; Wire EDM). Further accessibility analysis is required; this current work is only based on visibility. So, one may argue that the analysis results are not of much use; however, this is only a simple example. The benefits of the analysis may be more readily understood if, instead of a simple hole, the “feature” was the *Acetabulum* of a human pelvis (the “hip socket”). In that case, there are numerous complex bone surfaces forcing the setup selection down a very narrow solution path. Allowing more options to access each bone surface may make it possible to find at least one solution; if not a “best” solution may be derived that will allow access to the most surfaces in one setup.

As a more practical example, an industrial linkage represented with an STL file with 2196 facets was tested (Fig. 15a). The axis of rotation software was executed and the results indicated that no axis of rotation exists for the entire linkage. Though no axis of rotation was found to cover all 2196 facets, there are spherical regions that have been identified to satisfy over 99% of the facets on the linkage model. Three spherical areas (clusters 1, 2, and 3), that satisfy over 99% of facets, are displayed in Fig. 15b and their details are provided in Table 1**.** Each cluster is one or a group of axes listed with the number of axes and the number of facets that they make visible through rotations. The axes indicated on the hemisphere indicate that rotations at or around the x- or y-axis of the part will work for the most surfaces, which should be intuitive by observation. It has been further verified by machining the part in the lab using a 3-axis Fadal VMC using a programmable 4th axis indexer.

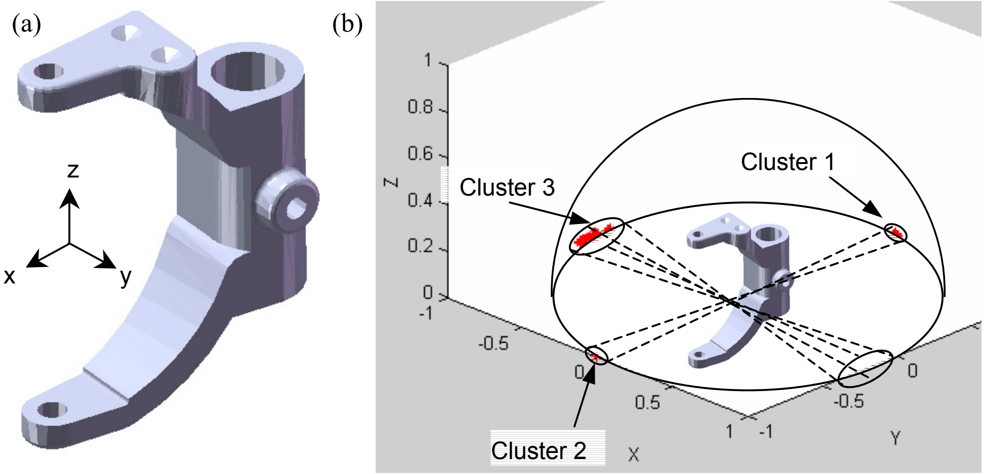


Fig. 15 Example industrial component, (a) STL model of the part, (b) Axes of rotation results in clusters of axes illustrated on the hemisphere, w.r.t. the part model

Table 1 – Axes of rotation results for the Linkage model

Cluster Number	Number of Axes	Number of Facets Covered	Percentage of Facets Covered
1	120	1 axis covers 2178 facets	99.18 %
		25 axes cover 2177 facets	99.13 %
		94 axes cover 2176 facets	99.09 %
2	1	1 axes covers 2176 facets	99.09 %
3	18	2 axes cover 2178 facets	99.18 %
		4 axes cover 2177 facets	99.13 %
		12 axes cover 2176 facets	99.09 %

One will note that this model *should* have at least one axis that satisfies all facets in the model. Recall, this is a facetted model created in the form of an STL model. Tessellation processes are inherently approximate, for example, the "straight" surfaces in a thru-hole become slightly tipped triangular facets. Moreover, these representations are prone to errors, from degenerate facets, to flipped normals, to numerical round-off errors. Hence, a further challenge is to balance the problem of needing a fine resolution polygonal model for accurate calculations, but without egregious increases in computation time. The computational complexity of computing axes for a model with n-facets is $O(n^2)$. In these three examples; the block with thru-hole, block with prismatic pockets and the Linkage had computation times of 12, 27 and 1257 seconds, respectively. There exist many opportunities to improve the method, and perhaps reduce computation times; however, that was outside of the scope of this paper. In addition, one could pre-process the polygonal models to avoid propagating tessellation errors into the analysis. However, the authors chose to present these examples as-is, to reveal the limitations and challenges of conducting analysis using a feature-free approach. Regardless, these three example parts illustrated the effectiveness of the axis algorithm and provide support for the use of feature-free analysis in certain process planning problems.

7. Conclusion

In this paper a feature-free approach to determine feasible axes of rotation for 4-axis indexed processes is presented. Without relying on feature recognition, the approach can calculate setup orientations for arbitrarily shaped parts. It begins with visibility computation of facets comprising the geometric model. Next, the Visibility Cone is rasterized into great arcs and then each visibility arc is mapped onto axes of rotation using their geometric relation. This approach can also provide an expanded space for searching optimal axes of rotation in terms of minimum stock size, machining time, surface roughness, etc., regardless of whether a globally feasible axis exists. Information about potentially feasible axes of rotation also provides valuable feedback to designers by presenting the approximate manufacturability of a geometric model on a specific machine setup. Therefore the goals of this axis of rotation approach were twofold: (1) enable process planning for non-feature-based objects, and (2) provide the fundamental starting point for a potentially new Design for Manufacture (DFM) tool. As future research, the authors would like to focus on providing re-design suggestions to the designer if an axis of rotation does not exist for a geometric model. It is proposed that, knowing the limited feasible set of axes, a second map could be generated; one that illustrates the geometry on the part that is most severely limiting the choice of setups. This may allow a designer to focus efforts on changes to problem areas of the part that are forcing the use of expensive multi-axis machines and setups, when fewer may be possible.

References

[Allada and Anand 1995] Allada,V. and Anand, S., 1995, “Feature-Based Modelling Approaches For Integrated Manufacturing:State-Of-The-Art Survey And Future Research Directions,” International Journal of Computer Integrated Manufacturing, 8, pp. 411- 440.

[Allada and Anand 1996] Allada, V. and Anand, S., 1996, “Machine Understanding Of Manufacturing Features,” International Journal of Production Research, 34(7), pp.1791–1819.

[Balasubramaniam et. al. 2000] Balasubramaniam M, Laxmiprasad P, Sarma S, Shaikh Z., 2000, "Generating5-Axis NC Roughing Paths Directly From A Tessellated Representation," Computer-Aided Design; 32(4) pp.261–77.

[Chen et. al. 1993] Chen, L. L., Chou, S. Y., and Woo, T. C., 1993, "Separating and Intersecting Spherical Polygons: Computing Machinability on Three-, Four-, and Five-Axis Numerically Controlled Machines," ACM Trans. Graphics, 12_4_, pp. 305–326.

[Chu and Gadh 1996] Chu, Chi-Cheng Peter; Gadh, Rajit, 1996, "Feature-Based Approach For Setup Minimization Of Process Design From Product Design," Computer Aided Design, v 28, n 5, pp. 321-332

[Demey et. al. 1996] Demey, S., Brussel, H. V., and Derache, H., 1996, "Determining Setups for Mechanical Workpieces," Robotics and Computer-Integrated Manufacturing, V. 12 No.2, pp. 195–205.

[Dhaliwal et. al. 2003] Dhaliwal S, Gupta SK, Huang J, Priyadarshi A., 2003, "Algorithms For Computing Global Accessibility Cones," ASME Journal of Computing and Information Science in Engineering, 3(3) pp.200–9.

[Ferreira and Liu 1988] Ferreira, P. M., and Liu, C. R., 1988, "Generation of Workpiece Orientations for Machining Using A Rule-Based System," Robotics and Computer-Integrated Manufacturing, V. 4 No. 3/4, pp. 545–555.

[Frank et. al. 2004] Frank, M.C., Wysk, R.A., and Joshi, S.B., 2004, "Rapid Planning for CNC Machining – A New Approach to Rapid Prototyping", Journal of Manufacturing Systems, SME, v23, No. 3, pp. 242-255

[Gologlu 2004] Gologlu, Cevdet, 2004, "Machine Capability And Fixturing Constraints-Imposed Automatic Machining Setups Generation," Journal of Materials Processing Technology, v 148, n 1, pp. 83-92

[Gan et. al. 1994] Gan, J. G., Woo, T. C., and Tang, K., 1994, "Spherical Maps: Their Construction, Properties, and Approximation," Journal of Mechanical Design, 116, pp. 357–363.

[Haghpassand and Oliver 1995] Haghpassand, K., and Oliver, J. H., 1995, "Computational Geometry for Optimal Workpiece Orientation," Journal of Mechanical Design, 117_2A_, pp. 329–335.

[Li and Frank 2007] Li, Y. and Frank, M.C., 2007, “Computing Non-Visibility of Convex Polygonal Facets on the Surface of a Polyhedral CAD Model”, Computer Aided Design, Vol. 39, No. 9, pp. 732-744.

[Miao et. al. 2002] Miao, Huikang K.; Sridharan, Nandakumar; Shah, Jami J., 2002, “CAD-CAM Integration Using Machining Features,” International Journal of Computer Integrated Manufacturing, v 15, n 4, pp. 296-318.

[Ong and Nee 1996] Ong, S.K.; Nee, A.Y.C., 1996, “Fuzzy-Set-Based Approach For Concurrent Constraint Setup Planning,” Journal of Intelligent Manufacturing, v 7, n 2, pp. 107-120.

[Ong et. al. 2002] Ong, S. K., Ding, J., and Nee, A. Y. C., 2002, “Hybrid GA and SA Dynamic Setup Planning Optimization,” International Journal of Production Research, 40_18_, pp. 4697–4719.

[Salomons et. al. 1993] Salomons, O.W., van Houten, F. J. A.M. and Kals, H. J. J., 1993, “Review Of Research In Feature-Based Design,” Journal of Manufacturing Systems, 12, pp.113-132

[Suh and Kang 1995] Suh, S. H., and Kang, J. K., 1995, “Process Planning for Multi-axis NC Machining of Free Surfaces,” International Journal of Production Research, 33_10_, pp. 2723–2738.

[Tang et. al. 1992] Tang, K., Woo, T., and Gan, J., 1992, “Maximum Intersection of Spherical Polygons and Workpiece Orientation for 4- and 5-Axis Machining,” Journal of Mechanical Design, 114, pp. 477–485.

[Yen et. al. 2001] Yen, David W.; Zhang, Y.; Hu, W.; Rong, Y., 2001, “Graph-Based Setup Planning And Tolerance Decomposition For Computer-Aided Fixture Design,” International Journal of Production Research, v 39, n 14, pp. 3109-3126.

[Zulkifli and Meeran 1999] Zulkifli, A. H. and Meeran, S., 1999, “Feature Patterns In Recognizing Non-Interacting And Interacting Primitive, Circular And Slanting Features Using A Neural Network,” International Journal of Production Research, 37(13), pp.3063–3100

CHAPTER 5. MACHINABILITY ANALYSIS FOR 3-AXIS FLAT END MILLING

A paper published in *ASME Journal of Manufacturing Science and Engineering*, v 128, n 2, 2006

Ye Li and Matthew C. Frank

Abstract

This paper presents a method for geometric machinability analysis. The implementation of the strategy determines the machinability of a part being processed using a plurality of 3-axis machining operations about a single axis of rotation for setup orientations. Slice file geometry from a Stereolithography (STL) model is used to map machinable ranges to each of the line segments comprising the polygonal chains of each slice. The slices are taken orthogonal to the axis of rotation, hence, both 2D and 3D machinability analysis is calculated for perpendicular and oblique tool orientations, respectively. This machinability approach expands upon earlier work on 2D visibility analysis for the rapid manufacturing and prototyping of components using CNC machining.

1. Introduction

Machinability analysis is taking an increasingly important role as complex surfaces are used in the design of a wide variety of parts. Current Computer Aided Manufacturing (CAM) software is readily capable of generating toolpaths given a set of surfaces of a part and a cutting orientation (3-axis machining). However, determining the setup orientation can be difficult and moreover, it may be very challenging to determine if the part can be created using machining at all. An appropriate setup orientation can guarantee an effective cutting of the surface, while an inappropriate one will leave too much material in certain regions. The advancement of 5-axis computer numerically controlled (CNC) milling machines seems to alleviate this situation; however, often the cost and/or difficulty of programming a 5-axis machine have limited their widespread use. 3-axis machines, as economical and technologically mature pieces of equipment, have been paid special attention with respect to complex surface machining if assisted with multi-setup devices (e.g. a programmable indexer). Suh and Lee **[1]** used a 3-axis machine with a rotary-tilt type indexer to provide an alternative to 5-axis ball end milling. Suh et al. **[2]** provided a theoretic basis for machining with additional axes. Recently Frank et al. **[3]** employed a 3-axis milling center with a 4th axis indexer as an effective rapid prototyping machine. End mills have been shown to offer a better match to the part surface geometry, a higher material removal rate, and a longer tool life compared to ball-mills **[4].** Ip and Loftus **[5]** demonstrated the competency of an inclined end mill machining strategy on 3-axis machines in producing low curvature surfaces. However, to machine a surface with large curvature variation, it is necessary to determine a set of machining orientations and carry out multiple 3-axis machining operations in a sequential manner with respect to each of those orientations. Therefore an effective machinability analysis is of critical importance to the successful implementation of multiple orientation 3-axis machining for creating complex parts.

Many researchers have studied machinability analysis and its closely related workpiece setup problem. Most of the approaches are based on visibility, which is essentially line-of-light accessibility. Su and Mukerjee **[6]** presented a method to determine machinability of polyhedral objects. A convex enclosing object is constructed to make each face of the part orthogonally visible to the planes of the enclosing object. The part is then considered to be machinable from the normal-vector directions of the enclosing object planes. Later, computational geometry on the sphere was utilized to analyze visibility by Chen and Woo **[7]** who performed pioneering work on computational geometry algorithms that could be used for determining workpiece set-up and machine selection. Tang et al. **[8]** formulated the problem of workpiece orientation as finding the maximum intersection of spherical polygons. Gan et al. **[9]** discussed the properties and construction of spherical maps and presented an efficient way to compute a visibility map from a Gaussian map. Chen et al. **[10]** partitioned the sphere by spherically convex polygons to solve the geometric problem of determining an optimal workpiece orientation for 3-, 4- and 5-axis ball end milling. A visibility map is generated by using the normal vectors of a specified *portion* of the surface of a part, therefore it cannot guarantee global accessibility. Yang et al. **[11]** computed visibility cones based on convex hull analysis, instead of relying on visibility maps. Yin et al. **[12]** defined complete visibility and partial visibility, and presented a C-space based method for computing visibility cones. A sculptured surface is approximated by its convex hull **[11]** and the spherical algorithms **[7,13]** are used in the approach of Yin **[12]**. The convex hull may in some cases have a significant deviation from the true surface. Suh and Kang **[14]** constructed a binary spherical map to compute the point visibility cone in order to algebraically solve machining configuration problems, including workpiece setup orientation. The part surface is decomposed into triangular patches. An occupancy test of the patches is conducted on a triangular-represented unit sphere to generate global visibility. Dhaliwal et al **[15]** presented a similar approach for computing global accessibility cones for polyhedral objects, but with exact mathematical conditions and algorithms. Balasubramaniam et al. **[16]** analyzed visibility by using computer hardware (graphics cards). Frank et al. **[17]** analyzed 2D global visibility on STL slices and searched the necessary machining orientations for 4th axis indexable machining by executing a *Greedy* search algorithm. All these visibility-based approaches determine the necessary condition for machinability; however, they ignore tool geometry and therefore true accessibility (machinability) is not guaranteed. Figure 1 shows that the accessibility cone (α,β) based on line-of light visibility cannot guarantee the true accessibility using a sized tool in machining a segment $\overline{ij}$.

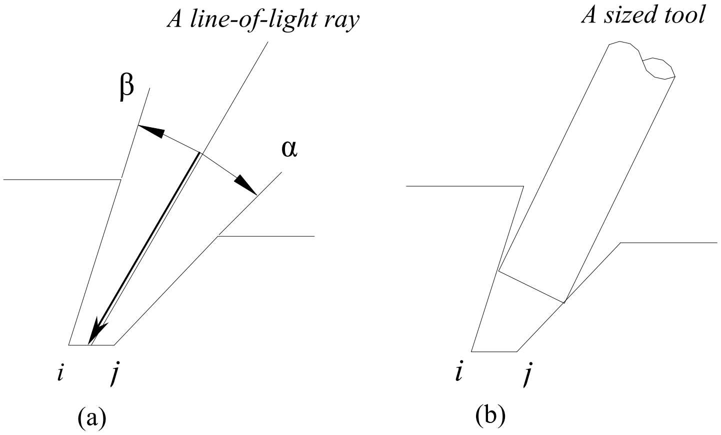
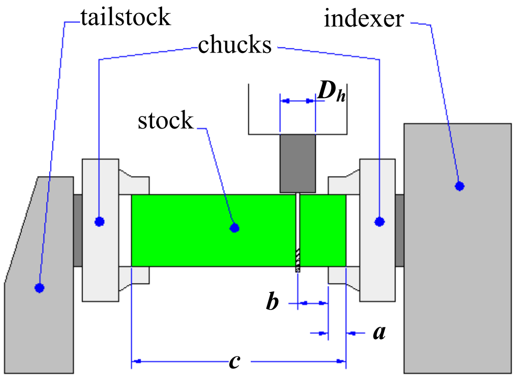


Fig. 1 Accessibility based on light ray and a sized tool

Su and Mukerjee **[6]** took into account the cutter information by constructing a new part model through offsetting the original part surface by the amount of the cutter radius. Machinability was further guaranteed by checking the topology of this offset part surface. This method is effective for the machinability analysis of a ball end cutter, but not for that of a flat end cutter, because the effective radius of a flat end cutter is variable with the change of tool tilting angle. Haghpassand and Oliver **[18]** and Radzevich and Goodman **[19]** considered both part surface and tool geometry. However, tool size was not taken into account due to the fact that Gaussian mapping does not convey any size information of the part surface and/or the tool. Balasubramaniam et al. **[16,20]** verified tool posture from visibility results by collision detection before interpolating the toolpath for 5-axis machining. Over the past years, feature-based technologies have been an active field among the manufacturing research community. Regli**[21],** Regli et al **[22],** and Gupta and Nau **[23]** discussed feature accessibility and checked it by calculating the feature accessibility volume and testing the intersection of the feature accessibility volume with the part. Gupta and Nau **[23]** recognized all machining operations that could machine the part, generated operation plans, and checked and rated different plans according to design needs. A comprehensive survey paper on manufacturability by Gupta et al. **[24]** reviewed representative feature-based manufacturability evaluation systems. Recently, Shen and Shah **[25]** checked feature accessibility by classifying the feature faces and analyzing the degree of freedom between the removal volume and the workpiece. The MEDIATOR system reported by Gaines et al. **[26]** used the knowledge of manufacturing equipment to identify manufacturing features on a part model. Accessibility is examined by testing the intersection of removal volumes with the part. Faraj **[27]** discussed the accessibility of both 2.5 D positive and negative features. Other researchers presented featured-based approaches to determine workpiece setups **[28-31]**.

Although feature-based approaches are capable tools to handle feature-based design, they cannot lend themselves to free-form surfaces where definable features may not exist. In addition, feature-based approaches suggest that all the geometric elements comprising of a feature are treated together as an entity. This actually imposes a constraint to the analysis of a part model. For example, it might be feasible to machine a portion of a part feature in one orientation and then finish the remaining surfaces of the feature in one or more successive orientations. The current problem that this paper addresses is based on a rapid machining strategy proposed by Frank et al. **[3]** whereby a part is machined with a plurality of 3-axis machining operations from multiple setup orientations about a single axis of rotation.

The strategy is implemented on a 3-axis CNC milling machine with a 4th axis indexer (Fig. 2). Round stock material is fixed between two opposing chucks and rotated between operations using the indexer. For each orientation, all visible surfaces are machined using simple layer-based toolpath planning. By setting the collision offset (b) (shown in the Fig. 2) on each side of the workpiece, the implementation of rapid machining can avoid the risk of collision between tool holders and the holding chucks. The diameter of largest tool (Dtmax) used to calculate the collision offset (b) makes the setting of collision offset for each new part unnecessary. The *feature-free* nature of this method suggests that it is unnecessary to have any surface be completely machined in any particular orientation. The goal is to simply machine ALL surfaces after ALL orientations have been completed. The number of rotations required to machine a model is dependent on its geometric complexity. Figure 3 illustrates the process steps for creating a typical complex part using this strategy.



Stock length: $\boldsymbol{c} = L_p + 2a + 2b$

where: L_p = Part length

a = Clamping length

b = Collision offset (x) = $.5D_h + .5D_t max$

D_h= Diameter of tool holder

$D_t max$= Diameter of largest tool

Fig. 2 Setup for rapid machining

Currently, the necessary cutting orientations are determined by 2D visibility maps with tool access restricted to directions orthogonal to the rotation axis. Cross sectional slices of the geometry from an STL model are used for 2D visibility mapping. The visibility of those slices approximates the visibility of the entire surface of the part along the axis of rotation since the slices are generated orthogonal to that axis. The above literature review suggests that existing approaches to machinability cannot calculate the set of orientations for setups such that one can machine ALL *machinable* surfaces after ALL orientations, because 1) either 2D or 3D visibility cones employed by the visibility-based approaches convey no size information of the tool and workpiece and therefore cannot guarantee true accessibility; or 2) the feature-based approaches cannot cope with complex (freeform) surface machining, because few traditional features can be identified on parts with freeform surfaces.

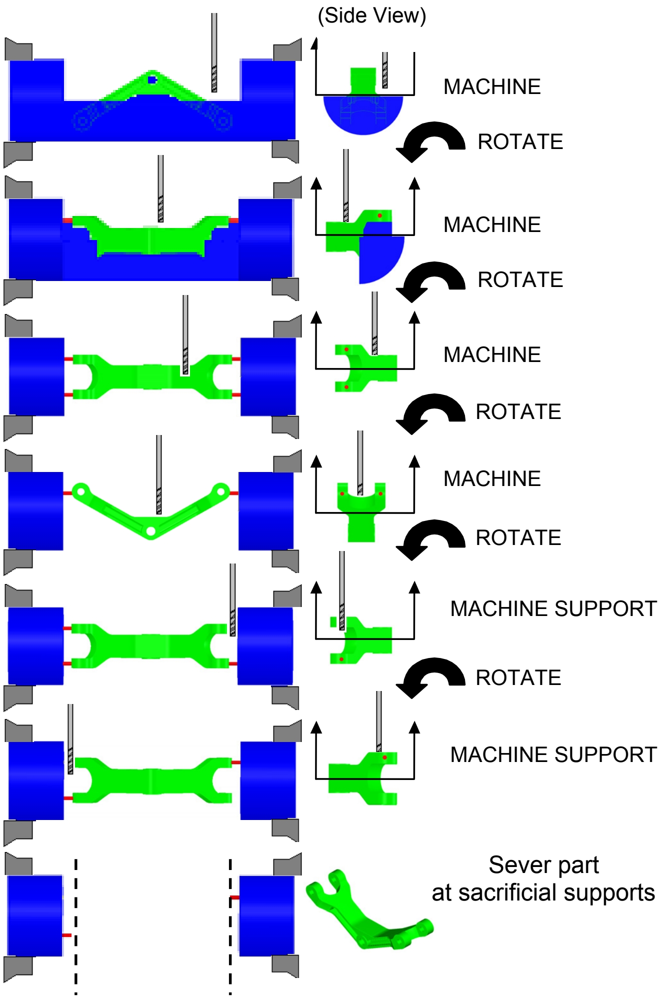


Fig. 3 Process steps for rapid machining

An effective machinability analysis method is a prerequisite to the successful implementation of multi-setup 3-axis end milling in order to achieve the needs of 4- and perhaps 5-axis machining. An effective machinability analysis method will determine, given a machining orientation and an end mill of a particular size, how much of the part surface can be machined with respect to this machining orientation. The focus of this paper is to present a feature-free machinability analysis that can determine the number of setups required to completely machine the surfaces of a part with one axis of rotation setups. The machinability analysis method presented in this paper is unlike any previous work in its completely feature-free treatment of the part geometry. We reduce the surfaces of the part down to simple line segments on the slices, therefore any CAD model can be exported as an STL file and studied. This approach is done because we are only assuming that the part is machined about one axis of rotation, therefore it is much simpler to simply analyze the 2D slices rather than 3D surface geometry.

The remainder of this paper is organized as follows. In the second section, definitions that are used throughout this paper are presented. The third section discusses the machinability analysis method in further detail. The fourth section presents the implementation of the machinability analysis approach. Lastly, conclusions and future research endeavors are provided.

2. Definitions

Although previous researchers have defined the concepts of *Visibility* and *Machinability* in their work, similar definitions are provided first in this section to clarify the difference between *Visibility* and *Machinability*. Next, the concept of *Tool Space, Obstacle Space* and *Machinable Range* are introduced. A condition to determine the existence of *Machinability* is also derived. The definitions provided in this section are used for the subsequent discussion in the remainder of this paper.

- *Visibility*: A point p on a surface S ($p \in S$) is visible by a light ray emanated from an external point q if $\overrightarrow{pq}$ suffices the condition of $\overrightarrow{pq} \cap (S\text{-}p)=\Phi$.
- *Machinability*: A point p on a surface S ($p \in S$) is machinable by a certain type and size of tool $T(CL,\alpha)$ if $p \in T(CL,\alpha)$ and $T(CL,\alpha) \cap (S\text{-}p) = \Phi$. $T(CL,\alpha)$ represents the tool surface at the cutter location CL, approaching from the orientation α.

By definition, machinability shares the same concept of accessibility with visibility, but differs in the sense that machinability takes into account the size and shape of the cutting tool instead of treating it simply as a line-of-light. Therefore machinability can guarantee true accessibility while visibility is only a necessary condition of machinability. Hence, the aggregate of orientations satisfying machinability is a subset of that satisfying visibility. In other words, machinability can guarantee visibility, but not vice versa.

Unlike the expression of visibility in angular orientations, the bundle of which forms a cone, there are two parameters used to describe machinability. They are the cutter location CL and the approaching orientation α, if the type and size of a cutter are specified. Machinability with respect to an approaching orientation α exists only if there is a cutter location that allows the cutting tool to approach and touch the point p without intersecting any other part surface.

Similar to the concept of the visibility of a feature, the machinability of a feature (a line, a curve, or a patch of surface that is geometrically composed of a set of points) is the intersection of the machinability of each point belonging to that feature. Similar to the concept of *Partial Visibility* (*PV*), *Partial Machinability* (*PM*) of a feature can also be defined in addition to the concept of *Complete Machinability* (*CM*).

- *Partial Machinability* (*PM*): A feature is partially machinable along an orientation α if there exists at least one point on that feature such that no cutter location *CL* exists for it to suffice the condition of $p \in T(CL,\alpha)$ and $T(CL,\alpha) \cap (S\text{-}p) = \Phi$.
- *Complete Machinability* (*CM*): A feature is completely machinable along an orientation α if for each point on that feature at least one cutter location *CL* can be found to guarantee the condition of $p \in T(CL,\alpha)$ and $T(CL,\alpha) \cap (S\text{-}p) = \Phi$.

Notice that *Complete Machinability* (*CM*) may exist for either a point or a feature, while *Partial Machinability* (*PM*) exists only for a feature, because a point can only be said to be either machinable or non-machinable.

If machinability exists with repect to an approaching orientation α, the number of feasible cutter locations *CL*s may vary with different points on a surface. Points with more feasible cutter locations *CL*s translates to easier machining because the more possible *CL*s provide more options for tool path and setup planning. The need to measure the space of cutter locations leads to the concept of *Tool Space*.

- *Tool Space* (*TS*): The aggregate of all feasible cutter locations to cut a point *p* from an orientation α forms a region called *Tool Space,* written as $TS\ (p,\ \alpha) = \{CL\text{: } p \in T(CL,\alpha)$ and $T(CL,\alpha) \cap (S\text{-}p) = \Phi\}$.

Tool Space of a feature *F* is the union of the *Tool Space* of every point belonging to *F*, that is, $TS\ (F) = \{\cup\ TS\ (p,\ \alpha)\text{: } p \in F\}$. A *Tool Space* (*TS*) reaches its maximum value *Maximum Tool Space* (*MTS*) when there is no obstacle around the geometric entity. Here we consider the entire part surface except the portion under consideration to be obstacles. Thus the corresponding space for obstacles is defined as *Obstacle Space*.

- *Obstacle Space* (*OS*): The aggregate of all unfeasible cutter locations with respect to an orientation α due to the existence of an obstacle *i* (*Obi*) is called the *Obstacle Space* of obstacle *i*, written as $OS\ (i,\ \alpha) = \{CL\text{: } T(CL,\alpha) \cap\ Obi \neq \Phi\}$.

The cutter cannot enter the domain of *Obstacle Space* because it will gouge into the obstacle.

Tool Space can be computed by subtracting all the *Obstacle Space*s from *Maximum Tool Space*.

$$TS=MTS-\sum_{i}OS \tag{1}$$

If the computed *Tool Space* using the above equation is not empty, then machinability exists; otherwise the geometric entity is non-machinable. The machinability analysis method presented in this paper is based on this equation. *Tool Space* is actually a measure of machinability since it tells the existence of machinability and the magnitude of machinability, if it exists.

Once the *Tool Space* is determined, the *Machinable Range* resulting from it can be obtained.

- *Machinable Range* (*MR*): The maximum machinable portion of a feature given the *Tool Space* is called *Machinable Range* (*MR*) of that feature, written as $MR=\{p: p\in F$ and $TS(p, \alpha)\neq \Phi\}$.

The above definitions will be used throughout the remainder of this paper.

3. Machinability Analysis

The machinability analysis approach presented in this paper is based on the concept of Configuration Space (C-space). The concept of C-space first applied in robotic spatial motion planning was documented by the work of Lozano-Perez **[32].** The basic idea of C-space is to find the aggregate of the valid spatial configurations for a moving mechanism in an environment with obstacles around it. Recently C-space has been applied in tool path planning for multi-axis machining. Choi et al. **[33]** presented a C-space based approach to generate 3-axis NC tool paths for sculpture machining by transforming the designed part surface and stock-surface into elements in C-space and treating the cutter as a moving object in the safe space. The C-space is represented and computed using a Z-map model of the part. Choi and Ko **[34]** incorporated C-space into computer automated process planning (CAPP) for freeform die-cavity machining. Morishige et al. **[35]** used C-space to generate tool paths for 5-axis ball end milling. Jun et al. **[36]** optimized tool orientations for 5-axis flat end milling by a search method in C-space. C-space of the cutting tool, which is defined as tool space in Section 2 of this paper, provides the safe space for tool path planning, therefore tool paths based on C-space are always gouge-free and collision-free. The tool space can be seen as the aggregate of all possible tool paths. If tool space exists, then at least one tool path can be generated to machine the corresponding geometric point on the part surface, and hence this point has machinability. Therefore, by testing the tool space of each point on a surface, the machinability of this surface can be theoretically determined.

The input for the machinability method of this paper is the slice file of an STL model of the part along the intended axis of rotation. An STL model is an approximation of the part surface by using triangular facets and is currently the de-facto standard file format for rapid prototyping systems. The tessellation process to create an STL model can keep the approximation error, the deviation between the part surface and the tessellation triangles, within a specified tolerance. The size and shape of each triangle created is adaptive to its local region on the part surface. Therefore, compared to the Z-map model employed by Choi et al. **[33],** which is essentially a virtually equal-spacing sampled model, the slice file of an STL model is a more precise and efficient representation of the original part surface. The slicing process for the STL model of the part, which may be either a feature based part or a freeform geometry, breaks the part surface into many line segments comprising the polygonal chains of each slice. These line segments are essentially the only representation of a "feature" in the method presented in this paper. In this manner, it is not important that any set of segments be machinable in any particular setup orientation. The intent is to map the segments or portions of segments machinable from each orientation α, in order for a minimum set of orientations to be found such that all segments are machined after all setup orientations are completed.

Each line segment of the slice file is either perpendicular or oblique with respect to the tool approach orientation. If the perpendicular case occurs, the obstacles and their corresponding obstacle spaces for each point on that line segment are the same. This is actually a static-obstacle case. A 2-dimensional C-space analysis can be performed for all those points at the same time, considering all the adjacent segments above the segment to be obstacles. Section 3.1 discusses the perpendicular case in more detail. However, if the line segment to be checked is oblique with respect to the tool approach orientation, the obstacles for each point on that segment are variable. This dynamic-obstacle environment creates difficulty for the machinability analysis process. The solution for this case will be presented in Section 3.2.

3.1 Perpendicular Case

The coordinate system used in this paper is consistent with that of a 3-axis milling center, whereby the slicing of the part occurs along the X-axis of the machine coordinate frame and each slice is in the Y-Z plane (Fig. 4(*a*)). We use $\overline{P_{i,j}P_{i,j+1}}$ to represent a segment that is currently undergoing the machinability analysis process, where *i* denotes the number of the slice on which that segment resides, and *j* and *j+1* denotes the two consecutive points forming that line segment. In the case where the segment $\overline{P_{i,j}P_{i,j+1}}$ is perpendicular to the tool cutting orientation, all the segments on slice *i* and its adjacent slices with a portion having a greater height than that of $\overline{P_{i,j}P_{i,j+1}}$ along the tool cutting orientation are obstacles. The obstacle spaces associated with these obstacles remain unchanged for the analysis of every point on $\overline{P_{i,j}P_{i,j+1}}$. In this situation, the cutting tool is moving in an environment with static obstacles. The problem is simplfied as finding the tool space for segment $\overline{P_{i,j}P_{i,j+1}}$ as a geometric primitive on a 2-dimensional plane, instead of analyzing each point separately. Figure 4(*a*) shows that there are *left* obstacles and *right* obstacles on each side of segment $\overline{P_{i,j}P_{i,j+1}}$. Obstacles may also exist on the slice where $\overline{P_{i,j}P_{i,j+1}}$ resides. Plane π contains segment $\overline{P_{i,j}P_{i,j+1}}$ and is perpendicular to the tool cutting orientation *α*. Any segment that has a portion above plane π is considered an obstacle. Figure 4(*b*) is the top view of Fig. 4(*a*). The boundary of the maximum tool space and the obstacle spaces are constructed by offsetting segment $\overline{P_{i,j}P_{i,j+1}}$ and obstacle segments by the amount of the tool radius. The curves in the dash-dot line (_._._._._._)denote the boundaries of left obstacle space($\sum_m OS\ (L_m,\ \alpha)$) and right obstacle space($\sum_n OS\ (R_n,\ \alpha)$). The curves in the dashed line (- - - - - - - - - - -) denote the boundaries of current slice obstacle space (*OS* (*i, α*)) and the closed curve in the continuous line (__________)denotes the boundary of the maximum tool space (MTS). The actual Tool Space (TS) represented by the shaded region in Fig. 4(*b*) is computed using Eq. (1) as follows:

$$\text{TS}=\text{MTS} - \sum_{m} OS\,(L_m,\,\alpha) - \sum_{n} OS\,(R_n,\,\alpha) - OS\,(i,\,\alpha)$$

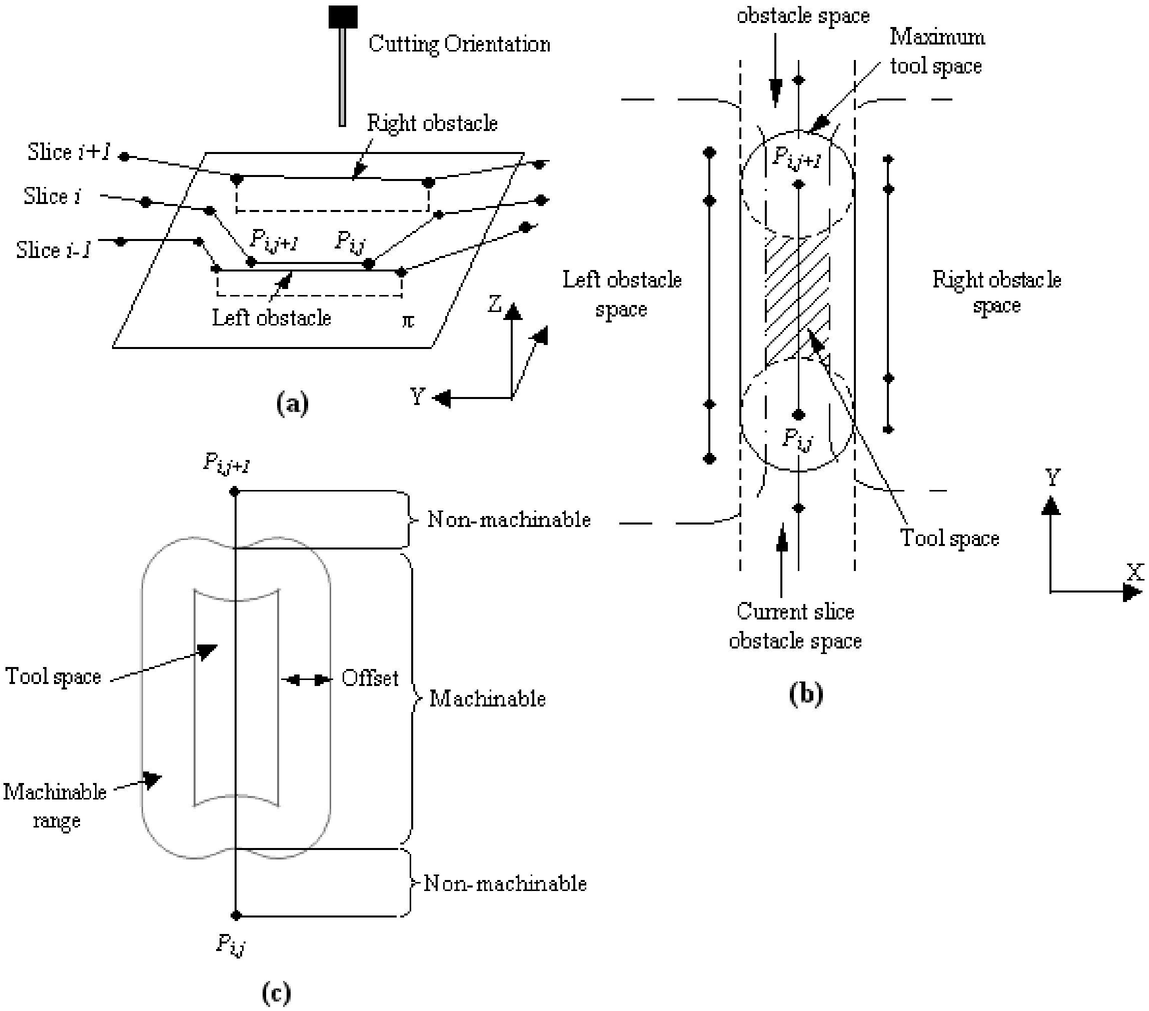


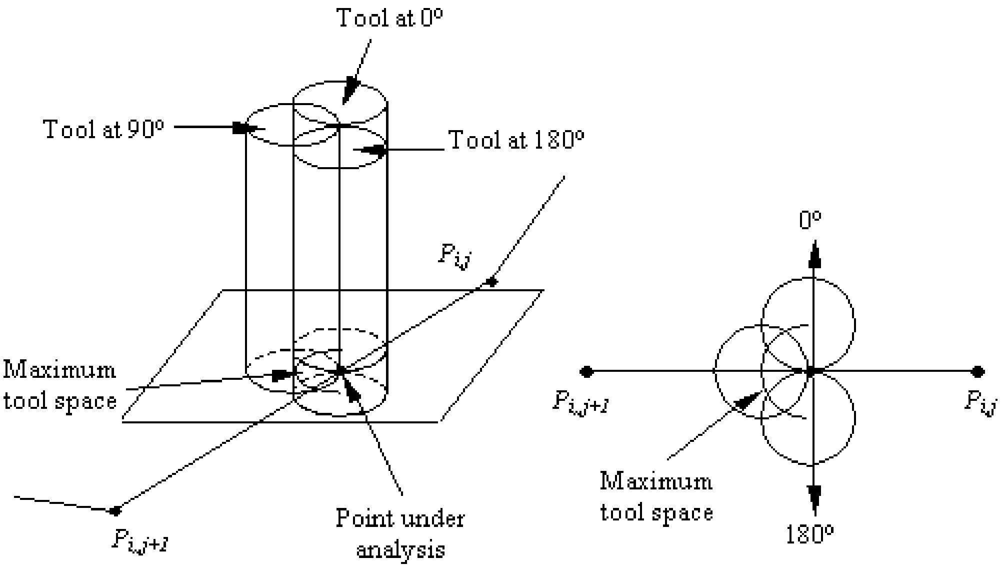
Fig. 4 Illustration of machinability for perpendicular case

If Tool Space results in an empty set, then the segment $P_{i,j}P_{i,j+1}$ contains no machinable portion. If Tool Space exists, then the Machinable Range (MR) boundary can be obtained by offsetting the boundary of Tool Space by the amount of the tool radius (Fig. 4(*c*)). It should be noted that the Tool Space may consist of several sub-regions in practice (The Tool Space shown in Fig. 4(*b*). consists of only one region). Therefore, the Machinable Range may also be separated into sub sections. If the Machinable Range subsections cover the entire segment, $\overline{P_{i,j}P_{i,j+1}}$, that is, $\overline{P_{i,j}P_{i,j+1}} \subseteq \sum_{\lambda} MR_{\lambda}$, then segment $\overline{P_{i,j}P_{i,j+1}}$ has *Complete Machinability*; otherwise it only has *Partial Machinability.* Figure 4(*c*) shows that $\overline{P_{i,j}P_{i,j+1}}$ has *Partial Machinability* and the portions outside of the Machinable Range boundary are non-machinable portions of segment $\overline{P_{i,j}P_{i,j+1}}$.

3.2 Oblique Case

3.2.1 Dynamic-Obstacle environment

For most cases, the cutting orientation is not perpendicular, but oblique to the line segment being analyzed for machinability. This makes the machinability analysis significantly different from the previously described perpendicular case that only had static obstacles. The fundamental reason is that the static-obstacle machinability analysis approach based on 2-dimensional C-space does not work for the oblique case, which is characterized by dynamic-obstacles. The maximum tool space for machining each point under the oblique case is invariantly a half-circle arc, which is shown in Fig. 5. However, the obstacle spaces that the points on $\overline{P_{i,j}P_{i,j+1}}$ are subject to are dynamically changing as the point under analysis



(a) Maximum tool space of a point

(b) Top view of maximum tool space

Fig. 5 Illustration of maximum tool space under oblique cutting

is moving along $\overline{P_{i,j}P_{i,j+1}}$. Figure 6 shows segment $\overline{P_{i,j}P_{i,j+1}}$ on slice i and an obstacle segment $\overline{P_{i+n,k}P_{i+n,k+1}}$ on adjacent slice $i+n$. Polygon $P_{i+n,k}P_{i+n,k+1}P'_{i+n,k+1}P'_{i+n,k}$ is the obstacle polygon resulting from segment $\overline{P_{i+n,k}P_{i+n,k+1}}$ with respect to the cutting orientation α. It is clear that the effective obstacles affecting point P_1 and P_2, denoted by Ob_1 and Ob_2 respectively, are different, even though they are caused by the existence of the same obstacle segment $\overline{P_{i+n,k}P_{i+n,k+1}}$. Hence, the obstacle spaces associated with these two obstacles are also different. The variation of obstacle spaces is due to the fact that the heights of the points on $\overline{P_{i,j}P_{i,j+1}}$ with respect to the cutting orientation α are different from one another. The obstacle spaces of a certain height are determined by the projections of the obstacle segments above that height on the plane perpendicular to the cutting orientation α at that height. This is actually a problem of 3-dimensional C-space. The Tool Space for $\overline{P_{i,j}P_{i,j+1}}$ could be computed by constructing a 3-dimensional C-space. However, since the "part surfaces" being worked on consist of segments from STL slice geometry, there is no information about what kind of feature $\overline{P_{i,j}P_{i,j+1}}$ resides on and/or the local surface description in the vicinity of $\overline{P_{i,j}P_{i,j+1}}$. Therefore, testing machinability by constructing a 3-dimensional C-space for each segment is inappropriate.

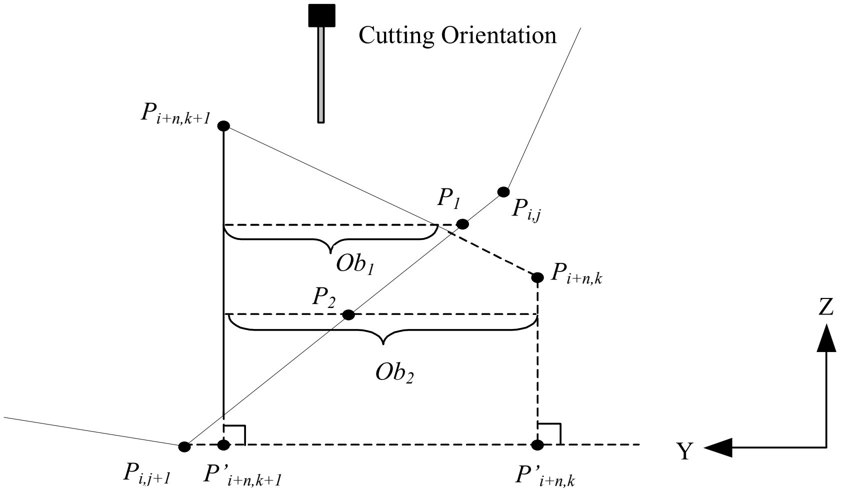


Fig. 6 Variation of effective obstacles

3.2.2 Relative movement of effective obstacles

Although the entire 3-dimensional C-space will not be constructed for each line segment, the effective obstacle spaces of each obstacle segment at different heights can be considered to have relative movement with the point under analysis if one sets a convention for machinability analysis. Referring to the example in Fig. 6, if one analyzes the point along the direction traversing the line segment $\overline{P_{i,j}P_{i,j+1}}$, the relative linear movement of the effective obstacles shows three stages where machinability is affected. They include stage 1 where the obstacle space begins to gouge into the maximum tool space (Fig. 7(*a*)) and the gouged tool space increases (Fig. 7(*b*)) until it reaches its maximum gouged tool space arc, $\theta_{mL}\theta_{mU}$ (Fig. 7(*c*)), stage 2 where the obstacle space maintains its maximum gouged tool space arc $\theta_{mL}\theta_{mU}$ (Fig. 7(*c*) to Fig. 7(*d*)); and stage 3 where the obstacle space begins to move away from O_m, the point being examined. Gouged tool space then begins to decrease (Fig. 7(*e*)) until the obstacle space does not affect machinablity (Fig. 7(*f*)). In Fig. 7, Δ denotes the slicing spacing; *k*Δ is the distance from the slice *i* to slice *i*+*k*, where the obstacle segment is located. The variable *d* is the relative distance of the effective obstacle segment to the segment under analysis. *EF* represents the effective obstacle, Point *F* moves along edge $\overline{P_{i+n,k+1}P_{i+n,k}}$ and $\overline{P_{i+n,k}P'_{i+n,k}}$, and point *E* moves along edge $\overline{P_{i+n,k+1}P'_{i+n,k+1}}$ of the polygon $P_{i+n,k}P_{i+n,k+1}P'_{i+n,k+1}\ P'_{i+n,k}$ in Fig. 6. The variable d_s denotes the distance from *F* to O_m along the Y axis, at which tool space begins to be gouged by the obstacle space of *EF*; while d_m denotes the distance from *F* to O_m along the Y axis, at which the maximum gouged tool space is reached (Fig. 7(*c*)), and also denotes the distance from *E* to O_m along the Y axis, at which gouged tool space begins to reduce (Fig. 7(*d*)). From Fig. 7. it can be seen that if there exists a point $p \in EF$ with its distance to O_m along the Y axis equal to d_m, then the maximum gouged tool space can be attained. The variable d_e denotes the distance from point *E* to O_m along the Y axis, at which the tool space becomes unaffected by the obstacle space of *EF*. Notice that each obstacle segment may have all or only one or two of these three typical stages in practice depending on the relative location of the obstacle segment and the segment under analysis. This will be discussed further in section 3.2.3.

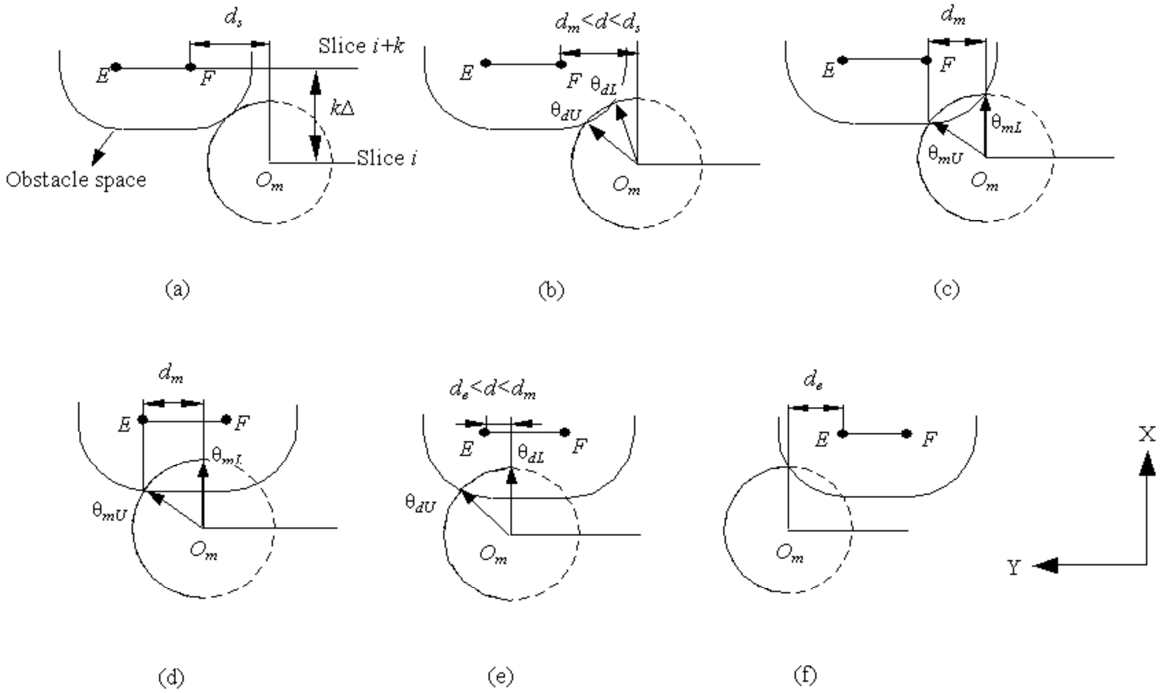


Fig. 7 Variation of machinable range due to the existance of an obstacle segment

For the above case, once the stage the effective obstacle belongs to and the value of d is known, the gouged tool space arc (θ_{dL}, θ_{dU}) can be computed as follows:

For Stage 1 where $d_m < d < d_s$

$$\theta_{dL} = \arccos(\frac{k\Delta(k^2\Delta^2 + d^2) + \delta}{2(k^2\Delta^2 + d^2)R})$$

$$\theta_{dU} = \arccos(\frac{k\Delta(k^2\Delta^2 + d^2) - \delta}{2(k^2\Delta^2 + d^2)R})$$

For Stage 2 where $d=d_m$

$$\theta_{dL} = 0$$

$$\theta_{dU} = \arccos(\frac{k\Delta - R}{R})$$

For Stage 3 where $d_e < d < d_m$

when $d_e < d < 0$

$$\theta_{dL} = 0$$

$$\theta_{dU} = \arccos(\frac{k\Delta(k^2\Delta^2 + d^2) + \delta}{2(k^2\Delta^2 + d^2)R})$$

when $0 \le d < d_m$

$$\theta_{dL} = 0$$

$$\theta_{dU} = \arccos\left(\frac{k\Delta(k^2\Delta^2 + d^2) - \delta}{2(k^2\Delta^2 + d^2)R}\right)$$

$$d_s = \sqrt{(2R)^2 - (k\Delta)^2}$$

$$d_m = \sqrt{R^2 - (k\Delta - R)^2}$$

$$d_e = -\sqrt{R^2 - (k\Delta - R)^2}$$

$$\delta = \sqrt{k^2\Delta^2(k^2\Delta^2 + d^2)^2 - (d^2 + k^2\Delta^2)^3 + 4d^2R^2(d^2 + k^2\Delta^2)}$$

R is the tool radius.

3.2.3 Characterization of the relative movement

The above discussion demonstrates that the relative movement of the effective obstacle with respect to the point under analysis provides the information necessary to calculate the gouged tool space resulting from the corresponding effective obstacle space, and thus the Tool Space can be computed. Therefore, if the relative movement of each effective obstacle with respect to the point under analysis can be precisely determined, then machinability can be computed without constructing a 3-dimensional C-space.

Geometric transformation

The relative movement of each effective obstacle with respect to the point under analysis can be precisely determined by performing a geometric transformation to both the segment to be analyzed and the obstacle segment.

Consider an inclined line segment $\overline{P_{i,j}P_{i,j+1}}$ ($P_{i,j}(y_{i,j}, z_{i,j})P_{i,j+1}(y_{i,j+1}, z_{i,j+1})$) on slice *i*, and suppose that $z_{i,j}$ is greater than $z_{i,j+1}$ ($z_{i,j} > z_{i,j+1}$) (Fig. 8.). The parameter *t* is from the parametric representation of $P_{i,j}P_{i,j+1}$. The parameter *t* is 0 at the end point with smaller Z coordinate and is 1 at the end point with greater Z coordinate. For the segment $\overline{P_{i,j}P_{i,j+1}}$ shown in Fig. 8, *t* is 0 at $P_{i,j+1}$ and is 1 at $P_{i,j}$. Point $E(y_E,z_E)$ is an arbitrary point in the plane Y-Z and lies on line *AB*, determined by point $A(y_A,z_A)$ and $B(y_B,z_B)$. Point *E* can be

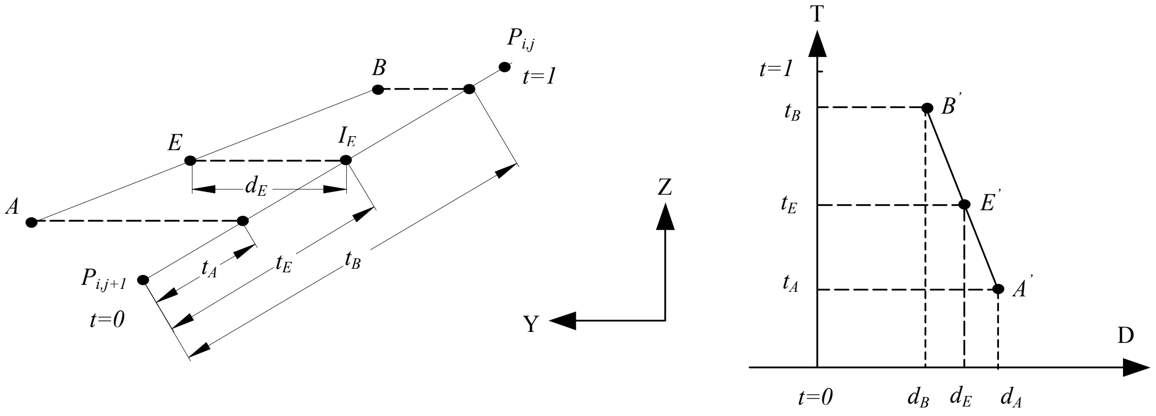


Fig. 8 Geometric transformation

transformed and mapped uniquely as a point $E'(d_E, t_E)$ in plane D-T as shown in Fig. 8. The d_E is the horizontal distance from E to line $\overline{P_{i,j}P_{i,j+1}}$ along the Y-axis. It is the distance from E to $I_E(y_{iE,} z_E)$, the intersection point of line $Z=z_E$ and the line where segment $\overline{P_{i,j}P_{i,j+1}}$ lies. The t_E is the parametric value at the intersection point I_E. Similarly A and B are mapped as A' and B' in plane D-T. The transformation from a 2D plane Y-Z to the 2D plane D-T (D-T transformation) can be represented as a map F: $R^2 \longrightarrow R^2$ of the form

$$F(p) = M * p + N$$

for all point $p \in R^2$, where $M = \begin{bmatrix} 1 & \dfrac{y_{i,j+1} - y_{i,j}}{z_{i,j} - z_{i,j+1}} \\ 0 & \dfrac{1}{z_{i,j} - z_{i,j+1}} \end{bmatrix}$ $N = \begin{bmatrix} \dfrac{z_{i,j+1}y_{i,j} - z_{i,j}y_{i,j+1}}{z_{i,j} - z_{i,j+1}} \\ -\dfrac{z_{i,j+1}}{z_{i,j} - z_{i,j+1}} \end{bmatrix}$ for line segments

with $z_{i,j+1} < z_{i,j;}$ $M = \begin{bmatrix} -1 & \dfrac{y_{i,j+1} - y_{i,j}}{z_{i,j+1} - z_{i,j}} \\ 0 & \dfrac{1}{z_{i,j+1} - z_{i,j}} \end{bmatrix}$ $N = \begin{bmatrix} \dfrac{z_{i,j+1}y_{i,j} - z_{i,j}y_{i,j+1}}{z_{i,j+1} - z_{i,j}} \\ -\dfrac{z_{i,j}}{z_{i,j+1} - z_{i,j}} \end{bmatrix}$ for those with $z_{i,j+1} > z_{i,j.}$

Mapping an obstacle polygon on D-T plane

Based on the D-T transformation, an obstacle segment on adjacent slice $i+m$ $Q_{i+m,k}Q_{i+m,k+1}$ and its associated obstacle polygon $Q_{i+m,k}Q_{i+m,k+1}Q'_{i+m,k+1}\, Q'_{i+m,k}$ (Fig. 9) can be mapped onto D-T plane as segment $Q^*_{i+m,k}\, Q^*_{i+m,k+1}$ and obstacle polygon

$Q^*_{i+m,k}\, Q^*_{i+m,k+1}\, Q'^*_{i+m,k+1} Q'^*_{i+m,k}$ (Fig. 10). The coordinates of $Q^*_{i+m,k}, Q^*_{i+m,k+1}, Q'^*_{i+m,k+1}, Q'^*_{i+m,k}$ are $(d_{i+m,k}, t_{i+m,k})$, $(d_{i+m,k+1}, t_{i+m,k+1})$, $(d_{i+m,k+1}, 0)$, and $(d_{i+m,k}, 0)$ respectively. Any line segment formed by truncating a horizontal line in the D-T plane by the boundary of the obstacle polygon $Q^*_{i+m,k}\, Q^*_{i+m,k+1}\, Q'^*_{i+m,k+1} Q'^*_{i+m,k}$ is actually the effective obstacle of $Q_{i+m,k}Q_{i+m,k+1}$ at the height t along the T-axis, as shown in Fig. 10. The end point with a smaller D value can be considered as a *front* point and the one with a larger D value can be considered as a *rear* point. The front points and rear points of all the effective segments resulting from $Q_{i+m,k}Q_{i+m,k+1}$ form the front point trajectory and the rear point trajectory. In Fig. 10, the front point trajectory is $Q^*_{i+m,k}\, Q'^*_{i+m,k}$ and the rear point trajectory is $Q^*_{i+m,k}\, Q^*_{i+m,k+1}$ and $Q^*_{i+m,k+1}\, Q'^*_{i+m,k+1}$.

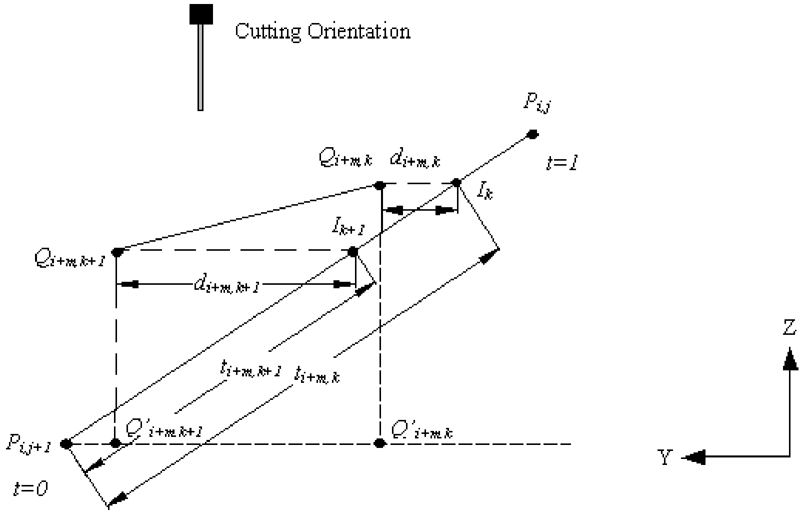


Fig. 9 An obstacle polygon in Y-Z plane

The three stages discussed in section 3.2.2 can also be reflected clearly on the obstacle polygon in plane D-T, as illustrated in Fig. 10. $\overline{JK}$, along the front point trajectory, truncated by the line $d=d_s$ and $d=d_m$, corresponds to stage 1, $\overline{KG}$, along the line $d=d_m$ truncated by the front point trajectory and the rear point trajectory corresponds to stage 2, and $\overline{GH}$, along the trajectory of the rear point, truncated by line $d=d_m$ and $d=d_e$, corresponds to stage 3. As mentioned previously, not all the obstacle segments possess all of these three

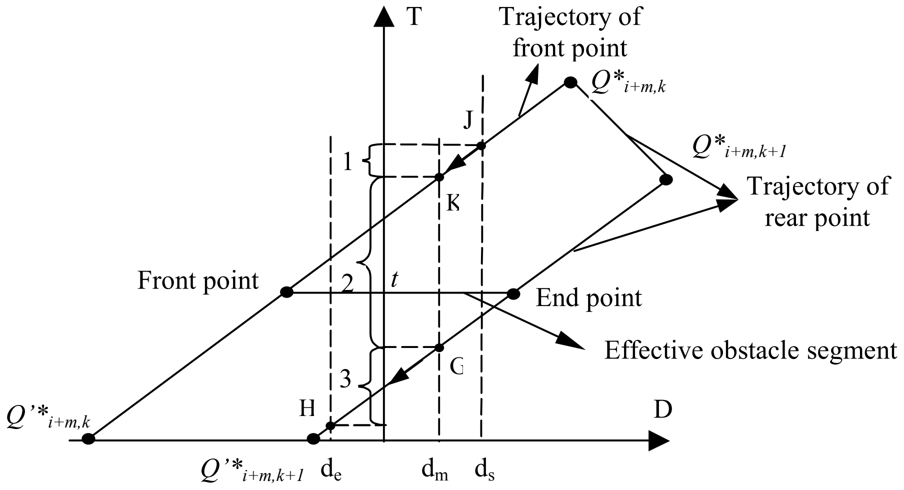


Fig. 10 An obstacle polygon in D-T plane

stages. How many stages an obstacle segment may have depends on the shape of the obstacle polygon in the D-T plane. The mapping of these three stages onto the plane D-T can provide a precise calculation for the gouged tool space, because given a *t* value, the corresponding *d* value, which denotes the relative distance of the segment under analysis and one obstacle segment, can be easily mapped on plane D-T and therefore the gouged tool space can be precisely calculated with the *d* value, using the method described in section 3.2.2.

So far we have shown that *d* is a function of *t* and gouged tool space is a function of *d*. Therefore, the gouged tool space can also be considered as a function of *t*. This leads to the construction of a graph of machinability versus the parameter *t* for each line segment to be examined.

3.2.4 Machinability graph

Figure 11 illustrates the composition of the machinability graph for line segment $\overline{P_{i,j}P_{i,j+1}}$ under the interference of one obstacle $Q_{i+m,k}Q_{i+m,k+1}$. It consists of the same three stages as those shown in Fig. 10 and is bounded by two curves: Upper boundary curve θ_U and Lower boundary curve θ_L. The region bounded within these two curves is the gouged tool space obstructed by the obstacle $Q_{i+m,k}Q_{i+m,k+1}$ and is denoted as $OS_{i+m,k,k+1}$. The rectangular frame shown in Fig. 11 with length 1 and width π is the maximum tool space

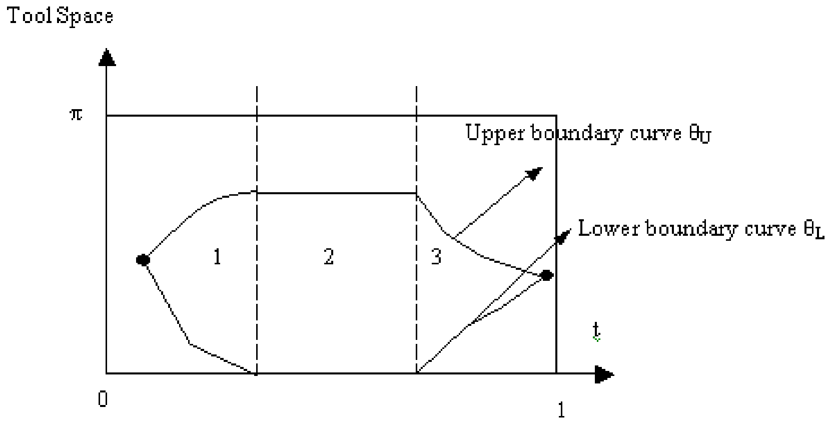


Fig. 11 Machinability graph

(MTS). Once the gouged tool space for each obstacle segment is obtained, the machinability analysis can be conducted by computing $MTS - \sum_{n}\sum_{k} OS_{i+n,k,k+1}$. If the subtraction operation yields an empty result, then the segment $\overline{P_{i,j}P_{i,j+1}}$ is not machinable with respect to the machining orientation. Otherwise this segment is at least partially machinable.

Since the upper boundary curve θ_U and lower boundary curve θ_L are not of regular shapes, the analytic computation of tool space is not feasible. In this paper, a sweeping line method was used to incrementally check the existence of tool space. The steps to the method are as follows:

Step 1: Assign t_s to t. (t_s can be either 1 or 0, depending on the incremental direction).

Step 2: Check and store the intersections of line $t=t_s$ with the upper and lower boundary curves of each obstacle segment $Q_{i+m,k}Q_{i+m,k+1}$.

Step 3: If there is no intersection, then $t=t_s$ is machinable. Go to Step 8. If there are intersections, sort all the intersections in decreasing order into a sequence.

Step 4: If the maximum intersection is less than 180^o, or the minimum intersection is greater than 0^o, then $t=t_s$ is machinable. Go to Step 8. Otherwise search the reverse order pair in the intersection points list. The reverse order pair is defined as a lower intersection point (LIP, an intersection with a lower boundary curve) immediately in front of an upper intersection point (UIP, an intersection with an upper boundary curve) in the sorted intersection points sequence.

Step 5: If there is no reverse order pair, then $t=t_s$ is non-machinable. Go to step 8: Otherwise store the reverse order pairs.

Step 6: For each reverse order pair *j*, check the number of net upper intersection points (Num_NUIP) before reverse order pair *j*(denoted by <*j*) and the number of net lower intersection points (Num_NLIP) after reverse order pair *j* (denoted by >*j*) in the intersection points sequence.

Num_NUIP(<*j*) =number of UIP (<*j*) - number of LIP(<*j*)

Num_NLIP (>*j*) = number of LIP(>*j*) - number of UIP(>*j*)

Step 7: If Num_NUIP=1 and Num_NLIP=1, then $t=t_s$ is machinable. If Num_NUIP>1 and Num_NLIP>1, then $t=t_s$ is non-machinable.

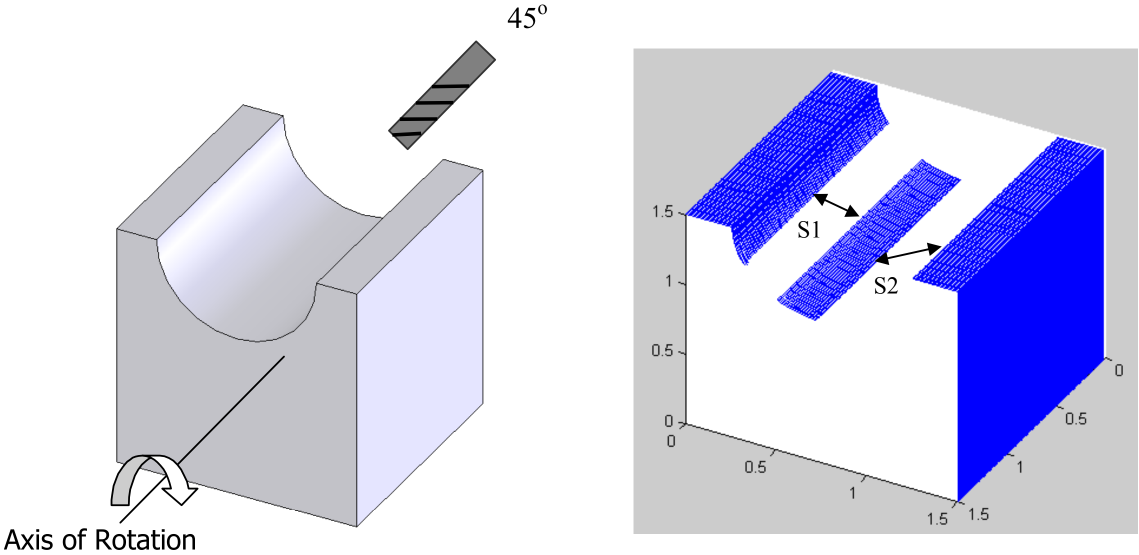
Step 8: Update *t* by an increment and check if stopping criterion is met. If not met, go to Step 1.

4. Implementation

To validate the approach proposed in this paper, the machinability algorithms were implemented in C programming language on a Pentium IV, 3.06 Ghz PC running window XP. The machinability software uses slice-visibility data, the size of the flat-end tool chosen and a specified cutting orientation as inputs. It generates the machinable portion of each slice segment with respect to the cutting orientation as output. We present two example part surfaces to verify the machinability analysis approach. The first part was chosen to be quite simple so that the non-machinable regions should seem intuitive to the reader. The second part is a more complex part, a toy "jack", and this part is evaluated with 3D inspection software that is used for reverse engineering.

4.1 Example 1

Fig. 12(*a*) shows a block with a half cylindrical extruded cut. We chose the direction of 45 degrees on a plane orthogonal to the axis of rotation as the cutting orientation. The flat end tool diameter is set to be 0.25inch (6.35mm). The slice spacing is 0.01 inch (0.254mm). Results from the machinability analysis are displayed in Fig. 12(*b*), which indicates that there are two non-machinable regions, denoted as S1 and S2. Since this model is an extrusion, the geometric shape along the axis of rotation does not change. Therefore, the results of the machinability analysis of this part should be the same on every slice along the axis of



(a) Half cylinder extrusion pocket model

(b) Machinable surface

Fig. 12 Machinability of a half cylinder extrusion pocket

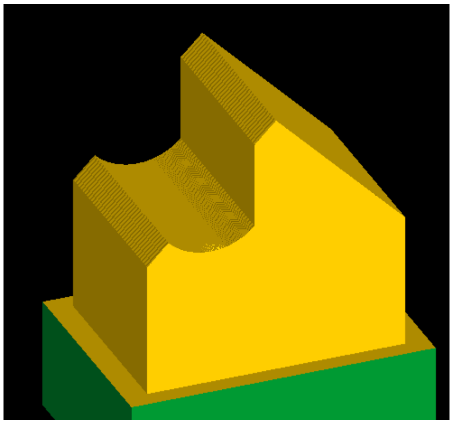
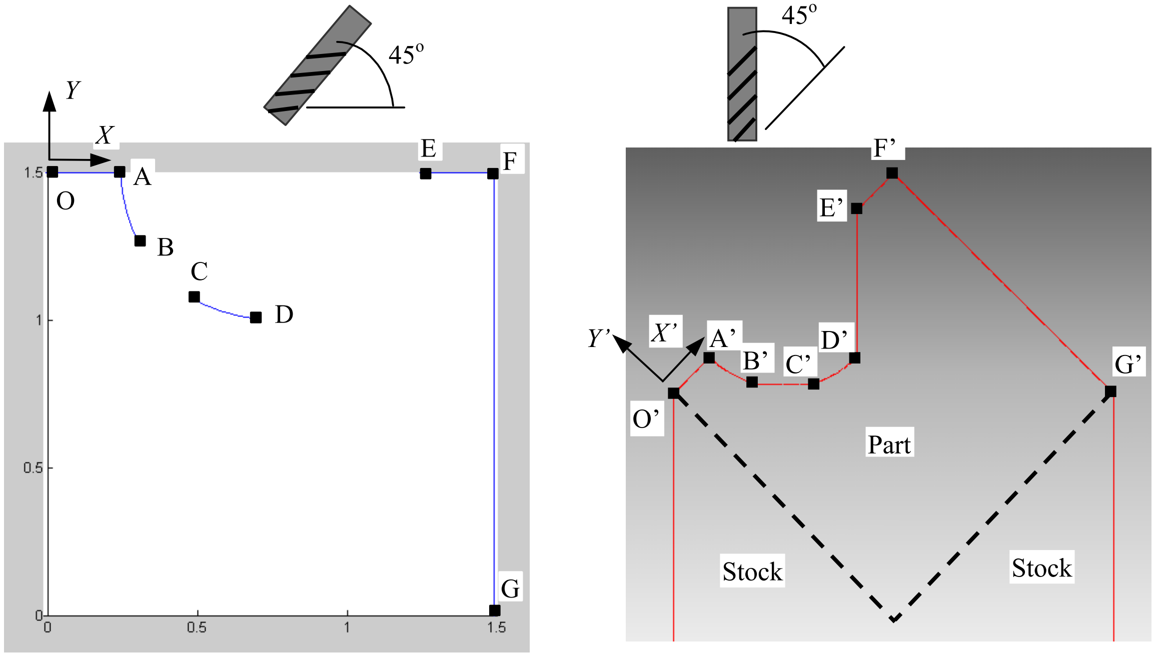


Fig. 13 Screen shot of example part virtually machined in MasterCAM

rotation. The machinable profile of one slice is displayed in Fig. 14(*a*). O-A-B, C-D and E-F-G are machinable regions while B-C and D-E are non-machinable regions. This part was also virtually machined in Mastercam. A screen shot of the part being virtually machined from a block of material is shown in Fig. 13. The virtually machined volume was saved as an STL file and imported into Rapidform 2004(reverse engineering software). Figure 14 (*b*) shows a cross-sectional profile of this STL by using the inspection function of Rapidform. The orientation of the profile displayed in Fig. 14 (*b*) is 45 degree with respect to the horizon, the same direction as the machining set-up shown in Fig. 13. O'-A'-B', C'-D' and E'-F'-G' are



(a) Result of our machinability analysis

(b) Result from virtual machining

Fig. 14 2D views of machinable profiles

Table 1 Boundary point coordinates of non-machinable regions

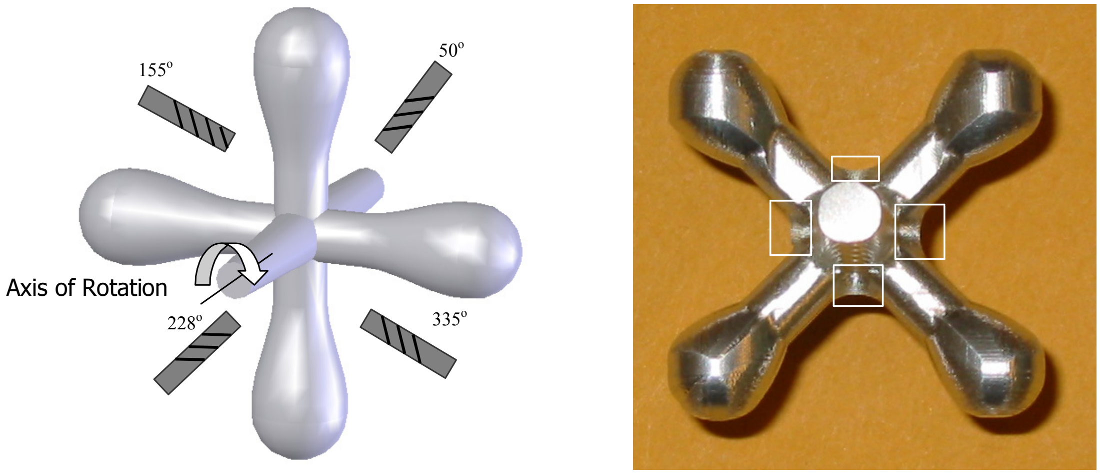
Point	Result of machinability analysis (in.)	Result of virtual machining (in.)
B, B'	$B(0.3173, -0.2481)$	$B'(0.3170, -0.2506)$
C, C'	$C(0.4973, -0.4294)$	$C'(0.4956, -0.4291)$
D, D'	$D(0.7500, -0.4983)$	$D'(0.7513, -0.4982)$
E, E'	$E(1.2500, 0.0000)$	$E'(1.2496, 0.0000)$

the machinable regions while B'-C' and D'-E' are the non-machinable regions in Fig. 14 (*b*). To verify the results from the machinability analysis, two local coordinate systems are set up on point O (X-O-Y in Fig. 14 (*a*)) and O' (X'-O'-Y' in Fig. 14 (*b*)), respectively. Coordinates of the non-machinable boundary points B, C, D, E are computed from the results of the machinability analysis software we developed and then the coordinates of B', C', D', E' are measured in RapidForm 2004. The coordinates of the points from each approach are shown in Table 1. The data are very close, and the error is within that which can be expected using an STL approximation and a line-sweeping algorithm for finding machinable regions.

4.2 Example 2

The second part model used as an example is a toy "Jack" that is analyzed first for visibility and then for machinability using the approach of this paper. Figure 15 (*a*) shows the STL-model of the "Jack". The visibility software developed previously **[17]** first processed this model and gave the result that the model is 100% visible through four orientations {50^{o}, 155^{o}, 228^{o}, 335^{o}}. To demonstrate the deficiency of visibility analysis, the "Jack" model was machined with these four orientations using a 0.125 inch (3.175mm) diameter flat-end tool. Figure 15 (*b*) shows the machined "Jack" with non- machinable regions indicated by the four rectangles.

Since the shape of slices of the "Jack" model varies along the axis of rotation, the non-machinable regions change accordingly and therefore are not regular shapes. To demonstrate that the machinability analysis approach can predict the non-machinable regions, we virtually machined the "Jack" from a 50-degree orientation in Mastercam and saved the result as an STL file after virtual machining. The file was imported into RapidForm 2004 and was overlapped with the original CAD model. Using the inspection function of Rapidform, we are able to map the deviations of the machined STL model to the original part model. Figure 16 illustrates the virtually machined surface (in MasterCAM) by rotating the part model to a 50-degree orientation. The region marked by the circle contains the non-



(a) "Jack" model (b) Machined "Jack"

Fig. 15 Machining result of a "Jack" model

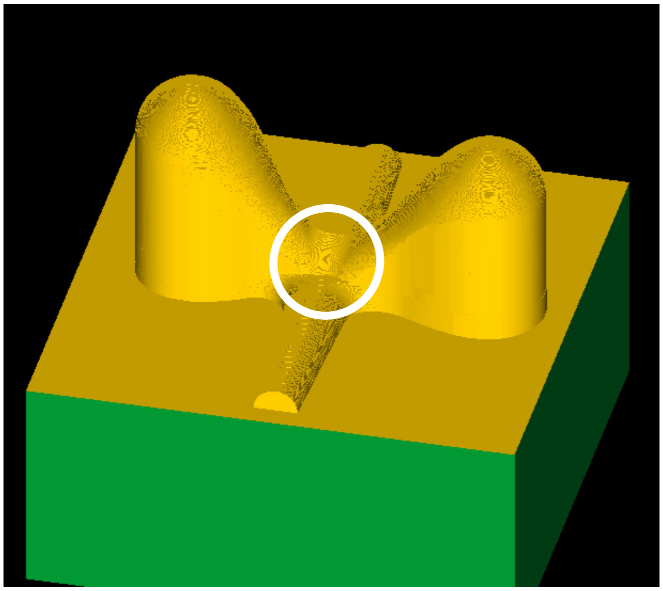
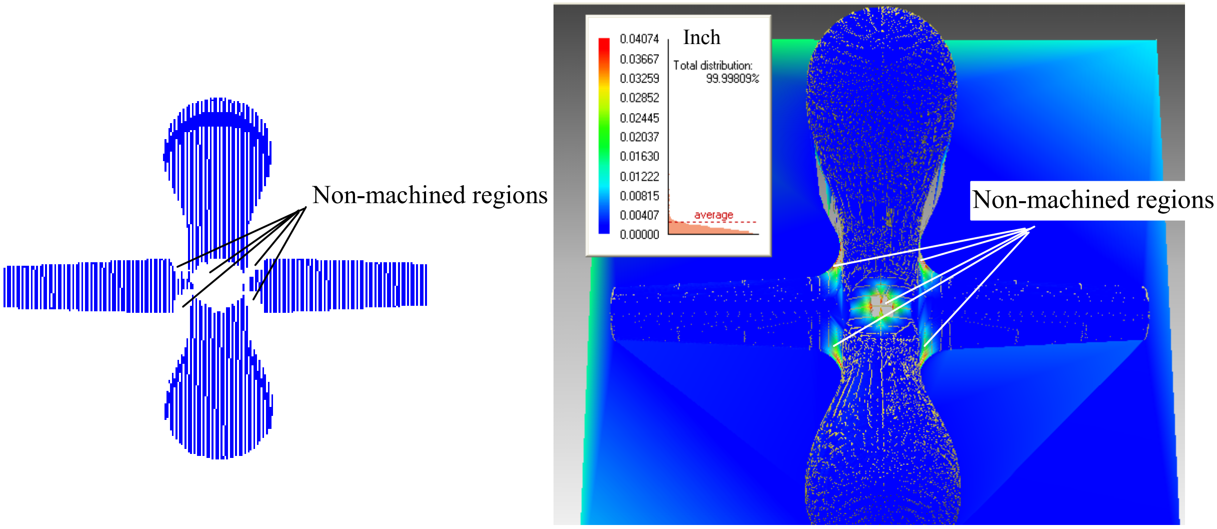


Fig. 16 Machined "Jack" surface in Mastercam

machinable surfaces. Figure 17 (*b*) illustrates the deviation results from RapidForm 2004. The part model is displayed in point shading mode to indicate the surface profile. Figure 17(*a*) illustrates the results from our machinability analysis software, which corresponds well with the graphical display in Fig. 17 (*b*).

The above examples indicate that the machinability software not only predicts machinable/non-machinable regions for a sliced STL model but also determines the exact coordinate locations of its machinable/non-machinable portions. Therefore, given a cutting orientation and a sized tool, the precise locational information of the machinable and non-machinable regions can be obtained, thus providing a tool for product designers in evaluating the manufacturability of a particular design. Another application could be found in determining the minimum number of setups for machining a part using an indexable 4-axis machine with respect to one rotation axis. This could be realized by first running the machinability software for cutting orientations from 0^o up to 360^o at the interval of a specified incremental angle, and then searching for the minimum number of setup orientations using an optimization algorithm. In this manner, the machinability analysis presented in this paper is not relegated to a *verification* tool, rather, the information it provides can be used to calculate machining setup information and for toolpath planning.



(a) Result of machinability analysis

(b) Graphical display of deviation in RapidForm

Fig. 17 Identification of non-machinable regions for “Jack”

5. Conclusions and Future Research

This paper presents a machinability analysis approach for 3-axis flat end milling based on the concept of C-space. The approach is intended to find the true accessibility, which is often approximated by the visibility of line-of-light. The input information is the STL slice file of a geometric model with restriction to one axis of rotation. The model can be either feature-based or feature-free, as the machinability analysis approach presented in this paper does not require feature recognition. Instead, it analyzes basic geometric primitives, line segments from the STL slice file. In addition to checking 2D machinability, the analysis approach can handle 3D machinability by constructing a machinability graph. The results of this analysis yields the portions of the surface segments machinable from a given orientation and tool. The expectation is that the machinability results can be used to determine the minimum set of setup orientations necessary to machine all surface segments using simple 3-axis milling operations. Currently our approach is restricted to 3-axis flat end milling. The machinability analysis of ball-end mills and fillet-end mills may be the focus of future research. In addition, how to extend the current research to the application of 4-axis or perhaps 5-axis milling will likely be another consideration.

References

[1] Suh, S. H., and Lee, J. J., 1998, “Five-axis Part Machining with Three-axis CNC Machine and Indexing Table,” ASME Journal of Manufacturing Science and Engineering, **120** (1), pp. 120-128.

[2] Suh, S. H., Lee, J. J., and Kim, S. K., 1998, “Multiaxis Machining with Additional Axis NC System: Theory and Development,” International Journal of Advanced Manufacturing Technology, **14**(12), pp. 865-875.

[3] Frank, M.C., Wysk, R.A., and Joshi, S.B., 2004, “Rapid Planning for CNC Machining – A New Approach to Rapid Prototyping”, Journal of Manufacturing Systems, **23** (3), pp. 242-255

[4] Vickers, G. W., and Quan, K. W., 1989, “Ball-mills Versus End-mills for Curved Surface Machining,” ASME Journal of Engineering for Industry, **111**, pp. 22-26.

[5] Ip, W. L. R., and Loftus, M., 1993, “The Application of An Inclined End Mill Machining Strategy on 3-axis Machining Centers,” International Journal of Machine Tool and Manufacture, **33**(2), pp. 115-133.

[6] Su, C. J., and Mukerjee, A., 1991, “Automated Machinability Checking for CAD/CAM,” IEEE Transactions on Robotics and Automation, **7**(5), pp. 691-699.

[7] Chen, L. L., and Woo, T. C., 1992, “Computational Geometry on the Sphere with Application to Automated Machining,” ASME Journal of Mechanical Design, **114**, pp. 288-295.

[8] Tang, K., Woo, T., and Gan, J., 1992, “Maximum Intersection of Spherical Polygons and Workpiece Orientation for 4- and 5-axis Machining,” ASME Journal of Mechanical Design, **114**, pp. 477-485.

[9] Gan, J. G., Woo, T. C., and Tang, K., 1994, “Spherical Maps: Their Construction, Properties, and Approximation,” ASME Journal of Mechanical Design, **116**, pp. 357-363.

[10] Chen, L. L., Chou, S. Y., and Woo, T. C., 1993, “Separating and Intersecting Spherical Polygons: Computing Machinability on Three-, Four-, and Five-axis Numerically Controlled Machines,” ACM Transactions on Graphics, **12**(4), pp. 305-326.

[11] Yang, W., Ding, H., and Xiong, Y., 1999, “Manufacturability Analysis for A Sculpture Surface Using Visibility Cone Computation,” International Journal of Advanced Manufacturing Technology, **15**(5), pp. 317-321.

[12] Yin, Z. P., Ding, H., and Xiong, Y.L., 2000, "Visibility Theory and Algorithms with Application to Manufacturing Processes," International Journal of Production Research, **38**(13), pp. 2891-2909.

[13] Gan,J.G., 1990, "Spherical Algorithms for Setup Orientations of Workpiece with Sulptured Surfaces," Ph.D. thesis, University of Michigan, Ann Arbor, MI.

[14] Suh, S. H., and Kang, J. K., 1995, "Process Planning for Multi-axis NC Machining of Free Surfaces," International Journal of Production Research, **33**(10), pp. 2723-2738.

[15] Dhaliwal, S., Gupta, S. K., Huang, J., and Priyadarshi, A., 2003, "Algorithms for Computing Global Accessibility Cones," ASME Journal of Computing and Information Science in Engineering, **3**(3), pp. 200-209.

[16] Balasubramaniam, M., Laxmiprasad, P., Sarma, S., and Shaikh, Z., 2000, "Generating 5-axis NC Roughing Paths Directly From A Tessellated Representation," Computer Aided Design, **32**(4), pp. 261-277.

[17] Frank, M.C., Wysk, R.A., and Joshi, S.B., 2006, "Determining Setup Orientations From The Visibility Of Slice Geometry For Rapid Computer Numerically Controlled Machining", Journal of Manufacturing Science and Engineering, v 128, n 1, February, 2006, pp. 228-238

[18] Haghpassand, K., and Oliver, J. H., 1995, "Computational Geometry for Optimal Workpiece Orientation," ASME Journal of Mechanical Design, **117**(2A), pp. 329-335.

[19] Radzevich, S. P., and Goodman, E. D., 2002, "Computation of Optimal Workpiece Orientation for Multi-axis NC Machining of Sculptured Part Surfaces," ASME Journal of Mechanical Design, **124**(2), pp. 201-212.

[20] Balasubramaniam, M., Sarma, S. E., and Marciniak, K., 2003, "Collision-free Finishing Toolpaths From Visibility Data," Computer Aided Design, **35**(4), pp. 359-374.

[21] Regli, W. C., 1995, "Geometric Algorithms for the Recognition of Features from Solid Models". Ph.D. Thesis, The University of Maryland, College Park, MD.

[22] Regli, W. C., Gupta, S. K., and Nau, D. S., 1995, "Extracting Alternative Machining Features: An Algorithmic Approach," Research in Engineering Design," **7**(3), pp. 173-192.

[23] Gupta, S. K., and Nau, D. S., 1995, "Systematic Approach to Analysing the Manufacturability of Machined Parts," Computed Aided Design, **27**(5), pp 323-342.

[24] Gupta, S. K., Regli, W. C., Das, D., and Nau, D. S., 1997, "Automated Manufacturability Analysis: A Survey, " Research in Engineering Design, **9**(3), pp. 168-190.

[25] Shen, Y. and Shah, J.J., 1998, "Recognition of Machining Features Based on HSPCE Decomposition, Feature Composition, and Process Centered Classification," ASME Journal of Mechanical Design, **120**(4), pp. 668-677.

[26] Gaines, D.M., Castano, F, and Hayes, C.C., 1999, "MEDIATOR: A Resource Adaptive Feature Recognizer that Intertwines Feature Extraction and Manufacturing Analysis," ASME Journal of Mechanical Design, **121**(1), pp. 145-158.

[27] Faraj, I., 2003, "Manufacturing Features: Verification Interaction Accessibility and Machinability," International Journal of Production Research, **41**(10), pp. 2249-2272.

[28] Ferreira, P. M., and Liu, C. R., 1988, "Generation of Workpiece Orientations for Machining Using A Rule-based System," Robotics and Computer Integrated Manufacturing, **4**(¾), pp.545-555.

[29] Demey, S., Brussel, H. V., and Derache, H., 1996, "Determining Set-ups for Mechanical Workpieces," Robotics and Computer Integrated Manufacturing, **12**(2), pp.195-205.

[30] Wu, H. C., and Chang, T. C., 1998, "Automated Setup Selection in Feature-based Process Planning," International Journal of Production Research, **36**(3), pp. 695-712.

[31] Ong, S. K., Ding, J., and Nee, A. Y. C., 2002, "Hybrid GA and SA Dynamic Set-up Planning Optimization," International Journal of Production Research, **40**(18), pp. 4697-4719.

[32] Lozano-Perez, T., 1981, "Automatic Planning of Manipulator Transfer Movements," IEEE Transactions on Systems, Man, and Cybernetics, **SMC-11**(10), pp. 681-698.

[33] Choi, B. K., Kim, D. H., and Jerard, R. B., 1997, "C-space Approach to Tool-path Generation for Die and Mound Machining," Computer Aided Design, **29**(9), pp. 657-669.

[34] Choi, B. K., and Ko, K., 2003, "C-space Based CAPP Algorithm for Freeform Die-cavity Machining," Computer Aided Design, **35**(2), pp. 179-189.

[35] Morishige, K., Takeuchi, Y., and Kase, K., 1999, "Tool Path Generation Using C-space for 5-axis Control Machining," ASME Journal of Manufacturing Science and Engineering, **121**(1), pp. 144-149.

[36] Jun, C. S., Cha, K. D., and Lee, Y. S., 2003, "Optimizing Tool Orientations for 5-axis Machining by Configuration-space Search Method," Computer Aided Design, **35**(6), pp. 549-566.

CHAPTER 6. FUTURE WORK AND CONCLUSION

Future Work

The three manufacturability indicators presented in this research hierarchically analyze the manufacturability of non feature-based objects. Potential manufacturing difficulty is then fed back to a designer upon detection for further modification. This is a process where CAD models are *checked* for manufacturability. In the case of feature-based models, this *checking* process might be sufficient, followed by modification of any non-manufacturable features based on the designer's knowledge and the parametric information for each type of feature. However, in the environment of non feature-based geometries, perhaps manufacturability analysis would not stop at the *checking* stage; rather, it would have the capability of guiding the designer towards a more manufacturable design. The subtle difference is assumed to be on the very nature of the design models. In feature-based models, there are definable boundaries and inter-relationships between the features (a bolt-hole array, for example). In contrast, a non feature-based model, such as an organic shape reverse engineered from a sculpture does not have the same distinct boundaries, intersections or parametric forms. Hence, a further assumption is made; that the redesign of such a model would also be organic, and not be restricted along rigid boundaries and parametric forms. For future work, this dissertation proposes a new redesign process for non feature-based models. The sections below describe a critical new concept for redesign called the *Multi-Layer Visibility Map*.

Multi-Layer Visibility Map

Visibility analysis is a very capable tool to analyze manufacturability for material removal processes and forming processes. Contemporary research on visibility relies on the concept of the unit sphere, on which both visibility and its complementary set, non-visibility, are represented and manipulated as spherical surface patches. Computation of visibility on the unit sphere is then conducted on part of the unit sphere that is occupied by visibility patches. Current visibility based approaches implicitly assume the use of *one-layer* visibility on a unit sphere, where non-visibility areas are not distinguished.

In Fig. 6.1, three non-visibility regions (represented by three different colors) from three obstacles are merged and it turns out that there is no visible direction on the hemisphere. Each invisible direction is treated equally in a single-layer visibility map. However in Fig. 6.2, non-visible regions are distinguished and the intersections of non-visibility regions are identified. The intersections of non-visibility regions suggests that those directions are blocked by more than one layer of obstacles, and that in order to make them visible, a designer would need to modify more than one obstacle. This clearly illustrates that the *single layer* visibility map has the limitation of describing the potential effort involved in *re-designing* a CAD model, since it does not distinguish the magnitude of non-visibility.

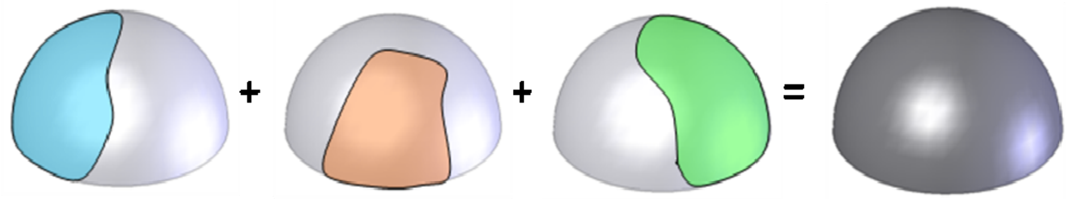


Fig. 6.1 Single-layer visibility map where sum is null set

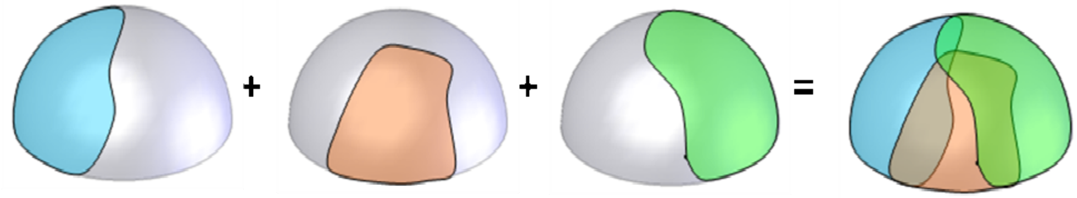


Fig. 6.2 Multi layer visibility map with multi layers of visibility

A *multi-layer* visibility map is constructed by mapping non-visibility cones from all obstacles on the surface of a unit sphere separately. Visible regions are left open; while non-visible regions are covered by one- or multi-layer patches. Such an approach allows the analysis of layers of obstacles that cause non-visibility. A *multi layer* visibility map aims to differentiate non-visible directions with the magnitude of non-visibility; hence it would assist the designer to identify the potentially best method for modifying a design with the minimum amount of changes. As an analogy, all neighboring co-workers in an office of cubicles may be *invisible* to a particular worker; however, some workers are *more* or *less* invisible, based on the number of cubicle walls between them. Hence, there would be more or less work involved to make some particular pairs of workers visible to each other, based on how many walls would need to be moved.

Re-design based on Multi-Layer Visibility Map

Based on the proposed *multi layer* visibility map, a methodology of re-design for increased visibility should be developed as a re-design software module. This corresponds to the first manufacturability indicator – visibility. Upon the detection of non-visible regions, the re-design process will operate through the re-design module embedded into CAD/CAM software. The re-design process would entail two stages:

(1) Construction of *multi layer* visibility maps for selected non-visible regions of the CAD models.

(2) Uniform shrinking of non-visibility cones determines directions to improve visibility that require the least amount of surface changes (Fig. 6.3).

The uniform shrinking process will shrink each non-visibility cone inward, until a visible direction emerges. Such a direction is the accessible direction for material removal processes, with the least amount of surface changes. With the re-design module functioning as an interactive design mechanism, the designer could of course *override* this result and ask the re-design module to continue the uniform shrinking process to search for more favorable solutions.

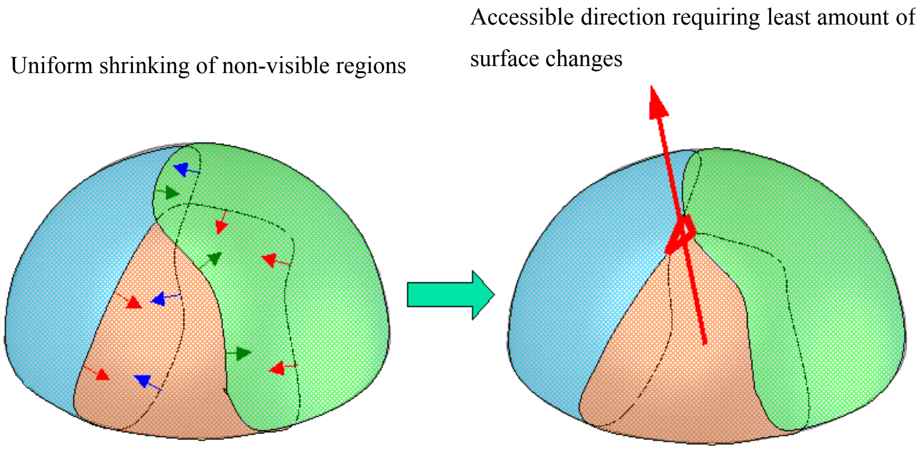

Uniform shrinking of non-visible regions
Accessible direction requiring least amount of surface changes

Fig. 6.3 Re-design for increased visibility

Similar to what has been discussed in re-design for increased visibility, re-design for an axis of rotation could also be developed, intending to improve the visibility around an axis of rotation. The existence of axes of rotation is another measure to determine the manufacturability of a design. Re-design for axis of rotation will reduce the process planning effort and potential errors from re-fixturing the workpiece. Re-design for an axis of rotation is only required when no feasible axis is found. Initially, it would be proposed that visibility will be restricted to be around the desired axis of rotation for this re-designing process (Fig. 6.4). This is based on the goal of making the minimum changes such that an axis of rotation becomes feasible.

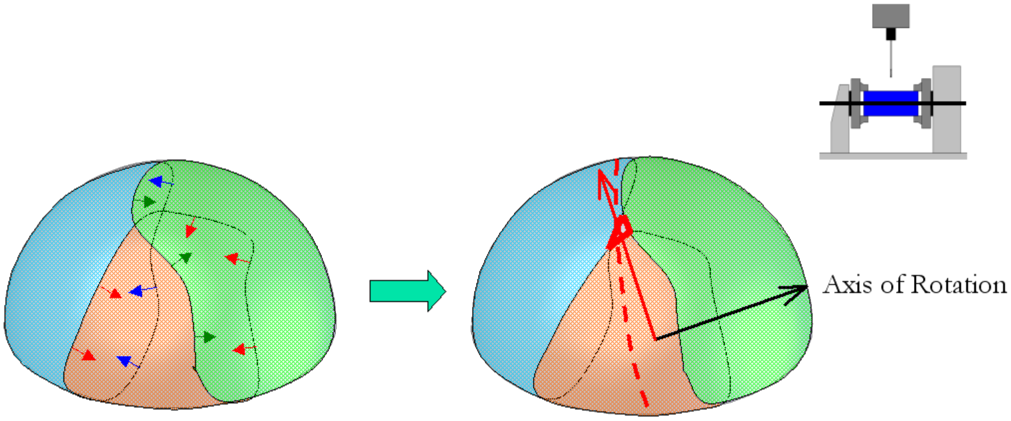


Fig. 6.4 Re-design for 4-axis indexed machining

Bio-Embodied Multi-Layer Visibility Map

As a particular application, a *multi layer* visibility map could find niche application in biomedical engineering, where it could be extended to an *Embodied Multi-Layer Visibility Map* to incorporate bio-information. Medical treatments targeted at internal organs, such as radiation therapy, require penetrating accessibility, which may make Multi-Layer Visibility Mapping a suitable tool. An Embodied Multi-Layer Visibility Map would be created by adding the dimensions of bio-information (e.g. attenuation of radiation beams, organ tolerance to radiation, etc.) onto each non-visible layer of the *Multi-Layer Visibility Map*. Similar to the re-designing process, Embodied *Multi-Layer Visibility Maps* will be constructed at selected point grids on morbid tissue. In contrast to the methods above, a *non-uniform* spherical shrinking process would be implemented on the Embodied Multi-Layer Visibility Map, because different layers of bio-structure are of different importance from a medical perspective (Fig. 6.5).

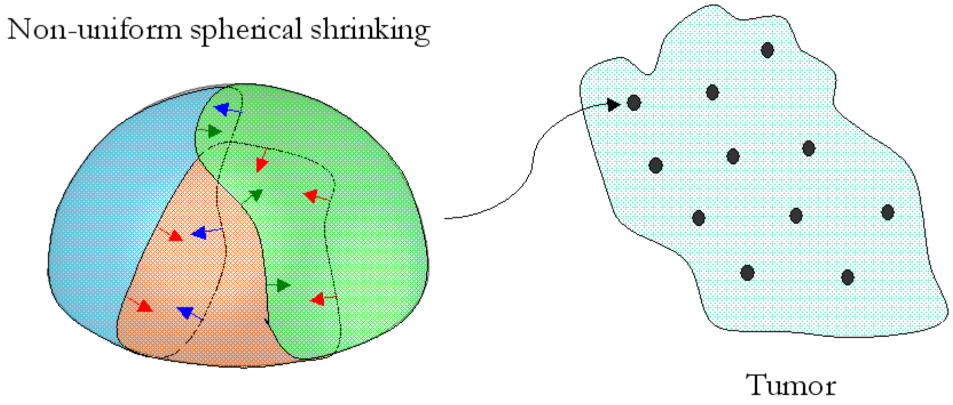


Fig. 6.5 Embodied Multi-Layer Visibility Map

This research would seek to create a tool used in the optimization of radiation therapy with minimum good tissue invasion.

Examples of particular research tasks include:

- Creating *Multi-Layer Visibility Map* for shapes of organs from medical images

- Embodying *Multi-Layer Visibility Map* with radiation-intensity tolerance of organs
- Selecting an optimal set of beam orientations to deliver dose of radiation therapy

Conclusion

In this dissertation, methods for computing three indicators of manufacturability are presented which are not dependent on feature recognition; therefore they can be applied to non feature-based objects. These three indicators are *visibility*, *axis of rotation* and *geometric machinability*. The computations of visibility and axis of rotation are conducted on facets of polygonal models. Though STL files are used in this dissertation, the method itself is not limited to triangular-facetted polygonal models, but is applicable to arbitrary convex planar facets. Similarly, the computation of geometric machinability was conducted on slices of polygonal models. As such, no feature recognition is required for any of the three indicators presented in this work.

Computational results of these three manufacturability indicators can be used either alone or together in a hierarchical manner, to render manufacturability information to designers. Working hierarchically, the methodology starts from computing visibility which describes the potential manufacturability. The existence of visibility for each facet of a polygonal CAD model is the necessary condition for the later two stages of manufacturability analysis. Any non-visibility at this step could be used as feedback, informing the designer of a potential problem. Upon confirmation of the existence of visibility, the manufacturability analysis process can then continue to the next stage; to evaluate accessibility on a specific machine setup. In this work a 4-axis indexed machine setup is used, and the manufacturability translates to finding the existence of axes of rotation, which can guarantee visibility around the axes. The axes of rotation are mapped from visibility information through a rasterization process on the unit sphere. Inexistence of feasible axes of rotation may also suggest the need for more complex machining setups, such as 5-axis machining. The third step of the methodology is to compute geometric machinability, which is related to the cutting tools and the specific cutting direction. At this stage, manufacturability is analyzed for the compatibility of the cutting tool shape and size with surfaces on the CAD model. Non-machinable regions can be determined with specific coordinates.

The manufacturability analysis methods presented in this dissertation do not rely on feature analysis; however they are still applicable to feature-based models as long as the model is represented as a polygonal model. This allows manufacturability analysis of hybrid models that are comprised of both *feature-based* surfaces and other free-form surfaces. Currently, the manufacturability analysis methods are developed for a 4-axis indexed machining setup using flat end mills. Extending the methodology to other machining setups and a general cutting tool could also be part of future work. In closure, this dissertation should lay the groundwork for several new research efforts in feature-free manufacturability analysis, new methods to evaluate design problems, and practical applications that solve problems important to both industry and society.

REFERENCES

[Adalier and Tsatsoulis 1992] Adalier, M. and Tsatsoulis, C., 1992, “Redesigning For Manufacturability Using Reinred”, Applied Artificial Intelligence, v 6, n 3, Jul-Sep, pp. 285-302

[Alexander et. al. 1998] Alexander, P.; Allen, S.; and Dutta, D., 1998, “Part Orientation and Build Cost Determination in Layered Manufacturing”, Computer Aided Design, v 30, n 5, Apr, pp. 343-356

[Balasubramaniam et. al. 2000] Balasubramaniam, M., Laxmiprasad, P., Sarma, S., and Shaikh, Z., 2000, “Generating 5-axis NC Roughing Paths Directly From A Tessellated Representation,” Computer Aided Design, 32(4), pp. 261-277

[Balasubramaniam et. al. 2003] Balasubramaniam, M., Sarma, S. E., and Marciniak, K., 2003, “Collision-free Finishing Toolpaths From Visibility Data,” Computer Aided Design, 35(4), pp. 359-374

[Bandyopadhyay et. al. 1993] Bandyopadhyay, B.P., Hoshi, T., Latief, M.A., and Hanada, T., 1993, “Development Of A Fixture-Free Machining Center For Machining Block-Like Components”, Journal of Materials Processing Technology, v 39, n 3-4, Nov, pp. 405-413

[Barton et. al. 1996] Barton, R.R., Joo, Y., and Ham, I., 1996, “Feedback Of Manufacturing Experience For DFM Design Rules”, CIRP Annals - Manufacturing Technology, v 45, n 1, pp. 115-120

[Boothroyd 1994] Boothroyd G., 1994, “Product Design For Manufacture And Assembly”, Computer Aided Design, v 26, n 7, July, pp. 505-520

[Chen et. al. 1993] Chen, L. L., Chou, S. Y., and Woo, T. C., 1993, “Separating and Intersecting Spherical Polygons: Computing Machinability on Three-, Four-, and Five-axis Numerically Controlled Machines,” ACM Transactions on Graphics, 12(4), pp. 305-326

[Chen and Woo 1992] Chen, L. L., and Woo, T. C., 1992, “Computational Geometry on the Sphere with Application to Automated Machining,” ASME Journal of Mechanical Design, 114, pp. 288-295

[Choi et. al. 2001] Choi, D.S., Lee, S.H., Shin, B.S., Whang, K.H., Yoon, K.K., and Sarma, S.E., 2001, “A New Rapid Prototyping System Using Universal Automated Fixturing With Feature-Based CAD/CAM”, Journal of Materials Processing Technology, v 113, n 1-3, Jun 15, pp. 285-290

[Corbett 1986] Corbett, J., 1986, “Design For Economic Manufacture”, CIRP Annals, v 35, n 1, pp. 93-97

[Das et. al. 1996] Das D., Gupta S. K., and Nau D. S, 1996,"Generating Redesign Suggestions To Reduce Setup Cost: A Step Towards Automated Redesign", Computer Aided Design, Vol. 28, No. 10, pp. 783-782

[Demey et. al. 1996] Demey, S., Brussel, H. V., and Derache, H., 1996, "Determining Set-ups for Mechanical Workpieces," Robotics and Computer Integrated Manufacturing, 12(2), pp.195-205

[Desa et. al. 1987] Desa, S., Nagurka, M. L., and Ghosal, A. 1987, "Product Redesign for Performance, Manufacture, and Assembly: a Rational Methodology Towards Total System Design", Proceedings of the 1987 International Conference on Engineering Design - International Congress on Planning and Design Theory., Boston, MA, USA, pp. 463-472

[Dhaliwal et. al. 2003] Dhaliwal, S., Gupta, S. K., Huang, J., and Priyadarshi, A., 2003, "Algorithms for Computing Global Accessibility Cones," ASME Journal of Computing and Information Science in Engineering, 3(3), pp. 200-209

[Dumitrescu and Szecsi 2002] Dumitrescu, R.and Szecsi, T., 2002, "Implementing Design For Manufacture Rules", Technical Paper - Society of Manufacturing Engineers, MS, n MS02-124

[Faraj 2003] Faraj, I., 2003, "Manufacturing Features: Verification Interaction Accessibility and Machinability," International Journal of Production Research, 41(10), pp. 2249-2272

[Fauvel 1994] Fauvel, O. R. , 1994, "Object-Oriented Design For Manufacture", Journal of Intelligent Manufacturing, v 5, n 1, Feb, pp. 1-11

[Ferreira and Liu 1988] Ferreira, P. M., and Liu, C. R., 1988, "Generation of Workpiece Orientations for Machining Using A Rule-based System," Robotics and Computer Integrated Manufacturing, **4**(¾), pp.545-555

[Finn 1994] Finn F., 1994, "A Seven Step Procedure for Design for Manufacture", World Class Design to Manufacture, Vol. 1 No. 2, pp. 23-30

[Frank et. al. 2002] Frank, M.C., Joshi, S., and Wysk, R.A., 2002, "CNC-RP: A Technique for Using CNC Machining as a Rapid Prototyping Tool in Product/Process Development", Proceedings of the Industrial Engineering Research Conference, Orlando, FL

[Frank et. al. 2003] Frank, M.C., Joshi, S., and Wysk, R.A., 2003, "Rapid Prototyping As An Integrated Product/Process Development Tool: An Overview Of Issues And Economics", Journal of Chinese Institute of Industrial Engineers. Vol. 20, No.3, pp. 239-245

[Frank et. al. 2004] Frank, M. C., Wysk, R.A., and Joshi, S. B., 2004, “Rapid Planning For CNC Milling-A New Approach For Rapid Prototyping”, Journal of Manufacturing Systems, v 23, n 3, pp. 242-255

[Frank et. al. 2006] Frank, M.C., Wysk, R.A., and Joshi, S.B., 2006, “Determining Setup Orientations from the Visibility of Slice Geometry for Rapid CNC Machining”, ASME Journal of Manufacturing Science and Engineering, v 128, n 1, pp. 228-238

[Gaines et. al. 1999] Gaines, D.M., Castano, F, and Hayes, C.C., 1999, “MEDIATOR: A Resource Adaptive Feature Recognizer that Intertwines Feature Extraction and Manufacturing Analysis,” ASME Journal of Mechanical Design, 121(1), pp. 145-158

[Gan 1990] Gan,J.G., 1990, “Spherical Algorithms for Setup Orientations of Workpiece with Sulptured Surfaces,” Ph.D. thesis, University of Michigan, Ann Arbor, MI

[Gan et. al. 1994] Gan, J. G., Woo, T. C., and Tang, K., 1994, “Spherical Maps: Their Construction, Properties, and Approximation,” ASME Journal of Mechanical Design, 116, pp. 357-363

[Gibson and Shi, 1997] Gibson, I. and Shi, D., 1997, “Materials Properties And Fabrication Parameters In Selective Laser Sintering”, Rapid Prototyping Journal, 3, pp.129–136

[Gupta and Nau, 1995] Gupta, S. K., and Nau, D. S., 1995, “Systematic Approach to Analysing the Manufacturability of Machined Parts,” Computed Aided Design, 27(5), pp. 323-342

[Gupta et. al. 1997] Gupta, S. K., Regli, W. C., Das, D., and Nau, D. S., 1997, “Automated Manufacturability Analysis: A Survey” Research in Engineering Design, 9(3), pp. 168-190

[Haghpassand and Oliver, 1995] Haghpassand, K., and Oliver, J. H., 1995, “Computational Geometry for Optimal Workpiece Orientation,” ASME Journal of Mechanical Design, 117(2A), pp. 329-335

[Hague et. al. 2004] Hague, R., Mansour, S., and Saleh, N., 2004, “Material And Design Considerations For Rapid Manufacturing”, International Journal of Production Research, v 42, n 22, Nov 15, pp. 4691-4708

[Hassold 1995] Hassold, R., 1995, “CNC Machining As A Rapid Prototyping Technique”, Modern Machine Shop, v 68, n 5, Oct, pp. 68-73

[Hayes and Gaines 1996] Hayes, C. C., and Gaines, D. M., 1996, “Using Near-Misses From Feature Recognition To Generate Redesign Suggestions For Increased Manufacturability”, Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition, Nov 17-22, Atlanta, GA, USA, pp. 67-77

[Hayes 1996] Hayes, C.C., 1996, “Plan-Based Manufacturability Analysis And Generation Of Shape-Changing Redesign Suggestions”, Journal of Intelligent Manufacturing, v 7, n 2, Apr, pp. 121-132

[Hodgson and Pitts 1991] Hodgson, B.A. and Pitts, G., 1991, “Design For CNC Manufacture*”*, in Corbett J, Dooner M, Meleka J and Pym C (eds). Design for manufacturing strategies, principles and techniques, Addison-Wesley, pp.56-69

[Huang and Mak 1999] Huang G.Q. and Mak K.L, 1999, “Design For Manufacture And Assembly On The Internet”, Computers in Industry 38, pp.17–30

[Hur et. al. 2001] Hur, S.M., Choi, K.H., Lee, S.H. and Chang, P.K., 2001, “Determination Of Fabricating Orientation And Packing In SLS Process”, Journal of Materials Processing Technology, 112, pp. 236–243

[Ji and Lau 1999] Ji, P. and Lau, K.H. , 1999, “Design For Manufacturing: A Dimensioning Aspect”, Journal of Materials Processing Technology, v 91, n 1, pp. 121-127

[Karapatis et. al. 1998] Karapatis, N.P., Van Griethuysen, J.P.S.and Glardon, R., 1998, “Direct Rapid Tooling: A Review Of Current Research”, Rapid Prototyping Journal, v 4, n 2, pp. 77-89

[Koren et. al. 1999] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel,H., 1999, “Reconfigurable Manufacturing Systems”, CIRP Annals - Manufacturing Technology, v 48, n 2, pp. 527-540

[Landers et. al. 2001] Landers, R.G., Min, B.K., and Koren, Y. 2001,“Reconfigurable Machine Tools”, CIRP Annals - Manufacturing Technology, v 50, n 1, pp. 269-274

[Lenau 1996] Lenau, T., 1996, “Missing Element In Design For Manufacture”,CIRP Annals - Manufacturing Technology, v 45, n 1, pp. 105-108

[Levy et. al. 2003] Levy, G. N., Schindel, R., and Kruth, J.P., 2003, “Rapid Manufacturing And Rapid Tooling With Layer Manufacturing (LM) Technologies, State Of The Art And Future Perspectives”, CIRP Annals - Manufacturing Technology, v 52, n 2, pp. 589-609

[Lin et. al. 2003] Lin, J., Thimm, G., and Britton, G.A., 2003, “Alternative Design Specifications Of Prismatic Parts For Manufacturability”, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, v 217, n 4, pp. 569-572

[Liu et. al. 1995] Liu, T.I., Yang, X.M.and Kalambur, G.J., 1995, “Design For Machining Using Expert System And Fuzzy Logic Approach”, Journal of Materials Engineering and Performance, v 4, n 5, Oct, pp. 599-609

[Masood and Rattanawong 2002] Masood, S.H.and Rattanawong, W., 2002, “A Generic Part Orientation System Based On Volumetric Error In Rapid Prototyping”, International Journal of Advanced Manufacturing Technology, v 19, n 3, pp. 209-216

[Merz et. al. 1994] Merz, R., Prinz, F. B., Ramaswami, K., Terk, M., and Weiss, L., 1994, ‘‘Shape Deposition Manufacturing,’’ Proceedings of the Solid Freeform Fabrication Symposium, University of Texas at Austin, August 8–10

[Montero et. al. 2001] Montero M., Roundy S., Odell D., Ahn S.H., and Wright P.K., 2001, “Material Characterization of Fused Deposition Modelling (FDM) ABS by Designed Experiments”, Proceedings of the SME RPA conf., 15-17 May 2002, Cincinnati USA

[Mukerjee and Jain 1997] Mukerjee, A. and Jain, N. K., 1997, “Feature-Less Approach To Process Planning” Proceedings - IEEE International Conference on Robotics and Automation, v 4, pp. 2747-2752

[Ong et. al. 2002] Ong, S. K., Ding, J., and Nee, A. Y. C., 2002, “Hybrid GA and SA Dynamic Set-up Planning Optimization,” International Journal of Production Research, 40(18), pp. 4697-4719

[Ong et. al. 2003] Ong, S.K., Sun, M.J., and Nee, A.Y.C. , 2003, “A Fuzzy Set AHP-Based DFM Tool For Rotational Parts”, Journal of Materials Processing Technology, v 138, n 1-3, Jul 20, pp. 223-230

[Pham and Gault 1998] Pham, D.T. and Gault, R.S., 1998, “Comparison Of Rapid Prototyping Technologies”, International Journal of Machine Tools & Manufacture, v 38, n 10-11, Oct-Nov, pp. 1257-1287

[Poli 2001] Poli C., 2001, “Design For Manufacturing A Structured Approach”, Butterworth-Heinemann, Boston, Massachusetts

[Radstok 1999] Radstok E., 1999, “Rapid Tooling”, Rapid Prototyping Journal, Volume 5. Number 4, pp. 164-168

[Radzevich and Goodman 2002] Radzevich, S. P., and Goodman, E. D., 2002, “Computation of Optimal Workpiece Orientation for Multi-axis NC Machining of Sculptured Part Surfaces,” ASME Journal of Mechanical Design, 124(2), pp. 201-212

[Relvas and Simoes 2004] Relvas, C. and Simoes, J.A., 2004, “Optimization Of Computer Numerical Control Set-Up Parameters To Manufacture Rapid Prototypes”, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, v 218, n 8, August, pp. 867-874

[Regli 1995] Regli, W. C., 1995, “Geometric Algorithms for the Recognition of Features from Solid Models”. Ph.D. Thesis, The University of Maryland, College Park, MD

[Regli et. al. 1995] Regli, W. C., Gupta, S. K., and Nau, D. S., 1995, "Extracting Alternative Machining Features: An Algorithmic Approach," Research in Engineering Design," 7(3), pp. 173-192

[Sarma and Wright 1997] Sarma, S. E. and Wright, P. K., 1997, "Reference Free Part Encapsulation: A New Universal Fixturing Concept", Journal of Manufacturing Systems 16 (1), pp. 35-47

[Schmidt 1997] Schmidt, Joseph. W., 1997, " CNC machining - the Other Rapid Prototyping Technology". SAE Special Publications, v 1233, 970368, Automotive Concurrent/Simultaneous, Engineering, pp. 89-91

[Segal and Campbell 2001] Segal, J.I.and Campbell, R.I., 2001, "A Review Of Research Into The Effects Of Rapid Tooling On Part Properties", Rapid Prototyping Journal, v 7, n 2, pp. 90-98

[Shen and Shah 1998] Shen, Y. and Shah, J.J., 1998, "Recognition of Machining Features Based on HSPCE Decomposition, Feature Composition, and Process Centered Classification,"ASME Journal of Mechanical Design, 120(4), pp. 668-677

[Stauffer et. al. 2003] Stauffer, L., Rule, R. and Ren, H., 2003, " A Template For Design For Manufacture Guidelines", ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference, Sep 2-6 2003, Chicago, IL

[Stoll 1991] Stoll, H.W. 1991, "Design For Manufacturing: An Overview", in Corbett J, Dooner M, Meleka J and Pym C (eds). Design for manufacturing strategies, principles and techniques, Addison-Wesley, pp.107-130

[Stoll 1997] Stoll, Henry W., 1997, "Design For Life-Cycle Manufacture", Proceedings of the 1997 ASME International Mechanical Engineering Congress and Exposition, Nov 16-21, Dallas, TX, pp. 35-47

[Su and Mukerjee 1991] Su, C. J., and Mukerjee, A., 1991, "Automated Machinability Checking for CAD/CAM," IEEE Transactions on Robotics and Automation, 7(5), pp. 691-699

[Suh and Kang 1995] Suh, S. H., and Kang, J. K., 1995, "Process Planning for Multi-axis NC Machining of Free Surfaces," International Journal of Production Research, 33(10), pp. 2723-2738

[S.Wesley and Li 1996] S.Wesley, C. and Li L., 1996, "A Knowledge-Based Design Critique System For Manufacture And Assembly Of Rotational Machined Parts In Concurrent Engineering", Computers in Industry, 32, pp. 117-140

[Tang et. al. 1992] Tang, K., Woo, T., and Gan, J., 1992, “Maximum Intersection of Spherical Polygons and Workpiece Orientation for 4- and 5-axis Machining,” ASME Journal of Mechanical Design, 114, pp. 477-485

[Taylor 1997] Taylor G. Don, 1997, “Design For Global Manufacturing And Assembly”, IIE Transactions, Journal of Design and Manufacturing, 29, pp. 585-597

[Tu et. al. 1998] Tu, Y., Yang, W., and Xiong, Y., 1998, “A Concurrent Manufacturing Strategy For One-Of-A-Kind Products With Complicated Sculptured Surfaces”, International Journal of Advanced Manufacturing Technology, v14, pp.93-98

[Wang et. al. 1999] Wang, F., Marchetti, L. and Wright, P. K., 1999, “Rapid Prototyping Using Machining”, Technical Paper - Society of Manufacturing Engineers. PE, n PE99-118, p PE99-118-1 - PE99-118-6

[Wu and Chang 1998] Wu, H. C., and Chang, T. C., 1998, “Automated Setup Selection In Feature-Based Process Planning” International Journal of Production Research, **36**(3), pp. 695-712

[Yan and Gu 1996] Yan, X. and Gu, P., 1996, “Review Of Rapid Prototyping Technologies And Systems”, CAD Computer Aided Design, v 28, n 4, Apr, pp. 307-318

[Yang et. al. 1999] Yang, W., Ding, H. and Xiong, Y., 1999, “Manufacturability Analysis For A Sculptured Surface Using Visibility Cone Computation”, International Journal of Advanced Manufacturing Technology, v 15, n 5, pp. 317-321

[Yin et. al. 2000] Yin, Z. P., Ding, H. and Xiong, Y.L., 2000, “Visibility Theory and Algorithms with Application to Manufacturing Processes,” International Journal of Production Research, 38(13), pp. 2891-2909

[Zhou and Gaines 2003] Zhou X. and Gaines D. M., 2003, “Arm: A Tool For Identifying And Repairing Un-Machinable Shapes In Designs”, Robotics and Computer Integrated Manufacturing 19, pp. 343–353

ACKNOWLEDGEMENTS

The completion of my PhD dissertation would not be possible without the help from those people who supported me in all aspects of my life. I would like to take this opportunity to express my most sincere thankfulness to them. First and foremost, my parents Guangcheng Li and Bingli He for them bringing me up and giving me their immense love. I am very blessed to have such great parents who constantly give me the best they have for my life and education.

Next, I would like to deeply thank my adviser Dr. Matthew Frank for his guidance, support and contribution throughout this research and the completion of this dissertation. He is always trying to help me in every way. I wish every PhD student had such a great major professor. I am also thankful to my committee members for their comment and advice on my work. My special thank goes to Dr. Frank Peters who mentored me through the *Preparing Future Faculty* class and gave me a lot of professional advice.

My thankfulness extends to two faculty members in the Department of Agricultural and Bio-System Engineering, Dr. Joseph Chen who shared with me his valuable teaching skills and professional experience, and Dr. Steven Mickelson who provided me the teaching opportunity at the Industrial Technology program.

I cannot forget to thank those students in my research group, Wutthigrai Boonsuk, Xiaoming Luo, Alex Renner and Jerry Cao. They gave me many helps in developing the software for the research, and the group discussion with them was inspirational.

Last but not least, I would like to thank all the brothers and sisters in the Church in Ames. It is them who encouraged and strengthened me while I was in sorrow; it is them who rekindled my hope for a better tomorrow in Jesus Christ.