COMPUTER-AIDED CMM INSPECTION PLANNING AND VERIFICATION

COMPUTER-AIDED CMM INSPECTION PLANNING AND VERIFICATION

By

ANIS LIMAIEM, B.Sc. Hons (Mech. Eng.), M.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctorate of Philosophy

McMaster University

(c) Copyright by Anis Limaiem, September 1996

DOCTOR OF PHILOSOPHY (1996) MCMASTER UNIVERSITY

(Mechanical Engineering) Hamilton, Ontario, Canada

TITLE: Computer Aided CMM Inspection Planning and Verification

AUTHOR: Anis Limaiem
B.Sc. Hons (Mech. Eng.) (Ecole National d'Ingenieurs, Tunis)
M.Sc. (Mech. Eng.) (Ecole Polytechnique, Montreal)

SUPERVISOR: Dr. Hoda A. ElMaraghy

NUMBER OF PAGES: xxiv, 202

To My Parents

ABSTRACT

Inspection planning and verification procedures are central activities in any inspection using Coordinates Measuring Machines (CMMs). The objective of tactile CMM inspection planning is to establish the best sequence of inspection steps with a detailed inspection procedure for each inspection feature or cluster of measurement points. Algorithms are then used to interpolate these points and generate their mathematical model(s). Complex surfaces may need to be decomposed into several patches which are interpolated separately then joined in a single model by determining their intersections. These models (curves and surfaces) are then used by tolerance analysis algorithms to verify if tolerance specifications are met by comparing the interpolated model representing the actual surface and the CAD model representing the theoretical surface.

This dissertation presents a computer-aided CMM inspection planning system as well as new tools for the interpolation and manipulation of measured features. The inspection planning system is modular and integrates all planning tasks.

A new algorithm is developed for determining inspection accessibility domains (or cones). An improved discrete accessibility algorithm is developed for probes with discrete possible orientations. A novel formulation of the problem of measurement points clustering and probe orientation selection in terms of operations sequencing and resources allocation is devised and adopted. A method for optimum clustering and sequencing of measurement points has been developed. The criteria in this case are the minimum number of clusters, the minimum number of resources used and finally the minimum distance travelled by the probe. A collision-free shortest probing path algorithm is enhanced. A modular Computer-Aided Inspection Planning (CAIP)

system which integrates inspection planning tasks was developed and validated. Examples of actual parts have been used, tested and simulated.

A new method for the incorporation of uncertainties as well as linear constraints in the interpolation model based on dual Kriging interpolation has been developed. A new curve/surface formulation of dual Kriging as a combination of interpolation profiles is proposed, hence extending its use for solids and n-D entities interpolation, as well as sweeping, skinning and blending. In addition, dual Kriging was generalized to incorporate NURBS and B-splines. Finally, geometric algorithms, as opposed to numerical, analytical or differential algorithms for the intersection and manipulation of curves and surfaces are developed. Algorithms for the intersection of parametric/implicit and parametric/parametric entities (curves and surfaces) as well as for the projection of points on curves and surfaces have been developed, implemented and validated.

The results of this work are intended to fill voids which exist in previous works in inspection planning and verification. These are: the integration of the different tasks involved in CMM inspection planning in order to develop an automated and robust inspection planner based on formalized and integrated approaches for accessibility analysis, optimum measurement operations sequencing and resources allocation, and accurate representation and manipulation of measured curves and surfaces.

ACKNOWLEDGMENT

I would like to thank my supervisor, Dr. Hoda A. ElMaraghy, for her guidance and support throughout the course of this research. I would like also to express my gratitude to the members of my supervisory committee Dr. Allan Spence, Dr. Mateusz P. Sklad and Dr. Robert Sowerby. I am very grateful to Dr. Hoda ElMaraghy, the Mechanical Engineering Department, and McMaster University for the Research Scholarships and Teaching Assistantships I have received for financial support. I would like to thank the Mechanical Engineering Department secretaries Jane Mah, Louise Perry and Betty Bedell-Ryc and the FMS secretary Barbara Nethercot.

I would like to thank everyone who helped me during my stay in the Flexible Manufacturing Center, McMaster University, namely Todd Pfaff the system engineer.

I would like to extend my thanks to every one who helped me during may stay in the Intelligent Manufacturing Systems Lab, namely Dr. Waguih ElMaraghy, Dan Corrin the system engineer and Patricia Jackson and Penny Lappan the administrative secretaries.

My final thanks go to my colleagues and dear friends, Dr. Ashraf Nassef, Dr. Hassen Zghal, El-Houssaine Waled and all my freinds in the IMS Lab.

Finally, I present this work to my parents as well as my brothers and sisters.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter introduces the Computer-Aided CMM Inspection Planning and Verification System developed in this thesis. The first section of this chapter describes the background and motivation of this work. The second section defines the problem of inspection planning and geometric modeling. Finally, section three presents the objectives and an overview of the thesis.

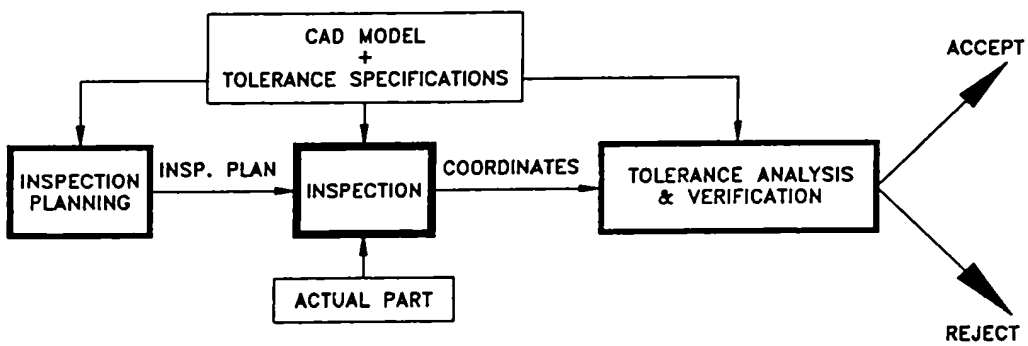


Figure 1.1 Inspection Planning and Verification

1.1 BACKGROUND AND MOTIVATION

Inspection using Coordinate Measuring Machines (CMMs) is the central activity in the inspection planning and verification procedure (Figure 1.1). Inspection planning is the upstream off-line activity which establishes a detailed inspection procedure for the workpiece. The downstream activity is tolerance verification which generally takes place on-line and is based on interpolation algorithms as well as tolerance analysis and verification algorithms.

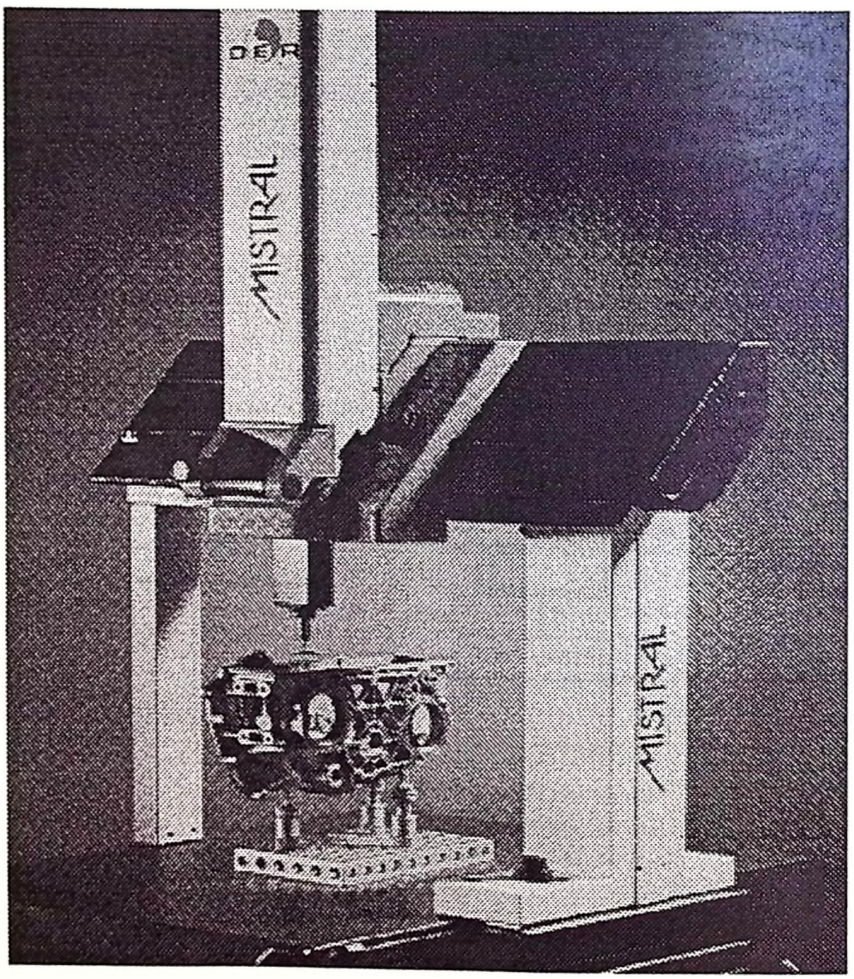


Figure 1.2 Coordinate Measuring Machine

1.1.1 Inspection Planning

Inspection planning is a very difficult and complex task usually performed by experienced operators. In modern manufacturing characterized by low volume, high variety production, close tolerances and high quality products, the speed and accuracy of the traditional approaches to dimensional inspection and lack of integration with CAD can create bottlenecks in production. More automated inspection process planning systems and better decision support tools for human planners are needed. This can be achieved by adopting and developing an integrated Computer-Aided Inspection Planning (CAIP) approach (ElMaraghy, H. A. and ElMaraghy, W. H., 1994). The anticipated advantages include better utilization of resources, improved quality control and flexibility, as well as rationalization of decisions due to improved logic formalization.

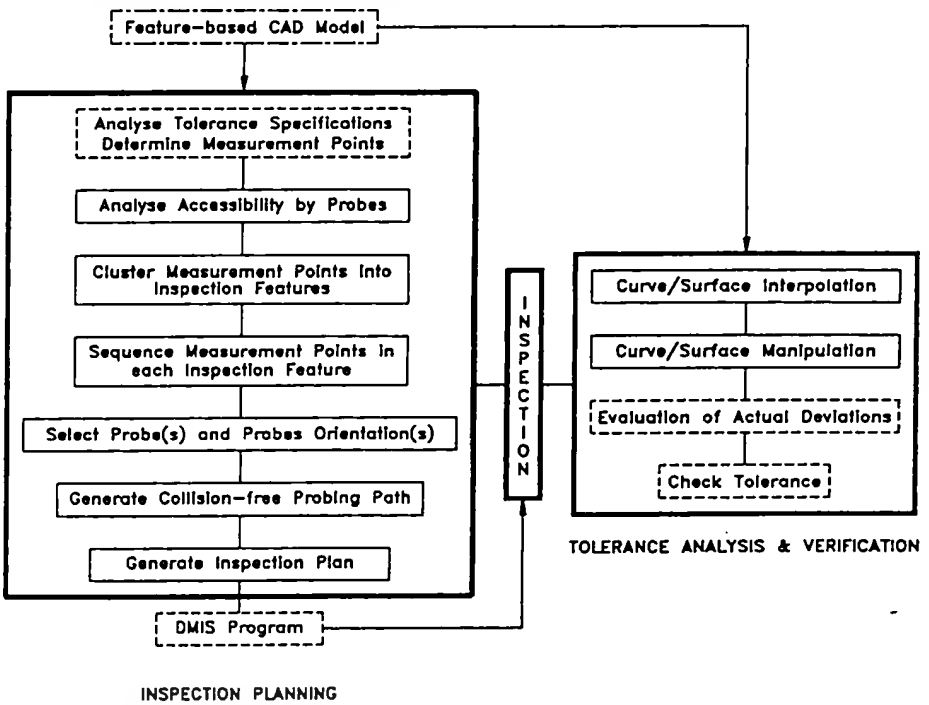


Figure 1.3 Main Functions of a Tactile Inspection Planning System

The objective of tactile CMM inspection planning is to establish the best sequence of inspection steps with a detailed inspection procedure for each inspection feature or cluster of measurement points. This can be achieved in seven main steps (Figure 1.3): 1) analyze tolerance specifications in order to determine the measurement points necessary for the inspection, 2) analyze the accessibility of the measurement points by the probe(s) and determine their accessibility domains, 3) group the measurement points into clusters or inspection features using the same set-up, same probe and same probe orientation. 4) sequence measurement steps according to stated criteria, 5) select most suitable probes and probe orientations, 6) generate collision-free probing paths, and finally, 7) generate an inspection plan and a DMIS program.

The need to automate inspection planning appeared with the advent of programmable CMMs and more accurate and flexible machines (i.e. motorized probe heads). In addition to the growing importance of quality control and the need to conform to existing quality standards as well as the generalization of integrated design-manufacturing-inspection environment in industries. Indeed, the efficiency of the inspection plan is evaluated in terms of cost which is closely related to inspection cycle time. The issue of minimizing the cost of the inspection plan is therefore very important. The expertise of the planner and the CMM operator has to be complemented with mathematical tools which can handle efficiently and optimally critical and time consuming tasks such as the analysis of the accessibility of inspection features by measurement probes, the clustering and sequencing of measurement operations while minimizing set-up changes, probe changes and probe orientation changes and the optimized planning of the probing path. On the other hand, more accurate representations of measured features (curves/surfaces) allow better evaluation of tolerances, thus reducing rejection and reworks.

In order to achieve the goal of automating the inspection process plan, a rationalization and formalization of the different tasks and the knowledge related to each step in the inspection process is vital. Many attempts to achieve the goal of automating inspection planning have been reported in the literature. However, major challenges remain, such as 1) the optimization of the inspection

plan with or without considering cycle time constraints and 2) a general procedure for checking the accessibility of the inspected features of parts. Heuristics have been used extensively to find acceptable solutions; however, the need for optimization is a priority where tight time constraints are present in order to improve productivity and competitiveness. The optimization issue has been considered by previous works locally where the choice of strategic parameters is made sequentially, for example, selecting part orientation followed by examining feature accessibility and generating measurement points. However, a better approach would be to use a least commitment principle, which better describes the highly coupled problem especially with a global optimization goal. This avoids the early elimination of good solutions.

The main deficiencies are found in high level inspection planning, especially in the accessibility analysis of measurement points. In fact, no robust general approach for the inspection probe accessibility analysis has been proposed yet; however, methods for the computation of the accessibility domains were developed in the special case of polyhedral objects. The issue of optimizing inspection plans has not been considered properly. Only feasible plans were sought and heuristics were used extensively. Even though attempts have been made to optimize the inspection plans at the low level, the results are insignificant from a global viewpoint. A formalized inspection planning approach is needed in order to achieve an automated and integrated computer-aided inspection planning (CAIP) system.

1.1.2 Geometric Modeling

Coordinate measuring machines are very accurate and powerful tools for the measurement of distances between identified points. They generally produce 3-D data points or coordinates. Algorithms are then used to interpolate these points and generate their mathematical model(s). Complex surfaces may need to be decomposed into several patches which are interpolated separately then joined in a single model by determining their intersections (see Figure 1.4). These models (curves and surfaces) are then used by tolerance analysis algorithms to verify if tolerance

specifications are met.

Another field where contributions are needed is in the modeling, manipulation and interpolation of complex measured surfaces. This is very important if we consider the fact that the verification of specified geometric dimensions and tolerances is based on the comparison of the interpolated model representing the actual surface and the CAD model representing the theoretical surface. In addition, the verification of tolerances generally involves complicated and long iterative calculations. The cost of these calculations depends heavily on the interpolation model as well as the methods used. These are generally analytical algorithms for solving non-linear equations and can be unstable as in the case of Non Uniform Rational B-splines (NURBS), Bezier and Coons curves and surfaces (Menq et al., 1992a, Piegl and Tiller, 1987). This is especially important for on-line verifications systems. In addition, it is important to include measurement errors in the surface interpolation model in order to better describe the actual surface with a smooth model and overcome the problem caused by random errors which generate perturbations in the interpolated surface.

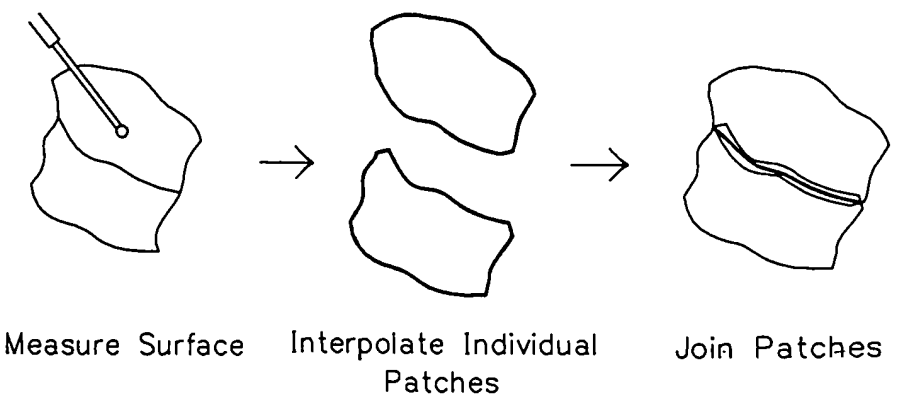


Figure 1.4 Steps in Inspection and Verification

1.2 PROBLEM DEFINITION

The objective of this research is to develop an automated and integrated inspection planner for coordinate measuring machines and a set of tools for the interpolation and manipulation of measured surfaces.

The part dealing with inspection planning deals with five main tasks:

1. Analyzing the Accessibility of Measurement Points and Features
2. Clustering of Measurement Points
3. Sequencing of Measurement Points
4. Generation of Collision-free Probe Path
5. Integration of Previous Tasks in a CAIP System

The second part dealing with the manipulation of measured surfaces includes the following tasks:

1. Surface Modeling
2. Surface Manipulation

1.2.1 Inspection Planning

Accessibility Analysis may be considered the most important task in high level inspection planning. The lack of formalized and general analysis methods was the main problem in previous inspection planning efforts. In addition, existing approaches impose restrictive abstractions and simplifications which generally eliminate good solutions at an early stage of the planning. A general accessibility analysis method which will determine all possible inspection probe orientations that can safely access a given measurement point is needed. The actual shape and size of the probe should also be considered and post-verification should be avoided.

Measurement points clustering is part of high level inspection planning. It is the foundation of any optimization procedure. Two cases are considered:

- Clustering without precedence constraints
- Clustering with precedence constraints

The problem is to generate the optimum clusters of measurement points according to certain criteria including: 1) minimum set-up changes, 2) minimum probe changes and 3) minimum probe orientation changes. Precedence constraints between inspection features may also be considered. Examples: 1) inspect tight or critical tolerances before the loose or less important tolerances in order to avoid costly late rejections. 2) Inspect datum features first in order to define the datum reference frame. Precedence constraints generally depend on the software used to analyze and verify the tolerances.

The clustering process produces clusters of measurement points not arranged in any particular sequence especially when no precedence constraints are imposed. The aim is then to sequence the measurement points properly within these clusters such that the overall distance travelled by the probe is minimized.

Collision-free probe path generation is the final part in the inspection planning process. The path between each two points without collision or interference with the part and its environment has to be determined. The objective is to generate the shortest collision-free probe path.

It is important to integrate the previous tasks into one modular CAIP system with an effective user interface. The input to the system would consist of the part solid model and its orientation, measurement points (their number and location) as well some characteristics of the CMM and the probes, and eventually approximated representations of the part and its environment. The output would include the clusters of measurement points with the specified probe and its orientation(s) as well as the probe tip path.

1.2.2 Geometric Modeling

The objective is to interpolate a surface from a set of measurement points (obtained with a CMM) while considering a maximum deviation imposed by the possible errors at the measured points, or interpolate a set of curves obtained using a scanning probe.

The problem of determining the intersection of two geometric entities is encountered each time curves and surfaces are used. The existing methods rely mainly on analytical approaches and are very sensitive to the type of interpolation used (eg. NURBS) as well as the initialization of the algorithms. The aim is to develop a set of general algorithms to compute the intersection of implicit/parametric and parametric/ parametric curves and surfaces.

1.3 OVERVIEW OF THE THESIS

This dissertation is divided into nine chapters and three appendices.

Chapter 1 presents the background and motivation of this research as well as the problem definition and this overview of the thesis.

Chapter 2 describes some relevant aspects of CMM technology. It provides a review of the important works related to CMM inspection planning and related to the modeling and manipulation of measured surfaces. It concludes with a discussion of this survey.

In chapter 3 the method developed for applying accessibility analysis to inspection planning is described for both exact as well as approximate approaches.

Chapter 4 presents a method integrating the sequencing of measurement operations and allocation of resources is described for the case of imposed precedence constraints. A method for clustering measurement points when no precedence constraints are considered is also described. This chapter concludes with a discussion of the previous approaches.

In chapter 5, an introduction to the problem of path planning and collision avoidance and their complexity is presented. Then, the approaches adopted for work space discretization, objects

approximation and interference detection are described. A discussion is provided at the end of the chapter.

Chapter 6 generalizes the concept of dual Kriging applied to geometric modeling and presents its theoretical foundations. An overview of dual Kriging, its different applications and main advantages are introduced. Dual Kriging applied to the interpolation of parametric curves, surfaces and solids is described. The introduction of uncertainties and linear constraints in the interpolation model is presented as a direct application of measured surfaces modeling followed by a general discussion.

Chapter 7 deals with the intersections in geometric modeling. The methods used to solve the problem of curve/curve, curve/surface and surface/surface intersection are overviewed. An algorithm designed for the intersection of implicit/parametric objects is presented. In addition a novel method for the orthogonal projection of a point onto a curve and onto a surface is described. An algorithm for the intersection of parametric curves and surfaces based on successive orthogonal projections is described. A significant enhancement of the intersection algorithm using successive orthogonal projections based on the intersection of tangent vectors or planes is presented. A discussion of the different methods is provided.

Chapter 8 presents the structure and implementation of the Computer-Aided CMM Inspection Planning System developed in this research. Two examples of actual parts are used to test, validate and discuss previous approaches and algorithms.

The dissertation is concluded in chapter 9 with a discussion of the achievements and contributions, conclusions and future research.

The thesis includes three appendices. Appendix A contains a description of the branch & bound algorithm applied to the Travelling Salesman Problem. Appendix B presents Dijkistra's algorithm. Finally, Appendix C provides detailed examples of application of dual Kriging for curve, surface and solid interpolation.

CHAPTER 2

LITERATURE SURVEY

This chapter reviews some of the important aspects of CMM technology in the first section. A review of the important works which emerged in the fields of CMM inspection planning and geometric modeling and curve/surface manipulation is provided in the second and third section respectively. A discussion of this survey is presented in section four.

2.1 CMM TECHNOLOGY

Coordinate Measuring Machines (CMMs) are versatile, have gained widespread use, and are the most precise devices when compared with manual and non-contact measurement devices. They offer a measurement accuracy ranging from 0.001" to 0.0001" (ElMaraghy, H. A. and ElMaraghy, W. M., 1994), with an excellent repeatability. These machines can be operated in four modes: manual, teach mode, off-line interactive programming and automated planning and program generation. CMMs main advantages are: 1) reduced set-up and fixture cost, 2) faster inspection process (measurement time cut by up to 24:1 ratio (Anon, 1990)), thus allowing more sampling, and 3) easier statistical process control, thus reducing scrap and rework.

A CMM is basically a Cartesian robot. It can have one of four main construction designs including Cantilever, Bridge, Horizontal and Gantry type (Figure 2.1).

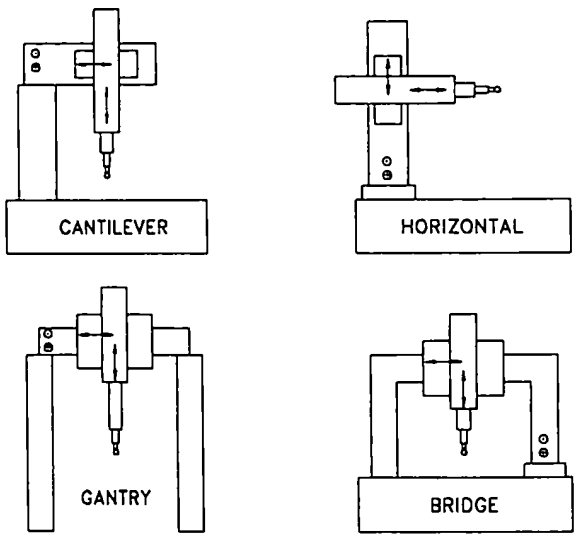


Figure 2.1 Different Types of CMMs

Touch probes are most commonly used for measurements. They can have single or multi-tips of various size and stylus length. Motorized probes, such as Renishaw indexable head, allow changing the orientation of the probe stylus in small incremental rotations about two axes. They can be easily integrated within a manufacturing cell to provide timely feedback for improving production quality. CMMs have many advantages compared to non-contact inspection devices such as laser scanners. They are more accurate and have a uniform precision over a wide range, whereas non-contact devices are less accurate with a trade-off between range and accuracy. However, non-contact devices (i.e. laser scanners) are not sensitive to normal temperature variations, and allow for better control of the amount of data generated, compared to the large amounts of data generated and the need for data reduction. Touch probes can reach many internal features that are otherwise not accessible by laser beams, the movement of parts during measurements is not required, thus making inspection of large and bulky parts easier; however, no fixturing is needed for non-contact inspection.

2.1.1 Inspection using CMMs

The procedure followed by a CMM operator for the inspection of a mechanical part based on a set of specified Geometric Dimensioning and Tolerances (GD&T) depends heavily of the experience of the CMM operator and the inspection software utilized with the CMM. The contributions of the operator is especially critical when no detailed inspection plan is provided, in this case, the contributions of the operator are mainly in high level inspection planning (i.e. orientation of the part, sequencing of inspection feature, orientation of the probe ...). On the other hand, the inspection software organizes the lower level inspection for each inspection feature (i.e. alignment of the part with the reference frame of the CMM, number of measurement points for each type of tolerance, ...). A typical inspection procedure is composed of the following tasks:

* Choose the orientation (or set-up) of the part according to the features to be inspected, their datum and according to the geometry of the part.
* Align the reference frame of the part with the reference frame of the CMM. The operator generally takes some measurement points on the workpiece, an algorithm makes the alignment by determining the translation and rotations necessary to transform from the reference frame of the part to the reference frame of the machine and vice-versa.
* Choose the probe
* Choose a probe orientation with which the maximum number of measurements may be performed.
* For each of the inspection features accessible in a given orientation, follow the procedure dictated by the software. For each tolerance, take some measurement points (by moving the probe manually) which are then used by the software to check if the tolerance is verified.

In the case of an automated inspection procedure (i.e. NC CMMs), the operator has the minimum interaction in the inspection process.

2.1.2 Programming CMMs

CMMs can be operated in four modes:

1. Manual: the operator moves the probe by hand so that the probe tip touches all of the specified measurement points. Then, the encoders determine the position of each measurement point.

2. Teaching mode: the probe is moved manually from one measurement point to the next, while the computer records the probe path.

3. Off-line interactive programming: in this case the CMM program is generated without physically engaging the CMM. Design data and expert knowledge are used to generate the inspection plan which is then translated to either a machine language or a standard language interface such as the ANSI standard Dimensional Measuring Interface Specifications (DMIS) (ASME, 1990). Path planning has to be performed using off-line simulators.

4. Automated Planning and Program Generation: in this case the CMM program is generated automatically. A formalized and systematic procedure (or strategy) is adopted for the inspection plan generation. Expert knowledge as well as clustering methods for grouping inspection features, feature accessibility analysis and collision-free path planning have to be combined in order to automatically generate an inspection plan.

2.1.3 Measurement Uncertainty for CMMs

There are 25 sources of uncertainty in CMM measurements as reported in (Huang and Wu, 1992, Pahk and Kim, 1993 and Phillips et al., 1993). They can be grouped into four main categories, as follows:

1. Probe Properties

- Effective stylus diameter (set by probe calibration)

* Offset vector between different styli (set by multi-styli calibration)
* Probe lobbing (systematic direction-dependent probe error)
* Probe repeatability
* Stylus bending
* Indexable probe head repeatability
* Stylus (or probe) changer repeatability
* Probe spatial frequency response and frictional effects

2. CMM Properties

* Errors in rigid body geometry (at standard temperature)
* Non-rigid body geometry errors (quasi-static conditions)
* CMM part loading effects
* CMM dynamic behavior
* CMM repeatability
* Algorithm accuracy
* Thermally induced errors in a uniform and constant, but non-standard temperature environment
* Other environmental factors
* Variations in utility services: air pressure, electrical power and water supply.

3. Part Properties

* Part dynamics (part bending under probing force and vibration)
* Part fixturing
* Part thermal properties

4. User Selected Properties

* CMM operating parameters
* Sampling strategy (incomplete part geometry information)

* Part location and orientation

It is very important to take into account these errors and to minimize their effects on measurement quality and precision. This can be achieved physically by reducing or eliminating the causes of uncertainty, by direct corrections and compensation in the measurement results or by errors calibration.

2.2 INSPECTION PLANNING

Many attempts to achieve the goal of automating inspection planning have been reported in the literature. However, major challenges remain such as: 1) optimizing the process plan with or without cycle time constraints and 2) developing a general procedure to check the accessibility of the inspected features of parts. The optimization issue has been considered by previous works locally in a sequential approach where the choice of strategic parameters is made sequentionally, for example, part orientation followed by feature accessibility, then generation of measurement points. However, another approach using a least commitment principle, which better describes the highly coupled problem especially with a global optimization goal, would be preferable. This avoids eliminating good solutions early in the search.

2.2.1 The Work of ElMaraghy and Gu

ElMaraghy and Gu (1987) developed the first expert system for inspection planning (see Figure 2.2). They created a generative task planning method for CMMs based on a feature-oriented modeling approach using a feature based modeler: Intelligent Product Design and Manufacturing (IPDM) (ElMaraghy, 1991). Inspection features are grouped according to their datums, and are assigned inspection priorities based on the nature and the magnitude of the related tolerances. Their accessibility in a given part orientation is also checked.

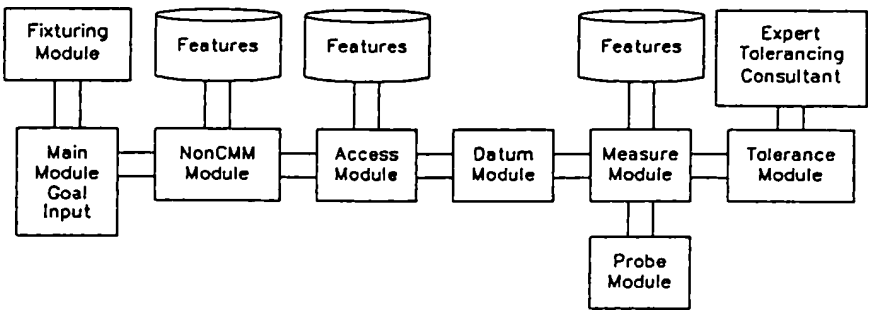


Figure 2.2 Inspection Planning System Model (ElMaraghy and Gu, 1987)

The expertise of human inspection planners has been transcribed into expert rules and syntactic pattern recognition and used for clustering features to be inspected according to the above criteria. The CMM characteristics as well as the inspected part function and geometric properties were also considered while generating the inspection plan.

The inspection planning is realized at three main stages:

1. Suitability for CMM Inspection: The system analyses the specified tolerances or the desired measurement accuracy of each feature to decide whether to use a CMM or a more accurate and/or specialized inspection equipment. For inspection using CMMs the system decides on the type and size of probes based on the specified tolerances or the desired measurement accuracy and the accessibility of the features.

2. Features Clustering: Important features in terms of function and geometric accuracy are inspected first. The search for datums is the starting point of an inspection sequence planning based on geometric tolerances and their datums. After all features in one orientation are examined, the measurement planning phase takes place. An appropriate probe is chosen to carry out the measurement. Probe selection is based on the feature's geometric properties, location of the part and corresponding tolerances according to

predefined rules.

3. Feature Accessibility: This task is determined and inspection steps are arranged so that the changes of the part and/or the probe orientation are identified, planned and minimized.

ElMaraghy and Gu (1987) heuristically optimized the inspection plan; however, cycle time optimization has not been addressed. Alternate part orientation and consequently alternate inspection plans were considered. However, features accessibility should be performed before features clustering in order to have a least commitment strategy. In addition, a more rigorous approach is needed to determine features accessibility and part orientation.

2.2.2 The Work of Menq and his Team

Menq and his team (Sahoo and Menq, 1991, Menq et al., 1992, Ge et al., 1992) developed a Computer-Integrated Dimensional Inspection System for objects having sculptured surfaces. A recent accessibility analysis approach based on a ray-tracing algorithm was proposed by Lim and Menq (1994). The length of the probe is assumed to be infinite and all possible angles of the probe head (which must be finite) are investigated by ray tracing. The 3-D accessibility cone is transformed into a 2-D map where only the orientations of the probe expressed by two angles in the spherical coordinate system are considered. This map is further digitized to lay out the discrete set of possible angles. Corrections are performed to take into account the actual shape of the probe. A heuristic search is finally performed in order to determine the optimal probe direction for a set of inspected points.

The system consists of five modules:

1. Inspection Specification Module: This translates functional requirements, tolerances, manufacturing parameters and CMM constraints into the inspection specification.

2. Automatic Inspection Planning: This generates the probe path taking into account

manufacturing accuracy and tolerance specifications. The path is verified against collision, where-upon three levels of interference are considered: interference with the probe, with the stylus and with the column. If a collision is detected, then a heuristic modification of the CMM path takes place. Measurement data, the design model and inspection attributes are processed by the comparative analysis module to generate an inspection report. A statistical sampling plan is used to determine the inspection sample size based on manufacturing accuracy and tolerance specification. A sensitivity analysis is also performed to avoid singularity when using the localization algorithm (Menq et al, 1992).

3. CMM Verification: Here the inspection path is dynamically simulated using a CATIA workcell. In addition to generating a collision-free path, the necessary time for part inspection and consequently cost estimation are computed.

4. CMM Execution: This module is composed of two major parts: part alignment and automatic inspection. Part alignment is realized by either the traditional 3-2-1 method or by a CAD model-based localization algorithm that facilitates the integration of the CMM in the CIM environment.

5. Comparative Analysis: Each measured surface is compared with the surface specified by design, then a validation based on the tolerance for features and tolerance envelope for surface and curve profiles is performed. An optimal matching algorithm is used for eliminating the offset error between measured and design data to obtain the minimal total deviation. A decomposition into deterministic and random error can be performed.

The core of the system is a knowledge-based inspection planner. Its strength lies in generating a collision-free probe path for inspecting complex surfaces, verifying them using dynamic simulation, and creating the algorithm for matching measured and design data. The choice of a normal orientation of the probe for each measured point imposes a re-setting of the probe at

each measured point, which increases considerably inspection time and consequently inspection cost. In addition, the issues of generating the best inspection sequence and analyzing of feature accessibility problems rigorously have not been addressed.

2.2.3 The Work of Spyridi and Requicha

Spyridi and Requicha (1990, 1991, 1993, 1994), present and discuss an overall strategy for inspection planning. Given a polyhedral solid model of the part and associated tolerancing information, the planner produces a partially-ordered collection of set-ups and a partially-ordered set for each set-up. The planner follows the principle of least commitment which considers as many options as possible at each stage of the planning. For each feature to be inspected, all accessible directions are computed, then a minimal set of directions that enable inspection of all the features is determined.

An overview of the approach is as follows: given a solid model of the part and associated tolerancing information, the planner produces a partially-ordered collection of set-ups and a partially-ordered set for each set-up. The planner follows the principle of least commitment which considers as many options as possible at each stage of the planning. Accessibility considerations play a crucial role in probe selection and workpiece orientation. For each feature to be inspected, all accessible directions are computed, then a minimal set of directions that enable inspection of all the features is determined. The set of directions are modelled by "direction cones".

The problem is solved in two stages: 1) accessibility analysis: finding the cones of accessibility direction for the feature, and 2) clustering: selecting a minimal set of directions sufficient to inspect all the features.

The accessibility analysis is performed at two levels: 1) at the local level, where obstacles in the immediate neighborhood of a point are considered, 2) at the global level, where the entire workpiece is taken into account.

The clustering problem, being NP-complete, does not generally lead to a unique solution;

therefore, few solutions are generated, and a heuristic function is used to select the "best one".

The main limitations of this approach are: 1) it is valid only for polyhedra objects, 2) the used probe abstraction represents it as a half infinite line, while real probes have a finite length, and thus in some cases feasible solutions are not considered; this abstraction can be replaced by the one considering the longest possible probe, 3) it is possible that an entire feature cannot be measured with a single straight probe, 4) a verification module is required to check the validity of the results for real probes, 5) it is an expensive method in terms of calculations and computations, 6) the method was applied in the simple case of polyhedra objects, and finally 7) the constraint of determining all accessible directions for each feature (for every point of the feature) does not avoid early, and potentially erroneous, commitments. On the contrary, it over constrains the set of solutions. The accessibility analysis is worthwhile only for this set of discrete points, because only a finite set of measured points is considered for inspection using CMMs.

2.2.4 The Work of Fiorentini et al.

Fiorentini et al. (1992) developed a CAIP. The CMM they used has a horizontal chuck, a rotating table, an automatic tool magazine, and a Renishaw touch probe. The necessary information is extracted by a CAD system. The workpieces treated come from the car industry, and do not include sculptured surfaces. The workpiece clamps are already known.

The logical steps of the CAIP are: Inspection Feature (IF) extraction, IF analysis, sensor selection and path planning.

Decision tasks include: IF selection, probe selection, points distribution, fixtures and clamp design (not treated), and probe path generation. The major topics bearing on these goals are: IF definition, extraction and sorting, accessibility analysis, correlation of surface points distribution vs. measuring precision, and optimal path generation and simulation.

The definition proposed for an IF is: informative content of a tolerance indication together with the technological information necessary to plan its inspection and evaluation. The number of

Ifs is equal to the number of tolerances in the drawing. Two kinds of information are considered: geometric and topologic (from a CAD representation of the part), and technological (from the CIM database related to the product). Faces are taken as the basic carrier of the technological information necessary for the inspection process and the information evolves with the workpiece history.

An IF model where information is distributed on three levels is proposed: IF level, IFF (Inspection Form Feature) level and XF (Extended Face) level. The following definitions have been proposed :

XF: is defined by the mathematical description of its surface and its boundary plus its technological parameters.

IFF: is a peculiar characteristic of a geometric entity and a combination of one or more geometric entities. It can be reconstructed from the surface estimation obtained from an inspection process.

IF: refers to one or more geometric features. It contains a tolerance relation concerning one or more IFFs.

This model is the basis for the CAIP system. It is a simple extension of the boundary representation used in most CAD systems. The object is decomposed into a set of IF-IFF-XF instances. Rules then build up the datum reference frame, configure probes, align the workpiece reference to that of the robot and selects the features to be inspected.

In (Fiorentini et al. 1992), a cycle time constraint was adopted. The cycle time is estimated by a simulation of the plan execution. Two levels of estimation are considered: 1) a lower bound of the cycle time, what a fixed average single measuring time is used to estimate the cycle time, and 2) a more refined estimation of the time for the movement between points, alignment procedure time and point measuring time.

When the cycle time constraint is not respected, a reduction of the list of inspected IF and the related measuring points is performed better: according to respecting a set of security Ifs (i.e. a minimum number of features). A utility score depending on the tolerance grade and the process capability of the manufacturing process is assigned to each feature not in the security list. This score will orient the phase of IF elimination if the cycle time constraint is violated.

This prototype is based on the extraction of features from a boundary representation CAD model. Technology attributes such as tolerances and datums must be provided in order to complement the geometrical and topological data extracted from the CAD model. Only simple features have been extracted and tested. In addition, solving the problem of cycle time constraints by reducing the number of features is not generally an efficient approach. The optimization of the inspection plan should be the first to be considered for the optimization of the cycle time. Choosing the security list from the start as the set of inspection features is better since they are by definition the smallest set of features that must be inspected.

2.2.5 The Work of Medland and Mullineux

An automatic feature-based inspection planning system (RASOR) was developed (Medland and Mullineux, 1992). It is able to select features to be measured on the basis of: 1) two possible strategies, preferred order or minimum path length, and 2) feature information such as the significance of the feature, the requirement of different probe types and attitudes to reach the feature, any special need to achieve the necessary accuracy and also the capability specified for the manufacturing process involved.

A strategic selection of the appropriate probe arrangement is adopted, the least preferred activities being: changing of the probe type during the process, introducing new combination of shank length and ball size, measurement interruption for re-calibration, and finally orienting the probe to a new attitude. All the decisions are made on the assumption that the minimum level of interruption to the measuring cycle is the preferred solution.

Feature accuracy selection implies determining the number of probing points required to measure a particular feature type to a required accuracy. The relationship between accuracy and the number of required measurement points has been found experimentally for a number of standard features and probes.

The CMM inspection process includes: 1) identifying features, 2) measurement strategy planning, 3) CMM control, and 4) measurements evaluation. Errors observed in measurement with contact probe are caused by: 1) repeatability, 2) probe length and ball size, 3) approach directions, and 4) probe re-setting. These errors are considered when making decisions during the plan generation.

This system is modular and implemented on a manufacturing network where communications are achieved through file exchange within a CIM environment.

The optimization of the inspection plan in terms of the best sequence has not been considered. In addition, the choice of a single orientation for each inspected point contributes to over-constraining the problem and thus interesting solutions may be ignored.

2.2.6 The Work of Merat et al.

Merat et al. (1991), developed a Rapid Design System (RDS). Their objective was to reduce time and cost from design to inspection. A combined generative/retrieval planning system is utilized for automating inspection planning.

Geometry and tolerances are represented as features. Inspection fragments are generated based on rules and methods used in industrial practices. Different inspection plan fragments (IPF) are associated with each tolerance. The final choice of the IPF will result in an overall time-efficient plan.

The following features are used: 1) form features (i.e. part geometry), 2) Geometric Dimensioning and Tolerancing features, 3) inspection features which contain the basic operations done during inspection, 4) manual operations measurement, evaluation and comparison module,

5) planning features (IPF), which are linked to GD & T features and contain the coordinates of the points to be sampled as well as probe orientation.

The steps used to generate a part inspection plan are: 1) generate all IPFs, 2) request a group measurement by part orientation, 3) select an IPF to measure each tolerance feature, 4) eliminate redundant measurement requests, 5) create "specification objects" from "request objects", 6) sequence the operations, 7) plan a CMM collision-free probe path, and 8) generate the CMM code.

This approach is being implemented for very simple parts consisting of blocks, holes and slots. Its main limitations are: lack of consideration of features interaction, and lack of path planning and plan optimization. In addition, features accessibility is not included, and inspection steps for various features are not prioritized or clustered to generate an optimal sequence.

2.2.7 The Work of Brown and Gyorog

This Inspection Process Planning Expert (IPPEX system) developed by Brown and Gyorog (1990) uses a product geometric modeler coupled with a dimensional and tolerance modeler.

The overall system is divided into nine main activities : 1) define/obtain the inspection task, 2) decompose the task (select of work elements and their inspection methods), 3) determine dimensional measurement equipment, 4) determine set-ups (orientation, fixtures and clamps), 5) determine the probe configuration, 6) determine an inspection plan, 7) generate/simulate the probe path, 8) produce a DMIS control program, and 9) produce support information.

As for the expert system environment, several decision-making domains have been created to support the system in the selection of: 1) dimensional measuring equipment, 2) workpiece orientation, 3) workpiece fixture, 4) set-up task, 5) probe configuration, 6) inspection planning, and 7) inspection technique.

Five levels in the system for hierarchical inspection process are considered: 1) the machine level (the machine is selected), 2) the set-ups level (the part is oriented and fixtured), 3) sensor

level (the probe identification and configuration are detected and defined), 4) work element planning level (a reference, a feature and an entity planning level are defined), and 5) action level (the sequence of actions necessary to measure each work element is planned).

In this system, the optimization of the inspection sequence or cycle time is not addressed, and no rigorous accessibility analysis is performed.

2.2.8 The Work of Tao and Davies

Tao and Davies (1992), developed INSPEC(P), a knowledge-based inspection planning system for 2-1/2D prismatic components in which surfaces are either parallel or perpendicular to the cutter axis.

The system is basically structured as follows: features including tolerance annotations are automatically extracted from wireframe models through IGES, and a six-part subsystem performs the following functions: 1) tolerance and profile matching, 2) set-up planning, 3) operation planning, 4) transformation of coordinates, 5) NC program code output, and 6) probe path display and inspection planning. Prolog was used to build the knowledge base, which is composed of a number of rules and facts.

In the probe operation process, two different tasks are considered: measurement procedure is implemented through reasoning and calculations.

The set-up is decided for tolerance measurement according to the directions from which the tolerance may be inspected. Rates defining a priority among the directions are calculated for all possible directions for all the tolerances. The directions with the greatest rates are chosen. An optimum measurement sequence is obtained through the elimination of redundancy in measurements. Different tolerances might require the measurement of the same feature more than once. A contact-point test is performed to verify if the measurement point belongs to the measurable zone.

The application of the above system is limited to 2-1/2D parts. The only optimization criteria considered is the elimination of redundancy in measurement operations. Redundancy may appear if each tolerance is verified independently of the others, however, measurement points may serve to verify more than one tolerance.

2.2.9 The Work of Dodini et al.

Two stages are followed to generate the inspection plan (Dodini et al, 1994): 1) accessibility analysis, and 2) clustering. During the accessibility analysis the set of constraints that the probe must satisfy to inspect the entire surface properly is analyzed. This work deals with the particular case of complex probe configurations such as the 5-way star probe.

2.2.10 The Work of Jones and Ulsoy

This work deals essentially with low level inspection planning (Jones and Ulsoy, 1994). An optimization strategy for maximizing CMM productivity is proposed. The problem is formulated in terms of non-linear programming. The variables are the acceleration, the approach distance and the approach rate. The constraints are measurement quality and available motor and other machine specific characteristics. The objective is to minimize measuring time. Experiments demonstrated a 27 percent reduction in time for optimized speed compared to the default speed. Another interesting application of this work is sensitivity analysis, i.e. how changes to a machine's design will affect productivity.

Other less important works related to inspection planning and verification are described in (Chen et al., 1994, Chivate and Jablokow, 1993, Hsieh et al., 1993, Koshnevis and Yeh, 1993, Menderos et al, 1994, Oetjens, 1989, Sostar, 1994, Walker and Wallis, 1992).

The surveyed systems are summarized in Table 2.1.

Table 2.1 Characteristics of Inspection Planning Systems

	1. ElMaraghy and Gu (1987)	2. Menq et al. (1991-1994)	3.Spyridi and Requicha (1991-1994)	4. Fiorentini et al. (1992)
Type	Expert System	Automatic	Automatic	Knowledge Based
CAD Model	Feature-based	3-D Complex Surfaces	Polyhedral Parts	Feature-based
Inspection Type	CMMs and non-CMMs	CMMs	CMMs	CMMs
Measurement Points Generation	-	Statistical, Accuracy and Tolerance Dependent	-	Statistical
Accessibility Analysis	Heuristics	Discrete Ray Tracing	Continuous	Heuristics
Operations Sequencing	Heuristics	Heuristics	Heuristic Search	Heuristics
Collision-free Path Planning	-	Heuristics	-	-
Plan Simulation	-	Yes	-	Yes
Alternate Plans	-	-	Yes	-
Optimization	Heuristics	Heuristics	Heuristics	-

Table 2.1 Characteristics of Inspection Planning Systems (Cont.)

	5. Medland and Mullineux (1992)	6. Merat et al. (1991-1994)	7. Brown and Gyorog (1990)	8. Tao and Davies (1992)
Type	Automatic	Generative/ Retrieval	Expert System	Expert System
CAD Model	Feature-based	Feature-based	Feature-based	Feature-based
Inspection Type	CMMs	CMMs and non-CMMs	CMMs	CMMs
Measurement Points Generation	Accuracy and Tolerance -	Predefined	-	Heuristics
Accessibility Analysis	-	-	Discrete	-
Operations Sequencing	Strategies	Time Efficient	Heuristics	Heuristics
Collision-free Path Planning	-	Heuristics	Heuristics	Heuristics
Plan Simulation	-	-	Yes	-
Alternate Plans	-	-	-	-
Optimization	-	-	-	-

2.3 GEOMETRIC MODELING

The tactile inspection process is followed by the analysis and verification of tolerance specifications. Data points or coordinates are used to represent or approximate the actual inspected feature using mathematical models. These models are then compared to the CAD models in order to verify if tolerance specifications are met.

Complex surfaces are frequently encountered in inspected parts especially in mould design. They are more difficult to inspect and usually need to be decomposed into patches which are interpolated separately then joined and intersected. The tools to be used for representing and manipulating these surfaces are very important, they have to represent closely the actual surface and to be efficient and fast.

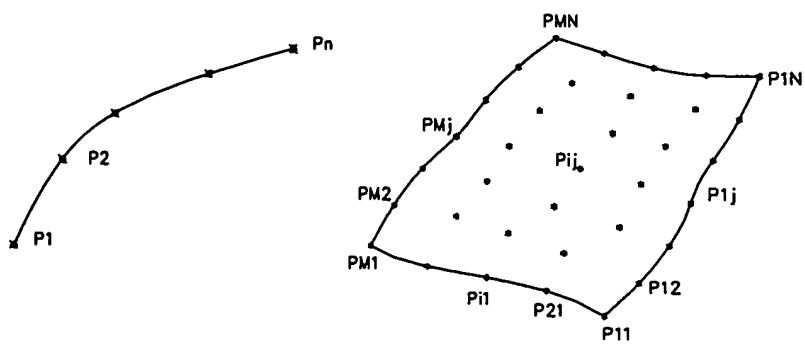


Figure 2.3 Curve and Surface Interpolation

2.3.1 Curve and Surface Representation

There are three main applications of curve and surface modelling: 1) design, where only a general or qualitative idea about the desired shape is available, 2) interpolation, where the geometric model has to pass exactly through a set of data points, and finally 3) data fitting, where a large number of data points is used, and the geometric model has to approximate these points

with a given accuracy. The last two applications are generally used to model data generated using CMMs (de Boor, 1978, Kuriyama, 1994, Morteson, 1985, Piegl, 1991, Sarkar and Menq, 1991).

The most popular techniques for curve and surface representation are: piecewise polynomial interpolation, spline interpolation, Bezier, B-splines and Non-Uniform Rational B-spline (NURBS) curves and surfaces, and Coons surfaces.

In the case of Bezier, B-spline and NURBS techniques, the model is built using a set of control points that form a control polygon or polyhedra representing an approximation of the curve or surface. Note that the B-spline is a generalization of Bezier, and NURBS is a generalization of B-splines and may be represented using tensorial products. One of the major problems of the Bezier technique is the degree of the model which is directly related to the number of control points. This problem is eliminated in the case of B-spline and NURBS techniques. The NURBS technique has the additional property (or feature) of representing classical analytical models such as conics, and surfaces of revolution. These methods are also invariant under scaling, rotation, translation and shear as well as parallel and perspective projection.

The two previous classes of techniques use a set of discrete data or control points. Coons surface (Zeid, 1990), however, use a set of curves (i.e. an infinite number of data points) to generate the surface. This technique is very useful for the generation of blending surfaces or patches between existing surfaces and can ensure up to C^2 continuity (i.e. second order derivatives continuity) (Zeid, 1990).

2.3.2 Curve and Surface Manipulation

A large number of methods have been proposed in the literature for determining the intersection set between two geometric objects. Some methods are general, in the sense that they can be applied to any type of surface. The approach of Marciniak (1990) is based on solving a set of simultaneous nonlinear equations by Newton's method and on minimizing the distance between

the two objects. The same approach can be found in Faux and Pratt (1980), who apply these algorithms to generate cutter paths for numerically controlled machines. A general method for parametric piecewise surfaces intersection was described by Asteasu and Orbegozo (1991), requiring only C^1 surface continuity of the surface. An initial set of points is calculated by a method of enclosing boxes. From these isolated points, spans of the real intersection are calculated by an incremental method based on differentiating the parametric equations of the two surfaces. Another way of finding initial points is proposed by Mullenheim (1991) for a surface-surface intersection algorithm.

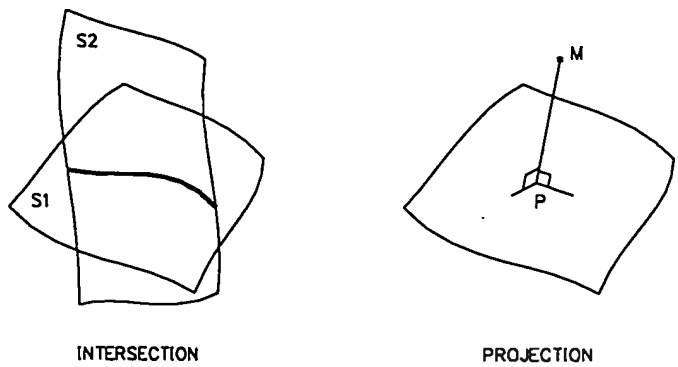


Figure 2.4 Examples of Tools for Surface Manipulation

Since a fairly large number of CAD systems represent free form surfaces by a set of parametric patches, intersection algorithms for parametric patches play an important role. This problem has been treated by various authors and a survey of this topic is reported in Pratt and Geisow (1986). The algorithm performs an adaptive and recursive decomposition of surfaces which is a function of their morphology, then a selection process filters out all irrelevant surface pairs. This method of surface intersection, which is independent of surface derivatives, is based on a triangular decomposition of relevant surface patches. Although this approach can be applied to any kind of free-form surface, it presents two technical difficulties: (1) the accuracy of the triangular

decomposition depends on the smoothness of the surface and on the size of the triangles; (2) the intersections of the two sets of triangles must be sorted and re-organized to generate curves.

A certain number of methods were devised for particular types of surfaces. For example, the method for surface/surface intersection of Aziz and Bata (1990) is valid only for Bezier surfaces. It is based on a recursive subdivision technique of their convex hulls. Hedrick and Bedi (1990) have proposed a method for two bi-quadratic or bi-cubic surfaces. The method is robust, but it is limited to parametric surfaces defined by quadratic or cubic polynomials. Note that a special method for the intersection of a plane and a natural quadric was proposed by Johnstone and Shene (1992). An original approach adopted by Klass and Kuhn (1992) consists of finding the intersection between two spline surfaces by using rolling balls of decreasing radius simultaneously tangent to the two surfaces. The intersection is obtained when the radius of the rolling ball becomes zero.

2.4 DISCUSSION

Review of previous work in inspection planning indicates that many problems have been solved only for special cases; but many more remain unsolved or their proposed solutions are expensive and/or very difficult to perform. In addition, heuristics were used extensively to find acceptable solutions; however, the need for optimization is a priority where tight time constraints are present, in order to improve productivity and competitiveness.

The formalization of the different tasks and the knowledge related to each step in the inspection process is vital. The main deficiencies are found in high level inspection planning, especially in the accessibility analysis of measurement points. In fact, no robust general approach for the inspection probe accessibility analysis has been proposed yet; however, methods for the computation of the accessibility domains were developed in the special case of polyhedral objects.

The issue of optimizing the inspection plans has not been considered properly: only feasible plans were sought and heuristics are used extensively. Even though attempts were made to optimize the inspection plans at a low level, the result is insignificant from a global viewpoint. A formalized inspection planning approach is needed in order to achieve an automated and integrated computer-aided inspection planning (CAIP) system.

Another field where contributions are needed is in the modelling, manipulation and interpolation of the measured complex surfaces. This is very important if we consider the fact that the verification of specified geometric tolerances is based on the comparison of the interpolated model representing the actual surface and the CAD model representing the theoretical surface. In addition, the verification of tolerances generally involves complicated and long iterative calculations. The cost of these calculations depends closely on the interpolation model as well as the methods used. These are generally analytical algorithms for solving non-linear equations and can be unstable in the case of NURBS, Bezier, and Coons entities (Piegl, 1991). This is especially important for on-line verification systems. In addition, it is important to include measurement errors in the surface interpolation model in order to better describe the actual surface with a smooth model and overcome the problem caused by random errors which introduces perturbations in the interpolated surface.

CHAPTER 3

ACCESSIBILITY ANALYSIS FOR INSPECTION PLANNING

Accessibility analysis is a very important aspect of Automated CMM Inspection Planning. It minimizes unnecessary changes in probe orientation during inspection, and hence maximizes the number of features inspected using the same probe orientation. This characteristic is more important to an effective inspection plan, than other factors such as the distance travelled by the probes, since probe changes and re-orientation are the most expensive operations in terms of calibration, set-up, etc. In this chapter two approaches for accessibility analysis are presented: 1) A continuous accessibility analysis based on a new algorithm yielding the exact accessibility domain of any point in the space is described in section 3.1, and 2) A discrete accessibility analysis where only predefined orientations of the probe are investigated is presented in section 3.2. Afterwards, a discussion of the two methods is provided in section 3.3.

3.1 CONTINUOUS ACCESSIBILITY ANALYSIS

This section presents a general method for features accessibility analysis used in the dimensional inspection of mechanical parts by CMMs. It is based on the Intersection of Concentric Spherical Shells (ICSS) centered at the Measurement Point. This method can be applied to any type of solid or surface and is not limited by continuity or derivability constraints. Accessibility domains can be generated easily as a limit case of the ICSS. This method has been generalized to include inspection with bent probes. Modifying it to take into account the actual shape of the probe is also considered. It may be used to analyze the accessibility or probe approach directions for any point

within or outside any object. It is also possible to determine the common accessibility domain for a set of points directly or by comparing the accessibility domain of each point. This method may have other applications in fields such as machining.

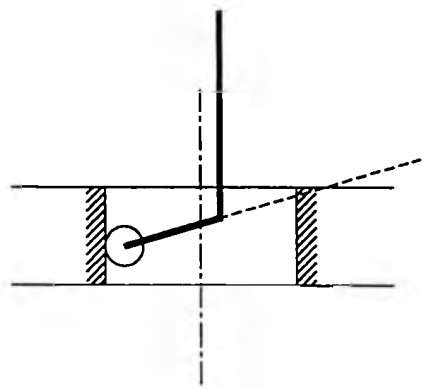


Figure 3.1 Elimination of Possible Orientations if an Infinite Half Line Abstraction of Inspection Probe is used

3.1.1 Related Works

The accessibility issue in inspection planning had not been addressed effectively in previous works (Brown and Gyorog, 1990, Elber, 1994, Fiorentini et al., 1992, Menq et al., 1992, Merat et al., 1991). Spyridi and Requicha (1990, 1991) were the first to adopt a systematic accessibility analysis for inspected features. They used algorithms based primarily on the computation of Gaussian images and Minkowski (or sweeping) operations (Ghosh 1988, Spyridi and Requicha 1990). The accessibility is analyzed at two levels: 1) locally where obstacles in the immediate neighborhood of a point are considered, and 2) globally where the entire workpiece is examined. The main limitations of their approach are: 1) the probe is abstracted as a half infinite line; however, real probes have a finite length, thus in some cases useful solutions are not considered and points not accessible by a straight infinite probe cannot be measured (see Figure 3.1); 2) it may

not be possible to measure an entire feature with a single straight probe; yet measuring a discrete set of points within this feature would be feasible; 3) a verification module to check the validity of the results for real probes is needed; 4) the method is computationally intensive; and finally 5) it was applied to only the simple case of polyhedra objects.

An accessibility analysis approach based on a ray tracing algorithm was recently proposed by Lim and Menq (1994). The length of the inspection probe is assumed to be infinite and all possible angles of the probe head, which in reality is of finite length are investigated by ray tracing. The probe and its type are determined before accessibility analysis. The 3-D accessibility cone is transformed into a 2-D map where only the orientations of the probe expressed by two angles in a spherical coordinate system are considered. Corrections are made to take into account the actual shape of the probe. A heuristic search is finally performed in order to determine the optimal probe direction for a set of inspected points. The main limitations of this method are: 1) the infinite half line probe is abstracted as in (Spyridi and Requicha, 1990), 2) the probe geometry and its resolution (i.e. possible orientations) are specified before applying accessibility analysis, which restricts the solutions, and ignores useful solutions using other probes, since the digitized accessibility maps for probes with different resolution cannot be compared; and finally 3) the search for the optimal inspection angle for a set of measured points is based on heuristics, which suffers from the drawbacks of any expert systems approach. Compromises were made between the safest angle criterion and the fastest inspection path; however, more critical criteria such as minimal probe orientation changes and minimal number of probes used were not considered.

3.1.2 The ICSS Method

The adopted probe abstraction is a line segment of length R which represents the total length of the probe. The problem is to find the set of probe directions which permit access to the

measurement point(s) without interference with neighboring obstacles such as the workpiece, fixtures and clamps.

An optical analogy to this problem is the following: find the projected image (i.e. the lit region) on a spherical screen of radius R centered at the measured point where a point light source M exists (see Figure 3.2). The probe direction will be any straight line segment joining the measured point and the projected image.

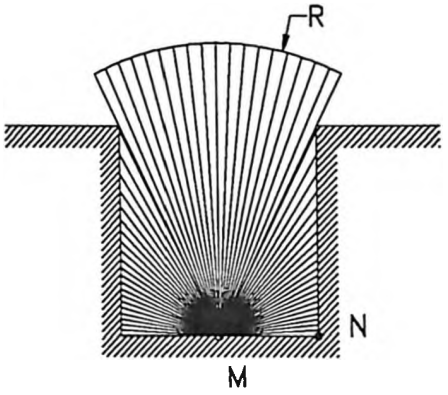


Figure 3.2 Optical Analogy of the Inspection Accessibility Problem

It is clear that the results of Spyridi and Requicha (1990) using an infinite half line abstraction of the probe can be found by considering a large enough length R so as to surround all obstacles in the measurement environment.

Principle of the ICSS Method

This method is based on recurrent geometric transformations. Basically spherical scaling and solids intersection are applied initially to the intersection of the complement space of the part and its surrounding objects in the vicinity of the measurement point, represented as a spherical shell of external radius, R, equal to the probe length and an internal radius $r0$, equal to the

maximum abstracted probe length for which the accessibility domain is maximum. The thickness of the shell is decreased at each iteration, and at the limit, a surface shell representing the accessibility domain is obtained. This shell is equivalent to the projected image in the optical analogy. Assume that:

M is the point to be accessed by a probe of length R.

D is the part's (and eventually fixtures and clamps) complement space.

The accessibility domain A_R is the set of directions MB such that:

$$\|MB\| = R \ , \ MB \subset D \tag{3.1}$$

An interesting property of the accessibility domain $A_R(M)$ is:

$$A_R(M) \subseteq A_r(M) \ , \ \forall \ r \leq R \tag{3.2}$$

Therefore, $A_R(M)$ can be stated as:

$$A_R(M) = \bigcap_{r=r_0}^{R} A_r(M), \ 0<r_0<R \tag{3.3}$$

Using (3.3) to determine $A_R(M)$ yields theoretically an infinite number of intersections. For this reason, spherical shells of thickness e ($e > 0$) will be used.

$r0$ Definition

$I_r(M)$ is the intersection of a half-sphere (or sphere in the case of a point on a surface singularity i.e. vertex, edge, or a point in the space not lying on a specified surface) of radius r centered at M, with domain D.

$S(I_r(M), r')$ is the spherical scaling of $I_r(M)$ at point M with the ratio r'/r.

$r0$ is the maximum abstracted probe length for which the accessibility domain is maximum, i.e.

$$r0 = \max \{ r / I_r(M) \subset S(I_r(M), r'), \forall r' > r \} \quad (3.4)$$

$r0$ must satisfy:

$$S(I_{R-r0}(M), r0) \subset I_{r0}(M), \forall R \geq r0 \quad (3.5)$$

Property (3.5) suggests a dichotomy procedure to find $r0$. This is due to the fact that if (3.5) is not satisfied for a given $r0$, then it will not be satisfied for any $r0' > r0$. The steps used to determine or approximate $r0$ are:

step 1 Initialization $r0^1 = R / 2$

step 2 Verify relation (3.5)

step 3 if (3.5) is satisfied then $r0^{i+1} = (r0^i + R)/2$ else $r0^{i+1} = r0^i / 2$

step 4 if $\text{abs}(r0^{i+1} - r0^i) > \varepsilon$ goto step 2 else end.

The following procedure is applied to reduce the thickness of the spherical shell $S_0(r0<r<R)$ Intersect the spherical shell $S_0(r0<r<r1)$ with D, then scale the result with an amplitude $R/r1$ and intersect it with $S_0(r0<r<R)$, keeping the center of the shells at the same point each time. A solid $S_1(r_1<r<R)$ is obtained. However, $r1$ has to verify

$$R - r1 = \frac{R}{r1}(r1 - r0) \quad (3.6)$$

or

$$r1 = \sqrt{R\,r0} \quad (3.7)$$

The thickness of the shell is:

$$e_1 = R - r1 = R - \sqrt{R\,r0} \quad (3.8)$$

The importance of ensuring that $r0 > 0$ is clear at this level. In effect, an $r0 = 0$ will not lead to a decrease in the shell thickness, and the method will not converge. In addition, the larger

r0 is, the faster the convergence (for a fixed accuracy) to the accessibility domain. Note that *r0* is at least equal to the radius of the probe ball; otherwise, the inspection is impossible.

Each point in the spherical shell represents a possible orientation. The intersection procedure successively eliminates impossible orientations.

The sphere of radius *r0* represents the maximum number of directions of length *r0* that can access the measurement point. If we consider one specific ray of length *r0* fired from *M* (see Figure 3.3), the scaling operation extends the ray with a length $e = r1 - r0$. Two cases are possible, 1) the segment (M_1M_2) is included in (D), 2) the segment (M_1M_2) is not included in (D). Only the directions of the second type are eliminated by the successive intersection operations.

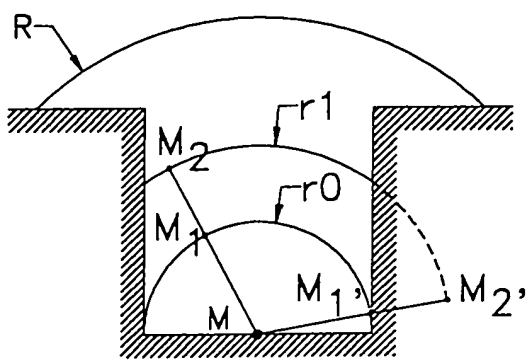


Figure 3.3 Steps of the ICSS Method

If the same procedure is applied to solid S_1, choosing $r2 = \sqrt{(Rr1)}$, the thickness of the equivalent solid (shell) will be $e2 = R - r2 = R - \sqrt{(Rr1)}$. Equations (3.7) and (3.8) can be generalized to:

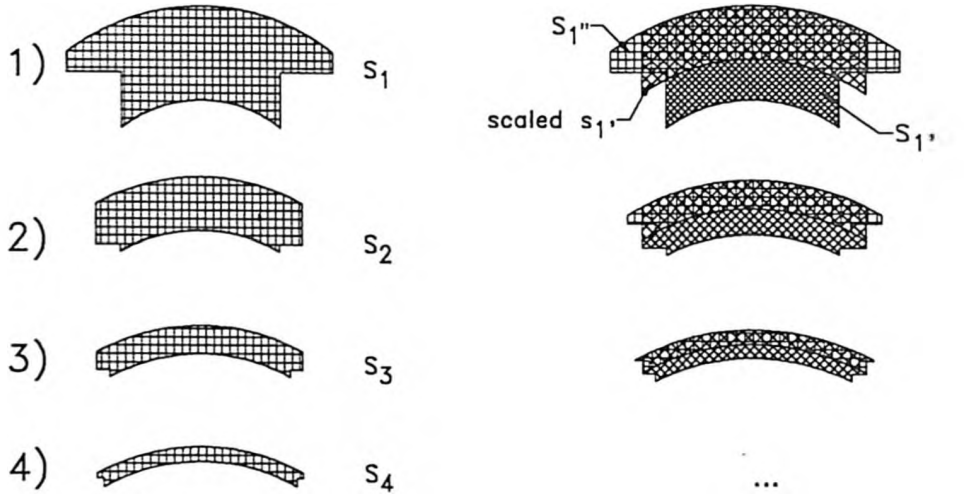
$$r_n = \sqrt{Rr_{n-1}} \tag{3.9}$$

and

$$e_n = R - r_n = R - \sqrt{R\, r_{n-1}} = R - \sqrt{R(R - e_{n-1})} \tag{3.10}$$

or

$$\begin{aligned} R - e_n &= (R(R - e_{n-1}))^{\frac{1}{2}} \\ &\Updownarrow \\ 1 - \frac{e_n}{R} &= (1 - \frac{e_{n-1}}{R})^{\frac{1}{2}} = (1 - \frac{e_1}{R})^{\frac{1}{2^{n-1}}} = (\frac{r0}{R})^{\frac{1}{2^n}} \end{aligned} \tag{3.11}$$



$$S_i = S_{i-1}'' \cap \text{scaled}(S_{i-1}')$$

Figure 3.4 The First Four Iterations of the ICSS Method

From the equation (3.11), we notice that $e_n \to 0$ when $n \to \infty$, hence, when $n \to \infty$, the exact accessibility domain is generated. However, for a desired accuracy, or for a thickness e of the final shell, the number of iterations will be:

$$n = integer\left[\frac{1}{Log2}\ \frac{Log\frac{r0}{R}}{Log(1-\frac{e}{R})}\right] \tag{3.12}$$

Figure 3.4 shows the results of the first four iterations applied to the example shown in Figure 3.2. The accessibility domain may be represented using spherical coordinates parameters θ and φ. A graph $\theta = f(\varphi)$ may be generated. The accessibility domain is then given by $\theta < f(\varphi)$ or $\theta > f(\varphi)$ (see Figure 3.5).

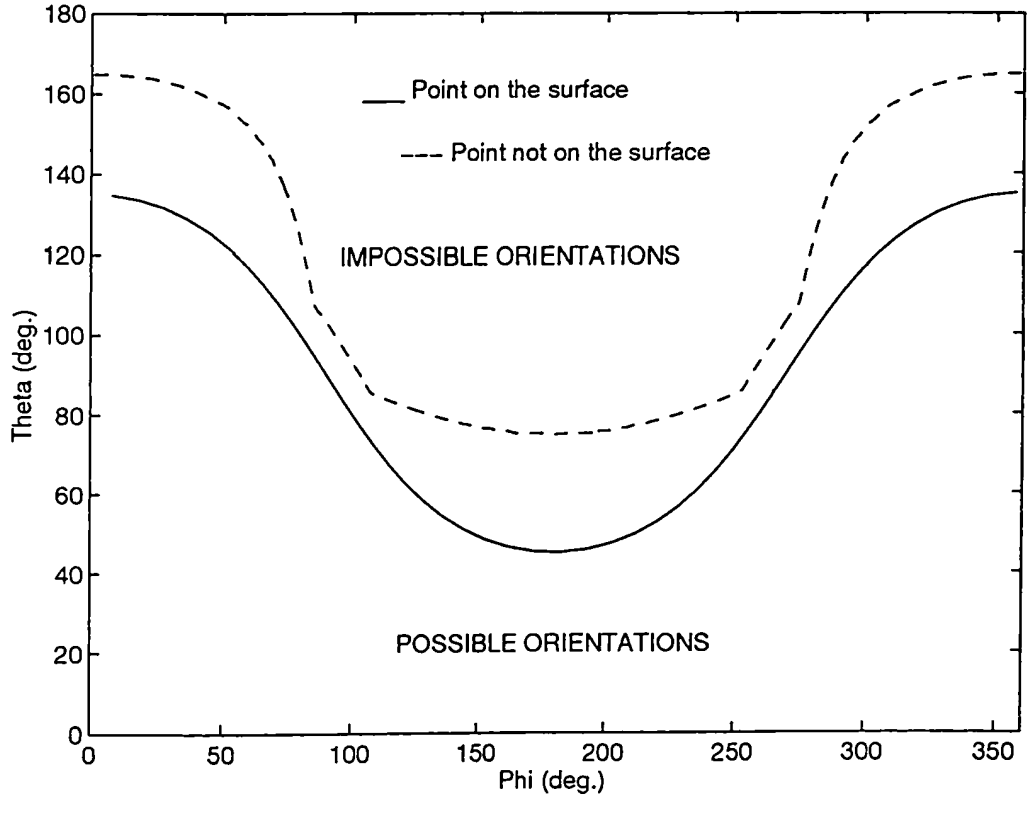


Figure 3.5 Implicit Representation of the Accessibility Domain

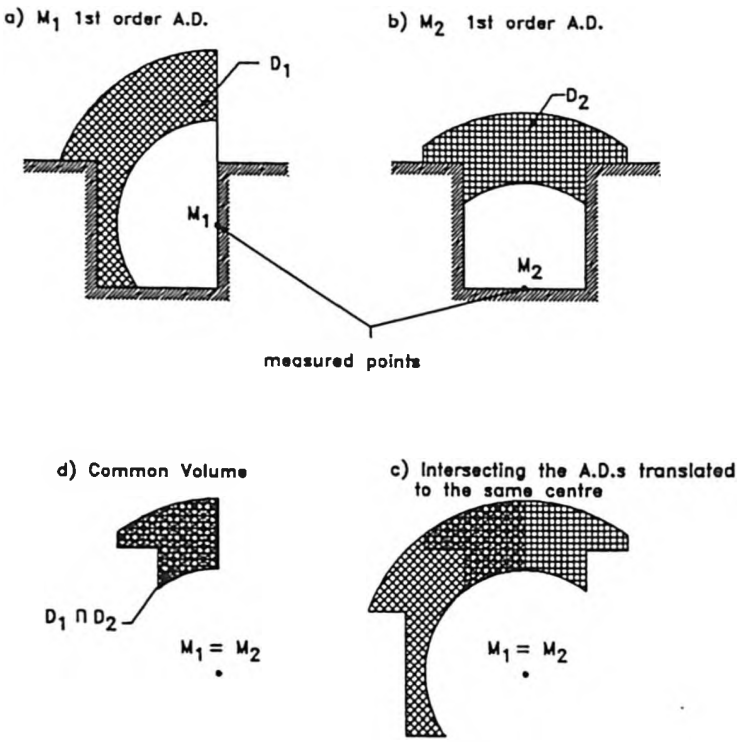


Figure 3.6 Steps of the Comparative Accessibility Procedure in the Case of Two Measurement Points

Remarks:

1. When the surface in the vicinity of the measured point is derivable, the accessibility domain is at most a half-spherical surface, where the tangent plane in this case is the boundary of the accessibility domain.

2. This method can be applied to any point in the 3-D space, not necessarily on a surface or solid.

3. Even though the inspection of singularity points (vertices and edges) where surface tangents are discontinuous is in general not considered, the ICSS method may be adapted to these particular cases by choosing each time a suitable value of *r0*.

Comparative Accessibility

In this case, the accessibility domains of two or more measurement points are compared (or intersected) in order to identify any common accessibility directions. The accessibility of different points can be obtained by applying the ICSS procedure to the first order (i.e. the 1st iteration of the ICSS) common accessibility domain as illustrated in Figure 3.6. The common *r0* must be in this case the minimum of the *r0* of the points to be compared.

Another approach consists of generating the accessibility domain for each measurement point using the parameters θ and φ and then comparing the results.

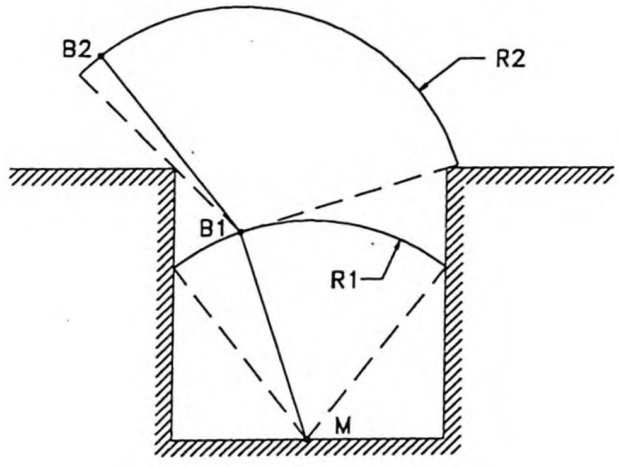


Figure 3.7 Principle of the Bent Probe Accessibility Analysis Approach

3.1.3 Application to Bent Probes

The ICSS method can be applied easily to bent probes (Figure 3.7). First, apply this method for a straight probe of length $R = R_1$, second, for a fixed orientation MB_1 in $A_{R1}(M)$, compute $A_{R2}(B_1)$, the accessibility domain in $B1$; and third, choose an orientation $A_{R2}(B_1)$ following given criteria such as minimum probe changes and minimum number of probes used. This procedure may

be repeated for as many segments as needed. It is performed exactly as in the case of a straight probe of length $R = R_1$.

The use of bent probes is of particular interest for cases where the inspection of a whole feature or a set of points by a single straight probe is impossible.

3.1.4 Modification of Accessibility Domain for Actual Probes

Actual probes are not dimensionless. If we consider the largest-half sphere of radius $r0$ centered at the measurement point, and the minimum cone of half angle $\alpha 0$, totally enclosing the probe less its segment of radius $r0$ (see Figure 3.8a), here appears another application of $r0$ which permits obtaining more accurate and more realistic enclosing cones of the measurement probe, as compared to the safety cone proposed in (Lim and Menq, 1994). A simple intersection test of the probe cone with the accessibility cone suffices when only a verification of the orientation is needed. This type of correction is represented by an offset of magnitude $\alpha 0$ of the curve $\theta = f(\varphi)$ in the direction of possible orientations.

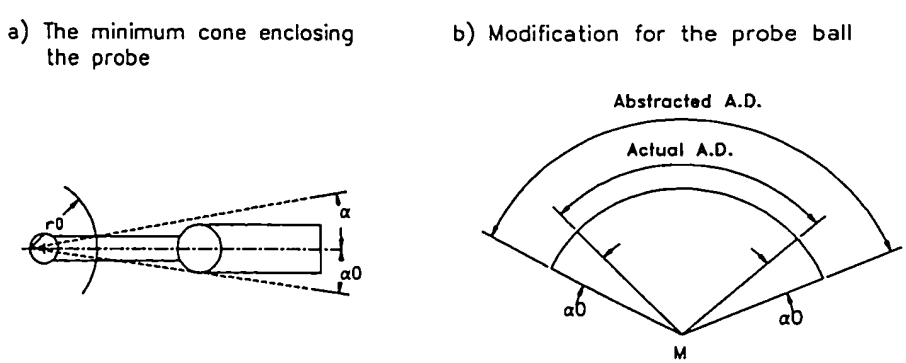


Figure 3.8 Principles of Modifications of Accessibility Domains

The same corrections can be applied to bent probes, considering each time a segment of

the bent probe and having initially determined the minimum enclosing cone for each segment. This correction can be refined further if we observe that the accessibility domain has to be determined not at the measurement point but at the center of the probe ball in contact with the measurement point (see Figure 3.8b). This correction cannot be performed by the method described by Spyridi and Requicha (1990) since the center of the ball probe does not belong to the actual solid or surface used to analyze accessibility. However, with the ICSS method, it is possible to perform this type of refinement easily. This modification generally enlarges the abstracted accessibility domain (see Figure 3.8b), hence directions parallel to the tangent plane are not eliminated. The effect of this correction on the curve $\theta = f(\varphi)$ is generally a non-uniform offset (i.e. an offset with a non-constant magnitude) in the direction of possible orientations. Figure 3.12 shows the effect of these corrections.

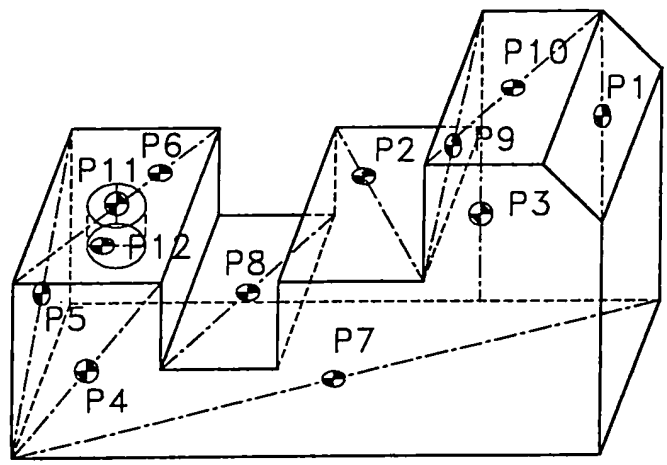


Figure 3.9 Example Part and the Inspected Points *P1* ... *P12*

3.1.5 An Example

The accessibility domains of 12 measurement points on the workpiece shown in Figure 3.9 have been generated. External obstacles such as tools and fixtures have not been considered. A

straight probe of length 40 mm is utilized.

The accessibility domains of points *P1*, *P3*, *P4*, *P5*, *P6*, *P7* and *P10* are deduced directly without using the ICSS method (in these particular cases $r0 = R$). Their accessibility domains are half-spheres oriented along the surface normal direction.

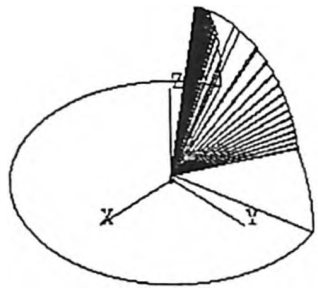


Figure 3.10a *P2* Accessibility Cone

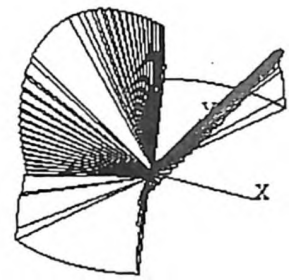


Figure 3.10b *P8* Accessibility Cone

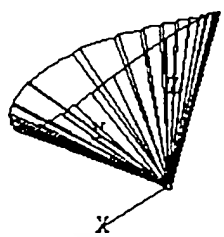


Figure 3.10c *P11* Accessibility Cone

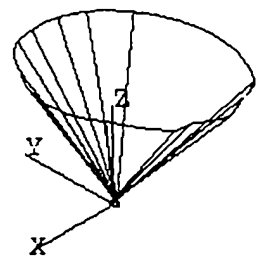


Figure 3.10d *P12* Accessibility Cone

The accessibility domains of points *P2*, *P8*, *P9*, *P11* and *P12* are generated using the ICSS method in six iterations. Figure 3.10 shows the boundaries of the accessibility cones of *P2*, *P8*, *P11* and *P12* and Figure 3.11 shows their implicit representation in the plane. The shaded area in Figure 3.11 shows the intersection of the accessibility domains of points *P2*, *P8*, *P9*, *P11* and *P12*.

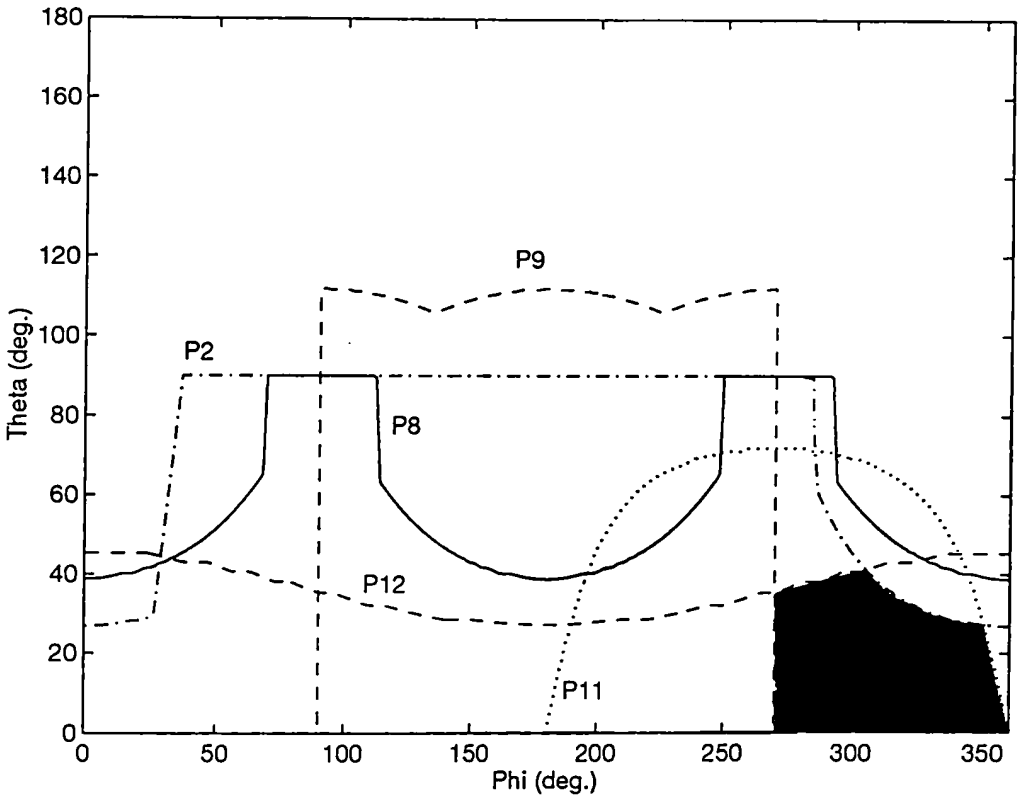


Figure 3.11 Implicit Representation of the Accessibility Domains of Points *P2*, *P8*, *P9*, *P11* and *P12*

Figure 3.12 shows an example of application of the corrections that can be performed over the accessibility domain of point *P1*. The continuous curve represents the boundary of the accessibility domain of an abstracted probe obtained for a point lying on a surface. The dash-dot curve represents the accessibility domain of a point offset from the measurement point by the radius of the probe tip. The dashed curve is an example where a safety cone angle (10 deg.) is considered for the probe. Finally, the combination of previous corrections produces the dotted curve. The accessibility domains of these points may be compared in order to determine their intersection (i.e. the set of possible common orientations). These results have been generated using the ACIS Geometric Modeler (Spatial Technology, 1993).

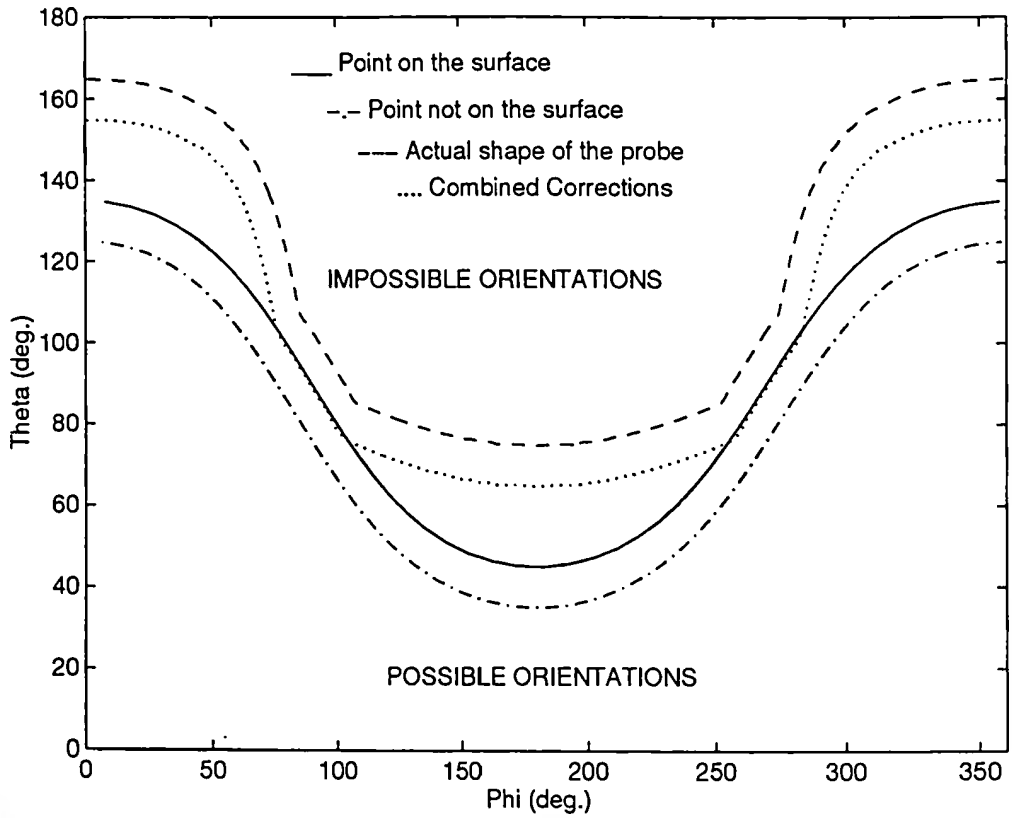


Figure 3.12 Corrections of the Implicit Accessibility Domain of Point *P1*

3.2 DISCRETE ACCESSIBILITY ANALYSIS

The possible orientations of measurement probes are generally finite, and depend on the resolution of the probe, which is usually expressed in terms of two rotations angles (Figure 3.13). The discrete approach is particularly useful when the probe(s) used is (are) known in advance. In addition, the accessibility domain obtained takes into account the actual shape of the probe, hence eliminating post-corrections. The comparative accessibility analysis of multiple points is easier since we deal with finite sets. The algorithm developed for this approach is based on ray tracing techniques.

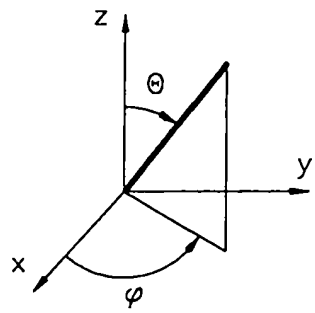


Figure 3.13 Spherical Coordinates System

3.2.1 Ray Tracing Algorithm

This section presents an overview of the ray tracing algorithm. More details may be found in (Zeid, 1990). Ray tracing has been utilized in a variety of applications: visual realism to generate line drawings with hidden solids removed, shaded picture, and solid analysis (mass properties). The basis of ray tracing is very simple. It consists of the intersections of a ray (i.e. straight line) which is best defined in a parametric form as a point and a direction vector with an object (solid or surface). It is based on line/surface intersections algorithms.

3.2.2 Description of the Method

This method is based on ray tracing algorithms implemented using the ACIS Solid Modeler (Spatial Technology, 1993). The CMM probe is composed of a stylus and a head. The stylus may be oriented according to two independent angles with predefined ranges and resolutions. The orientation of the probe is described by the two spherical angles θ and φ , their range θ_{min} and θ_{max}, φ_{min} and φ_{max} and their increments $\Delta\theta$ and $\Delta\varphi$. Orientation changes may be performed manually or automatically as in the case of the PH9A Renishaw probe. Figure 3.14 shows the adopted probe abstraction, it includes the probe head as well the probe stylus. The rays utilized

have, in addition to their origin and direction, a radius (i.e. the ray is not a line segment but a cylinder). It is also possible to determine the distance between the origin of the ray and the first entity hit by the ray. These characteristics permit taking into account the actual size and geometry of the probe, hence generating the exact accessibility domain.

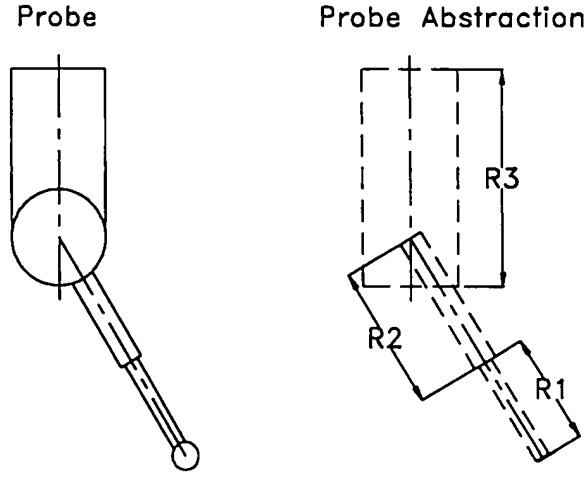


Figure 3.14 Probe Abstraction

The set of all possible orientations of the probe is scanned, and only orientations without interference are retained. Note that in this case, a collision occurs when the ray interferes with the environment within any probe segment. This overcomes the problem by considering infinite ray length which may cause perfectly good orientations to be ignored as a result of interference which occurs beyond the probe. The set of all orientations that the probe can assume is represented with a binary matrix $[b_{kl}]$. The term b_{lk} represents the orientation $\theta = (k\text{-}1)\,\Delta\theta$ and $\varphi = (l\text{-}1)\,\Delta\varphi$. A possible orientation corresponds to $b_{lk} = 1$ and an impossible orientation corresponds to $b_{lk} = 0$ (see Figure 3.15). Choosing a finite length of the probe has many advantages; it allows analyzing only the local neighborhood of the measurement points for each segment of the probe (see Figure 3.16), thereby, reducing calculations for complicated parts and their surroundings.

φ

θ

```
00000000000000000000000000000000000
00000000000000000000111111111111111
00000000000000000000111111111111110
00000000000000000000111111111111110
00000000000000000000011111000111100
00000000000000000000001110000011100
00000000000000000000000000000000000
00000000000000000000000000000000000
00000000000000000000000000000000000
00000000000000000000000000000000000
00000000000000000000000000000000000
00000000000000000000000000000000000
00000000000000000000000011111000000
00000000000000000000011111111111000
00000000000000000000111111111111100
00000000000000000000111111111111110
00000000000000000001111111111111111
00000000000000000001111111111111111
00000000000000000000000000000000000
```

Figure 3.15 Example of Discrete Accessibility Domain (where 1 represents a feasible orientation and 0 represents an unfeasible orientation for all values of φ and θ)

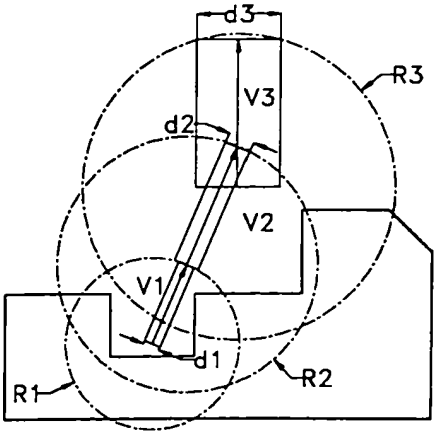


Figure 3.16 Subdivision of the Neighbourhood for Each Segment of the Probe

Accessibility domains of different measurement points may be compared and intersected in order to determine their common accessible orientations. It is also possible to determine directly

the discrete common accessible orientations for a set of points. This can be done in three steps:

Step 1 Subtract the part and its environment from a sphere of radius R (i.e. the maximum length of the probe and its head) centered at the measurement points (Figure 3.17a).

Step 2 Intersect the solids obtained in step 1 translated to the same center point (Figure 3.17b, c).

Step 3 The solid obtained in step 2 (Figure 3.17c) is subtracted from a sphere of radius R. Apply the discrete accessibility analysis to the previous solid (Figure 3.17d).

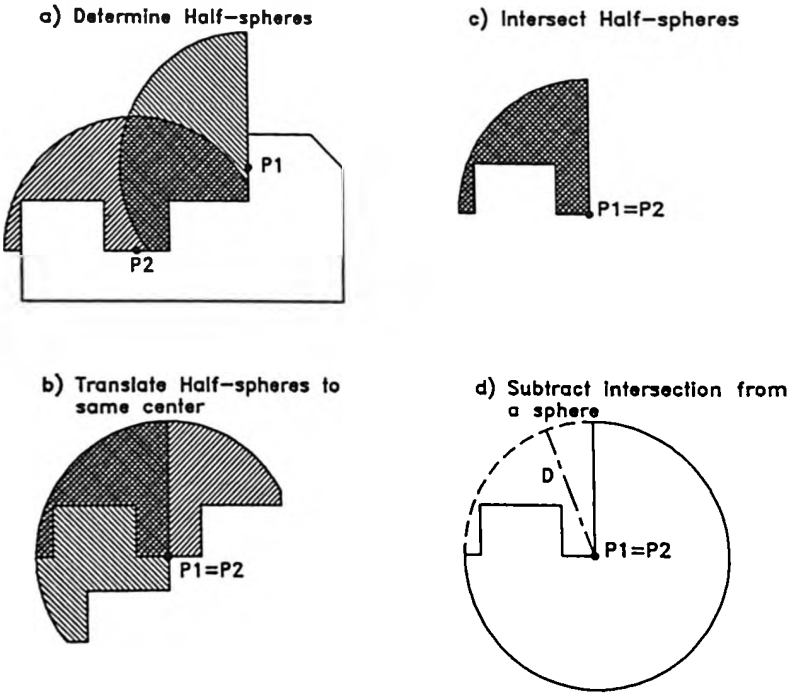


Figure 3.17 Steps of the Discrete Comparative Accessibility Analysis

The accessibility domain of the obtained sphere is identical to the intersection of the accessibility domains of the initial measurement points. This allows to check if a set of points is accessible with the same orientation(s) directly and without computing the accessibility domain of each point.

Principal Clusters Definition

This new notion of principal clusters is very useful for the following steps in inspection planning. It yields a Tool/Operation matrix representation which allows the problem of measurement sequence and probe orientations to be solved by the traditional methods of scheduling, sequencing and resources allocation.

Consider the following notations:

P_k measurement point k, $1 \le k \le N$

O_{ij} probe orientation i,j, $0 \le i \le I$ and $0 \le j \le J$, ($\theta = \Delta\theta\ i$, $\varphi = \Delta\varphi\ j$)

A_k accessibility domain of P_k; $A_k = \{O_{ij} / P_k$ accessible by probe orientation $O_{ij}\}$

C_{ij} $\{ P_k / O_{ij} \in A_k \}$

P the set of principal clusters

POINTS

CLUSTERS	1	2	3	4	5	6	7	8	9
1	1	1		1		1		1	
2		1	1		1				1
3						1	1	1	1
4			1		1	1			1
5				1		1	1	1	
6	1	1			1	1		1	

OPERATIONS

TOOLS	1	2	3	4	5	6	7	8	9
1	1	1		1		1		1	
2		1	1		1				1
3						1	1	1	1
4			1		1	1			1
5				1		1	1	1	
6	1	1			1	1		1	

Figure 3.18 Equivalence between the Points/Principal Clusters and the Tools/Operations Matrix Representation

The set of principal clusters P is determined by using the following algorithm:

Initialization

$P = \{\}$

Iteration

for each O_{ij}, $0 \le i \le I$ and $0 \le j \le J$ do

determine C_{ij}

if $\forall\ C_{kl} \in P$; $C_{kl} \not\subset C_{ij}$ and $C_{ij} \not\subset C_{kl}$ then

$P = P \cup \{C_{ij}\}$

else

for each $C_{kl} \in P$ do

if $C_{kl} \subset C_{ij}$ then

$P = (P - \{C_{kl}\}) \cup \{C_{ij}\}$

end

end

end

Given the set of principal clusters, it is possible to represent the problem by using a Tool/Operation matrix representation. Each principal cluster represents a tool (a probe with a set of orientations) and each point represent a measurement operation (see Figure 3.18).

3.2.3 Example

The same part and measurement points of the example presented in section 1 are considered. The characteristics of the probe (Renishaw PH9 for example) are:

angle θ : 75, 180 deg. step: 7.5 deg.

angle φ : -180, 180 deg. step: 7.5 deg.

With this resolution, the probe is able to inspect a part with a total number of probe orientations equal to: (105/7.5 + 1)(360/7.5) - 47 = 673. The total number is not 720 as mentioned in (Lim and Menq, 1994), since the orientations with θ = 90 deg. and φ = -180, 180 are identical.

Examples of discrete accessibility domains are shown in Figure 3.19.

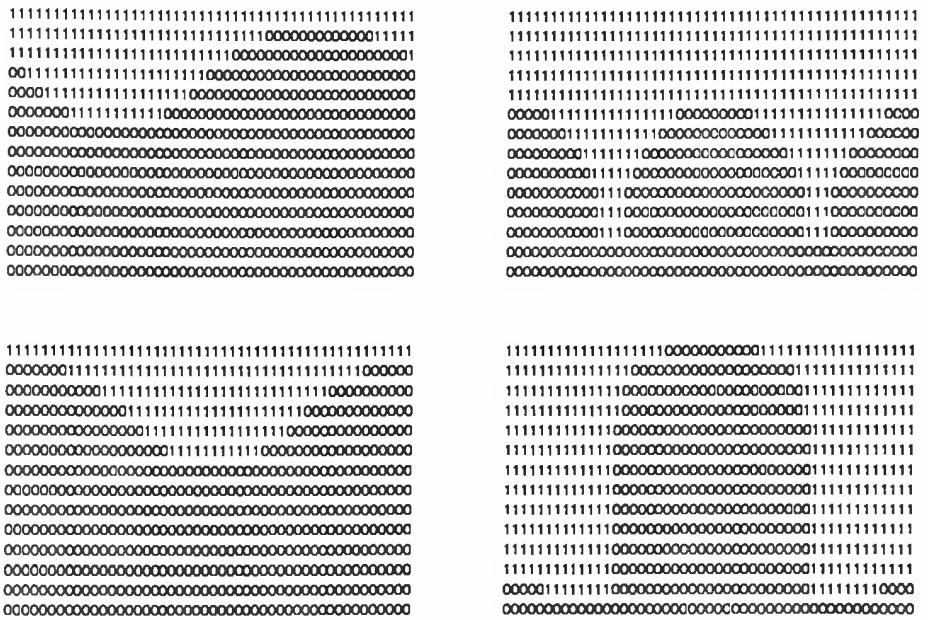


Figure 3.19 Discrete Accessibility Domains of Points *P11*, *P8*, *P12* and *P9*

3.3 DISCUSSION

The presented ICSS method has many advantages, the major ones being its generality and its simplicity. In effect, it can be performed using the solid model of the part (as well as its clamps and fixtures) and the standard geometric transformations available in any CAD system with exact representation of solids, surfaces and curves. It is not limited to a particular surface or solid model representation and can be generalized to handle bent probes. The accessibility domain can be generated for a given probe length (e.g. maximum length of available probes) and modified by applying corrections taking into account the actual location and the actual shape and geometry of the probe used. It is also possible to determine the common accessibility domain for a set of points

directly or by comparing the accessibility domain of each point. The precision of the results may be increased according to needs by increasing the number of iterations. Computation time depends of the complexity of the part to be inspected as well as the time efficiency of the Boolean algorithms of the CAD system

The discrete approach is particularly useful when the probe(s) used is (are) known in advance. However, if the continuous accessibility domains are already computed, it is very simple to retrieve the discrete accessibility domains.

Choosing a finite radius of the probe has many advantages in both continuous and discrete accessibility analysis. It permits analyzing only the local neighborhood of the measurement points, thereby reducing calculations for complicated parts and environments. In addition, feasible orientations are not ignored.

The previous considerations as well as the corrections taking into account the actual shape of the probe and its actual position leads to exact and robust accessibility analysis.

CHAPTER 4

CLUSTERING AND SEQUENCING OF INSPECTION OPERATIONS

The step following the accessibility analysis in the inspection planning methodology is measurement points sequencing and resources allocation (machines, set-ups, probes and probe orientations). This task is performed using an optimization search technique. The principal clusters determined by the accessibility analysis procedure are used for this purpose. The first section presents integrated measurement operations sequencing and resources allocation method applied to a case including precedence constraints. The second and third sections deal with the clustering and sequencing of measurement operations when no precedence constraints are imposed. Finally, a discussion of the results is provided in the fourth section.

4.1 CLUSTERING with PRECEDENCE CONSTRAINTS

In this section we present a method for sequencing operations that allocate resources such as machines, set-ups, probes and probe orientations changes while maintaining precedence constraints between operations.

If only one type of resource is involved, the proposed method minimizes resource changes, i.e. it minimizes the number of clusters using the same resource. Alternate solutions are generated and additional refinements are performed to minimize the number of different resources used in the sequence.

The case of multiple resources (i.e. machines, set-ups and probes and probe orientations) was handled by defining a resources hierarchy. The method followed for a single resource was adapted to this case, and the different types of resources are assigned in parallel.

Precedence constraints are generally due to the methodology followed for inspection, which in turn depends on the machine characteristics, the software used for tolerance verification and analysis and some expert rules followed by inspection operators. Examples of precedence considerations include: 1) important features in terms of function and accuracy are inspected first, and 2) inspect features with form tolerances first. The software used for tolerance verification and analysis may also dictate a certain procedure for measurements, reference frame definition and part localization.

This approach solves the problem of inaccessible features, i.e. those whose measurement points are not accessible with the same probe orientation, by splitting the features into accessible sub-features implicitly at the clustering stage. However, the clustering approach presented here minimizes the number of sub-features, hence minimizing probes changes and probe orientation changes.

4.1.1 The Operation Sequencing Problem

The optimization of the inspection plan depends on a great number of parameters and constraints (ElMaraghy, H. A. and ElMaragahy, W. M., 1992 and ElMaraghy, 1993). Analytical approaches based on mathematical programming techniques such as Integer Programming and Branch and Bound rapidly become computationally impractical even for medium-sized problems (Miller and Stockman, 1989 and Prabhu et al., 1991). In addition, any optimization technique requires the definition of a cost function. This function depends on many parameters, and its evaluation becomes very difficult especially at the stage where many parameters have not yet been determined (as in the case of a least commitment strategy). Moreover, the cost function depends

on the optimization criteria, where in general more than one criterion are considered (Agapiou, 1991). These criteria often conflict and are highly coupled. Therefore, depending on the context, a compromise has to be made among different criteria hierarchically ordered according to their importance.

Optimization Criteria

We have identified eight general optimization criteria in a general CMM inspection context. They are divided in two groups:

Principal Criteria

1. Machine changes
2. Set-up changes
3. Probe changes
4. Probe orientation changes

Secondary Criteria

5. The total number of machines used
6. The total number of set-ups used
7. The total number of probes used
8. The total number of probe orientations used

An absolute ordering of these two groups of criteria can be performed by analyzing their contribution to the cost function within a specific manufacturing context. However, the above order of criteria [1, 2, 3 then 4] and [5, 6, 7 then 8] is generally accepted. In effect, a machine change costs more than a set-up change (since a machine change implicitly involves a set-up change), and a set-up change costs more than a tool change which costs more than a probe orientation change. Similarly, an additional machine costs more than an additional set-up which in turn costs more than

an additional probe. Ordering criteria from the two different groups is more difficult and needs a precise analysis of the cost of each non-verified criterion. A non-verified criterion yields a resource (machine, set-up or probe) change in the case of principal criteria, and a non-minimum number of resources used in the case of secondary criteria.

Penalties respecting criteria hierarchy are assigned to each non-verified criterion to take into account the hierarchy of principal criteria. The simplest way to achieve this is to assign a decreasing integer power of two penalty to the criterion. For example, 2^3, 2^2, 2^1 and 2^0. These values ensure that machine changes have the highest penalties, then set-up changes, followed by probe changes, then probe orientation changes.

Two different approaches are possible: 1) penalty ($\{O_q^k\} \cup \{O_j^{k+1}\}$), a local approach where these penalties are calculated (locally) for the last operation of the sub-sequence and the set of possible next operations, and 2) penalty ($S_m^{k-1} \cup \{O_q^k\} \cup \{O_j^{k+1}\}$), a global approach where these penalties are calculated for the entire sub-sequence and the possible subsequent operations. In the global approach, the penalty of two set-up changes equals the penalty of one machine change, which in turn equals four tool changes; hence a compromise between the different resources can be accomplished. These penalties depend therefore on the context, and they can be modified if a global approach is adopted to take into account real costs (time and money) for non-verified criteria.

Remark: by sub-sequence, we mean a sequence i S_i^k of k operations ($(k < n)$, n being the total number of operations) which will be extended with an additional operation O_j^{k+1} (operation j ranked $k+1$):

$$S_i^{k+1} = S_i^k \cup \{O_j^{k+1}\} = S_m^{k-1} \cup \{O_q^k\} \cup \{O_j^{k+1}\} \qquad (4.1)$$

If a local approach is adopted, these penalties permit sorting the sub-sequences in only one pass. In effect, if they are not utilized, four sorting stages are necessary: first, the *S1* sub-sequences that do not cause machine changes (if possible) are chosen; then, these *S1* sub-sequences are sorted in order to select only the *S2* sub-sequences which do not cause set-up changes; next, an *S3* selection in the last *S2* sub-sequences according to probe changes is performed, finally, an *S4* selection is made in the last *S3* sub-sequences according to probe orientation changes.

Further refinements of the previous search are possible by considering the secondary criteria, as well as others such as minimizing the probe travelled distance.

Remark: A unified optimization criterion is without doubt the cost expressed in terms of money or time of each solution. If this criterion is considered from the beginning, expensive calculations will be needed for each alternative resource assignment. In addition, the search space cannot be reduced, which will result in a combinatorial explosion. However, by considering the previous main criteria, undesirable resources are eliminated, thus reducing considerably the search space.

4.1.2 Minimizing Resource Changes

In this section, a resource-change minimization method is presented. This method is based on a selective breadth-first search, where only selected branches of the search tree are expanded. Given a set of operations with alternative resources (one type of resource, i.e. machine, set-up or tool), and a set of precedence constraints between these operations, the aim is to generate the sequence(s) that minimize the resource change and assign only one resource to each operation.

Note that this method optimizes the sequences according to only one of the principal criteria (i.e. machine, set-up, probe changes or probe orientation changes).

Proposition

We propose to establish a sequence of inspection operations where the first task is fixed and (if possible) subsequent tasks which do not break the previously formed cluster are chosen. The resulting number of clusters will consequently be the minimum possible, given the fixed first operation. A minimum number of clusters yields a minimum number of resource changes.

Remarks: 1) The sequence of operations is established by choosing the first task from the possible first operations according to the precedence constraints, then the second operation in the sequence is chosen from among the operations liberated from their precedence constraints, and so on, until all operations are sequenced. 2) If a task uses the same resource as the previous task, they belong to the same cluster.

Proof

Assume that a penalty of one unit is associated with a resource change. An isolated operation will cause two resource changes, thus will be penalized with two units; a clustered operation; however, will not cause a resource change, and does not incur any penalty.

Suppose that (i-1) tasks among n have already been sequenced, and that at the stage i ($i < n$), two tasks are possible, Oj and Ok, Oj using the same resource as the previous task, and Ok using a different resource.

Six different cases are possible (Figure 4.1):

	Penalty
I. If Oj is chosen (before Ok)	
1. Ok will be clustered with other operations	**0**
2. Ok will not be clustered with other operations	**2**

II. If *Ok* is chosen (before *Oj*)

Ok is clustered with other operations

3. *Oj* will be clustered with other tasks **0**

4. *Oj* will not be clustered with other tasks **2**

Ok is not clustered with other operations

5. *Oj* will be clustered with other tasks **2**

6. *Oj* will not be clustered with other tasks **4**

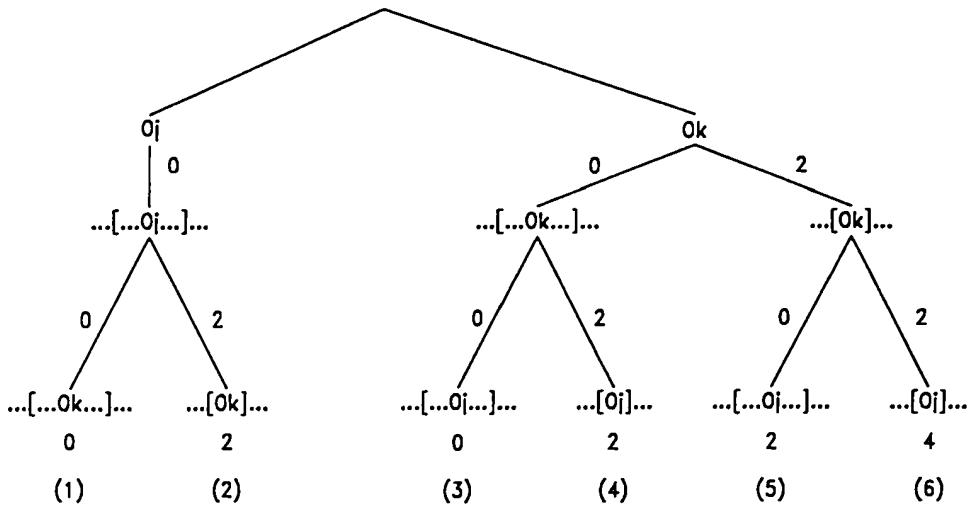


Figure 4.1 Principle of the Clustering Algorithm

The only case where choosing *Ok* before *Oj* may be better in terms of penalty than choosing *Oj* before *Ok* is when both hypotheses of cases (2) and (3) (see Figure 4.1) are simultaneously verified.

Note that an operation might not be clustered with another one, for the following reasons: 1) no further clusters might be possible using the same resource, and 2) precedence constraints might not allow the clustering.

Case (2) hypothesis implies that ***Ok*** cannot be clustered with other tasks when performed after task ***Oj***, and Case (3) hypothesis implies that ***Ok*** can be clustered with other tasks when performed before task ***Oj***.

Oj and ***Ok*** being independent (i.e. precedence constraints between them do not exist), these two hypotheses are contradictory and cannot be verified simultaneously. Choosing ***Ok*** before ***Oj*** will at most lead to a solution as good as that obtained by choosing ***Oj*** before ***Ok***; thus, always choosing ***Oj*** (the task that does not break the cluster) ensures the optimality of the final sequence.

Global optimal sequences will be generated by performing this procedure for all possible first tasks, and selecting at each step the best sub-sequences. Hence, the generation of optimal solution(s) is guaranteed. However, in general, not all optimal solutions are obtained.

This method may be summarized as:

Step 1

Initialization: Determine the set of all possible first operations (i.e. operations which do not have predecessors). They will form clusters of one operation.

Step 2

For each of previous clusters:

1. Determine the set of possible operations (i.e. operations whose predecessors have been already clustered).
2. Keep only those operations (active operations) that use the same resource as the last operation clustered. If no operation verifies the last condition, consider possible operations as active operations.
3. Extend the cluster with one operation from the active operations.

Step 3

If all operations are sequenced, then quit; otherwise, return to step 2.

The advantages of this method are:

1. An optimal solution is always guaranteed.
2. There is no backtracking in the search (as in A* and best first algorithms (Laperriere, 1992, Nielsson, 1973 and Rich, 1991)), hence reducing the storage of intermediate results and speeding the search.
3. Alternative optimal solutions are easy to generate.
4. The approach remains valid if there is no precedence.
5. Time is polynomial with exactly (*n*-1) expansions

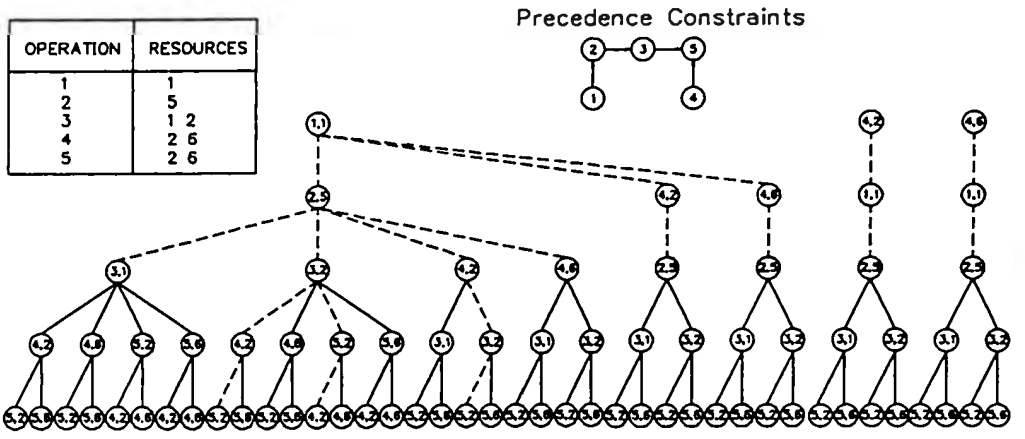


Figure 4.2 Example of Operations Sequencing and Resources Allocation

An example illustrating the basis of this sequencing method is shown in Figure 4.2, where dashed edges represent the best choices at each level of the decision tree. This example consists of five operations and four resources. The first possible operations are operations 1 and 4. The possible sequences of two operations are: **{[(1,1)(2,5)], [(1,1)(4,2)], [(1,1)(4,6)], [(4,2)(1,1)], [(4,6)(1,1)]}**. All second and third operations induce resource changes. However, four out of twenty possible sequences of four operations do not cause resource changes and three of forty possible sequences of five operations do not yield resource changes. They constitute the optimum sequences and use only three different resources.

4.1.3 Multicriteria Trade-Off Sequencing Algorithm

In the case of multiple resources, the same principle used for solving the single resource problem is applied, and the choice of the next operations is based on a comparison of the cumulative penalties of all possibilities. If the number of possible sub-sequences obtained is greater than a fixed number ***m***, further refinements can be applied. If it remains larger than the fixed value, only ***m*** sub-sequences are retained for next iterations. Limiting the maximum number of generated sub-sequences may cause the loss of some feasible solutions. However, this is often necessary in order to avoid a combinatorial explosion during the search. The value of this limit, ***m***, depends on the size of the problem and available computation capacity.

A characteristic of the assigned costs or penalties is that their value is dynamic (i.e. it depends on previous tasks); therefore, a representation in a graph form or making use of the graph algorithms is impossible.

This approach bears some semblance to the nearest-neighbor search (Foulds, 1984) and best-first search (Rich, 1991); however, no estimate of the additional cost of getting from the current node to the goal node is needed, and no backtracking is performed. Note that the solutions generated by the nearest-neighbor search are reported to cost about 6% more than the optimal

solutions (Foulds, 1984).

Large problems involving 100 different operations, 171 precedence constraints, up to 3 different machines, up to 2 different set-ups and up to 4 different tools per operation were solved. To avoid a combinatorial explosion, a maximum number of alternative sub-sequences m was specified. Table 4.1 presents the results obtained for different values of m. Calculations were performed on Sparc2 workstation using LISP language. Previous results permit verifying that the difference between the costs for different values of m is about 6% of the optimal cost. The cost for $m = 50$ may be considered the minimum, since it did not change for $50 \leq m$. We notice that for $m = 1$, a cost lower than that for $m = 2$ and $m = 5$ is obtained. This is due to the effect of neglecting, at earlier stages of the search, apparently non-interesting solutions (i.e. truncating the breadth of the search tree).

Table 4.1 Typical Example of Computation Time

m	CPU time	Cumulative penalties	% of cost
1	0.28 sec	168	3.70
2	0.29 sec	172	6.17
5	0.62 sec	169	4.32
10	1.26 sec	165	1.85
20	4.14 sec	165	1.85
50	9.35 sec	162	0.00
70	23.07 sec	162	0.00
100	23.57 sec	162	0.00

4.2 CLUSTERING WITHOUT PRECEDENCE CONSTRAINTS

As mentioned previously, the algorithm developed in the case of clustering with precedence constraints remains valid in this case and the algorithm is greatly simplified.

In the case of a single resource (e.g. probe orientation), the clusters are established as follows:

Step 1

Initialize by choosing the resources used by the maximum number of operations. The associated operations will constitute the largest initial clusters of operations.

Step 2

For each of the previous initial clusters:

1. Eliminate these operations from the set of active operations.
2. Determine the resources using the largest number of active operations. Their associated operations will constitute the following clusters.

Step 3

If the set of active operation is not empty, then return to step 2.

When sequencing measurement operations with the minimization of probe orientations changes, the tools/operations representation deduced from the accessibility analysis procedure is utilized.

In the case of multiple resource types, the optimization criteria (principal then secondary) are used to refine the solutions sequentially. The algorithm for single-resource allocation is utilized to allocate the most important resource. The solutions obtained are then refined according to the following next highest ranking resources.

4.3 SEQUENCING

In the case of clustering without precedence constraints, the solution generated by the clustering algorithm are not sequenced. A sequencing procedure has to be performed in order to choose the best sequence of operations.

In the case of inspection using CMMs, the optimization criterion considered is the probe's total travelled distance. Each cluster of measurement points is sequenced. This problem is similar to the classical Travelling Salesman Problem (TSP) (Whitley et al., 1989), hence can be solved using exact methods such as the branch and bound technique or approximated methods such as the insertion algorithm (Syslo et al., 1983).

The cities in the TSP problem represent the measurement points in clusters of points using the same probe orientation, the same probe, the same set-up and the same machine. The actual distance travelled by the probe between two measurement points is generally difficult to determine at this stage of the planning. It needs an analysis of the path which must be free of collision. For a cluster of N points, $N(N-1)/2$ different distances have to be determined for each probe orientation. This results in long and very expensive calculations. For the sake of simplicity, the distance considered between each two points is the direct distance or euclidian distance calculated directly from the coordinates.

4.3.1 Branch and Bound Technique

This algorithm is based on a tree search where at each step all possible solutions of the current problem are partitioned into two or more subsets, each represented by nodes in a decision tree. In this case, the solutions are partitioned at each step into two subsets: those that contain a specific edge (i, j) and those that do not. This branching is performed according to some heuristics which shorten the search to be conducted for the optimal solution. After branching, lower bounds are computed on the cost of each of the subsets. The next solution space to be searched is chosen

as the one with the smaller of the two lower cost bounds. This process is continued until a Hamiltonian cycle is obtained. Then, only those subsets of solutions whose lower bounds are smaller than the value of the current solution need to be searched.

The branch and bound algorithm is basically a method of exhaustive searching, and for a worst-case input, one may end up examining all possible solutions. That is a complexity as bad as $O(n!)$ (n-cities asymmetric TSP).

More details on the branch and bound algorithm may be found in Appendix A.

4.4 DISCUSSION

Clustering and sequencing in previously developed inspection planning systems is based on heuristics which generate only feasible or near-optimal solutions. In our approach the optimization is performed at two levels: 1) clustering by grouping the maximum number of measurement points in the same set-up, same probe and same probe-orientation, and 2) measurement points sequencing while minimizing the total travelled distance of the probes in each cluster.

A straight forward method based on a selective-breadth search which integrates the sequencing of operations and the allocation of resources has been presented. The optimality of this method in the case of a single resource has been proven. It generates inspection planning solutions with minimum number resource changes and minimum number of resources.

This previous method has been extended to multiple resource allocations. In this case, a near-optimal solution is obtained, and compromises taking into account a hierarchy defined among the resources are reached within an acceptable period of time (see Table 4.1). A maximum number of alternative sequences can be imposed to handle large problems where storage and computation capacities are limited. However, the storage capacity for the proposed method is lower than the storage capacity in cases where backtracking is needed.

The methods proposed in this chapter are not limited to inspection planning. They may be used in process planning in general. These methods overcome the problems caused by the exhaustive approach which can be applied only for small problems to avoid combinatorial explosion, and the heuristic approach which generates only feasible or near-to-optimal solutions. They are especially useful for a high-level process planner or at an early stage of the product development, and can be implemented as a sequencing and resource allocation kernel in a computer-assisted process planner, and the ability to generate alternative solutions adds flexibility to the resulting process plans.

CHAPTER 5

COLLISION-FREE INSPECTION PATH PLANNING

The path between each two points without collision or interference with the part and its environment has to be determined as the final part in the inspection planning process.

The first section of this chapter is an introduction, while the second section deals with the discretization of the probe's workspace. In section three an exact collision-free path planning approach is detailed. Section four presents an approximate collision-free path planning approach. Finally, section five discusses the results.

5.1 INTRODUCTION

The problem is to determine the shortest path joining two points, while avoiding collision and interference with the part and its environment (Campbel and Luh, 1980, Hu et al., 1993 and Latombe, 1991).

Dijkistra's shortest path, which is the most convenient algorithm for this kind of problem, has been used (Syslo et al., 1983). This solution has been preferred over the heuristic approach because of its generality and robustness.

Dijkistra's algorithm is $O(n^2)$ in the worst case (Syslo et al., 1983), then the network is given in the form of a weight matrix. If the network is sparse, i.e. the number of edges is much smaller than $n(n-1)$, it is possible to reduce the actual computation time by selecting a different data structure (e.g. adjacency lists) and by considering for updating only those nodes that are immediate successors of the recent node (see Appendix B).

Most previous works deal with 2-D path planning. A summary of the works related to 3-D path planning with their time complexity (i.e. an evaluation of the computation time function of the size of the problem) is given below.

In the case of Lozano-Perez's (1979, 1981 and 1983) configuration space approach in 3-D, the time complexity for computing each pseudo-obstacle is $O(v^2 \log v)$, where v is the order of the number of the vertices of the moving convex polyhedra and the polyhedra obstacles. The A* algorithm is then used to search for the solution path in a graph formed by connecting all the object vertices. The A* algorithm was shown to require $O(2^N)$ steps in the worst case for searching a graph with N nodes.

An algorithm based on optimization and used for a path search along a fixed set of edges is described in (Wong and Fu, 1986). The time complexity of the algorithm is $O(N^2)$ where N is the number of edges to be traversed. The algorithm finds the optimal three-dimensional path in terms of euclidian distance.

Note that in previous inspection planning systems, the path planning has been exclusively based on heuristics generating only feasible paths (see Table 2.1).

A direct and exact approach to this problem is to integrate the shortest path algorithm with a solid modeler and to verify at each step of the algorithm if there are first-order and second-order interferences. However, this solution requires a long computation time and is practically infeasible. The approach followed in this work is to approximate the probe, the part and the surrounding objects with a set of enveloping boxes.

5.2 WORKING SPACE DISCRETIZATION

Dijkistra's shortest path algorithm uses an adjacency matrix representing the topology of the nodes and their mutual distances. However, in this case, the working space is discretized using uniform cartesian mapping and parallelepiped elements (Figure 5.1); hence, each node has at most

26 adjacent nodes in 3-D, and the global adjacency matrix is very sparse. However, only the local adjacency matrix, which can be determined very easily without any need for storage if the nodes are numbered in a special manner, is used.

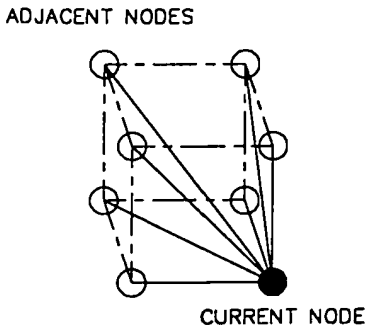


Figure 5.1 Parallelepiped Element of the Discretized Workspace

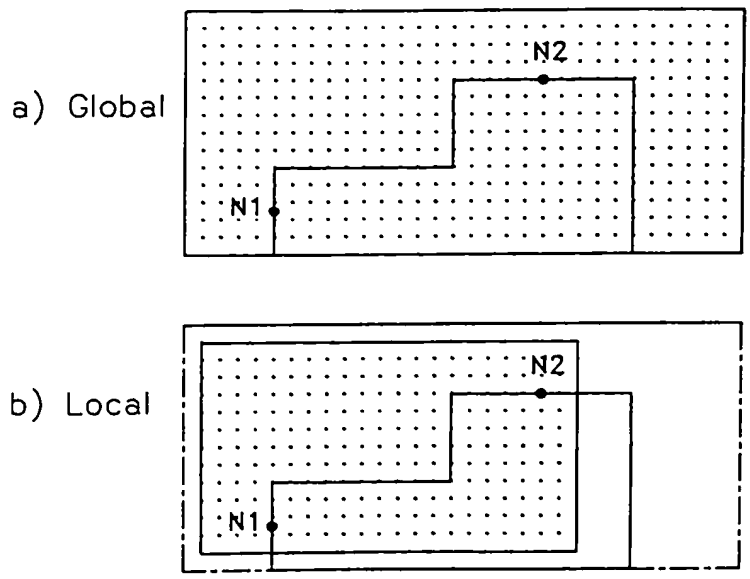


Figure 5.2 Workspace Discretization: a) Global and b) Local

Only the domain surrounding the two points is discretized (see Figure 5.2) in order to further reduce computations. The size of the domain depends of the size of the probe approximation. This is given by a heuristic formula depending of the size of the probe, its orientation and distance between the two points.

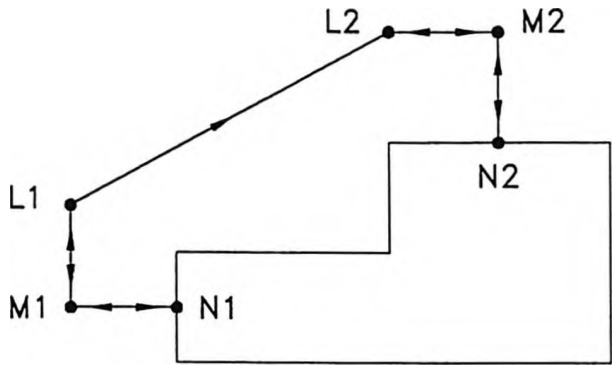


Figure 5.3 Scheme for Path Generation Between Two Measurement Points *N1* and *N2*

The path between two measurement points is decomposed into three path positions (Figure 5.3): 1) the approach path (decelerated and normal to the measurement point), 2) the drive path (accelerated), and 3) the retract path (decelerated and normal to the measurement point). Four points are then defined: the measurement points M_1, M_2 and the retract points N_1, N_2. The paths M_1N_1 and M_2N_2 are verified against collision by the accessibility analysis. In effect, the distance between these points may be fixed for all measurement points such that $|M_1N_1| = |M_2N_2| = d$. d is a measurement parameter which may be chosen using results of low level inspection planning optimization (Jones and Ulsoy 1994). It is sufficient to take a probe length equal to $R+d$ (R being the length of the probe and $d \prec R$) to ensure obtaining a collision-free path between these points. The reference of the machine is added to the clusters as an additional point (to reset the probe after each change in the set-up, the probe, or probe orientation.

5.3 OBJECTS APPROXIMATION

The part and the environment objects are decomposed into a set of Cartesian boxes manually (see Figure 5.4), this is the only part of the system which is not yet completely automated. An approach similar to that of the octree decomposition may be used to automate this procedure (Chien and Aggarwal, 1986 and Pujair and Reddy, 1989).

Octrees are hierarchical structures that reflect the recursive subdivision of objects into variably sized cubes. In octree encoding, the object is enclosed inside a cube. If the object does not uniformly cover the cube, then we subdivide the cube into eight octants. If any of the resultant octant is full (i.e. completely inside the object) or empty (i.e. completely outside the object), no further subdivision is made. If any of the octants is partially full, we subdivide it again into octants. We continue to subdivide the partially full octants until the resulting octants are either full or empty or until some predetermined level of subdivision is reached.

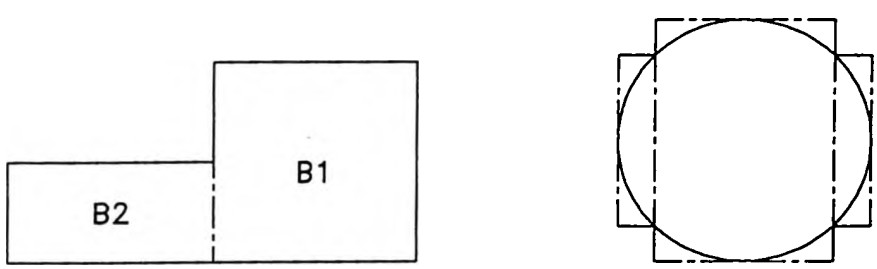


Figure 5.4 Examples of Objects Approximation

The approximation of the probe is automatic and depends of the orientation of the probe. In a first step, the probe is approximated using a set of spheres. These spheres approximate each cylinder of the probe model. This approximation is independent of the probe orientation. It will be used to generate the approximating boxes of the probe according to its orientation (see Figure 5.5).

A safety factor (i.e. 10 to 20%) that increases the size of boxes is applied when approximating the probe.

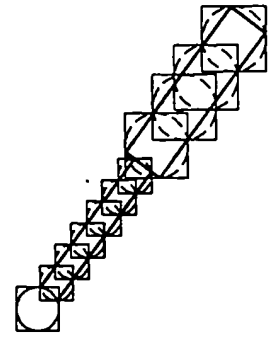


Figure 5.5 Probe Approximation

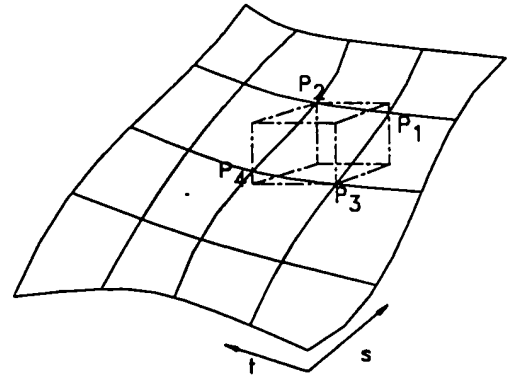


Figure 5.6 Approximation of Complex Surfaces

In the case of inspecting complex surfaces (Chapter 6), it is also possible to represent the surface using a set of boxes. This can be achieved by using an isoparametric mapping of the surface and by considering the enveloping box of each element of the mapping (see Figure 5.6). This approximation may be refined by choosing smaller steps in the isoparametric mapping of the

surface.

Note that it is very important to have the simplest representation of the object(s) with the minimum number of boxes. The main reason for this is to reduce computation time when checking for interference. Procedures are used to reduce the number of boxes by merging adjacent boxes which can be represented by a single box (see Figure 5.7). An example of application of these simplifications is when the probe is parallel to the coordinates axes.

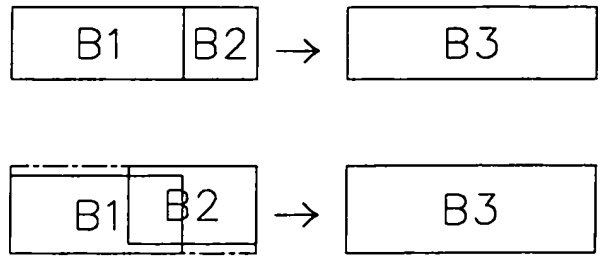


Figure 5.7 Examples of Merging Boxes

5.4 INTERFERENCE CHECKING

Two kinds of interferences were considered (Figure 5.8): 1) a first order (or static) interference, which occurs when the probe is at a specified position, and 2) a second order (or dynamic) interference, which occurs when the probe moves between two points; this interference occurs between the volume swept by the probe and the workpiece or other surrounding objects.

Offsetting all the faces of the part and surrounding objects by a fixed distance (valid for near-to-spherical objects) is an approach that does not correctly suit the case of the measurement probe, and which does not possess spherical symmetry since the length of the probe is larger than its diameter. In addition, the probe is not always in a vertical position. Therefore, in 3-D, the problem of planning the path of the measurement probe, cannot in general, be simplified to point path planning.

Simplifications used in Interference Checking

* If a first order interference is detected at a given node, then the node is removed from the list of possible nodes.
* If the box enveloping all the boxes of the probe does not interfere with an obstacle, then it is not necessary to check each elementary box of the probe for interference.
* If the enveloping box of the probe moving between two positions does not interfere with an obstacle, then it is not necessary to check each elementary box of the probe for interference.

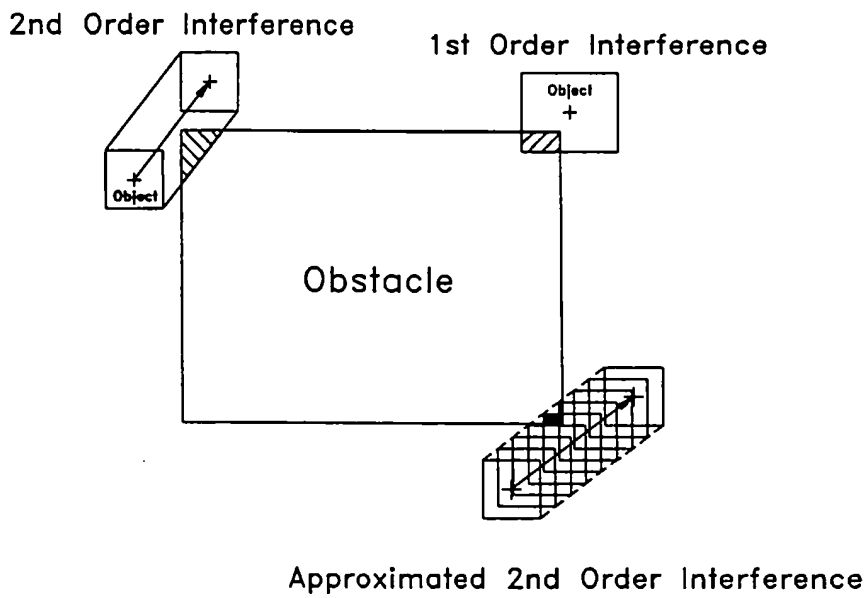


Figure 5.8 First Order and Second Order Interference Checking

5.4.1 Checking First Order Interference

This case can be detected easily. It is sufficient to compare each one of the boxes representing the moving probe to each box representing the part and its environment.

A box is defined by its two extreme corners $B\{[x_{min}, y_{min}, z_{min}]\ [x_{max}, y_{max}, z_{max}]\}$. Two boxes *B1* and *B2* do not overlap if and only if:

$$x^1_{min} > x^2_{max} \quad or \quad y^1_{min} > y^2_{max} \quad or \quad z^1_{min} > z^2_{max}$$
$$or \qquad (5.1)$$
$$x^2_{min} > x^1_{max} \quad or \quad y^2_{min} > y^1_{max} \quad or \quad z^2_{min} > z^1_{max}$$

The time complexity for computing the first order interference checking is $O(m\ n)$, where m is the number of boxes representing the probe and n the number of boxes representing the obstacles. An algorithm for reporting intersection of n boxes is proposed by Hoffmann (1989). The complexity of the algorithm is $O(n\ log^2(n) + J)$, where J is the number of actual intersections. This algorithm is, however, very difficult to implement.

5.4.2 Checking Second Order Interference

This type of interference occurs when moving from one point to another, and is caused by the object swept volume, which interferes with the part and its environment. This type of interference is more difficult to analyze. Both exact and approximated approaches were evaluated:

1) The swept volume of the box is compared to the boxes of the part and its environment. Calculations involving the evaluation and comparison of the position of the vertices, edges and faces of the box and the external faces of the swept volume are performed.

2) The approximated approach consists of dividing the path between the two points into small sub-paths, and checking at the intermediate nodes first order interferences (see Figure 5.6).

The time complexity for computing the second order interference checking is $O(k$

m n), where *m* is the number of boxes representing the probe, *n* the number of boxes representing the obstacles and *k* the number intermediate nodes.

The second approach was by far computationally more efficient, hence was selected. It should be noted that 1) each node where a first order interference is detected is removed from the list of possible positions, and 2) the path generated is the shortest one for the specified discretization.

5.5 DISCUSSION

This part of the inspection planning procedure is the most delicate and time consuming. It is part of the low level planning, hence the optimization of the path is very important for minimizing inspection cost.

Path planning in previously developed inspection planning systems was based mainly on heuristics. The path is generated then verified against collision by simulation. If a collision is detected, the user modifies it interactively or a set of heuristic rules are used to move the probe away from the interference region. The approach used in the proposed and developed system is based on a general algorithm which generates the shortest collision-free probing path, hence eliminating the need for post-verification and simulation. The only task which is not completely automated is the modelling approximation of the workpiece and its environment, however, existing techniques, such as octree decomposition, may be used to fully automate the path planning procedures.

CHAPTER 6

GEOMETRIC MODELING USING DUAL KRIGING

The objective of tactile inspection is to generate a set of points which are then used to approximate the inspected feature (i.e. curve or surface). The number and location of these points depend of the shape of the feature and the type and value of the dimensional or geometric tolerance to be verified. The inspection feature may be approximated using classical interpolation techniques such polynomial, Bezier, Spline, B-splines and NURBS. The model obtained is then analyzed and compared to the theoretical or CAD model in order to determine its conformity to tolerance specifications. The interpolation technique used in this thesis research is dual Kriging.

This chapter generalizes the concepts of dual Kriging applied to geometric modeling and presents its theoretical foundations. The equations of the parametric model are derived in a simple and novel way by considering the combination of interpolation profiles, hence permitting the interpolation of both discrete and continuous data as well as the interpolation in n-D. A new technique for the incorporation of uncertainties in the data points as well as special forms of linear constraints have been adapted to dual Kriging of parametric curves and surfaces.

Section 1 is an introduction. Section 2 presents the theoretical foundations of dual Kriging applied to the interpolation of positions and derivatives in 1-D. Invariance under affine transformation of the interpolation model is proved. Section 3 presents the application of dual Kriging to parametric surface modeling. In section 4 parametric Kriging is applied to the interpolation of solids. Section 5 examines the foundation of the nugget effect technique and its extension to parametric curve/surface modeling. In section 6, the incorporation of linear constraints

in Kriging formulation is described. Finally, a discussion is presented in section 7.

6.1 INTRODUCTION

Kriging is a statistical technique first proposed in 1951 by Krige for evaluating natural resources. Its mathematical foundations have been established by Matheron (1973). Kriging is the best linear unbiased estimator of a random function as presented in the mathematical framework of geostatistics. It requires, however, the solution of a new linear system for each interpolated value. Kriging has been extensively used for cartography and contour maps (Chiles, 1973, Chiles, 1977 and Trochu, 1993).

Dual Kriging is a refinement of the primary form. It provides an explicit interpolation formula and is computationally more efficient than the primary Kriging formulation. Dual Kriging is a general method that incorporates several interpolation techniques such as piecewise interpolation, cubic splines (Matheron, 1980), Bezier (Montes, 1990), B-splines and NURBS curves, surfaces and solids in a single formulation. Table 6.1 shows how a Kriging model, using a given set of parameters reduces to known interpolation methods. Even least squares methods can be derived as a limit case of Kriging. Dual Kriging was first proposed for curve and surface modeling by Gilbert et al. (1990), then was extended to solids by Limaiem and ElMaraghy (1995). Dual Kriging is a global interpolation method; however, by choosing the covariance function properly and incorporating a distance of influence in each data point, it becomes a local interpolation technique, where the connections between two points located far apart are discarded (Trochu, 1993).

Kriging interpolation is capable of fitting data points. If data points are known within a bounded error, it is possible to interpolate while considering this maximum deviation. The interpolant in this case will not exactly fit data points; however, the deviation from data points will be bounded by the maximum error. This is accomplished by introducing the so-called nugget-effect

in the Kriging system. Usually, the intensity of the maximum deviation is chosen so as to be proportional to the variance error (Trochu, 1993). Interesting applications of this feature is the interpolation of curves and surfaces from a set of measurement points. In effect, two types of errors may occur while measuring: 1) errors caused by physical effects (see chapter 2 section 2.1.2) which may be evaluated, and 2) random errors or noise. Exact interpolation of the measured data generally generates a poor quality model since noise is not eliminated. However, allowing the model to deviate from measured data by a maximum value filters the noise and gives a smoother model. Classical least squares filter the noise in the data, however, the deviation of the model from the data points may be large locally, this is not the case when dual Kriging includes the nugget effect which controls the deviations locally at each data point. Therefore, a smoother model is obtained allowing a more accurate representation of the actual feature (i.e. curve or surface).

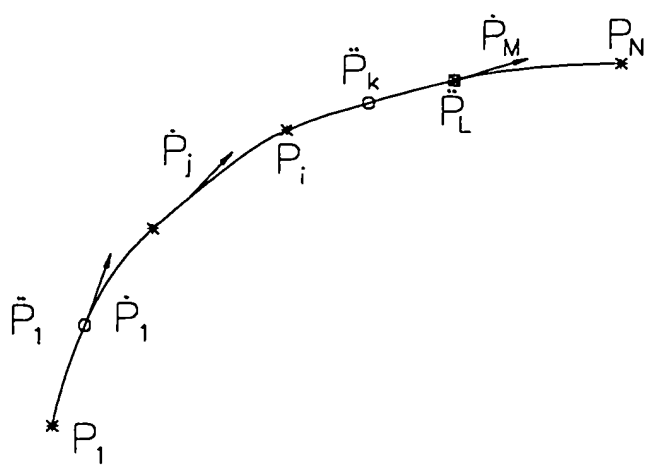


Figure 6.1 Curve Interpolation with Positions, First and Second order Derivatives

6.2 PARAMETRIC CURVES

The geometric model of a curve is defined in the three-dimensional space of cartesian coordinates x, y and z. The parametric equation is defined by three functions $P(t) = \{x(t), y(t) z(t)\}$.

Dual Kriging permits the automatic construction of equations of smooth parametric curves from a discrete number of points.

The primal formulation of Kriging with data points containing: N positions P_i, M first order derivatives $\dot{P}_j$, and L second order derivatives $\vec{P}_k$ (see Figure 6.1), is to find an interpolant $P(t)$ as a linear combination of the data points, i.e.

$$P(t) = \sum_{i=1}^{N} \lambda_i P_i + \sum_{j=1}^{M} \mu_j \dot{P}_j + \sum_{k=1}^{L} \gamma_k \vec{P}_k \tag{6.1}$$

λ_i, μ_j and γ_k are determined by minimizing the squared variance of the estimation error, i.e.

$$e(..\lambda_i..,..\mu_j..,..\gamma_k..) = E[P(t) - \sum_{i=1}^{N} \lambda_i P_i - \sum_{j=1}^{M} \mu_j \dot{P}_j - \sum_{k=1}^{L} \gamma_k \vec{P}_k]^2 \tag{6.2}$$

Additional constraints, called no-bias conditions in classical Kriging, are considered. These constraints state that the interpolant filters a given basis of finite dimensional subspace of $K+1$ functions $p_l(t)$ (e.g. polynomial, rational and trigonometric functions), i.e.

$$p_l(t) = \sum_{i=1}^{N} \lambda_i p_l(t_i) + \sum_{j=1}^{M} \mu_j \dot{p}_l(\dot{t}_j) + \sum_{k=1}^{L} \gamma_k \vec{p}_l(\vec{t}_k),\ 0 \le l \le K \tag{6.3}$$

This basis of functions generally describes the mean shape of the function.

The problem is solved by introducing the Lagrange multipliers δ_m, and minimizing

$$E = \frac{1}{2} e - \sum_{m=0}^{K} \delta_m (p_l(t) - \sum_{i=1}^{N} \lambda_i p_l(t_i) - \sum_{j=1}^{M} \mu_j \dot{p}_l(\dot{t}_j) - \sum_{k=1}^{L} \gamma_k \vec{p}_k(\vec{t}_k)) \tag{6.4}$$

After deriving E with regard to the different parameters λ_i, μ_j, γ_k and δ_m, we obtain

$$
\begin{aligned}
&\sum_{i'=1}^{N} \lambda_{i'} E[P_{i'} P_i] + \sum_{j=1}^{M} \mu_j E[P_i \dot{P}_j] + \sum_{k=1}^{L} \gamma_k E[P_i \vec{P}_k] + \sum_{m=0}^{K} \delta_m p_i(t_i) = E[P(t) P_i] \\
&\sum_{i=1}^{N} \lambda_i E[P_i \dot{P}_j] + \sum_{j'=1}^{M} \mu_{j'} E[\dot{P}_j \dot{P}_{j'}] + \sum_{k=1}^{L} \gamma_k E[\dot{P}_j \vec{P}_k] + \sum_{m=0}^{K} \delta_m \dot{p}_i(\dot{t}_j) = E[P(t) \dot{P}_j] \\
&\sum_{i=1}^{N} \delta_i E[P_i \vec{P}_k] + \sum_{j=1}^{M} \mu_j E[\dot{P}_j \vec{P}_k] + \sum_{k'=}^{L} \gamma_{k'} E[\vec{P}_{k'} \vec{P}_k] + \sum_{m=0}^{K} \delta_m \vec{p}_i(\vec{t}_k) = E[P(t) \vec{P}_k] \\
&\sum_{i=1}^{N} \delta_i p_i(t_i) + \sum_{j=1}^{N} \mu_j \dot{p}_i(\dot{t}_j) + \sum_{l=1}^{L} \gamma_k \vec{p}_i(\vec{t}_l) = p_i(t)
\end{aligned}
\tag{6.5}
$$

The intrinsic hypothesis in classical Kriging states that the covariance between two points $P(t)$ and $P(s)$ depends only on their Euclidian distance, i.e.

$$E[P(t_i) P(t_j)] = K(|t_i - t_j|) = K(h) \tag{6.6}$$

Remark: The choice of the *generalized covariance* function $K(h)$ determines if the solution of E is a minimum or a maximum. Christakos (1984) should be consulted for the conditions that $K(h)$ must verify in order to have a global minimum solution (non-negative definite system). However, in geometric modeling it is generally sufficient to have an invertible system. This is achieved by choosing $K(h)$ such that $K(|t-t_i|)$, $\dot{K}(|t-\dot{t}_j|)$, $\vec{K}(|t-\vec{t}_k|)$ and $p_l(t)$ are linearly independent for u in [0,1], $1 \le i \le N$, $1 \le j \le M$, $1 \le k \le L$ and $0 \le l \le K$.

Noting that:

$$\frac{dK(|t_i - t_j|)}{dt_i} = \frac{dK(h)}{dt_i} = \frac{dh}{dt_i}\frac{dK}{dh} = e(t_i,t_j)\dot{K}(h), \quad e(u,v) = \begin{cases} 1 & \text{if } u \ge v \\ -1 & \text{if } u < v \end{cases} \tag{6.7}$$

Using previous relations, we can deduce the following expressions:

$$
\begin{aligned}
&E[P(t)P_i] = K(|t-t_i|) = K_i \\
&E[P(t)\dot{P}_j] = e(\dot{t}_j,t)\dot{K}(|t-\dot{t}_j|) = e_j\dot{K}_j \\
&E[P(t)\vec{P}_k] = \vec{K}(|t-\vec{t}_k|) = \vec{K}_k \\
&E[P_iP_j] = K(|t_i-t_j|) = K_{ij} \\
&E[P_i\dot{P}_j] = e(\dot{t}_j,t_i)\dot{K}(|t_i-\dot{t}_j|) = e_{ji}\dot{K}_{ij} \\
&E[P_i\vec{P}_k] = \vec{K}(|t_i-\vec{t}_k|) = \vec{K}_{ik} \\
&E[\dot{P}_i\dot{P}_j] = -\ddot{K}(|\dot{t}_i-\dot{t}_j|) = -\ddot{K}_{ij} \\
&E[\dot{P}_i\vec{P}_j] = e(\dot{t}_i,\vec{t}_j)\bar{K}(|\dot{t}_i-\vec{t}_j|) = e_{ij}\bar{K}_{ij} \\
&E[\vec{P}_i\vec{P}_j] = K^{(4)}(|\vec{t}_i-\vec{t}_j|) = K^{(4)}_{ij}
\end{aligned}
\tag{6.8}
$$

Equations (6.5) can be written as:

$$
\begin{bmatrix}
\vdots & | & \vdots & | & \vdots & | & \vdots \\
\cdots\ K_{ii'} \ \cdots & | & \cdots\ e_{ji'}\dot{K}_{i'j} \ \cdots & | & \cdots\ \vec{K}_{i'k} \ \cdots & | & \cdots\ p_i(t_{i'}) \ \cdots \\
\vdots & | & \vdots & | & \vdots & | & \vdots \\
- - - & - & - - - & - & - - - & - & - - - \\
\vdots & | & \vdots & | & \vdots & | & \vdots \\
\cdots\ e_{j'i}\dot{K}_{ij'} \ \cdots & | & \cdots\ -\ddot{K}_{jj'} \ \cdots & | & \cdots\ e_{j'k}\bar{K}_{j'k} \ \cdots & | & \cdots\ \dot{p}_i(\dot{t}_{j'}) \ \cdots \\
\vdots & | & \vdots & | & \vdots & | & \vdots \\
- - - & - & - - - & - & - - - & - & - - - \\
\vdots & | & \vdots & | & \vdots & | & \vdots \\
\cdots\ \vec{K}_{ii'} \ \cdots & | & \cdots\ e_{ji'}\bar{K}_{ji'} \ \cdots & | & \cdots\ K^{(4)}_{kk'} \ \cdots & | & \cdots\ \vec{p}_i(\vec{t}_{k'}) \ \cdots \\
\vdots & | & \vdots & | & \vdots & | & \vdots \\
- - - & - & - - - & - & - - - & - & - - - \\
\vdots & | & \vdots & | & \vdots & | & 0 \ \cdots\ 0 \\
\cdots\ p_i(t_i) \ \cdots & | & \dot{p}_i(\dot{t}_j) \ \cdots & | & \cdots\ \vec{p}_i(\vec{t}_k) \ \cdots & | & \vdots\ \ddots\ \vdots \\
\vdots & | & \vdots & | & \vdots & | & 0 \ \cdots\ 0
\end{bmatrix}
\cdot
\begin{bmatrix}
\vdots \\ \lambda_{i'} \\ \vdots \\ - \\ \vdots \\ \mu_{j'} \\ \vdots \\ - \\ \vdots \\ \gamma_{k'} \\ \vdots \\ - \\ \vdots \\ \delta_i \\ \vdots
\end{bmatrix}
=
\begin{bmatrix}
\vdots \\ K_{i'} \\ \vdots \\ - \\ \vdots \\ \dot{K}_{j'} \\ \vdots \\ - \\ \vdots \\ \vec{K}_{k'} \\ \vdots \\ - \\ \vdots \\ p_i(t) \\ \vdots
\end{bmatrix}
\tag{6.9}
$$

or in compact form

$$\begin{bmatrix} K & \dot{K} & \ddot{K} & p \\ \dot{K}^T & \ddot{K}_0 & \ddot{K} & \dot{p} \\ \ddot{K}^T & \ddot{K}^T & K^{(4)} & \ddot{p} \\ p^T & \dot{p}^T & \ddot{p}^T & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \mu \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} k \\ \dot{k} \\ \ddot{k} \\ p \end{bmatrix} \tag{6.10}$$

The inversion of the previous system yields

$$\begin{bmatrix} \lambda \\ \mu \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} U & V & W & X \\ V^T & Y & R & S \\ W^T & R^T & Z & Q \\ X^T & S^T & Q^T & T \end{bmatrix} \cdot \begin{bmatrix} k \\ \dot{k} \\ \ddot{k} \\ p \end{bmatrix} \tag{6.11}$$

Recalling that

$$P(t) = \begin{bmatrix} P & \dot{P} & \ddot{P} \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \mu \\ \gamma \end{bmatrix} \tag{6.12}$$

and combining (6.11) and (6.12), we obtain

$$P(t) = [P\ \dot{P}\ \ddot{P}] \cdot \begin{bmatrix} U \\ V^T \\ W^T \end{bmatrix} \cdot [k] + [P\ \dot{P}\ \ddot{P}] \cdot \begin{bmatrix} V \\ Y \\ R^T \end{bmatrix} \cdot [\dot{k}] + [P\ \dot{P}\ \ddot{P}] \cdot \begin{bmatrix} W \\ R \\ Z \end{bmatrix} \cdot [\ddot{k}] + [P\ \dot{P}\ \ddot{P}] \cdot \begin{bmatrix} X \\ S \\ Q \end{bmatrix} \cdot [P] \tag{6.13}$$

or

$$P(t) = [b]^T \cdot [k] + [c]^T \cdot [\dot{k}] + [d]^T \cdot [\ddot{k}] + [a]^T \cdot [P] \tag{6.14}$$

The dual expression of $P(t)$ is then

$$P(t) = \sum_{i=0}^{K} a_i p_i(t) + \sum_{j=1}^{N} b_j K(|t-t_j|) + \sum_{k=1}^{M} c_k e(\dot{t}_k,t)\dot{K}(|t-\dot{t}_k|) + \sum_{l=1}^{L} d_l \vec{K}(|t-\vec{t}_l|) \tag{6.15}$$

The previous expression may be generalized to higher order derivatives by:

$$P(t) = m(t) + e(t) = \sum_{i=0}^{M} a_i p_i(t) + \sum_{j=0}^{D}\sum_{i=1}^{N_j} b_i^j (e(t_i^j,t))^j K^{(j)}(|t-t_i^j|) \tag{6.16}$$

In Kriging, the first summation $m(t)$ is called the *drift* represents the average shape of the model, and the second double summation $e(t)$ is called the fluctuation of the model. The coefficients a_i and b_i^j are determined by requiring that the interpolation passes through data points and verifies *Dth* order derivatives conditions, i.e.

$$P_m^k = P^{(k)}(t_m^k) = \sum_{i=0}^{M} a_i p_i^{(k)}(t_m^k) + \sum_{j=0}^{D}\sum_{i=1}^{N_j} b_i^j (e(t_m^k,t_i^j))^{j+k} K^{(j+k)}(|t_m^k-t_i^j|) \tag{6.17}$$

combined with the no-bias conditions

$$\sum_{j=0}^{D}\sum_{i=1}^{N_j} b_i^j p_i^{(j)}(t_i^j) = 0 \tag{6.18}$$

Remark

The same procedure is applied in the case of an implicit interpolant of the form $z = f(x,y)$. It is sufficient to replace P with z and t with the vector (x,y). The absolute value is replaced with a vectorial norm, such as the Euclidian norm.

If derivatives are not considered in the interpolation model then the parametric equations of Kriged curves can be written as follows:

$$P(t)=\sum_{i=0}^{M} a_i p_i(t) + \sum_{j=1}^{N} b_j K(|t-t_j|) \tag{6.19}$$

The parameters t_j represent a decreasing sequence of real numbers: $t_1 < t_2 < ... < t_N$. They may be determined by an approximation of the curve length calculated from $t_1 = 0$ by:

$$t_{i+1}=t_i+[(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2+(z_{i+1}-z_i)^2]^{\frac{1}{2}},\ 1\le i\le N-1 \tag{6.20}$$

These parameters are usually normalized so that t takes its value in the interval [0, 1]. Equation (6.19) represents a Kriged curve in which the terms in a_i model the average shape of the curve. The summation with the coefficient b_j is a correction about the average shape that allows the model to fit a set of N given data points.

Examples of curve interpolation with positions and derivatives data are presented in Appendix C.

6.2.1 Invariance by Affine Transformations and Translations

Invariance by affine transformations and translations property may be proved easily by considering the primal formulation of Kriging, which states that any point on a Kriged curve is determined from a linear combination of the curve data points

$$P(t) = \sum_{j=0}^{D} \sum_{i=1}^{N_j} \lambda_i^j P_i^j \tag{6.21}$$

where λ_i verifies the no-bias conditions

$$\sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j p_l^{(j)}(t_i^j) = p_l(t) \tag{6.22}$$

Applying a general affine transformation (e.g. scaling, rotation, etc.), followed by a translation $A[P] = L[P] + T$, to the Kriged curve, yields:

$$A[P(t)] = L[P(t)] + T = L[\sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j P_i^j] + T = \sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j L[P_i^j] + T \tag{6.23}$$

On the other hand,

$$\sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j A[P_i^j] = \sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j (L[P_i^j] + T) = \sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j L[P_i^j] + T\sum_{j=0}^{D}\sum_{i=1}^{N_j}\lambda_i^j \tag{6.24}$$

The conditions of no-bias in equation (6.22) states that $\Sigma\Sigma\lambda_i = 1$ for $p_l(t) = p_0(t) = 1$. This is because the constant function is always present in the basis of linear subspaces of functions. The affine image of a Kriged curve is obtained by transforming the data points (positions and derivatives) and leaving the Kriging parameters unchanged. The same property applies to surfaces and solids.

6.2.2 Equivalence to Least Squares Interpolation

The least squares interpolation of a set of data points by a linear combination of a basis of functions is obtained by considering the drift as a linear combination of these functions, and taking the covariance function to be equal to the function $K(0) = 1$, and

$K(h) = 0$ for $h \neq 0$.

In effect, equation (6.17) may be written in the following matrix form.

$$\begin{bmatrix} K & D \\ D^T & 0 \end{bmatrix} \begin{bmatrix} B \\ A \end{bmatrix} = \begin{bmatrix} P \\ 0 \end{bmatrix} \tag{6.25}$$

If $K = I$, the $N \times N$ identity matrix, then the previous equation yields:

$$\begin{cases} KB + DA = P \\ D^T B = 0 \end{cases} \rightarrow \begin{cases} A = (D^T D)^{-1} D^T P \\ B = 0 \end{cases} \tag{6.26}$$

which is the least squares interpolation of data points P, using the basis of functions defining the drift (Farebrother, 1988).

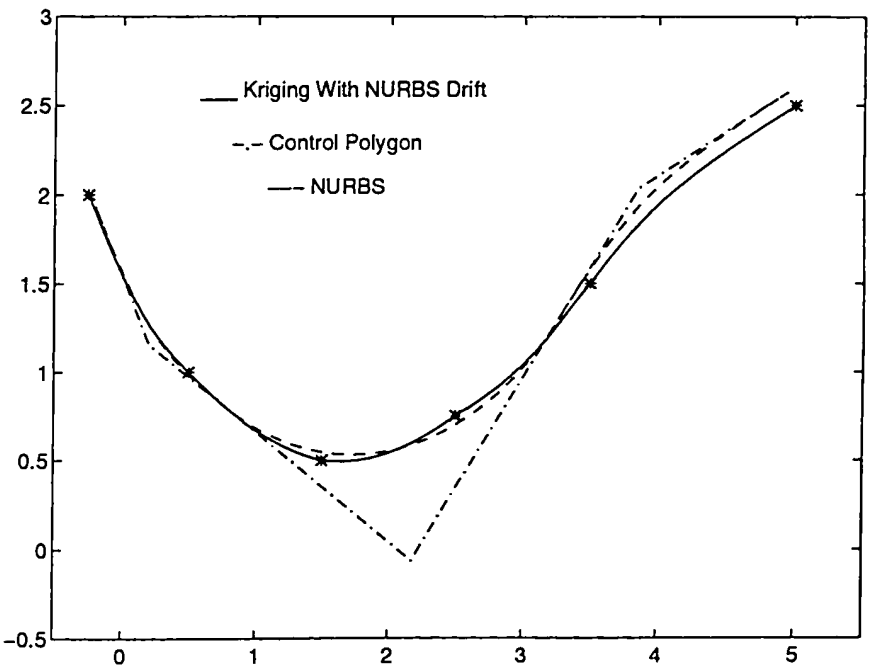


Figure 6.2 Kriging with a NURBS Drift

6.2.3 Equivalence with B-splines and NURBS

The equivalence with NURBS is quite easy to verify. Simply consider the drift as a linear combination of the B-spline functions, or the rational B-spline functions, and consider a null

covariance function. Then the a_i represent the control points of the NURBS curve. In the case of data interpolation using B-splines or NURBS, the least squares method is used (Piegl and Tiller, 1987).

Figure 6.2 shows an example of curve interpolation using a quadratic NURBS drift, the weight vector is [0.1 0.3 0.2 0.3 0.1] and the knots vector [0 0 0 1/4 2/3 1 1 1]. With a covariance $K(h) = h^{3/2}$, the continuous curve fits all the data points, and the dashed curve represents the NURBS created from the control points a_i obtained by Kriging.

6.2.4 Smoothness of Kriging Curves

The continuity of Kriging profiles depends on the continuity of the drift and the fluctuation of the model. Usually, the drift is C^n, the continuity is then limited by the fluctuation terms. The absolute value in the generalized covariance terms is generally the cause of eventual discontinuities. Polynomial and trigonometric drifts combined with a generalized covariance of the form Σh^{2k+1} are the most widely used Kriging profiles, their continuity is generally C^{2k}.

Table 6.1 Equivalence between Kriging and Other Interpolation Techniques

	Kriging Parameters		
Interpolation Method	Covariance	Drift	Derivatives
Piecewise Linear	K(h)=h	constant	-
Cubic Splines	$K(h)=h^3$	linear	-
Bezier	$K(h)=h^3$	linear	at end points
B-Splines	K(h)=0	B-spline basis functions	
NURBS	K(h)=0	NURBS basis functions	
Least Squares	K(0)=1 K(h)=0; h≠0	any	

6.2.5 Computation Efficiency

The computation efficiency of dual Kriging interpolation may not be readily obvious. Kriging interpolation involves the inversion of a full matrix, however, for NURBS and B-splines only a band matrix is inverted. These inversions are performed once for the entire Kriging model. When using existing interpolation methods, the recursive character of B-splines and NURBS interpolation methods requires significantly longer time.

A comparison between: i) the computation time required for the interpolation of a circle (7 control points) using quadratic NURBS and B-splines (note that the resulting curve is not an exact circle with B-splines) and a trigonometric drift combined with a linear covariance for Kriging interpolation using four points, and ii) the time required for the interpolation of an open curve using cubic B-spline and NURBS and Kriging with a cubic covariance and a linear drift. One thousand points were calculated by each method. In the first example, Kriging was about five times faster than NURBS and about two times faster than B-splines. The difference is even larger in the second example where Kriging was about twenty times faster than NURBS and four times faster than B-splines.

6.3 PARAMETRIC SURFACES

In the following sections we will consider the case where only the position coordinates of the data points are used. The case of profiles with derivatives may be treated by combining two profile of the general form defined by equations (6.16) to (6.18).

6.3.1 Interpolating a Grid of Data Points

A three-dimensional surface is described by a parametric equation of the form $P = P(u,v)$, with component functions $x = x(u,v)$, $y = y(u,v)$, and $z = z(u,v)$. A parametric surface is defined by two Kriging profiles **A** and **B** along the u and v directions respectively. A Kriging profile

consists of a drift and a generalized covariance that govern the shape of the surface. A grid of N_u by N_v data points defines the surface, that is N_v sections along the v direction, with each section defined by N_u data points $P_{ij}(x_{ij}, y_{ij}, z_{ij})$ (see Figure 6.3).

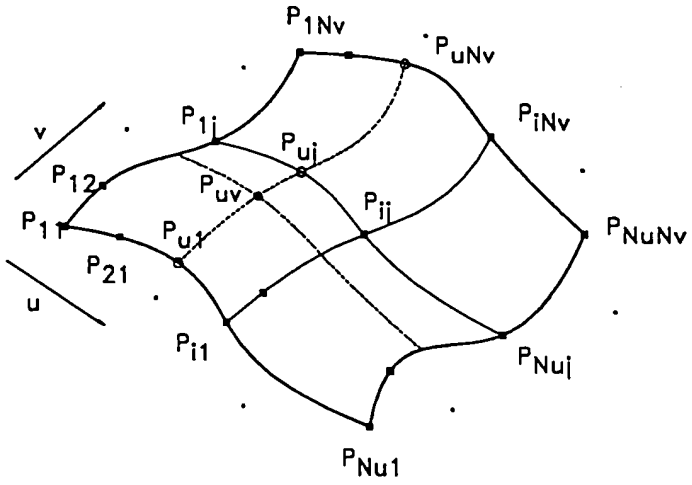


Figure 6.3 Discrete Surface Interpolation

The parametric equations of a curve in the u (respectively v) direction for a Kriging profile with a drift linear combination of $p_i(u)$ (respectively $q_k(v)$) and a generalized covariance $K_a(h)$ (respectively $K_b(h)$) can be written as follows:

$$P_{v_m}(u) = \sum_{i=0}^{M_u} a_i\, p_i(u) + \sum_{j=1}^{N_u} b_j K_a(|u-u_j|) \tag{6.27}$$

$$P_{u_n}(v) = \sum_{k=0}^{M_v} c_k\, q_k(v) + \sum_{l=1}^{N_v} d_l K_b(|v-v_l|) \tag{6.28}$$

u and v are determined in a manner similar to curves, using (6.19) along each profile.

It is possible to determine the coordinates of any point belonging to the generating profiles, that is points $P(u_m,v)$ or $P(u,v_n)$. However, in the general case where $P = P(u,v)$, another equation involving both profiles has to be defined. This is accomplished in the following steps (see Figure 6.2):

1. Determine a_i and b_j of equation (6.27) for each section by solving N_v linear systems: $P_m(u_j) = P_{ij}$, $1 \le j \le N_v$
2. Assuming that $u = const.$, we can determine N_v intermediate points $P_{iu} = P_m(u)$, $1 \le i \le N_u$.
3. Use previous intermediate points P_{iu} to create the isoparametric curve $u = const.$, and determine the corresponding c_k and d_l, then calculate $P_u(v)$ using (6.28).

Using matrix notation, the previous steps may be written as follows. Step 1 is equivalent to solving the system of equations

$$\begin{bmatrix} & \vdots & & | & p_0(u_1) & \cdots & p_{M_a}(u_1) \\ \cdots & K_a(|u_i-u_j|) & \cdots & | & p_0(u_j) & \cdots & p_{M_a}(u_j) \\ & \vdots & & | & p_0(u_{N_a}) & \cdots & p_{M_a}(u_{N_a}) \\ --- & --- & --- & - & - & - & - \\ p_0(u_1) & p_0(u_j) & p_0(u_{N_a}) & | & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & | & \vdots & \cdots & \vdots \\ p_{M_a}(u_1) & p_{M_a}(u_j) & p_{M_a}(u_{N_a}) & | & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_i \\ b_{N_a} \\ - \\ a_0 \\ \vdots \\ a_{M_a} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{1j} & P_{1N_v} \\ P_{i1} & P_{ij} & P_{iN_v} \\ P_{N_a1} & P_{N_aj} & P_{N_aN_v} \\ - & - & - \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix} \tag{6.29}$$

or in a compact form,

$$[K_A] \cdot [b] = [P] \tag{6.30}$$

where $[b] = \{b_1 \ldots b_i \ldots b_{Nu}\ a_0 \ldots\ aMu\}^T$. Solving for $[b]$ yields:

$$[b] = [K_A]^{-1} \cdot [P] \tag{6.31}$$

and step 2 yields:

$$P_{uv_j} = [\cdots\ K_a(|u-u_i|)\ \cdots\ p_0(u)\ \cdots\ p_{M_u}(u)]\,[K_A]^{-1} \begin{bmatrix} \vdots \\ P_{ij} \\ \vdots \\ - \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad 1 \le j \le N_v \tag{6.32}$$

The notation utilized on the R.H.S of equation (6.29) stands for the data points of the N_u sections considered along the v direction. In fact, coefficients a_i, b_j are vector coefficients, where each set of coefficients is obtained for a different section.

Finally, step 3 is equivalent to solving:

$$\left[\begin{array}{ccc|ccc} & \vdots & & q_0(v_1) & \cdots & q_{M_v}(v_1) \\ \cdots & K_b(|v_k-v_l|) & \cdots & q_0(v_l) & \cdots & q_{M_v}(v_k) \\ & \vdots & & q_0(v_{N_v}) & \cdots & q_{M_v}(v_{N_v}) \\ \hline q_0(v_1) & q_0(v_l) & q_0(v_{N_v}) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ q_{M_v}(v_1) & q_{M_v}(v_l) & q_{M_v}(v_{N_v}) & 0 & \cdots & 0 \end{array}\right] \begin{bmatrix} c_1 \\ c_j \\ c_{N_v} \\ - \\ d_0 \\ \vdots \\ d_1 \end{bmatrix} = \begin{bmatrix} P_{uv_1} \\ P_{uv_j} \\ P_{uv_{N_v}} \\ - \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.33}$$

then calculating

$$[P_u(v)]^T = [\cdots\ K_b(|v-v_k|)\ \cdots\ q_0(v)\ \cdots\ q_{M_v}(v)]\,[K_B]^{-1} \begin{bmatrix} \vdots \\ P_{uv_k} \\ \vdots \\ - \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.34}$$

Replacing P_{uv} by their value from equation (6.32), we obtain:

$$P(u,v)=\begin{bmatrix} \vdots \\ K_b(|u-u_i|) \\ \vdots \\ p_0(u) \\ \vdots \\ p_{M_u}(u) \end{bmatrix}^T [K_A]^{-1} \left[\begin{array}{ccc|ccc} & & & 0 & \cdots & 0 \\ & P_{ij} & & \vdots & \cdots & \vdots \\ & & & 0 & \cdots & 0 \\ \hline & & & - & - & - \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] [K_B]^{-1} \begin{bmatrix} \vdots \\ K_b(|v-v_k|) \\ \vdots \\ - \\ q_0(v) \\ \vdots \\ q_{M_v}(v) \end{bmatrix} \tag{6.35}$$

The above equation yields the parametric representation of a complex surface. The drift and covariance of each Kriging profile may be changed. This will affect the shape of the Kriged surface. For example, when $K(h) = h$, a piecewise linear surface is obtained. It can be shown also that Kriging with $K(h) = h^3$ and a linear drift is equivalent to bicubic spline interpolation. Note that a trigonometric drift is particularly adapted to surfaces of revolution.

Equation (6.35) may be written in a compact form as:

$$P(u,v) = [k_a(u)]^T\ [M]\ [k_b(v)] \tag{6.36}$$

where

$$[M] = [K_A]^{-1} \left[\begin{array}{ccc|ccc} & & & 0 & \cdots & 0 \\ & P_{ij} & & \vdots & \cdots & \vdots \\ & & & 0 & \cdots & 0 \\ \hline & & & - & - & - \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] [K_B]^{-1} \tag{6.37}$$

and

$$\begin{aligned} [k_a(u)] &= [\cdots\ k_a(|u-u_j|)\ \cdots\ p_0(u)\ \cdots\ p_{M_u}(u)]^T \\ [k_b(v)] &= [\cdots\ k_b(|v-v_j|)\ \cdots\ q_0(v)\ \cdots\ q_{M_v}(v)]^T \end{aligned} \tag{6.38}$$

For a given surface, $[M]$ is calculated once, and equation (6.36) is used to calculate the surface points.

6.3.2 Interpolating a Set of Curves

A similar approach may be used to generate a surface from a set of parametric curves $P_i(u)$ (see Figure 6.4). In this case, equation (6.34) is used and P_{uvk} is replaced with $P_i(u)$. The shape of the surface is controlled by the curves along u direction and the Kriging profile along v direction. The Kriging profile may have the general form of equation (6.16).

Ruled surfaces are a particular case of this procedure. They are obtained using a Kriging profile with a constant drift and a linear generalized covariance.

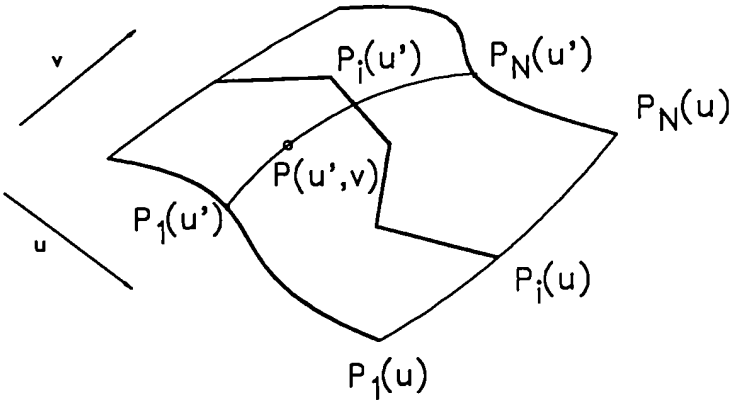


Figure 6.4 Continuous Surface Interpolation

This type of interpolation may be used for skinning operations as well as the interpolation of blending surfaces. Moreover, using traditional NURBS for skinning operations may destroy the continuity of the surface. This is not the case for skinning using Kriging if the curves to be skinned are continuous. On the other hand, skinning using dual Kriging is invariant under affine transformations. This is not generally the case for skinning with NURBS (Piegl, 1991).

It is also possible to interpolate a surface from a set of points where the number of data points is not necessarily equal along each isoparametric curve.

Examples of surface interpolation from a grid of data points and from a set of curves are provided in Appendix C.

6.4 PARAMETRIC SOLIDS

6.4.1 Interpolating a Grid of Points

A solid is described by a parametric equation of the form $P = P(u,v,w)$, with the component functions $x = x(u,v,w)$, $y = y(u,v,w)$, and $z = z(u,v,w)$. A parametric solid is defined by three Kriging profiles, A, B and C along the u, v and w directions respectively. A Kriging profile consists of a drift and a generalized covariance that govern the shape of the surface. A grid of N_u by N_v by N_w data points defines the solid. The parametric equations of a curve in the u (respectively v and w) direction for a Kriging profile with a drift linear combination of $p_k(u)$ (respectively $q_m(v)$ and $r_n(w)$) and a generalized covariance $K_a(h)$ (respectively $K_b(h)$ and $K_c(h)$) are utilized. The additional profile C is defined by the following equation:

$$P_{u_n v_m}(w) = \sum_{k=0}^{M_w} e_k \, r_k(w) + \sum_{l=1}^{N_w} f_l K_c(|w-w_l|) \qquad (6.39)$$

u, v and w are determined in a similar manner as for curves and surfaces, using equation (6.2) along each profile.

It is possible to determine the coordinates of any point belonging to the generating profiles, that is points $P(u_i,v,w)$, $P(u,v_j,w)$ and $P(u,v,w_k)$. However, in the general case where $P=P(u,v,w)$, an equation involving the three profiles has to be defined. This can be done in a manner similar to that of a surface. The steps illustrated in Figure 6.2 are:

1. Calculate the N_w matrices $[M_k]$ by using equation (6.37) for each section w = *const.*

2. Supposing u = *const.* and v = *const.*, determine N_w intermediate points P_{uvk} = $P_{uv}(w)$, $1 \le k \le N_w$.

3. Use previous intermediate points P_{uvk} to create the isoparametric curve u = *const.*, v = *const.*, and determine the corresponding Kriging coefficients. $P_{uv}(w)$ is then calculated using equation (6.41).

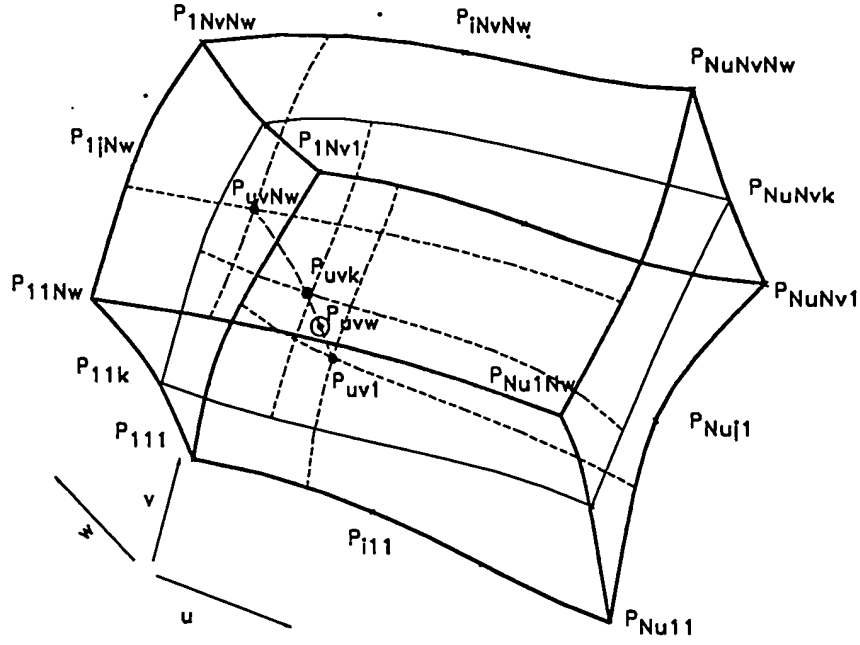


Figure 6.5 Solid Interpolation Using a Discrete Set of Points

Using matrix notation, the previous steps may be written as follows:

Step 1 is equivalent to calculating

$$[M_k] = [K_A]^{-1} \begin{bmatrix} & | & 0 & \cdots & 0 \\ P_{ijk} & | & \vdots & \cdots & \vdots \\ & | & 0 & \cdots & 0 \\ \text{-------} & | & - & - & - \\ 0 \ldots 0 & | & 0 & \cdots & 0 \\ \vdots \ldots \vdots & | & \vdots & \cdots & \vdots \\ 0 \ldots 0 & | & 0 & \cdots & 0 \end{bmatrix} [K_B]^{-1}, \ 1 \le k \le 1N_w \tag{6.40}$$

and step 2 yields:

$$P_{uvk} = [k_a(u)]^T \ [M_k] \ [k_b(v)] \tag{6.41}$$

The final expression of $P(u,v,w)$ is:

$$P(u,v,w) = [P_{uv}(w)]^T = [\cdots K_c(|w-w_k|) \cdots \ r_0(w) \cdots r_{M_w}(w)] [K_C]^{-1} \begin{bmatrix} \vdots \\ [k_a(u)]^T[M_k][k_b(v)] \\ \vdots \\ \text{----------} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.42}$$

The above equation yields the parametric representation of a complex solid formed by the combination of the three profiles A, B and C.

6.4.2 Interpolating a Set of Surfaces

The same procedure presented in section 3.2. is extended to the case of solid interpolation using a set of parametric surfaces. A Kriging profile is used along direction w, and $[k_a(u)]^T[M_k][k_b(v)]$ is simply replaced with $P_k(u,v)$ in equation (6.42).

6.4.3 Interpolating n-D Entities

In the case of n-D entities, the interpolation procedure is recurrent and is based on the following recurrent relation:

$$P(u_1,\cdots,u_n) = P_n(u_1,\cdots,u_n) = [k_n(u_n)]^T[K_n]^{-1}\begin{bmatrix} \vdots \\ P_{n-1}(u_1,\cdots,u_{n-1}) \\ \vdots \\ -- \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \tag{6.43}$$

Examples of applications of this generalization to n-D entities are:

1. Computer graphics and animation, the fourth parameter is in this case time and $P_{n-1}^{\ t}$ represents the solid/surface or curve at time t_i
2. Sweeping a solid along a curve or surface, the fourth (eventually fifth) parameter represent(s) the trajectory of the solid.

Detailed examples of solid interpolation are provided in Appendix C.

6.5 UNCERTAINTIES IN DATA POINTS

The coordinates generated by a CMM generally contain errors of two different types: 1) random errors, and 2) deterministic errors (see Chapter 2, section 1.2.1). The second type of errors can be evaluated and controlled. The first type however is not controllable and may be evaluated only if its probability distribution is known. With these uncertainties in the data points, an exact interpolation does not represent correctly the actual curves and surfaces. A model taking into account these errors is preferable. This can be achieved by interpolating with uncertainties in the data points.

First, the case of parametric curves will be considered, then this approach will be extended to surfaces.

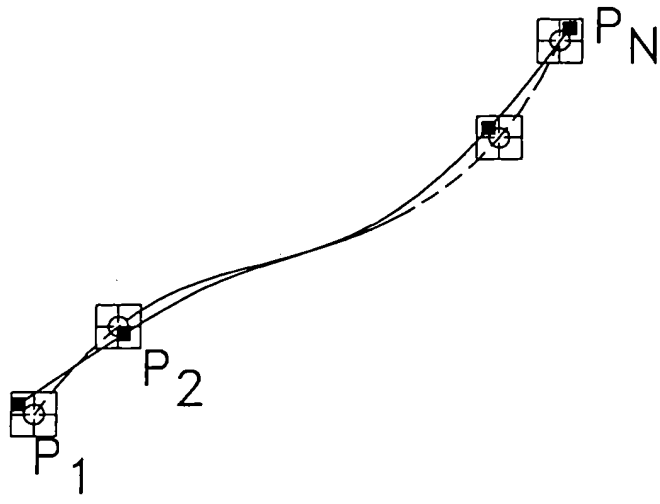


Figure 6.6 Interpolation with Uncertainties

We assume that the deviation e_k at each data point P_k is bounded by Δ_k (i.e. $|e_k| \leq |\Delta_k|$) (see Figure 6.6). If uncertainties are considered in the Kriging formulation, the interpolation model will no longer fit the data points, and equation (6.17) will be written as (with positions data only):

$$P_k - e_k = P(t_k) = \sum_{i=0}^{M} a_i p_i(t_k) + \sum_{j=1}^{N} b_j K(|t_k - t_j|) \qquad (6.44)$$

with the no-bias equations given in equation (6.4). The local deviation, $e_k = b_k d_k$, at each data point is assumed to affect only the diagonal terms of the Kriging submatrix, not the drift, because uncertainties in data points are assumed to be uncorrelated. The following linear system is obtained:

$$\begin{bmatrix} K(0)+d_1 & K(|t_1-t_l|) & \cdots & | & p_0(t_1) & \cdots & p_M(t_1) \\ \cdots & K(0)+d_k & \vdots & | & p_0(t_k) & \cdots & p_M(t_k) \\ & \vdots & K(0)+d_N & | & p_0(t_N) & \cdots & p_M(t_N) \\ \text{----} & \text{-------} & \text{----} & - & \text{-----} & -- & \text{-----} \\ p_0(t_1) & p_0(t_l) & p_0(t_N) & | & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & | & \vdots & \ddots & \vdots \\ p_M(t_1) & p_M(t_l) & p_M(t_N) & | & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_k \\ b_N \\ - \\ a_0 \\ \vdots \\ a_1 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_k \\ P_N \\ - \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.45}$$

or in a matrix format

$$\begin{bmatrix} C+dI & D \\ D^T & 0 \end{bmatrix} \begin{bmatrix} B \\ A \end{bmatrix} = \begin{bmatrix} P \\ 0 \end{bmatrix} \tag{6.46}$$

$[I]$ is the $N \times N$ identity matrix

However, equation (6.45) contains N additional unknowns, d_k. Since the number of equations is less than the number of variables, an iterative procedure is required to determine the d_k.

Statistically, the d_k represent the variance of the measurement errors on data points, it is called *Nugget-effect* in classical Kriging, i.e., $d_k = \sigma^2 = var[\varepsilon(t_k)] = E[\varepsilon(t_k)]^2$, and $|\varepsilon(t_k)| \leq |\Delta|$. Hence, it is possible to determine Δ if the statistical distribution of the errors is known. It can be shown that for a uniform distribution of the error between Δ and $-\Delta$, we have $\sigma^2 = \Delta^2/3$.

The proposed iterative procedure is as follows:

Initialization

$d_j^0 = 0$

$b_j(0)$ are determined by solving (6.45) without uncertainties.

$d_j^1 = \Delta / b_j(0)$ using these values in system (6.45) yields the deviation

$e_j^1 = b_j(d_j^1)\, d_j^1$

Iterations

Repeat until $|e_j^k| < |\Delta_j|$

$$d_j^k = \Delta_j / b_j(d_j^{k-1}) = \Delta_j / e_j^{k-1} d_j^{k-1}$$

$$e_j^k = b_j(d_j^k) d_j^k$$

solve system (6.45)

Different convergence criteria may be used. For constant $\Delta_j = \Delta$, $1 \leq j \leq N$, the criterion may be the maximum deviation for all data points, or their average absolute deviation. The convergence of this algorithm depends of the order or "degrees of freedom" of the Kriging profile. In effect, a linear profile (covariance and drift) is more difficult to deform than a cubic profile.

Furthermore, if $\Delta_j \rightarrow \infty$ it can be shown that the interpolation tends to the least squares solution. In effect, if $d_j \rightarrow \infty$, then $b_j \rightarrow 0$, and the interpolation is then reduced to the drift. The following is the proof that in this case the drift is the least squares approximation of the data points.

Writing equation (6.45) in a matrix format gives

$$\begin{cases} (C + dI)B + DA = P \\ D^T B = 0 \end{cases} \tag{6.47}$$

However, $d \rightarrow \infty$ implies that $B \rightarrow 0$, hence $D A = P$; that is,

$$A = (D^T D)^{-1} D^T P \tag{6.48}$$

which is the solution of the least squares approximation of data points P using basis functions $p_i(t)$ with coefficients A. $(D^T D)^{-1}$ exits because the $p_i(t)$ forms a basis of the subspace of functions. Another way to derive the least squares interpolation from the Kriging interpolation is by choosing a covariance such that $K(h) = 0$, for $h \neq 0$, and $K(0) = 1$.

In the case of surfaces, there are two independent profiles with two independent parameters. It is possible to consider the deviation for only one of the profiles, or divide it between the two profiles equally.

The first approach consists of taking each initial isoparametric of one of the profiles (i.e. composed of the original data points) independently and applying the same procedure as for curves. The initial data points are shifted to the deviated data points, which are then used to define the other profile. This is equivalent to step 1 in section 3, with $[b]$ determined using the iterative procedure. In a matrix notation, this is equivalent to calculating

$$S(u,v) = [k_u]\ [B_u(\Delta)][K_v]^{-1}[k_v]^T \tag{6.49}$$

The second approach involves dividing the maximum possible deviation Δ between the two profiles. This is similar to the first case but with half of the maximum deviation assigned for each profile. $[P]$ is decomposed into $[P] = [P][X][P]$, $[X]$ being the pseudo-inverse of $[P]$, that is, $[X] = [P^T]([P][P^T])^{-1}$. The deviated surface is then calculated using the following expression:

$$S(u,v) = [k_u][B_u(\frac{\Delta}{2})][X][B_v(\frac{\Delta}{2})][k_v]^T \tag{6.50}$$

Note that in the first approach iterations are performed only on one of the profiles, however in the second approach the iterations are performed on both profiles.

It is interesting to notice that the classical nugget-effect technique is limited to the implicit interpolant case, that is, surfaces of the form $z = f(x,y)$. In addition, only uncertainties in z are obtained easily using this technique, and in the case of uncertainties in x and y, an exact solution can only be found for the stationary case (constant drift), i.e. when $z(x, y) = A(x, y) + B$, with $E[A(x, y)] = 0$ and $B = const.$ An approximate solution may be found in the case of a linear drift (see Chiles, (1973 and 1977) for more details). These limitations, as well as the limited use of implicit representation of curves and surfaces in computer-aided design (Blechschmidt and Nagasuru, 1990) led us to adapting the nugget-effect technique for parametric representation. In addition, when closed curves, surfaces or solids are interpolated with uncertainties in positions, the model obtained may be discontinuous in position and/or derivatives at its joining points. This is solved by introducing two linear constraints: equality of position and derivatives in extremity points.

6.6 INTERPOLATING WITH LINEAR CONSTRAINTS

Introducing uncertainties in the data points may cause certain types of discontinuities in the case of closed curves and surfaces. This problem is overcome by introducing continuity constraints in the interpolation model.

The linear constraints considered are of the following form:

$$\mu P^{(k)}(t_i) + \nu P^{(l)}(t_j) + \delta = 0 \tag{6.51}$$

An example incorporating linear constraints in the Kriging interpolation with uncertainties is presented here. These constraints are in this case C^0 continuity and C^1 continuity at the extremity points of a closed curve or surface. The same procedure may be applied for higher order continuity or for other cases of linear constraints.

To incorporate constraints, the primal Kriging formulation must be used. The objective is to find $\lambda_1, \ldots \lambda_N, \alpha, \beta$ verifying (α, β being the Lagrange multipliers associated with the constraints) which verify that:

$$P(t) = \sum_{i=1}^{N} \lambda_i (P_i - e_i) + \alpha (P(0) - P(1)) + \beta \ (P'(0) - P'(1)) \tag{6.52}$$

where e_i is the uncertainties (or error) at point P_i. The continuity constraints are not limited to $t = 0$ and $t = 1$.

The objective is to minimize the squared variance of the estimation, that is:

$$e(\lambda_1, \cdots, \lambda_N, \alpha, \beta) = \frac{1}{2} E[P(t) - \sum_{i=1}^{N} \lambda_i (P_i - e_i) + \alpha (P(0) - P(1)) + \beta (P'(0) - P'(1))]^2 \tag{6.53}$$

Assume that the errors e_j are not correlated; Therefore,

$$\begin{aligned} &E[e_i e_j] = 0,\ i \neq j \\ &E[e_i P_j] = 0 \\ &E[e_i P(t)] = 0 \\ &E[e_i \dot{P}(t)] = 0 \end{aligned} \tag{6.54}$$

The intrinsic hypothesis states that:

$$E[P(t)P(s)] = K(|s-t|) \tag{6.55}$$

The following relations are then derived:

$$\begin{aligned} &E[P(t)]^2 = K(0) \\ &E[P_i P_j] = K(|t_i - t_j|) = K_{ij} \\ &E[(P_i - e_i)(P_j - e_j)] = K_{ij} + E[e_i e_j] \\ &E[P(0) - P(1)]^2 = 2(K(0) - K(1)) \\ &E[P'(0) - P'(1)]^2 = 2(K''(1) - K''(0)) \\ &E[P(t)P_i] = K(|t - t_i|) = K_i \\ &E[P(t)(P(0) - P(1))] = -K(t) - K(1-t) \\ &E[P(t)(P'(0) - P'(1))] = -K'(t) - K'(1-t) \\ &E[(P(0) - P(1))(P'(0) - P'(1))] = 0 \\ &E[P'(t)]^2 = \frac{d^2 E[P(t)P(t)]}{d^2 t} = -K''(t) \end{aligned} \tag{6.56}$$

Taking into account previous simplifications, the minimization of equation (6.53) is solved by deriving it with regard to $\lambda_1, \ldots \lambda_N$, α and β, that is, solving

$$\frac{\partial e}{\partial \lambda_i} = 0,\ \frac{\partial e}{\partial \alpha} = 0,\ \frac{\partial e}{\partial \beta} = 0 \tag{6.57}$$

The no-bias equations for a basis of functions $p_l(t)$, $0 \leq l \leq M$ are:

$$\sum_{i=1}^{M} \lambda_i p_l(t_i) + \alpha(p_l(0) - p_l(1)) + \beta(p_l'(0) - p_l'(1)) = p_l(t) \tag{6.58}$$

we obtain the following linear system

$$\begin{bmatrix} K & | & A & | & B \\ -- & -- & -- & -- & -- \\ A^T & | & 0 & | & C \\ -- & -- & -- & -- & -- \\ B^T & | & C^T & | & D \end{bmatrix} \cdot \begin{bmatrix} \vdots \\ \lambda_i \\ \vdots \\ \mu_0 \\ \vdots \\ \mu_M \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \vdots \\ K_i \\ \vdots \\ p_0(t) \\ \vdots \\ p_M(t) \\ K(t)-K(1-t) \\ -K'(t)-K'(1-t) \end{bmatrix} \tag{6.59}$$

where

$$K = \begin{bmatrix} \cdots & \vdots & \cdots \\ \vdots & K_a(|t_i-t_j|)+\sigma_{ij}^2 & \vdots \\ \cdots & \vdots & \cdots \end{bmatrix} \quad A = \begin{bmatrix} p_0(t_1) & \cdots & p_M(t_1) \\ \vdots & \cdots & \vdots \\ p_0(t_N) & \cdots & p_M(t_N) \end{bmatrix}$$

$$B = \begin{bmatrix} \vdots & \vdots \\ K(t_i)-K(1-t_i) & -K'(t_i)-K'(1-t_i) \\ \vdots & \vdots \end{bmatrix} \tag{6.60}$$

and

$$C = \begin{bmatrix} p_0(0)-p_0(1) & p_0'(0)-p_0'(1) \\ \vdots & \vdots \\ p_M(0)-p_M(1) & p_M'(0)-p_M'(1) \end{bmatrix} \quad D = \begin{bmatrix} 2(K(0)-K(1)) & 0 \\ 0 & -2(K''(0)-K''(1)) \end{bmatrix} \tag{6.61}$$

Therefore, the dual formulation is given by

$$P(t) = \sum_{i=0}^{M} a_i p_i(t) + \sum_{j=1}^{N} b_j K(|t-t_j|) + c(K(t)-K(1-t)) + d(K'(t)+K'(1-t)) \tag{6.62}$$

a_i, b_j, c and d are determined by solving the following system.

$$\begin{bmatrix} K & | & A & | & B \\ -- & - & -- & - & -- \\ A^T & | & 0 & | & C \\ -- & - & -- & - & -- \\ B^T & | & C^T & | & D \end{bmatrix} \begin{bmatrix} \vdots \\ b_l \\ \vdots \\ a_0 \\ \vdots \\ a_M \\ c \\ d \end{bmatrix} = \begin{bmatrix} \vdots \\ P_l \\ \vdots \\ 0 \\ \vdots \\ 0 \\ P(0)-P(1) \\ P'(0)-P'(1) \end{bmatrix} \tag{6.63}$$

Remark: Solving equation (6.57) does not necessarily lead to a minimum squared variance solution for any $K(h)$ function, i.e., matrix $[K]$ is not usually positive definite. Reference (Christakos, 1984) should be consulted for the conditions necessary to derive a non-negative definite matrix. Note that for parametric Kriging, $[K]$ has only to be invertible.

This procedure may be applied for any number of linear constraints. It is possible to impose higher order continuity constraints and obtain smoother closed surfaces.

In Figure 6.7 a cubic covariance combined with a constant drift are used. The dotted curve is obtained by exactly interpolating the data points shown by '+'. If a constant nugget effect (0.1 along x and y) is specified for each data point, the dash-dotted curve is obtained. The curve obtained in this case is not closed. If the tangents at end points are constrained to be equal, then the dashed curve is obtained. The curve is however still not closed. Finally, if in addition to previous considerations, position continuity is imposed, the continuous curve is obtained. The curve in this case is C^l. Note that the deviations of the data points from the curve were always smaller or equal to the maximum allowed deviation (0.1), this can be verified by the square domain surrounding each data point.

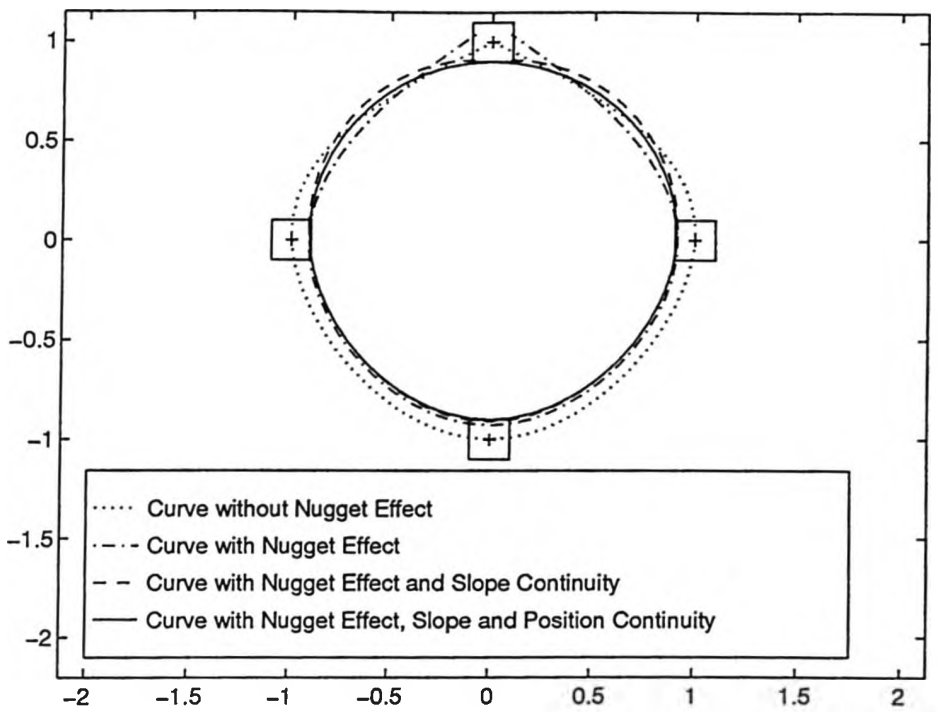


Figure 6.7 Interpolation with Linear Constraints

The previous steps may be applied to a surface. A linear drift and a cubic covariance is used for the circular profile and a constant drift and a linear covariance are utilized for the second profile of the surfaces shown in Figure 6.8. Figure 6.8a shows the surface obtained by interpolating exactly the data points. Figure 6.8b represents the surface obtained if a maximum deviation is allowed in each data point. Figure 6.8c shows the surface obtained if the tangents at end points are imposed to be equal. Finally, in Figure 6.8d the gap between the edges which appeared in the surface by introducing the nugget effect is eliminated by constraining the surface to be continuous at the end points.

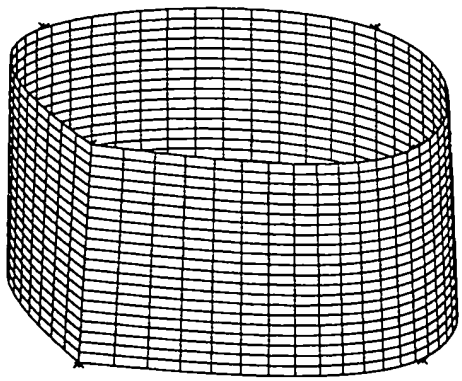


Figure 6.8a Classical Interpolation

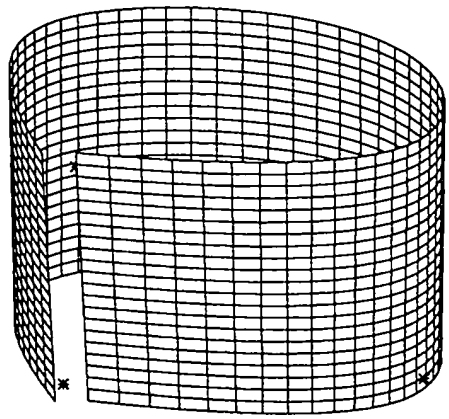


Figure 6.8b Interpolation with Nugget Effect

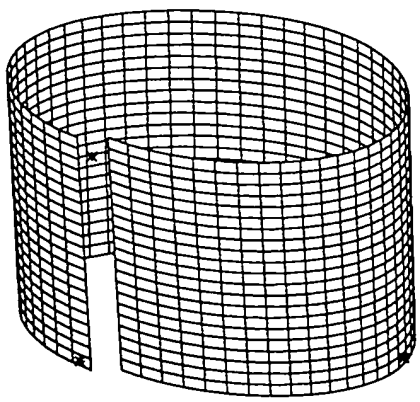


Figure 6.8c Nugget Effect and Tangents C ontinuity

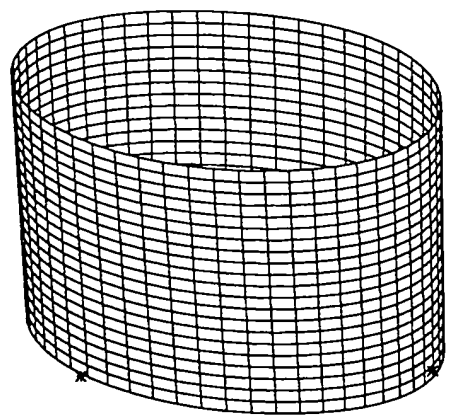


Figure 6.8d Nugget Effect, Tangents and Positions Continuity

6.7 DISCUSSION

A general formulation of parametric Kriging with positions and derivative data points has been presented. It has been shown that dual Kriging constitutes a general interpolation method incorporating some of the most commonly used interpolation techniques in a single formulation. A general geometric formulation for the generation of parametric surfaces and solids from discrete and continuous data as well as higher dimensionality entities has been defined. Dual Kriging is a new general interpolation technique which combines the advantages of the most popular interpolation techniques into a single formulation. It has been shown that it is possible to take into account, in dual Kriging interpolation, the possible errors in data points. Kriging with uncertainties in data points allows local control of the deviations in each data point as well as noise filtering. However, filtering methods such as the classical least squares allow only a global control of the deviations by minimizing the average deviation. The normal deviations of the surface model was verified to be bounded by the maximum allowable normal deviation. This feature is specific to Kriging interpolation, and has a statistical origin. It leads to bounded actual deviations and smoother curves and surfaces. It is also possible to consider certain types of linear constraints in the interpolation model. Interesting applications of this last feature are the fitting of smoother curves and surfaces by imposing higher order continuity for closed surfaces, and the construction of multiple continuous patches. The inclusion of possible deviations permits additional degrees of freedom for the curve which can be deformed to meet the imposed constraints.

Some of the advantages of dual Kriging are:

1. It presents a common mathematical form for representing and designing both standard analytical shapes (conics, quadrics, surfaces of revolution, etc.) and free form curves and surfaces.
2. Dual Kriging interpolation is invariant under scaling, rotation, translation and shearing, as well as parallel and perspective projection.

3. Fewer data points are needed to define traditional curves and surfaces using Kriging, as opposed to NURBS [70, 77]. For example, to represent a full circle, only three data points are required, hence, taking advantage of classical parametric representations, which are simpler and computationally more efficient compared with the seven points and ten knots needed for representing a circle using NURBS.
4. The interpolation passes through the data points, although controlled deviations are possible, and generally no control points, weights or knots are used.
5. Interpolating data with mixed types, such as positions, derivatives and linear constraints is possible.
6. It is possible to have mixed types of profiles. Example, surfaces of revolution are obtained by combining a general profile (such as NURBS) with a trigonometric profile.
7. It combines the interpolation of discrete data as well as continuous data (see sections 4.2 and 5.2).
8. It is possible to interpolate a grid of data points with a non-uniform number of points along each isoparametric curve.

CHAPTER 7

INTERSECTIONS IN GEOMETRIC MODELING

Once the measurement data has been interpolated, it is important to compare the obtained model and the original data points. Different tools are used for this purpose including: 1) orthogonal projection on curves and surfaces which are used for the determination of the normal deviation of a point from a curve or surface, and 2) intersection of curves and surfaces for the construction of a surface from a set of surface patches. The problem of finding the intersection of curves and surfaces arises in numerous computer-aided design applications and in the verification of specified geometric tolerances based on the comparison and manipulation of the interpolated model (based on measurements) which represents the actual surface and the CAD model representing the theoretical surface. The methods generally used rely on iterative numerical techniques based on the solution of a set of non-linear equations which are very sensitive to the type of interpolation used. These systems of equations are generally local and need adequate starting points in order to yield convergent solutions.

This chapter presents two general geometric algorithms to find the intersections of C^0 curves and surfaces. Section 7.1 presents an overview of related works in curve/surface manipulations. Section 7.2 explains the first intersection method which can be applied to two geometric objects defined respectively by their parametric and implicit equations. Section 7.3 describes the algorithm of orthogonal projection (OP), on which the general intersection method of section 7.4 is based. This method can be used with any kind of C^0 surfaces; if the surface is C^1, then it can be further refined into the so-called *conjugate tangent* approach to speed up the

algorithm. It is implemented successively for the intersection of two curves, of a curve with a surface and of two surfaces. Finally, section 6 presents the *conjugate tangent* approach derived from the *orthogonal projection* algorithm to accelerate convergence.

7.1 INTRODUCTION

This chapter presents two general methods for determining the intersection between geometric entities such as a line, curve, plane or complex surface. The first method can be applied when one object is defined by a parametric equation and the other by an implicit equation. Although this approach is general, it is not well adapted to cases of tangent intersections. Therefore, a more general method was devised, based on successive orthogonal projections between the two geometric objects. This approach is general since the same algorithm applies for any kind of geometric intersection: curve-curve, curve-surface and surface-surface. A line and a plane are simply particular cases of a curve or a surface, therefore they can be treated by the same algorithm, which is not based on the manipulation of mathematical equations but rather on geometric principles.

A large number of methods have been proposed in the literature for determining the intersection set between two geometric objects. Some methods are general, in the sense that they can be applied to any type of surface. The approach of Marciniak (1990) is based on solving a set of simultaneous nonlinear equations by Newton's method and on minimizing the distance between the two objects. The same approach can be found in Faux and Pratt (1980), who apply the algorithms to generate cutter paths for numerically controlled machines. A general method for parametric piecewise surfaces intersection was described by Asteasu and Orbegozo (1991), requiring only C^1 surface continuity of the surface. An initial set of points is calculated by a method of enclosing boxes. From these isolated points, spans of the real intersection are calculated by an incremental method based on differentiating the parametric equations of the two surfaces.

Another way of finding initial points is proposed by Mullenheim (1991) for a surface-surface intersection algorithm.

Since a fairly large number of CAD systems represent free form surfaces by a set of parametric patches, intersection algorithms for parametric patches play an important role. This problem has been treated by various authors and a survey on this topic is reported in Pratt and Geisow (1986). The algorithm performs an adaptive and recursive decomposition of surfaces which is a function of their morphology, then a selection process filters out all irrelevant surface pairs. This method of surface intersection, which is independent of surface derivatives, is based on a triangular decomposition of relevant surface patches. This approach, although it can be applied to any kind of free form surface, presents two technical difficulties: (1) the accuracy of the triangular decomposition depends on the smoothness of the surface and on the size of the triangles and (2) the intersections of the two sets of triangles must be sorted and reorganized to generate curves.

A certain number of methods were devised for the intersection of particular types of surfaces. For example, the method of Aziz and Bata (1990) is valid for Bezier surface intersections. It is based on a recursive subdivision technique of their convex hulls. Hedrick and Bedi (1990) have proposed a method for two biquadratic or bicubic surfaces. The method is robust, but limited to parametric surfaces defined by quadratic or cubic polynomials. Note that a special method for the intersection of a plane and a natural quadric was proposed by Johnstone and Shene (1992). An original approach adopted by Klass and Kuhn (1992) consists of finding the intersection between two spline surfaces by using rolling balls of decreasing radius simultaneously tangent to the two surfaces. The intersection is obtained when the radius of the rolling ball becomes zero.

7.2 INTERSECTION OF IMPLICIT/PARAMETRIC OBJECTS

The section presents a general method to find the intersecting set between two geometric objects, where one is defined by its parametric equation and the other by its implicit equation. The

geometric set defined parametrically will be referred to as the *parametric object*, and the set defined by an implicit equation of the form $f(M) = 0$, where M designates the reference coordinates of a point in the geometric space, is the *implicit object*. The following parametric objects will be investigated:

(i) a **curve**, defined for example in a three-dimensional space by:

$$x = g_1(t), \qquad y = g_2(t), \qquad z = g_3(t), \qquad t \in [0,1]$$

(ii) a **surface**, defined by:

$$x = g_1(s,t), \qquad y = g_2(s,t), \qquad z = g_3(s,t), \qquad s,t \in [0,1]^2$$

(iii) a **solid**, defined by:

$$x = g_1(r,s,t), \quad y = g_2(r,s,t), \quad z = g_3(r,s,t), \qquad r,s,t \in [0,1]^3$$

These various equations can be summed up in the notation $M = G(p)$ where M denotes the coordinates of a current point in the geometric space and p is the set of parameters, i.e., t, (s,t) or (r,s,t) respectively for the curve, surface or solid.

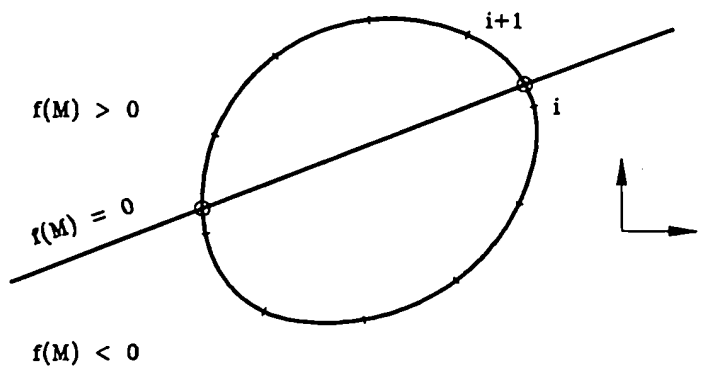


Figure 7.1 Intersection of Implicit and Parametric Entities

Two examples of application are given: (1) intersection of a line with a curve, and (2)

intersection of a plane with a surface. The proposed algorithm is based on the fact that an implicit object $f(M)$ divides the space into three regions: (1) a region where $f(M) < 0$; (2) a region where $f(M) = 0$, which defines the implicit object itself, and (3) a region where $f(M) > 0$. These three cases are illustrated in Figure 7.1. Note that the value of $f(P)$ may interpreted as the algebraic distance of point P from the implicit object $f(M) = 0$. The algorithm is based on a two-step procedure: (1) First, a set of isoparametric lines is drawn on the parametric object as shown in Figure 7.2 for the intersection of a plane with a cylinder. On each of these lines the intersection with the implicit object is located between two consecutive points for which $f(M)$ changes sign. (2) Secondly, an inverse interpolation is performed between the two points to find the exact position of the intersection. For this purpose, additional points are usually calculated between these two points, and the exact intersection can be found either by inverse kriging or by a dichotomy algorithm.

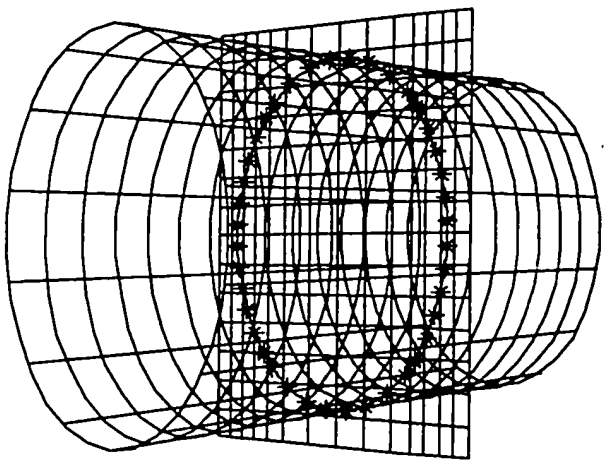


Figure 7.2 Intersection of a Plane with a Cylinder as an Application of Implicit and Parametric Entities Intersection Method

However, this methodology presents some limitations. Tangent intersecting points, as seen

in Figure 7.3 case b, cannot be obtained. In addition, if the density of points selected on the isoparametric curves is not high enough, in some cases the intersection will not be detected (see Figure 7.3 case c). Additional calculations involving derivatives would be needed to remedy these deficiencies. Although some problems could be overcome by increasing the number of subdivisions on the isoparametric curve, a more general methodology based on orthogonal projections and tangents will be presented in sections 4 and 5 respectively. The first method will now be illustrated by two applications: (1) intersection of a line with a parametric curve; and (2) intersection of a plane with a parametric surface.

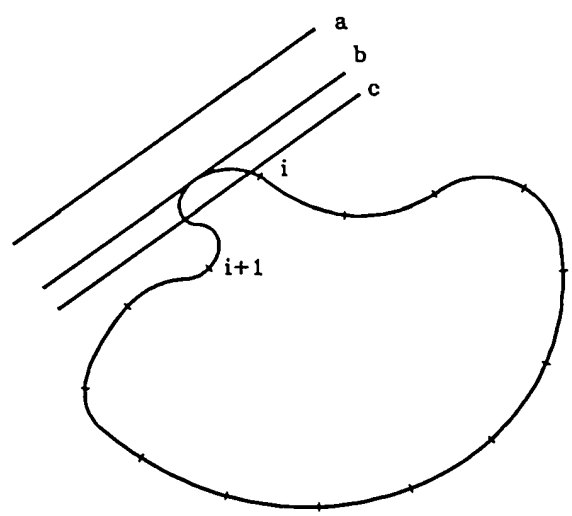


Figure 7.3 Cases where the Intersection is not Detected by the Implicit and Parametric Entities Intersection Method

7.2.1 Curve-Line Intersection

The simplest application of the above methodology consist of finding the intersection of a line with a planar curve defined by its parametric equation $x(t)$, $y(t)$ for $0 \leq t \leq 1$, (see Figure 7.1). If the implicit equation of the line is

$$ax + by + c = 0 \tag{7.1}$$

then the intersection with the curve is defined by the set of parameters t such that

$$ax(t) + by(t) + c = 0 \tag{7.2}$$

A two-step algorithm has been set up to solve equation (7.2):

Step 1

A sequence of $k+1$ points on the curve is calculated for a constant parameter step $\Delta t = 1/k$, say $x(i\Delta t)$, $y(i\Delta t)$ for $i = 0, 1, \ldots, k$. Each portion of the curve between two consecutive points i and $i+1$ is called an *arc*. After evaluating the algebraic distance from the line of each of the previous points, i.e.:

$$d_i = ax(i\Delta t) + by(i\Delta t) + c \tag{7.3}$$

it is possible to locate the intersecting arcs on the curve, if any intersection exists. Such arcs simply correspond to $d_i d_{i+1} \leq 0$.

Step 2

Once an intersecting arc is located on the curve, the exact position of the intersection may be calculated either by a dichotomy algorithm (i.e. bisection algorithm) (Burden, 1989) or by inverse kriging. The dichotomy algorithm consists in dividing the interval where the solution $d_k = 0$ is suspected to exist into two equal intervals then keeping the interval where a change in the sign of d_i is detected. The procedure is continued until a certain accuracy is reached. The parameters t are interpolated for a linear drift and cubic covariance as a function of $d(t) = ax(t) + by(t) + c$. From the set of data $(d(t_l),t_l)$ for $l = 1, \ldots, L$, a kriged interpolant $t = u(d)$ may be constructed, the parameter corresponding to the intersecting point being simply $t^* = u(0)$.

7.2.2 Surface - Plane Intersection

A slightly more complicated application of the same methodology is illustrated by the intersection of a plane with a surface defined by its parametric equation $x(s,t)$, $y(s,t)$, $z(s,t)$ for $0 \leq s \leq 1$ and $0 \leq t \leq 1$.

If the equation of the plane is:

$$ax + by + cz + d = 0 \tag{7.4}$$

then the intersection with the surface is defined by the set of parameters s and t such that:

$$a\,x(s,t) + b\,y(s,t) + c\,z(s,t) + d = 0 \tag{7.5}$$

The two-step procedure is simply:

Step 1

Compute a sequence of $k + 1$ points on the isoparametric curves $s = s_j$ for $j = 0, 1..., m$ for a constant parameter step $\Delta t = 1/k$, say $x(s_j, i\Delta t)$, $y(s_j, i\Delta t)$, $z(s_j, i\Delta t)$ for $i = 0, 1,..., k$. By evaluating the corresponding values of expression (7.5), namely:

$$e_i = a\,x(s_j, i\,\Delta t) + b\,y(s_j, i\,\Delta t) + c\,z(s_j, i\,\Delta t) + d \tag{7.6}$$

it is possible to locate the intersecting arcs on the surface whenever they exist. Such arcs simply correspond to $e_i e_{i+1} \leq 0$.

Step 2

As in the previous case, an inverse kriging of parameter $t = u(e)$ as a function of $e(t) = a\,x(s_j, t) + b\,y(s_j, t) + c\,z(s_j, t) + d$ is performed, and the parameter corresponding to the intersection being simply $t^* = u(0)$. The same procedure can be applied by interverting the roles of t and s. This yields a set of intersections on the plane, with at least one point for each s or t isoparametric curve. Note that the intersecting points obtained by this procedure

are not necessarily ordered. This is the case especially when a closed surface is intersected or when the isoparametric curves in s and t both intersect the surface. An appropriate sorting algorithm would have to be used if the intersection points are not ordered or if the intersection is composed of more than one curve. Note that the procedure can be accelerated by using a dichotomy algorithm method to find the exact intersection rather than inverse kriging.

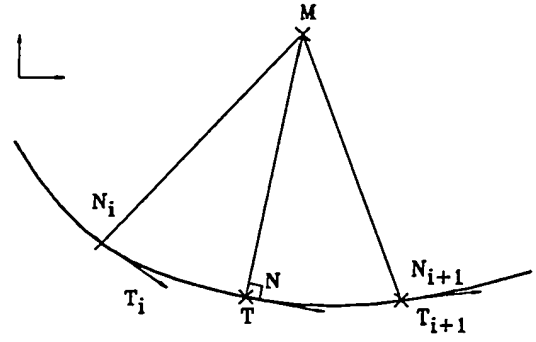


Figure 7.4 Determination of the Orthogonal Projection of a Point on a Parametric Curve

7.3 ORTHOGONAL PROJECTION ON A GEOMETRIC OBJECT

The orthogonal projection (OP) of a point on a parametric object is defined as the *nearest* point on the object from which a vector normal to the object will intersect the original point. A numerical procedure for finding the projection of a point on a parametric curve (planar or 3D) or a parametric surface will now be described.

7.3.1 Orthogonal Projection on a Parametric Curve

Let M be the point to be projected on a curve $C(t)$, while N_i denotes a current point on the

curve and T_i is the tangent vector at N_i. The OP N of M on the curve is by definition such that MN is orthogonal to T, the tangent unit vector at N, so $MN.T = 0$. The curve being continuous, it can be described by a set of discrete points N_i. By calculating the signs of the dot product $MN_i . T_i$, it is possible to locate the projection of M between two consecutive points on the curve. This simply corresponds to a change of sign between $MN_i . T_i$ and $MN_{i+1} . T_{i+1}$ as seen in Figure 7.5 for example. The next step consists of performing an interpolation by inverse kriging between the two points. Intermediate points are generated on the curve for a set of parameters t_i, their number depending on the desired accuracy. The curve parameters t_i are interpolated as a function of the scalar products $MN_i . T_i$ in order to construct a function $t = f(MN.T)$. The OP is obtained by calculating $t^* = f(0)$. If a point has more than one OP, then the nearest one will be chosen. In some cases, several points may be obtained that are all located at the same minimum distance from the original point. This is the case for example, for the projection of the centre of a circle on its perimeter. In this case, the algorithm does not necessarily converge. Figure 7.5 presents examples of OP on curves.

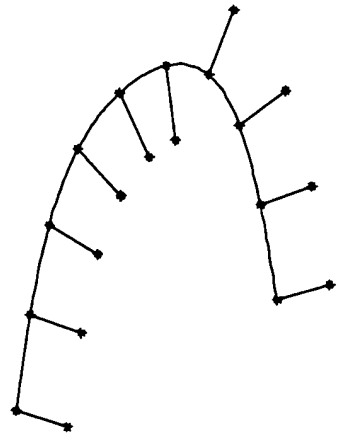


Figure 7.5 Examples of Orthogonal Projections on a Parametric Curve

7.3.2 Orthogonal Projection on a Parametric Surface

Let M be the point to be projected on a parametric surface $S(s,t)$ and denote by N_t a current point on the surface. In this case, two classes of isoparametric curves are considered on the surface, namely the curves obtained when one parameter only varies (t fixed, s variable; and conversely t variable, s fixed). Point M is first projected on an initial isoparametric curve, then on the isoparametric curve of the other class drawn from the last projected point (see Figure 7.6). The OP is obtained by projecting M alternatively on the two classes of isoparametric curves until two consecutive projections coincide with a given precision. The convergence of the algorithm can be demonstrated by a simple arguement.

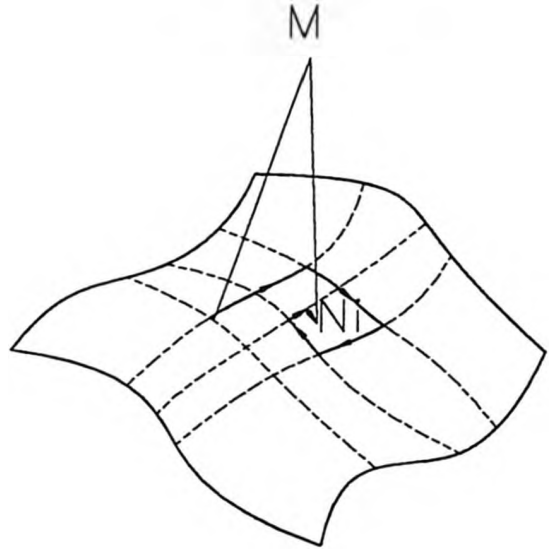


Figure 7.6 Convergence of the Proposed Projection Algorithm

Convergence of the Algorithm (see Figure 7.6)

Let $N_1(s_1,t_1)$ be the OP of M on the first isoparametric curve $s = s_1$. For any $N(s_1,t)$, we have $MN_1 \leq MN(s_1,t)$. Let $N_2(s_2,t_1)$ be the OP of M on the isoparametric curve $t = t_1$ passing

through N_1. For any $N(s,t_1)$, we have $MN_2 \leq MN(s,t_1)$. In particular, $MN_2 \leq MN(s_1,t_1) = MN_1$. By repeating the same process from N_2, we can construct a sequence of points N_1, N_2 , ..., N_i of decreasing MN_i distances. The minimum value will be attained for a prescribed precision e when $|MN_{i+1} - MN_i| \leq e$.

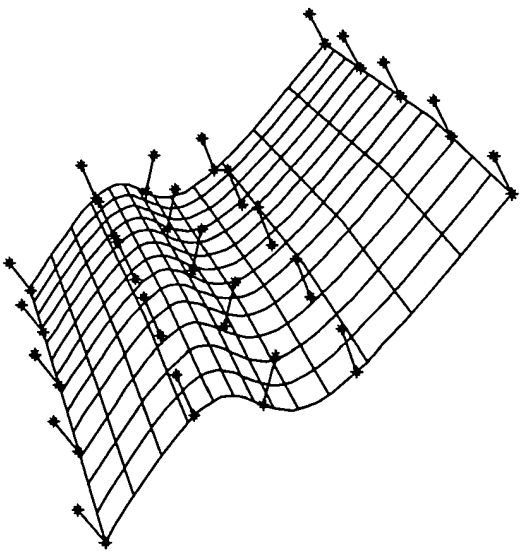


Figure 7.7 Examples of Orthogonal Projections on a Parametric Surface

The initial isoparametric curve could be chosen by considering the nearest one to the projected point. As in the case of orthogonal projections on a parametric curve, if multiple projections exist, then the algorithm should select the nearest one to the current point N_i. In order to obtain the minimum distance OP, the initial isoparametric curve must be located as close as possible to the current projected point N_i. This can be achieved by evaluating the distance between the point to be projected and a set of discrete points on the surface.

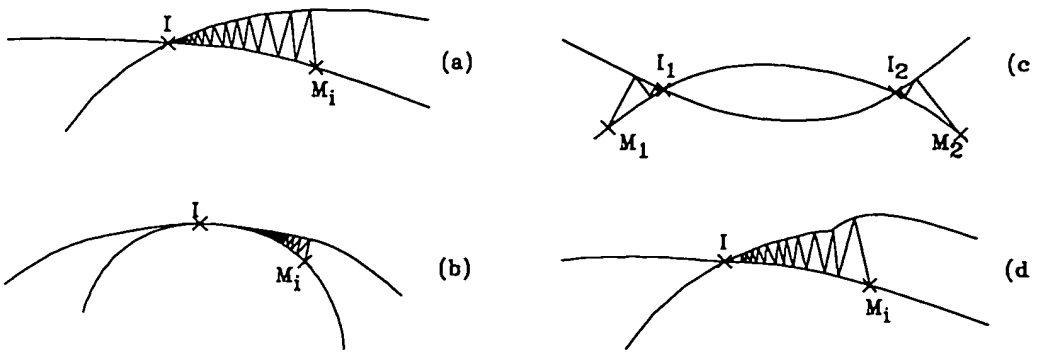


Figure 7.8 Different Cases of Intersection in the Orthogonal Projection Method

7.4 INTERSECTION BY THE METHOD OF ORTHOGONAL PROJECTIONS

The second intersection algorithm is based on the following property of orthogonal projections (OP): considering two parametric objects (curve or surface), successive orthogonal projections from an initial point located on one of the two objects either converge to the intersection (see Figure 7.8a), or yield the minimum distance between the two objects. A proof will be given for the intersection of two parametric geometric objects defined by their parametric equations. The method of orthogonal projection has been used previously for the iterative solution of linear systems (Lehning, 1986). For example, in the case of a system of three linear equations, where each equation represents a plane, an initial point can be projected orthogonally on the first plane, then the projection is re-projected on the second plane, and so on the third and on the first plane again, until the difference between two successive projections becomes smaller than a prescribed precision. Orthogonal projections on planes and lines can be obtained by a simple

analytic procedure, but in order to find orthogonal projections on complex parametric curves or surfaces, it is necessary to devise a special numerical procedure. Two versions of an algorithm for finding orthogonal projections on parametric curves and surfaces will be presented. The principle of this approach remains valid for implicit objects as well. However, in this case, the orthogonal projection is in general quite difficult to calculate.

The proposed methodology solves the intersection problem even if the two geometric objects are tangent (Figure 7.8b); if the two objects do not intersect, it yields the shortest distance between them (see Figures 7.13a and 7.13b). Besides, it is possible to obtain an ordered sequence of intersecting points. In addition, it can also be applied to more than two intersecting objects.

This section describes how the intersection between two geometric objects defined by their parametric equations can be constructed by successive orthogonal projections from one object to the other. The procedure will be illustrated by three examples of application: intersection of two curves, intersection of a curve with a surface and intersection of two surfaces.

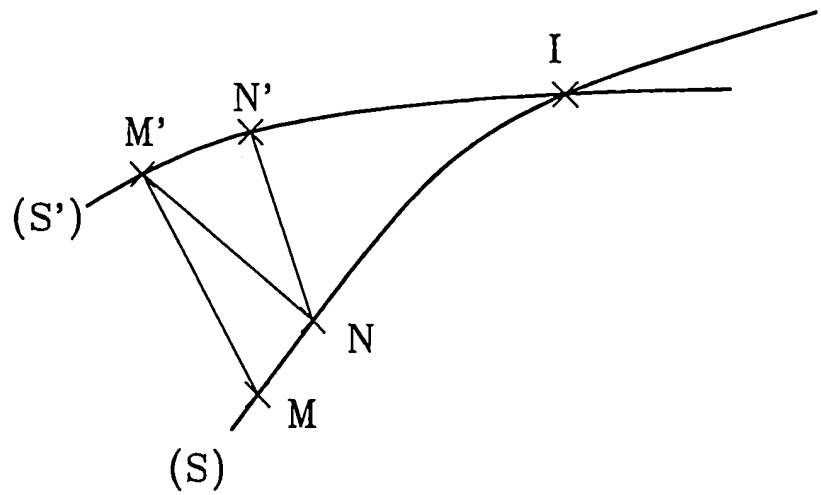


Figure 7.9 Intersecting Two Curves by Successive Orthogonal Projection

Hypothesis

Let S and S' denote the two geometric objects. It is assumed here that a single OP can be obtained on S' (resp. S) from any point of S (resp. S'). If more than one OP exist, only the portion of the object satisfying this hypothesis will be considered.

Proposition

Given the above hypothesis, successive orthogonal projections from an initial point located on one of two intersecting objects converge towards their intersection.

Proof

Let M be the initial point on object (S). From this initial point, the following OP are performed as illustrated in Figure 7.9:

(a) M' is the orthogonal projection of M on (S'), so $MM' \leq MN_i'$ for any N_i' on (S').

(b) N is the orthogonal projection of M' on (S), so $NM' \leq M_jM'$ for any M_j on (S).

(c) N' is the orthogonal projection of N on (S'), so $N'N \leq N_i'N$ for any N_i' of (S').

Inequality (b) leads to $NM' \leq MM'$, and (c) to $NN' \leq NM'$, which results finally in

$$NN' \leq MM'$$

Now we prove by contradiction that the above condition is in fact a strict inequality. Suppose that $MM' = NN'$, then $NM' \leq MM'$ and $MM' \leq NM'$; hence $MM' = NM' = NN'$. But $NM' = MM'$ means that M' has more than one OP of minimum distance on (S) (supposing that M is different from N), which contradicts the hypothesis. We can, therefore, conclude that MM' is not equal to NM', namely

$$NM' < MM' \text{ or } MN \leq k\, M'N \text{ with } 0 \leq k < 1$$

This latter inequality is precisely the condition of existence of a fixed point in the iterative procedure (Lehning, 1986). So the algorithm converges towards a fixed point, which belongs to the intersection of the two objects.

7.4.1 Intersection of Two Curves

In this case, the intersecting points are determined by successive OP from of a set of initial points, the number of which depends on the number of targeted intersections (For examples see Figures 7.8a, 7.8b, 7.8c).

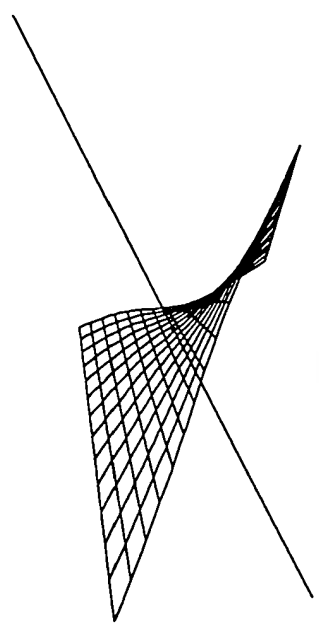


Figure 7.10 Examples of Intersection of a Parametric Surface with a Curve

7.4.2 Intersection of a Curve with a Surface

The intersection of a parametric curve and a parametric surface can be determined easily by successive OP from a certain number of regularly spaced initial points on the curve. Their

number depends on the targeted number of intersecting points. Figure 7.10 shows an example of a parametric surface intersecting with a curve.

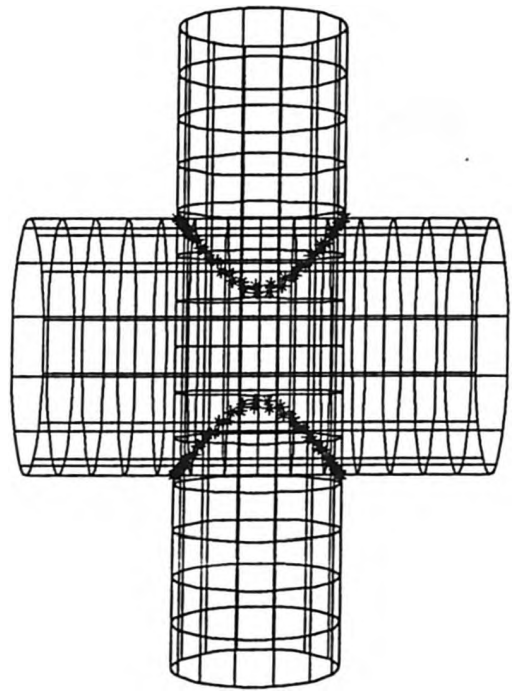


Figure 7.11 Intersection of Two Cylinders

7.4.3 Intersection of Two Surfaces

The intersection of two parametric surfaces can be treated by two different approaches: first, a surface-surface intersection by OP could be performed from an initial set of points on each surface; and secondly, multiple curve-surface intersections could be determined by considering the intersections of each isoparametric curve on one surface with another. The second approach was adopted in this thesis because it requires fewer OPs on surfaces and is less time-consuming than the first. In addition, OPs are easier to determine on curves than on surfaces. The method can be improved by considering the intersecting point of an isoparametric curve with the surface as the

initial point on the surface that will be projected on the immediately successive isoparametric curves. This speeds up the calculation, because initial points are chosen closer to the intersecting set. Examples of two surfaces intersecting are presented in Figures 7.11 and 7.12.

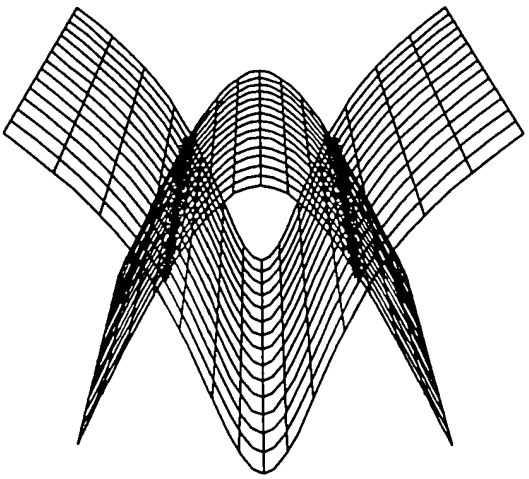


Figure 7.12 Intersection of Two Complex Surfaces

Remarks

1. This algorithm based on OP permits rigorous determination of cases where no intersection exists. The method converges towards the two nearest points on the two objects and yields the minimum distance between them (Figures 7.13a, 7.13b).

2. The algorithm remains effective even if the OP itself is not so precise, namely, if it is implemented with a loose accuracy factor.

3. The determination of the OP can be viewed as a minimization problem to find the minimum distance between the point to be projected and the object. Some other numerical techniques could also be used, such as least square methods.

4. The method of section 2 must be used when one of the intersecting objects is given by its implicit equation.

5. The speed of the algorithm can be improved considerably by using the *conjugate tangents* approach described in the next section.

6. Parametric curves and surfaces are generally defined with a fixed range of their parameters (i.e. $t_1 \le t \le t_2$, $s_1 \le s \le s_2$). In order for the successive projections to converge to the intersection(s), it may become necessary to extend the range of the parameters so that orthogonal projections are found.

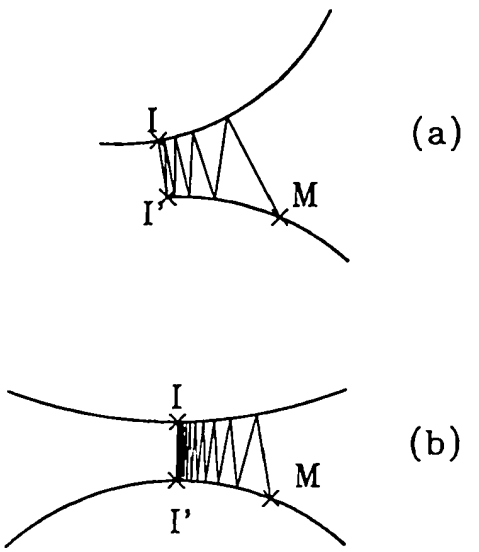


Figure 7.13 Two examples of Non-Intersecting Objects

7.5 CONJUGATE TANGENT APPROACH

Although the algorithm based on OP works satisfactorily, numerical experiments have shown that it can be accelerated by combining OP on each isoparametric curve of one surface with tangent planes on the other surface. Figure 7.14 illustrates the rapidness with which the alternate tangent and OP converge. This procedure is called the *conjugate tangent*, because iteration points are taken alternatively on tangent or normal vectors to each surface, in analogy with minimization

algorithms based on conjugate gradients (Avriel, 1976). The method will be described with the help of examples for finding the intersection of two planar curves, a curve and a surface or two surfaces. A pure tangent intersection approach could be utilized as shown in Figure 7.15. However, when the tangent does not intersect the curve or surface, the algorithm may be blocked as illustrated in Figure 7.16 where the tangent at point *M1* does not intersect the other curve. An OP is then needed to find another point, say *M2*, to ensure convergence of the algorithm as illustrated in Figure 7.16.

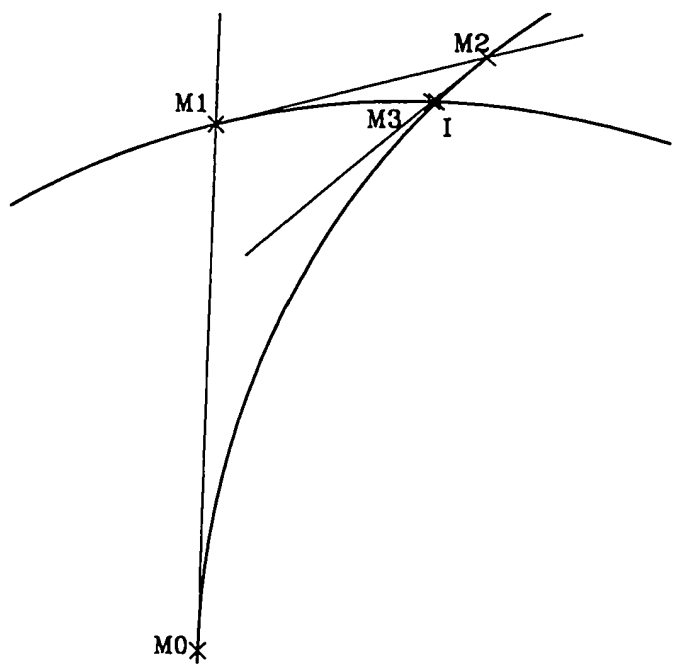


Figure 7.14 Intersecting Two Curves with the Pure Tangent Intersection Approach

7.4.1 Intersection of Two Planar Curves

The method converges rapidly when the intersections of the tangent vectors with the curves exist, as shown in Figure 7.14. When such intersections do not exist, as illustrated by the case of point *M1* in Figure 7.16, then we must come back to the OP: *M1* is projected on *M2*, and from *M2*, successive tangents yield *M3*, then *M4* located very close to an intersection. If the initial

point is far from the intersection, if no intersection exists or if there is more than one tangent intersection, successive OP must be used to get closer to a targeted intersection.

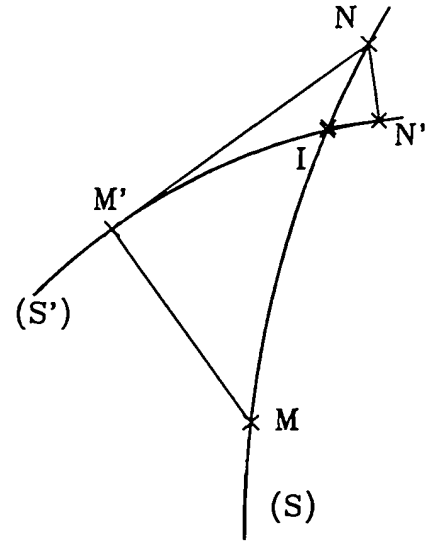


Figure 7.15 Intersecting Two Curves using the Conjugate Tangent Approach

7.4.2 Intersection of a Curve and a Surface

The principle of the method is identical to that for two planar curves, except that the tangent plane at a given point replaces the tangent. The tangent plane is intersected with the curve. The intersection is projected on the surface, and the curve is intersected again with the tangent plane at this point, and so on until two successive points lie within the same precision.

7.4.3 Intersection of Two Surfaces

The intersection of two parametric surfaces can be treated as a sequence of intersections of isoparametric curves with the surface.

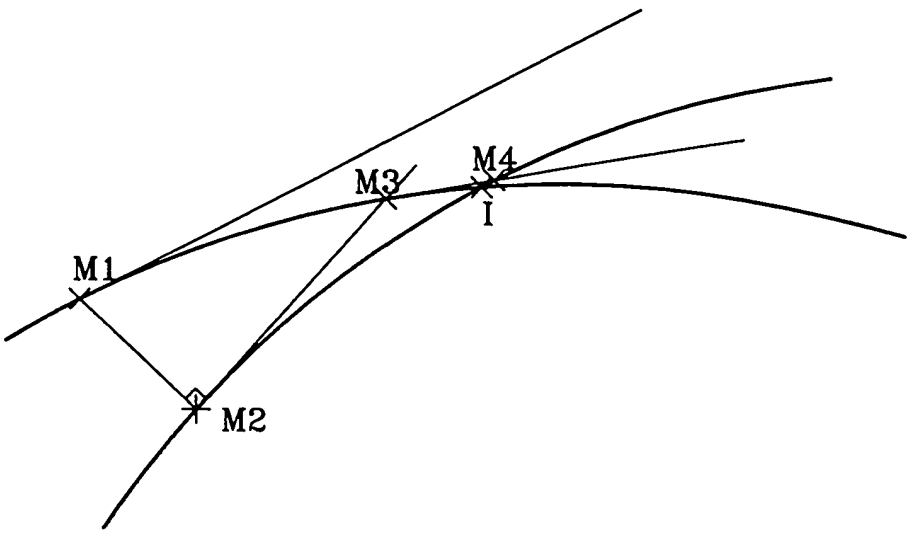


Figure 7.16 Using the Orthogonal Projection in the Case of a Non-Intersecting Tangent Vector

7.6 DISCUSSION

The two methods presented in this chapter for finding the intersection of geometric objects have been extensively used in various applications (Trochu, 1995 and Vafaeesefat, 1994), and were found to be reliable and robust.

The first method is interesting when one object is defined by its implicit equation and the other parametrically. Since this approach cannot handle the case of tangent objects, a second method is proposed based on successive *orthogonal projections* from one object to the other. The algorithm is general and can be applied to any type of geometric entities and intersection: curve-curve, curve-surface and surface-surface. Several application examples were given for arbitrary curves or surfaces. The algorithm is based on finding the orthogonal projection of a point on a geometric object. Convergence of the algorithm was demonstrated. The whole procedure could be

improved by considering enclosing boxes in order to speed up the determination of the nearest projected point on a curve or on a surface. Note also that for these intersection algorithms, being based on geometric considerations, a geometric interpolation of the iterative process can always be provided. Finally, the second method can be further improved in the so-called *conjugate tangents* approach by alternatively selecting iteration points on a tangent line or plane instead of using only orthogonal projections.

CHAPTER 8

THE COMPUTER-AIDED INSPECTION PLANNING (CAIP) SYSTEM

This chapter deals with the integration and implementation of the approaches and methodologies, proposed in this thesis, in a computer-aided inspection planning and verification system and its validation with application examples.

Section 8.1 presents the structure of the developed system. Section 8.2 describes the implementation of the different modules. Finally, examples of applications are presented and discussed in section 8.3.

8.1 THE CAIP SYSTEM STRUCTURE

The CAIP system is composed of two main parts dealing with two major tasks (see Figure 8.1):

* Inspection Planning
* Tolerance Verification

8.1.1 Inspection Planning

This part of the system consists of four modules:

- Accessibility Analysis Module
- Clustering Module
- Sequencing Module
- Path Planning Module

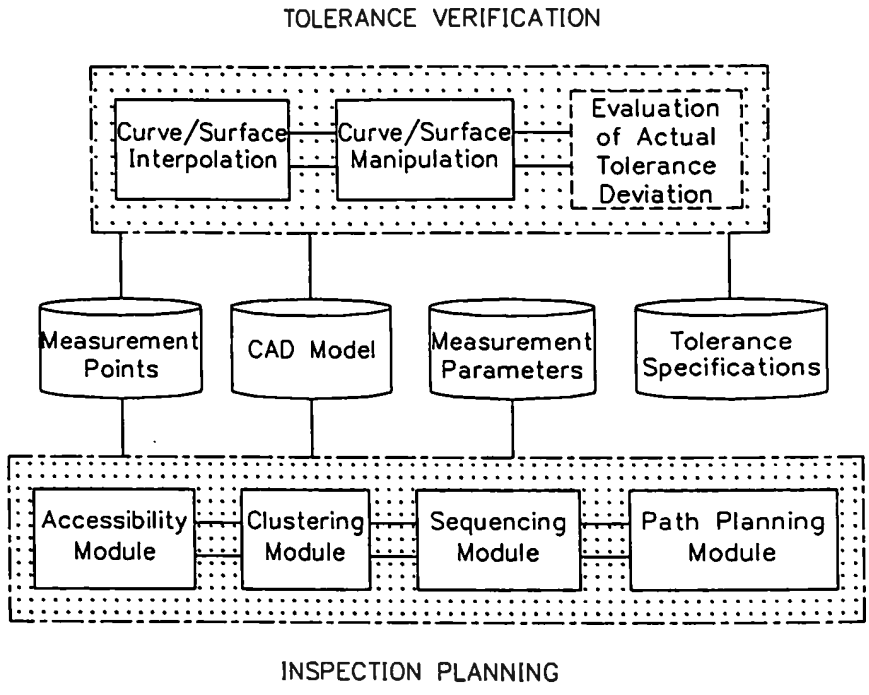


Figure 8.1 Structure of the CAIP System

8.1.1.1 Accessibility Analysis Module

This module performs the accessibility analysis of the inspection features. The input data are: the part and its environment solid models, the measurement points, their number and location (ASME Y14.5M, 1994, ASME Y14.5.1M, 1994, Hocken et al., 1993, Ligget, 1993 and Tandler, 1994), and measurement parameters (i.e. the dimensions of the probe, the machine workspace...). Two possible approaches may be adopted: 1) A continuous approach where the exact accessibility domain of each measurement point is determined, the probe is abstracted as a line segment, and corrections are performed to take into account the actual shape and position of the probe, and 2) a discrete approach where the accessibility is verified for a discrete number of orientations of the probe. These directions represent the resolution and the range of the probe. In this case, the probe

is approximated by a set of cylinders, and ray tracing techniques are used to check the accessibility in each possible direction. The set of all accessible directions is generated for each measurement point and for each probe. The principal clusters as defined in Chapter 4 are generated for further use in the clustering module.

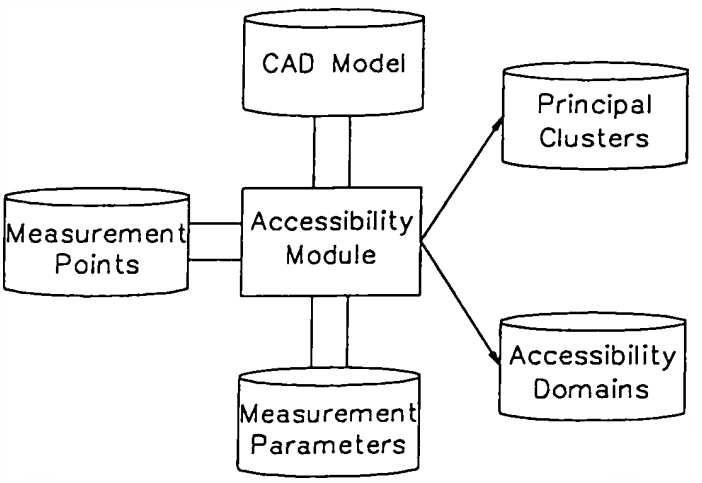


Figure 8.2 Accessibility Analysis Module

8.1.1.2 Clustering Module

The principal clusters obtained previously combined with precedence constraints are utilized to generate the optimum clusters of measurement points. These clusters are then used by the sequencing module.

The user may interactively modify the clusters. This is possible if and only if the accessibility domain of the new cluster (including the new point) is not void. These modifications are necessary when no collision-free path is found between two measurement points within the same cluster.

Figure 8.3 Clustering Module

8.1.1.3 Sequencing Module

This module generates the optimum sequence of measurement points. Two cases are considered: 1) one with precedence constraints, where a selective breadth-first search determines the best sequence(s) of measurement points yielding the optimization of criteria such as: minimum probe changes, minimum probe orientation changes and minimum number of probes used; 2) one without precedence constraints, where the clusters obtained by the previous module yield the optimization of criteria such as those considered with precedence constraints. Each cluster is considered independently and sequenced according to the additional criteria of the minimum travelled distance of the probe. This problem is solved using branch and bound techniques applied to the Travelling Salesman Problem (TSP).

The output of this module is the sequence of measurement points and the clusters they form.

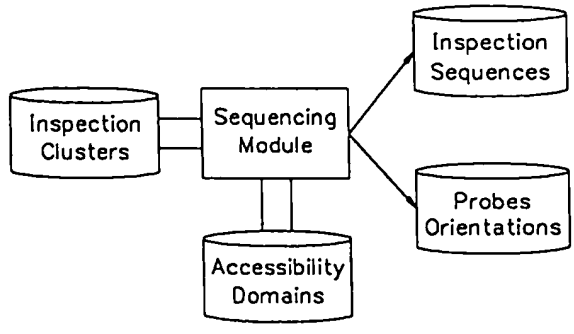


Figure 8.4 Sequencing Module

8.1.1.4 Path Planning Module

This module constitutes the final step in the inspection planning procedure. Using the sequence of measurement points, the probe orientation for each cluster, the models of the part and its environment, the shortest probe path between each two points is determined using Dijkistra's algorithm combined with an interference detection algorithm. The result of this module may be used to generate the CMM command program. The approximation of the part and the obstacles is automatic in the case of basic shapes such as cylinders and parallelepipeds.

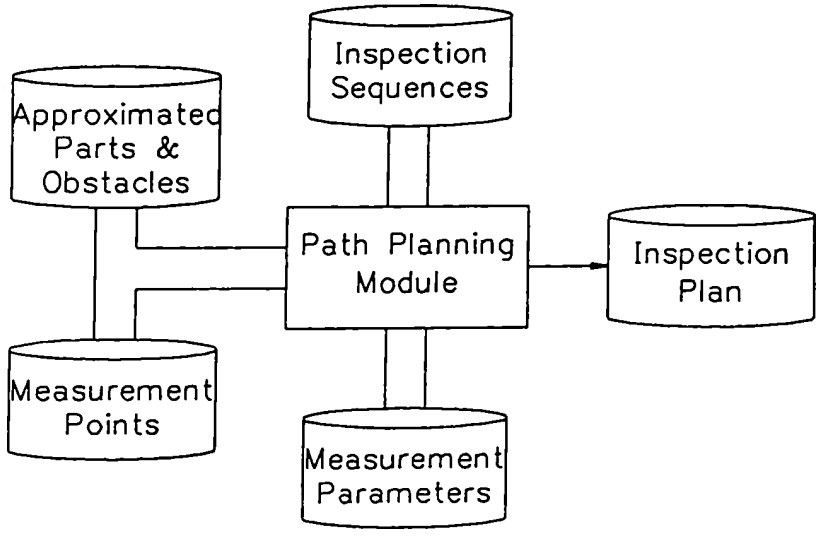


Figure 8.5 Path Planning Module

8.1.2 Tolerance Verification

This part of the system deals with the analysis and interpretation of the results of the measurements in order to evaluate the actual tolerance deviation. Dual kriging is used to interpolate the measured surface while considering the errors in measurement points coordinates, and geometric algorithms for the manipulation of curves and surfaces were developed in order to be used in a module for the evaluation of actual tolerance deviations. This last module is not part of the present work.

8.2 SYSTEM IMPLEMENTATION

This system was implemented using C++ and LISP languages. LISP was used only in the case of sequencing with precedence constraints.

ACIS Solid Modeler (Spatial Technology, 1993) was used for the Accessibility Analysis Calculations.

8.3 EXAMPLES

The first example of this section deals with the inspection planning for an actual part: flange. The second example deals with the interpolation of a sculptured surface from a grid of data points obtained using a CMM .

8.3.1 Example 1

The workpiece of this example was taken from ASME, Y14.5M-1994. It represents a typical example of a toleranced part. Forty measurement points distributed on each surface of the workpiece (see Figure 8.6) were used in the example of Figure 8.7. No precedence constraints were considered and only one probe is utilized in this example.

The origin of the coordinates system is the intersection of plane A and the axis of the workpiece (see Figure 8.7). The coordinates of the measurement points are given in Table 8.1.

They are distributed as follows (see Figure 8.6):

- P1, P2 and P3 on the planar surface (**SP1**).
- P4, P5 and P6 on the planar surface (**SP2**).
- P7, P8 and P9 on the planar surface (**SP3**).
- P10 on the right end of the thread (**SP4**).
- P11, P12 and P13 on the planar surface (**SP5**).

- P14, P15 and P16 on the planar surface (**SP6**).
- P17, P18 and P19 on the external cylindrical surface (**EC1**).
- P20, P21 and P22 on the cylindrical surface of one of the eight holes (**IC1**).
- P23, P24 and P25 on the external cylindrical surface (**EC2**).
- P26, P27 and P28 on the external cylindrical surface (**EC3**).
- P29, P30 and P31 on the external cylindrical surface (**EC4**).
- P32, P33 and P34 on the external cylindrical surface (**EC5**).
- P35, P36 and P37 on the internal cylindrical surface (**IC2**).
- P38, P39 and P40 on the internal cylindrical surface (**IC3**).

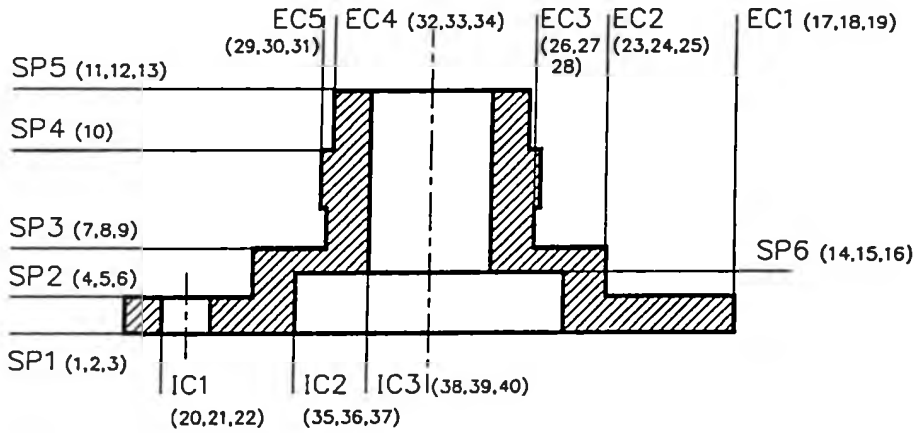


Figure 8.6 Measurement Points

Note that for a planar surface three measurement points are generally sufficient to determine its orientation and position. A larger number of points would be needed for form tolerances. The verification of the dimension (radius or diameter) of a cylindrical surface needs generally three coplanar points, however, for position and orientation tolerances at least four non-coplanar points are needed. For the verification of form tolerances, the number of measurement points depends of the algorithms used for the analysis and verification of the measurements.

P1	21.500	37.239	0.000	P21	3.400	41.400	2.800
P2	-37.239	21.500	0.000	P22	-3.400	41.400	2.800
P3	-37.239	-21.500	0.000	P23	0.000	28.550	10.800
P4	21.500	37.239	5.700	P24	24.725	-14.275	10.800
P5	-37.239	21.500	5.700	P25	-24.725	-14.275	10.800
P6	-37.239	-21.500	5.700	P26	0.000	17.875	17.200
P7	0.000	25.400	15.750	P27	15.480	-8.938	17.200
P8	22.000	-12.700	15.750	P28	-15.480	-8.938	17.200
P9	-22.000	-12.700	15.750	P29	0.000	19.375	26.900
P10	0.000	17.610	31.700	P30	16.779	-9.688	26.900
P11	0.000	13.000	41.200	P31	-16.779	-9.688	26.900
P12	11.258	-6.500	41.200	P32	0.000	15.850	36.450
P13	-11.258	-6.500	41.200	P33	13.727	-7.925	36.450
P14	0.000	16.000	9.500	P34	-13.727	-7.925	36.450
P15	13.856	-8.000	9.500	P35	0.000	10.000	15.850
P16	-13.856	-8.000	9.500	P36	8.660	-5.000	15.850
P17	0.000	49.750	2.800	P37	-8.660	-5.000	15.850
P18	43.085	-24.875	2.800	P38	0.000	22.263	4.750
P19	-43.085	-24.875	2.800	P39	19.280	-11.132	4.750
P20	0.000	47.000	2.800	P40	-19.260	-11.132	4.750

Table 8.1 Coordinates of Measurement Points

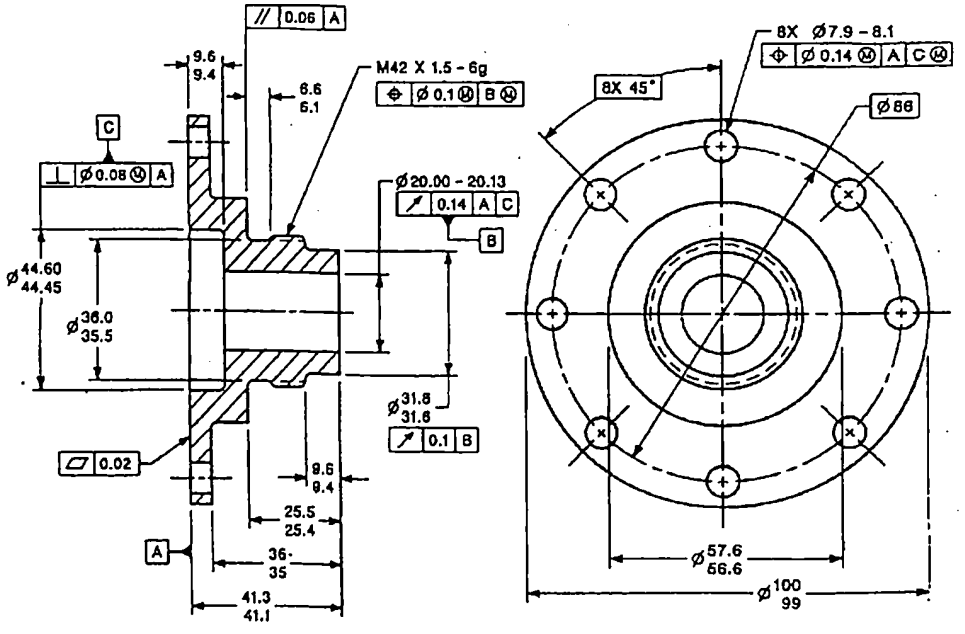


Figure 8.7 Example of Part (from ASME Y14.5M-1994)

In this example, the maximum number of measurement points per surface is limited to three for the sake a simplicity. Only one of the eight holes is considered.

Accessibility Analysis Module

The input consists of the solid model of the workpiece, the coordinates of the measurement points and the geometric characteristics of the probe(s).

The output is composed of: 1) a list of all principal clusters, a total number of 59 principal clusters were generated in this example (see Table 8.2), 2) the discrete accessibility domain of each measurement point for each probe, and 3) the normal vector at each measurement point.

Clustering Module

The input is the list of the principal clusters for each probe. The output is a list of optimum solutions from which the user chooses the most suitable one. Six alternative solutions were generated in this example, solution 2 was retained (see Figure 8.8). Each solution is composed of four clusters of measurement points, hence limiting the number of probe orientation changes to 3. It is composed of the following measurement points:

Cluster 1: 40

Cluster 2: 9 25 28 31 34

Cluster 3: 1 2 3 14 15 16 19 20 21 35 36 38 39

Cluster 4: 4 5 6 7 8 10 11 12 13 17 18 22 23 24 26 27 29 30 32 33 37

Sequencing Module

In this example no precedence constraints are considered. The input to this module is the list of clusters as well as the list of the measurement points and their coordinates. The module sequences each cluster independently (Figure 8.9) and outputs the sequenced clusters.

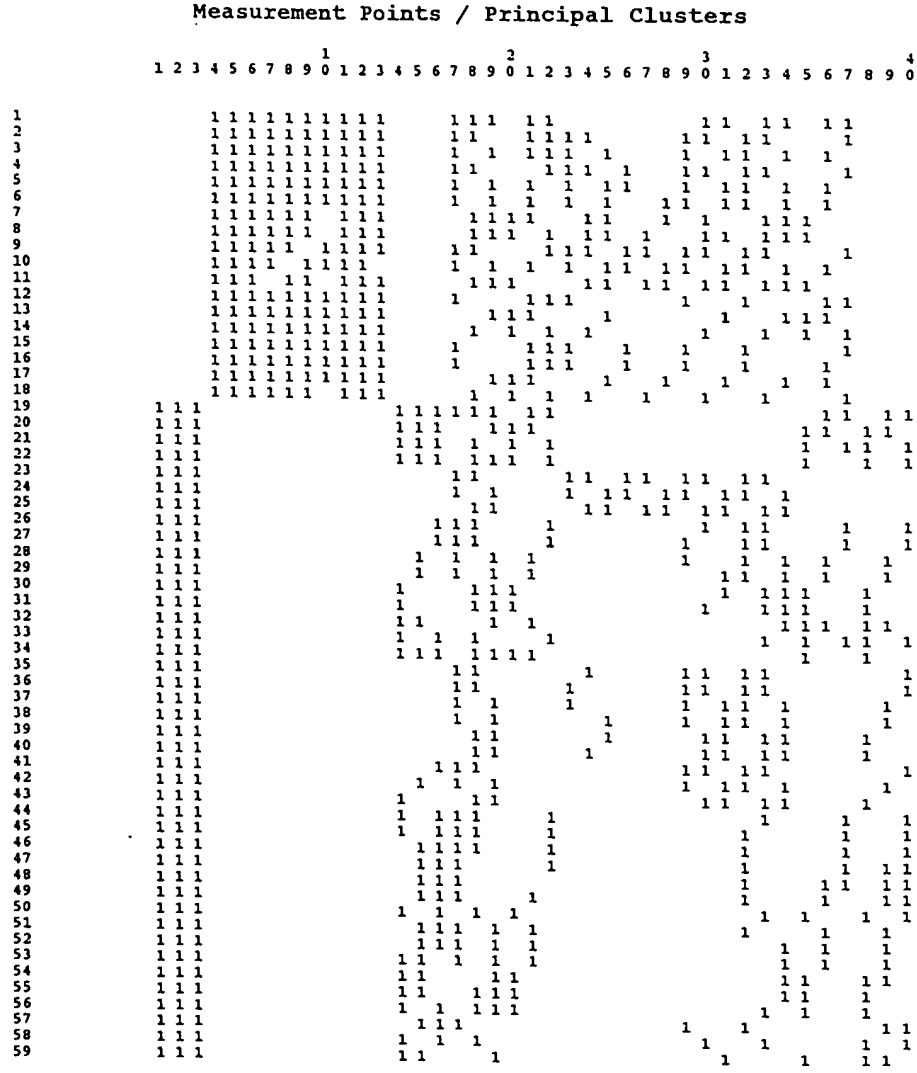


Table 8.2 Operations/Tools Representation of the Principal Clusters

```
Reading data from input file : ~/insp/exdata/clusters.dat
Reading completed ...
Printing 6 Solution(s) ...

SOLUTION 1
 Cluster 1 : operation(s) (039) with Probe 1
 Cluster 2 : operation(s) (08 024 027 030 033) with Probe 1
 Cluster 3 : operation(s) (01 02 03 014 015 016 018 020 022 035 037 038
040)              with Probe 1
 Cluster 4 : operation(s) (04 05 06 07 09 010 011 012 013 017 019 021
023 025
                   026 028 029 031 032 034 036) with Probe 1

SOLUTION 2
 Cluster 1 : operation(s) (040) with Probe 1
 Cluster 2 : operation(s) (09 025 028 031 034) with Probe 1
 Cluster 3 : operation(s) (01 02 03 014 015 016 019 020 021 035 036 038
039)     with Probe 1
 Cluster 4 : operation(s) (04 05 06 07 08 010 011 012 013 017 018 022
023 024              026 027 029 030 032 033 037) with Probe 1

SOLUTION 3
 Cluster 1 : operation(s) (039) with Probe 1
 Cluster 2 : operation(s) (024 026 027 030 033) with Probe 1
 Cluster 3 : operation(s) (01 02 03 014 015 016 018 020 022 035 037 038
040)              with Probe 1
 Cluster 4 : operation(s) (04 05 06 07 08 09 010 011 012 013 017 019
021 023              025 028 029 031 032 034 036) with Probe 1

SOLUTION 4
 Cluster 1 : operation(s) (039) with Probe 1
 Cluster 2 : operation(s) (024 027 028 030 033) with Probe 1
 Cluster 3 : operation(s) (01 02 03 014 015 016 018 020 022 035 037 038
040)              with Probe 1
 Cluster 4 : operation(s) (04 05 06 07 08 09 010 011 012 013 017 019
021 023              025 026 029 031 032 034 036) with Probe 1

SOLUTION 5
 Cluster 1 : operation(s) (040) with Probe 1
 Cluster 2 : operation(s) (025 027 028 031 034) with Probe 1
 Cluster 3 : operation(s) (01 02 03 014 015 016 019 020 021 035 036 038
039)              with Probe 1
 Cluster 4 : operation(s) (04 05 06 07 08 09 010 011 012 013 017 018
022 023              024 026 029 030 032 033 037) with Probe 1

SOLUTION 6
 Cluster 1 : operation(s) (040) with Probe 1
 Cluster 2 : operation(s) (025 026 028 031 034) with Probe 1
 Cluster 3 : operation(s) (01 02 03 014 015 016 019 020 021 035 036 038
039)              with Probe 1
 Cluster 4 : operation(s) (04 05 06 07 08 09 010 011 012 013 017 018
022 023              024 027 029 030 032 033 037) with Probe 1

End Printing Solution(s) ...

Writing solution to file
 Select the solution to output :  2

 Solution 2 Selected
```

Figure 8.8 Screen Dump of the Clustering Module

The sequenced clusters are:

Cluster 1: 40

Cluster 2: 5 9 25 28 31 34

Cluster 3: 1 21 20 2 3 19 16 15 39 36 35 14 38

Cluster 4: 4 8 18 24 27 30 12 33 11 32 10 29 7 23 26 37 13 6 5 22 17

These sequences are chosen so as to minimize the total distance travelled by the probe in each cluster of measurement points.

```
anis(220)time sequence
Runnning Measurement Points Sequencing Program

        Reading Input Data
        Reading Measurement Points Coordinates
        Reading Measurement Points Clusters

        Sequencing Cluster 1
        Sequencing Cluster 2
        Sequencing Cluster 3
        Sequencing Cluster 4

End of Measurement Points Sequencing Program
0.230u 0.170s 0:01.02 39.2% 0+130k 1+1io 5pf+0w
```

Figure 8.9 Screen Dump of the Sequencing Module

Path Planning Module

A preliminary task takes place in order to determine the list of probe orientations associated with each cluster of measurement points. Figure 8.10 and 9.11 shows the set of possible probe orientations of each cluster. The same scheme (θ, φ) used in Chapter 3 is adopted. The accessibility domains determined by the first module are intersected in order to generate the accessible orientations for each cluster. Previous results together with work space discretization parameters (i.e. size of discretized elements, clearance of the probe from the measurement point, resolution of the second order interference checking, safety factor, probe(s) characteristics and

workpiece approximation are used as an input to the path planning algorithms.

```
anis(223)time orient
Number of Measurement Points :  40

    Cluster 1

Probe 1

Point(s) : 40

000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
111111110000000000000000000000000001
111111111110000000000000000000001111
111111111110000000000000000000001111
111111111111000000000000000000011111
111111111111000000000000000000011111
111111111111000000000000000000011111
111111111111000000000000000000011111
111111111111100000000000000000111111

    Cluster 2

Probe 1

Point(s) : 9   25   28   31   34

000000000000000000000000000000000000
000000000000000000111111100000000000
000000000000000111111111111100000000
000000000000001111111111111100000000
000000000000011111111111111110000000
000000000000111111111111111111000000
000000000001111111111111111111100000
000000000001111111111111111111100000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000

    Cluster 3

Probe 1

Point(s) : 1   21   20   2   3   19   16   15   39   36   35   14   38

000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000001110000000000000
000000000000000000011111000000000000
000000000000000000011111000000000000
```

Figure 8.10 Possible Probe Orientation(s) of Each Cluster

Each sequenced cluster is considered independently. The first orientation in the list of possible orientations of the cluster is considered first. If a collision-free path for the entire cluster is not found then the following orientation in the list is considered.

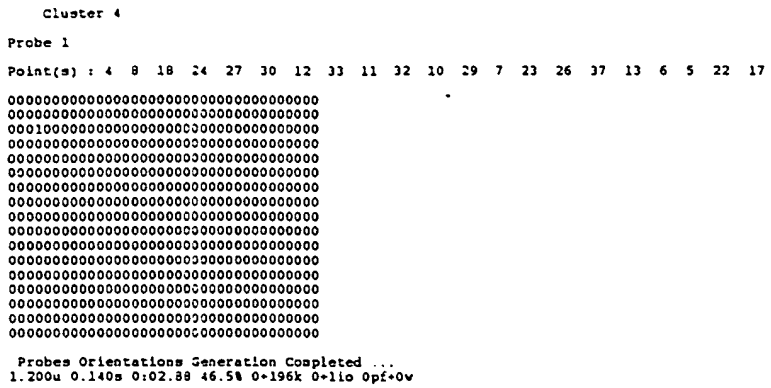


Figure 8.11 Possible Probe Orientation(s) of Each Cluster (Cont.)

If for a given cluster no collision-free path was found for the entire list of possible probe orientations (see Figure 8.12), the user interacts and re-moves the measurement point causing the problem (Measurement point number 37 in this example, see Figure 8.12) an places it in another cluster (cluster 1 in this example) which contains at least one common orientation, the path planning is then resumed. Figure 8.13 shows a screen dump of the program after modification of the clusters.

In the case of basic shapes such as cylinders and parallelepiped it is possible to automate the workpiece approximation procedure by decomposing the part into elementary shapes. Figure 8.14 shows the result of the approximation of this example's workpiece into six pipes.

The output of this module is composed of the characteristic points of each path between two measurement points (an algorithm is used to eliminate colinear points) and the orientation of the probe. The path may be simulated using any CAD system. AutoCAD was used in this case (see Figures 8.15 and 8.16).

Note that up to one million nodes (in the search graph) were used in the path planning of this example which would have needed and adjacency matrix of 10^{12} elements for their representation if the approach of local adjacency with special nodes numbering was not utilized.

```
anis(105)time path
  reading obstacles data

                CLUSTER 1

        Probe Orientation : Theta 100.00   Phi  0.00

                CLUSTER 2

        Probe Orientation : Theta 20.00   Phi 180.00
 Investigating Path Between Points 9 and 25,     The number of nodes is 18000
 Investigating Path Between Points 25 and 28,     The number of nodes is 13440
 Investigating Path Between Points 28 and 31,     The number of nodes is 120000
 Investigating Path Between Points 31 and 34,     The number of nodes is 81600

                CLUSTER 3

        Probe Orientation : Theta 170.00   Phi 210.00
 Investigating Path Between Points 35 and 38,     The number of nodes is 3888
 Investigating Path Between Points 38 and 14,     The number of nodes is 180000
 Investigating Path Between Points 14 and 1,     The number of nodes is 5808
 Investigating Path Between Points 1 and 21,     The number of nodes is 8976
 Investigating Path Between Points 21 and 20,     The number of nodes is 120000
 Investigating Path Between Points 20 and 2,     The number of nodes is 6336
 Investigating Path Between Points 2 and 3,     The number of nodes is 5616
 Investigating Path Between Points 3 and 19,     The number of nodes is 9504
 Investigating Path Between Points 19 and 16,     The number of nodes is 28800
 Investigating Path Between Points 16 and 15,     The number of nodes is 3888
 Investigating Path Between Points 15 and 39,     The number of nodes is 1000000
 Investigating Path Between Points 39 and 36,     The number of nodes is 8160

                CLUSTER 4

        Probe Orientation : Theta 30.00   Phi 30.00
 Investigating Path Between Points 4 and 8,     The number of nodes is 28800
 Investigating Path Between Points 8 and 18,     The number of nodes is 3840
 Investigating Path Between Points 18 and 24,     The number of nodes is 6240
 Investigating Path Between Points 24 and 27,     The number of nodes is 13440
 Investigating Path Between Points 27 and 30,     The number of nodes is 120000
 Investigating Path Between Points 30 and 12,     The number of nodes is 11088
 Investigating Path Between Points 12 and 33,     The number of nodes is 19968
 Investigating Path Between Points 33 and 11,     The number of nodes is 6240
 Investigating Path Between Points 11 and 32,     The number of nodes is 6048
 Investigating Path Between Points 32 and 10,     The number of nodes is 115200
 Investigating Path Between Points 10 and 29,     The number of nodes is 6912
 Investigating Path Between Points 29 and 7,     The number of nodes is 28800
 Investigating Path Between Points 7 and 23,     The number of nodes is 5616
 Investigating Path Between Points 23 and 26,     The number of nodes is 5040
 Investigating Path Between Points 26 and 37,     The number of nodes is 61440
NO PATH WAS FOUND BETWEEN POINTS 37 AND 26
WITH PROBE ORIENTATION : Theta 30.00   Phi 30.00

 TRYING ANOTHER ORIENTATION
2384.810u 8.330s 39:54.61 99.9% 0+0k 2+11io 0pf+0w
```

Figure 8.12 Screen Dump of Path Planning Procedure

If precedence constraints are imposed between measurement points, then the sequencing will no longer be performed by the Travelling Salesman algorithm but will be performed with the search algorithm presented in Chapter 4. Note that precedence constraints may increase the total number of clusters, resulting in additional costs linked to additional resources changes.

```
anis(108)time path
  reading obstacles data

                CLUSTER 1

        Probe Orientation : Theta 180.00   Phi  0.00
 Investigating Path Between Points 40 and 37,     The number of nodes is 8160

                CLUSTER 2

        Probe Orientation : Theta 20.00   Phi 180.00
 Investigating Path Between Points 9 and 25,     The number of nodes is 18000
 Investigating Path Between Points 25 and 28,     The number of nodes is 13440
 Investigating Path Between Points 28 and 31,     The number of nodes is 120000
 Investigating Path Between Points 31 and 34,     The number of nodes is 81600

                CLUSTER 3

        Probe Orientation : Theta 170.00   Phi 210.00
 Investigating Path Between Points 35 and 38,     The number of nodes is 3888
 Investigating Path Between Points 38 and 14,     The number of nodes is 180000
 Investigating Path Between Points 14 and 1,     The number of nodes is 5808
 Investigating Path Between Points 1 and 21,     The number of nodes is 8976
 Investigating Path Between Points 21 and 20,     The number of nodes is 120000
 Investigating Path Between Points 20 and 2,     The number of nodes is 6336
 Investigating Path Between Points 2 and 3,     The number of nodes is 5616
 Investigating Path Between Points 3 and 19,     The number of nodes is 9504
 Investigating Path Between Points 19 and 16,     The number of nodes is 28800
 Investigating Path Between Points 16 and 15,     The number of nodes is 3888
 Investigating Path Between Points 15 and 39,     The number of nodes is 1000000
 Investigating Path Between Points 39 and 36,     The number of nodes is 8160

                CLUSTER 4

        Probe Orientation : Theta 30.00   Phi 30.00
 Investigating Path Between Points 4 and 8,     The number of nodes is 28800
 Investigating Path Between Points 8 and 18,     The number of nodes is 3840
 Investigating Path Between Points 18 and 24,     The number of nodes is 6240
 Investigating Path Between Points 24 and 27,     The number of nodes is 13440
 Investigating Path Between Points 27 and 30,     The number of nodes is 120000
 Investigating Path Between Points 30 and 12,     The number of nodes is 11088
 Investigating Path Between Points 12 and 33,     The number of nodes is 19968
 Investigating Path Between Points 33 and 11,     The number of nodes is 6240
 Investigating Path Between Points 11 and 32,     The number of nodes is 6048
 Investigating Path Between Points 32 and 10,     The number of nodes is 115200
 Investigating Path Between Points 10 and 29,     The number of nodes is 6912
 Investigating Path Between Points 29 and 7,     The number of nodes is 28800
 Investigating Path Between Points 7 and 23,     The number of nodes is 5616
 Investigating Path Between Points 23 and 26,     The number of nodes is 5040
 Investigating Path Between Points 26 and 13,     The number of nodes is 5040
 Investigating Path Between Points 13 and 6,     The number of nodes is 5120
 Investigating Path Between Points 6 and 5,     The number of nodes is 5616
 Investigating Path Between Points 5 and 22,     The number of nodes is 5808
 Investigating Path Between Points 22 and 17,     The number of nodes is 24000
2698.810u 8.720s 45:08.40 99.9% 0+0k 3+11io 0pf+0w
anis(109)
```

Figure 8.13 Screen Dump of Path Planning Procedure

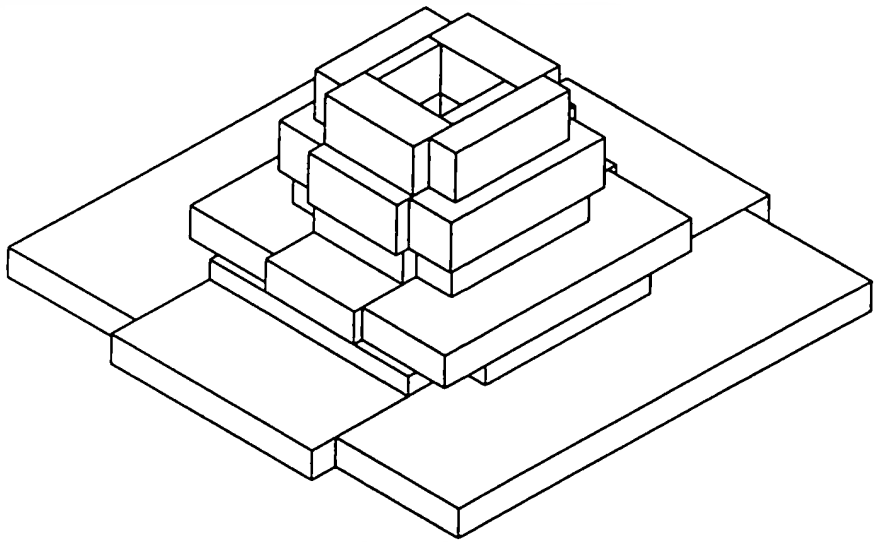


Figure 8.14 Part Geometric Approximation

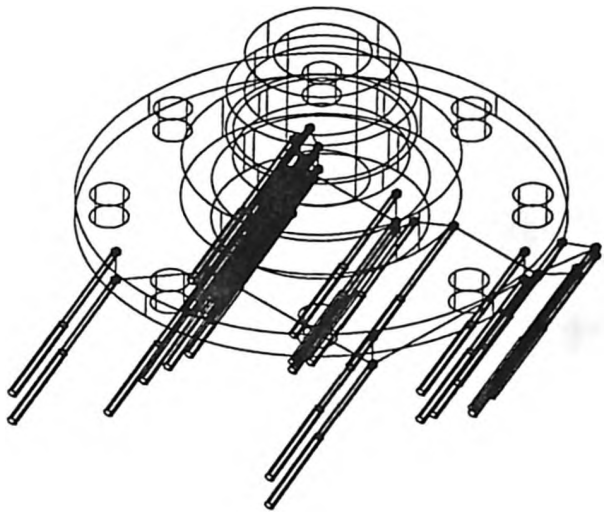


Figure 8.15 Isometric View of the Probe Path for the Inspection of **Cluster 3**

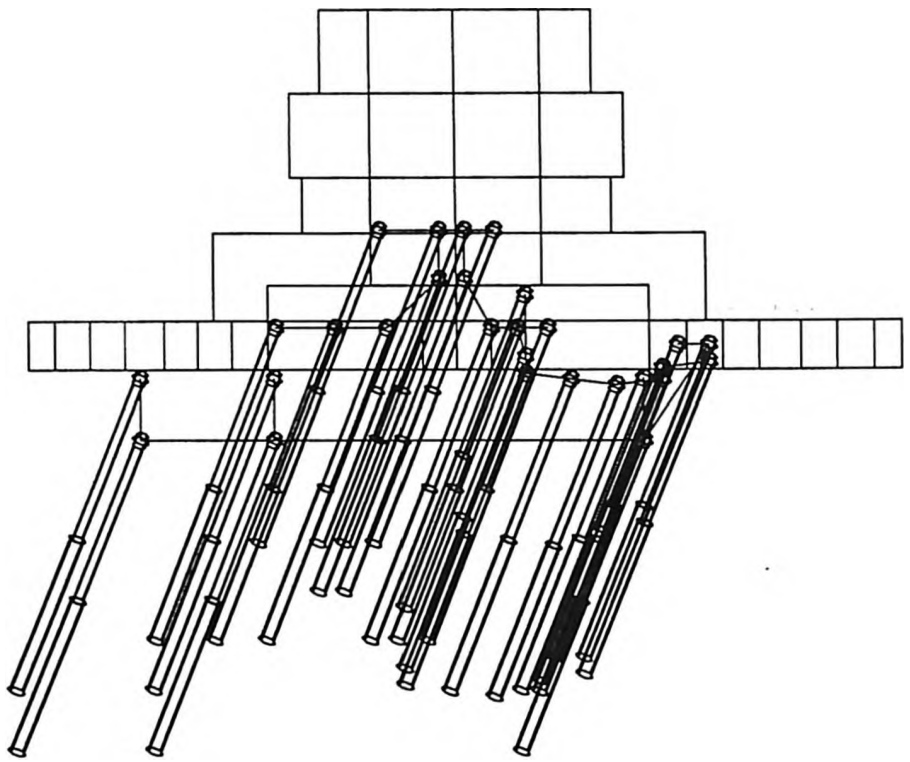


Figure 8.16 Side View of Probe Trajectories for Cluster 3

8.3.2 Logitech Mouse Inspection/Surface Fitting

In this example a patch of the top surface of a Logitech mouse (Figure 8.17) is scanned using discrete tactile measurements. The grid of 442 data points represented in Figure 8.18 was obtained automatically using a DEA CMM machine. Cross sections along the x-direction were scanned with a constant increment along the y-direction. Note that the grid of data points obtained is irregular, i.e. the number of points along the cross-sections is not uniform. A regular grid of 20 x 13 data points is extracted from the initial data (see Figure 8.19).

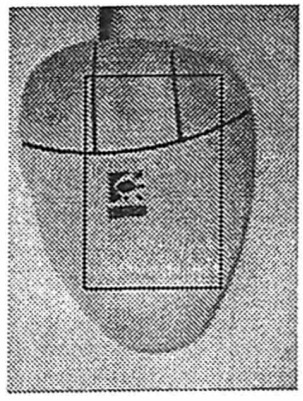


Figure 8.17 Logitech Computer Mouse

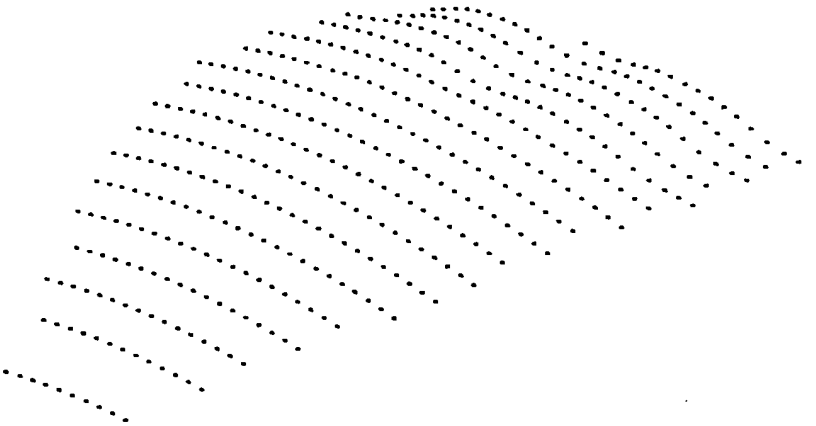


Figure 8.18 Data Points Measured on the Top Surface of the Mouse

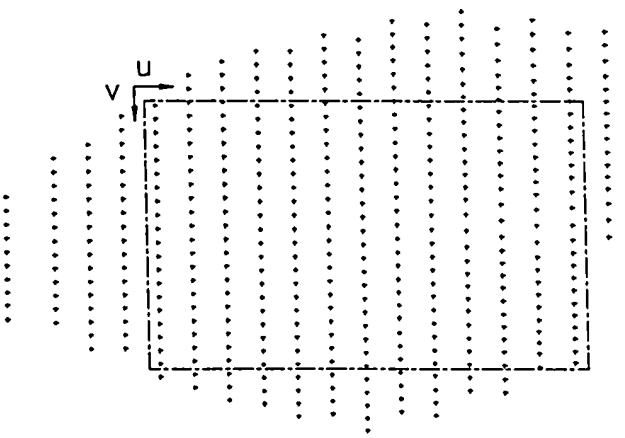


Figure 8.19 The Grid of Points (20 x 13) Extracted from the Initial Data

Two cases are considered: 1) interpolating the data points exactly, and 2) interpolating the data while considering a maximum deviation (uncertainty) of the data points from the model. In both cases the kriging profiles are composed of a linear drift and a cubic generalized covariance.

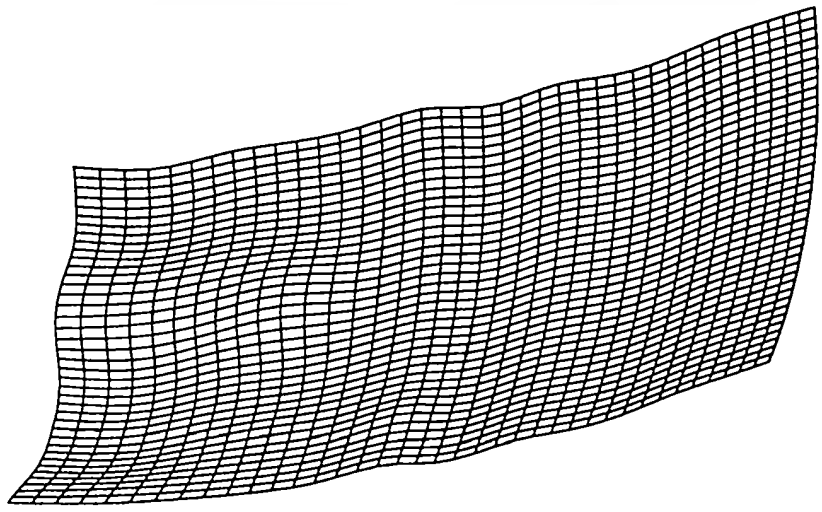


Figure 8.20 Interpolated Surface without Uncertainties

Exact Interpolation of the data points

Equation 8.35 is utilized to define the interpolation model. The surface in Figure 8.20 was obtained. We notice that the surface is not very smooth, this is particularly visible in the (x,z) and (y,z) planes projections (see Figure 8.21 and Figure 8.22).

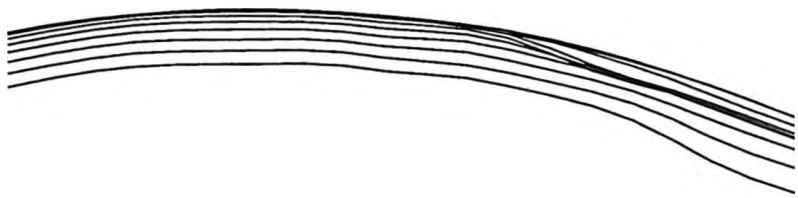


Figure 8.21 Isoparametrics Curves along the *u* Direction

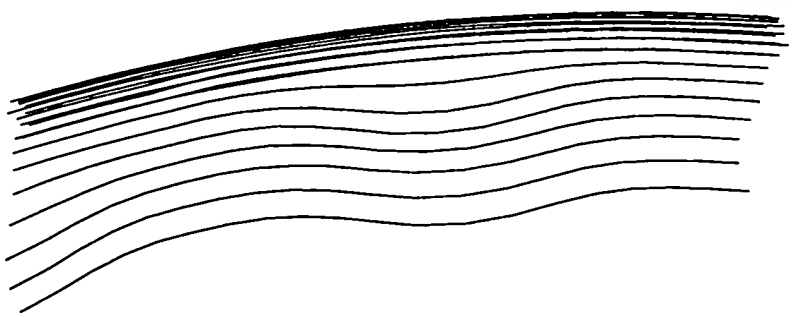


Figure 8.22 Isoparametric Curves along *v* direction

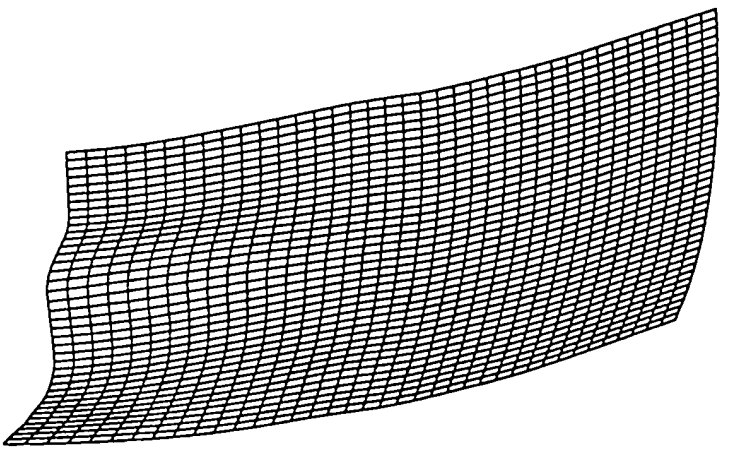


Figure 8.23 Interpolated Surface with Uncertainties

Interpolation with Uncertainties in the Data Points

The procedure proposed in Chapter 6 paragraph 5 is used to interpolate the grid of data points. Figures 8.21 and 8.22 show that the isoparametric curves in the u direction are less smooth than those in the v direction. therefore, only the isoparametric curves along u direction are allowed to deviate. Figure 8.23 shows the surface obtained with a maximum allowable deviation of 2mm along the x,y and z components. In this case, the isoparametric curves of the surface are smoother than those obtained with an exact interpolation of the data points. The actual maximum, minimum and average deviation of the data points from the interpolated surface are respectively 0.797, 0.022 and 0.373mm. These deviations are distributed along the coordinates components as follows:

	max	**min**	**average**
Δx	0.795	0.022	0.373
Δy	0.290	0.000	0.020
Δz	0.128	0.000	0.006

From the above results we can see that by introducing the uncertainties the surface tends to deform mainly along the x direction in order to have a smoother surface and to eliminate the noise and discontinuities.

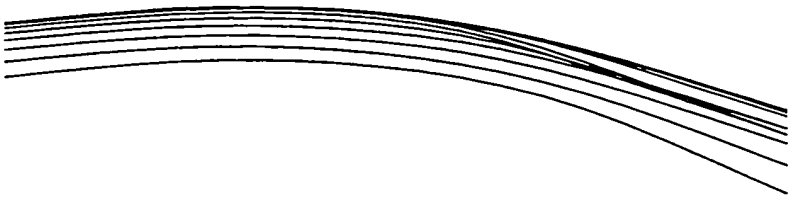


Figure 8.24 Isoparametrics Curves along the u Direction

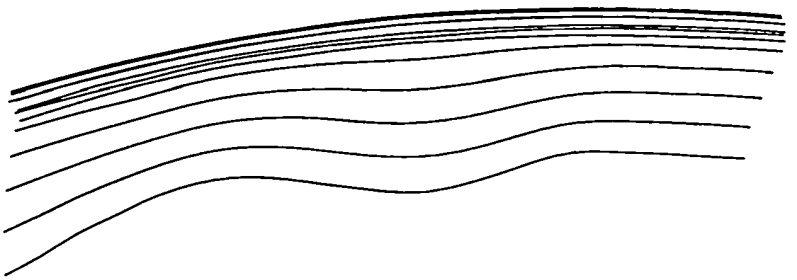


Figure 8.25 Isoparametrics Curves along the v Direction

Note that if a least squares interpolation is considered, then the surface obtained with linear drifts is a plane which is a poor representation of the of the actual surface.

8.4 DISCUSSION

Two examples have been presented in this chapter. In the example of CMM inspection planning, the results obtained are optimum and have been verified and simulated. It has been shown that user interaction may be considered advantageously in order to choose, adapt and

improve the multiple solutions proposed by the system.

The second example deals with the interpolation of a sculptured surface composed of 260 data points using a single surface patch. This example demonstrates the capabilities of the kriging method for the interpolation of large set of data points which would have needed a decomposition of the surface into several patches if interpolation methods such as Bezier were used. In addition, the uncertainties considered in the interpolation model filtered the noise in the data and allowed a better representation of the actual surface.

CHAPTER 9

CONCLUSION

This chapter is divided into three sections: contributions, conclusions and future research.

9.1 CONTRIBUTIONS

The reported research makes the following contributions to the fields of inspection planning and geometric modeling

9.1.1 Inspection Planning

* A new algorithm is developed for determining accessibility domains (or cones). This algorithm is general and provides the accessibility cone (continuous domain) for points in space or on surfaces, and is not limited to planar patches, unlike existing methods.
* An improved discrete accessibility algorithm is developed for probes with discrete possible orientations. The actual shape and position of the probe are considered from the outset. Additional corrections are not needed and bent probes may easily be considered using the same approach.
* A novel formulation of the problem of measurement points clustering and probe orientation selection in terms of operations sequencing and resources allocation is adopted.
* A method for optimum measurement points clustering and sequencing has been developed. The criteria are in this case the minimum number of clusters, the

minimum number of resources used and finally the minimum distance travelled by the probe. Precedence constraints are also considered and alternative plans are proposed.

* A collision-free probing path algorithm is enhanced. The distance between each two measurement points is minimized.
* A modular CAIP system integrating inspection planning tasks was developed and validated. Examples of actual parts have been solved, tested and simulated.

9.1.2 Geometric Modelling

* Uncertainties in measurement points as well as linear constraints are incorporated into the interpolation model using dual kriging.
* Parametric kriging as a combination of interpolation profiles is formulated, which permits the extension of parametric kriging to solids and eventually the interpolation of n-D entities.
* Dual kriging is generalized to incorporate B-splines and NURBS.
* Kriging, originally formulated for the interpolation of a grid of data points, is extended to interpolate a set of analytic curves/surfaces, hence modeling other classes of surfaces such as ruled surfaces, blending surfaces, sweeping and skinning surfaces.
* Geometric algorithms, as opposed to numerical, analytical or differential algorithms, for the intersection of curves and surfaces are developed. Algorithms for the intersection of parametric/implicit and parametric/parametric entities (curves and surfaces) as well as for the projection of points on curves and surfaces have been developed, implemented and validated.
* Various types of curves, surfaces and solids have been interpolated using dual kriging. Classical shapes such as revolution curves, surfaces and solids as well as

complex shapes have been modelled.

9.2 CONCLUSIONS

In this research we focused on two main issues related to inspection using CMMs: 1) planning, and 2) verification of measurement results.

The planning issue was addressed by the formalization of each step of the planning procedure with emphasis on the optimization of the inspection plan beginning with high-level planning then proceeding to low level-planning. A least commitment strategy was adopted throughout the planning in order to avoid eliminating good solutions. Accessibility analysis constitutes the basis of inspection planning. The abstraction of the probe and its environment were chosen so as to be the closest to reality while remaining in the frame of the least commitment strategy. Two approaches were developed: 1) a continuous approach generating the exact accessibility domain of a point in the space, given the characteristics of the probe, the workpiece to be inspected and the measurement environment; and 2) a discrete approach generating a discrete accessibility domain which depends of the resolution of the indexed probe. The results of the accessibility analysis are exploited in a novel way by formulating the problem of measurement points clustering in terms of a resources allocation problem, hence utilizing a new optimization algorithm that minimizes the number of clusters using the same probe and the same probe orientation. It is also possible to consider precedence constraints between measurement operations; in such a case, a sequence of measurement operations is generated. Where no precedence constraints are considered, an additional procedure for the sequencing of measurement operations in each cluster is needed. This problem has been solved by considering the shortest distance travelled by the probe when inspecting the cluster of measurement points as an optimization criterion. The last step in the planning procedure is the collision-free path planning of the probe. This part is the most delicate and most time-consuming. The shortest collision-free probe path between two measurement points

is determined by combining Dijkistra's shortest path algorithm with a collision detection algorithm. The workspace is discretized, and approximated representations of the probe, the workpiece and its environment are utilized in the algorithm.

The final process plan is composed of the sequence of measurement points, the probes and their orientations for each measurement point, and the path of the probe expressed as a sequence of linear segments.

The inspection plan is of high quality, since optimization criteria are followed throughout the planning. In addition, alternative plans are proposed which allow more flexibility in the inspection plan.

In the part dealing with the verification of measurement results, a set of tools for the interpolation and manipulation of measured surfaces has been developed. These tools may be utilized for the determination of the deviation of measured surfaces and design surfaces, hence allowing the verification of geometric tolerance specifications. Measured surfaces are interpolated using dual kriging which has been formulated to take into account measurement errors in a novel way as well as some types of linear constraints. This technique has been generalized and enhanced, hence extending its application to analytical solid modeling and continuous data interpolation. The main advantage of this technique compared to existing methods may be summarized in: 1) its generality since it encompasses several existing techniques, 2) its simplicity and flexibility which leads to an excellent computation efficiency, 3) the possibility to interpolate with derivatives data and with linear constraints as well as with specified uncertainties in the data. Tools for the manipulation of curves and surfaces based on geometric algorithms have also been developed. These permit overcoming some of the most critical problems encountered when using classical analytical methods.

9.3 FUTURE RESEARCH

There are a number of issues related to inspection planning and to the geometric modelling of measured curves and surface that need further investigation.

In inspection planning the issues are:

* Development of a knowledge-based system which will generate the measurement points, their location and their number according to the tolerance specified by designers for various features of the part. Alternatively, a computer interface which allows the user to interactively define these points.
* Improvements and acceleration of the performances of the path planning module. This can be achieved by automating the approximation procedure of the probe, the workpiece and its environment and by utilizing a parallel architecture for the computations of the shortest path and the collision detection procedures.
* The need for post-processor program in order to interface the inspection planner and the CMM machine. A module to determine the time it takes for inspection, the set-up, and the calibration is needed in order to evaluate the overall cost of the inspection and to eventually choose the least expensive inspection plans.
* Feedback inspection data to the CAD system to close the loop of product development.
* The integration of the inspection planner and a tolerance verification system.

In geometric modelling, the main issues to be investigated are:

* How dual kriging and NURBS interpolation compare, specifically when

representing or approximating general dual kriging using NURBS and vice-versa? This feature is very important for the compatibility of the proposed kriging models with existing CAD systems which utilize NURBS.

* Dual kriging has been used as an interpolation tool. The issue of data fitting using this technique is of great interest especially if large sets of data points are considered. This permits the representation of complicated surfaces without needing to decompose the surface into patches which are interpolated then joined.

* Transfinite interpolation using dual kriging may be performed at present only along one parameter of the surface. It is very important to extend the transfinite interpolation using dual kriging to the two surface parameters (similar to Coons surfaces). This will solve the problem of joining two different patches while respecting continuity constraints.

REFERENCES

ASME, *Dimensional Measuring Interface Specification (DMIS) : ANSI/CAM-I 101-1990*, New York, N.Y.

ASME Y14.5M, *1994 Dimensioning and Tolerancing.*

ASME Y14.5.1M, *1994 Mathematical Definitions.*

Agapiou, J.S., "Sequence of Operations Optimization in Single-Stage Multifunctional Systems," *Journal of Manufacturing Systems*, Vol. 10, No. 3, pp. 194-208, 1991.

Anon, "CMM cuts measurement times by 24:1 ratio," *Tooling & Production*, Vol. 55, No. 12, pp. 124-126, 1990.

Asteasu, C. and Orbegozo, A., "Parametric piecewise Surfaces Intersection," *Computer & Graphics*, Vol. 15, No 1, pp. 9-13, 1991.

Avriel, M., *Nonlinear Programming Analysis and Methods*, Prentice-Hall, 1976.

Aziz, N.M. and R. Bata, R., "Bezier Surface/Surface Intersection," *IEEE Computer Graphics & Applications*, pp. 50-58, January 1990.

Blechschmidt, J.L. and Nagasuru, D., "The use of Algebraic Functions as a Solid Modeling Alternative: An Investigation," *1990 ASME Design Technical Conferences*, Chicago, IL., pp. 33-41, 1990.

Brown, C.W. and Gyorog, D.A., "Generative Inspection Process Planner for Integrated Production," *Advances in Integrated Product Design and Manufacturing*, The Winter Annual Meeting of the American Society of Mechanical Engineers, Dallas, Texas, pp. 151-162, November 1990.

Burden, R.L. and Faires, J.D., *Numerical Analysis*, PWS-KENT Publishing Company, Boston, 1989.

Campbell, C.E and Luh, J.Y.S., "A Preliminary Study on Path Planning of Collision Avoidance for Mechanical Manipulators," *Tech. Report TR-EE 80-48*, Purdue University, Lafayette, IN, December 1980.

Campbell, C.E and Luh, J.Y.S., "Collision-free Path Planning for Industrial Robots," *Proc. IEEE Conference on Decision Control*, pp. 593-598, 1982.

Chen, Y.D., Tang, X.J., Ni, J. and Wu, S.M., "Automatic Digitization of Free-form Curve by Coordinate Measuring Machines," *International Journal of Production Research*, Vol. 32, No. 11, pp. 2603-2612, 1994.

Chien, C.H. and Aggarwal, J.K., "Volume/Surface Octree for the Representation of Three -Dimensional Objects," *Computer Vision, Graphics and Image Processing*, vol. 36, pp. 100-113, 1986.

Chiles, J.P., "How to Adapt Kriging to Non-classical Problems: Three Case Studies," *Advanced Geostatistics in the Mining Industry*, D. Reidel Publishing Company, Dordrecht-Holland, pp. 69-89, 1973.

Chiles, J.P. "Géostatistiques des phenomènes non stationnaires (dans le plan)," *thèse de docteur ingénieur*, Université de Nancy-I, France, 1977.

Chivate, P.N. and Jablokow, A.G., 1993, "Solid-model Generation from Measured Point Data," *Computer-Aided Design*, Vol. 25, No. 9, pp. 587-600, september 1993.

Christakos, G., "On the Problem of Permissible Covariance and Variogram Models," *Water Resources Research*, Vol. 20, No. 2, pp. 251-265, 1984.

de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.

Deutsch, C.V. and Journel, A.J., *GSLIB: Geostatistical Software Library and User's*, Guide, Oxford University Press, 1992.

Dijkistra, E.W., "A Note on Two Problems in Connection with Graphs," *Numerishe Matematik*, Vol. 1, pp. 269-271, 1959.

Dodini, A., Moroni, G., Palezzato, P. and Semeraro, Q., "Touch Probe Configuration for an Automatic Inspection System," *Manufacturing Systems*, Vol. 23, No. 2, pp. 123-128, 1994.

Elber, G., "Accessibility in 5-axis Milling Environment," *Computer-aided Design*, Vol. 26, No. 11, pp. 796-802, November 1994.

ElMaraghy, H.A. and Gu, P.H., "Expert System for Inspection Planning," *Annals of the CIRP*, Vol. 36/1/1987, pp. 85-89, 1987.

ElMaraghy, H.A., "Intelligent Product Design and Manufacture," *Chapter 7 in Artificial Intelligence in Design*, ed. D.T. Pham, Springer-Verlag, pp. 147-168, 1991.

ElMaraghy, H.A. and ElMaraghy, W.H., "Bridging the Gap Between Process Planning and Production Planning and Control," *Proc. 24th CIRP International Seminar on Manufacturing Systems*, Copenhagen, Denmark, pp. 1-10, 1992.

ElMaraghy, H.A., "Evolution and Future Perspectives of CAPP," *Annals of the CIRP*, Vol. 42, No. 2, pp. 1-23, 1993.

ElMaraghy, H.A. and ElMaraghy, W.H., "Computer-Aided Inspection Planning (CAIP)," Manufacturing Research and Technology 20, *Advances in Features Based Manufacturing*, Elsevier Science, B.V., pp. 363-395, 1994.

Farebrother, R.W., *Linear Least Squares Computations*, STATISTICS: Textbooks and Monographs, Marcel Decker inc, 1988.

Faux, I.D. and Pratt, M.J., *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester, UK, 1980.

Fiorentini, F., Moroni, G. and Palezzato, P., "Computer Aided Inspection Planning: Automatic Treatment of the Inspection Features," *25th ISATA Silver Jubilee Conference*, Florence, Italy, June 1992.

Fiorentini, F., Moroni, G. and Palezzato, P., "Feature Selection for an Automatic Inspection System," *24th CIRP International Seminar on Manufacturing Systems*, Copenhagen, Denmark, pp. 199-208, 1992.

Foulds, L.R., *Combinatorial Optimization for Undergraduates*, Springer-Verlag Inc., New York, 1984.

Ge, Q., Chen, B., Smith, P. and Menq, C.H., "Tolerance Specification and Comparative Analysis for Computer-Integrated Dimensional Inspection," *International Journal of Production Research*, Vol. 30, No. 9, pp. 2173-2197, 1992.

Ghosh, P.K. , "A Mathematical Model for Shape Description Using Minkowski Operators," *Computer Vision, Graphics, and Image Processing*, Vol. 44, pp. 239-269, December 1988.

Gilbert, R., Carrier, R., Benoit, C., Soulie, M. and Schiettekatte, J., "Application of Dual Kriging in Human Factors Engineering," *Advances in Industrial Ergonomics and Safety II*, ed. Biman Das, Taylor & Francis, London, 1990.

Hedrick, R.W. and Bedi, S., "Method for Intersection of Parametric Surfaces," *Transactions of the CSMI*, vol. 14, No 3, pp. 79-84, 1990.

Hocken, R.J., Raja, T. and Babu, U., "Sampling Issues in Coordinate Metrology," *Manufacturing Review*, Vol. 6, No. 4, pp. 282-294, December 1993.

Hoffmann, C. M., *Geometric & Solid Modelling: An Introduction*, Morgan Kaufmann, 1989.

Hsieh, Y.C., Drake, S.H. and Riesenfield, R.F. "Reconstruction of Sculptured Surfaces using Coordinate Measuring Machines," DE-vol. 65-2, *Advances in Design Automation*, Vol. 2, pp. 35-46, 1993.

Hu, T.C., Kahng, B. and Robins, G., "Optimal Robust Path Planning in General Environments," *IEEE Transactions in Robotics and Automation*, Vol. 9, No. 6, pp. 775-784, December 1993.

Huang, P.S. and Wu, S.M., "A Multi-Degree-of-Freedom Measuring System for CMM Geometric Errors," *Journal of Engineering for Industry*, Vol. 114, pp. 362-369, August 1992.

Jones, S.D. and Ulsoy, A.G., "An Optimization Strategy for Maximizing Coordinate Measuring Machine Productivity: PART I: Quantifying the Effects of Operating Speed on Measurement Quality," *First S.M. Wu Symposium on Manufacturing Science*, USA Venue, pp. 211-218, May 1994.

Jones, S.D. and Ulsoy, A.G., "An Optimization Strategy for Maximizing Coordinate Measuring Machine Productivity: PART II: Problem Formulation, Solution and Experimental Results," *First S.M. Wu Symposium on Manufacturing Science*, USA Venue, pp. 219-226, May 1994.

Johnstone, J.K. and Shene, C.K., "Computing The Intersection of a Plane and a Natural Quadric," *Computer & Graphics*, Vol. 16, No. 2, pp. 179-186, 1992.

Juran, J.M., *Quality Control Handbook*, New York, McGraw-Hill, 1974.

Klass, R. and Kuhn, B., "Fillet and Surface Intersections Defined by Rolling Balls," *Computer Aided Geometric Design*, 9, pp. 185-193, 1992.

Koshnevis, B. and Yeh, Z., "An Automatic Measurement Planning System for CMMs," *Manufacturing Review*, Vol. 6, No. 3, pp. 221-227, September 1993.

King, L.S.B. and Hutter, I., "Theoretical Approach for Generating Optimal Fixturing Locations for Prismatic Workparts in Automated Assembly," *Journal of Manufacturing Systems*, Vol. 12, No. 5, pp. 409-416, 1993.

Krige, D. G., "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," *J. Chem. Metal. Min. Soc. S. Afr.*, S2, pp. 119-139, 1951.

Kuriyama, S., "Surface Modelling with and Irregular Network of Curves Via Sweeping and Blending," *Computer-Aided Design*, Vol. 26, No. 8, pp. 597-606, August 1994.

Laperriere, L., "Generative Assembly Process Planning," *Ph.D. thesis*, McMaster University, 1992.

Latombe, J.C., *Robot Motion Planning*, Boston, MA: Kluwer Academic, 1991.

Lehning, H., *Analyse en dimension finie*, masson, Paris, 1986.

Ligget, J.V., *Dimensional Variation Management Handbook, A Guide for Quality, Design and Manufacturing Engineers*, Prentice-Hall Inc., 1993.

Lim, C.P. and Menq, C.H., "CMM Feature Accessibility and Path Generation," *International Journal of Production Research*, Vol. 32, pp. 597-618, March 1994.

Limaiem, A. and ElMaraghy, H.A., "An Integrated Resources Selection and Operation Sequencing Method for CAPP," *Proc. 27th CIRP International Seminar on Manufacturing Systems*, Ann Arbor, Michigan, pp. 312-319, 1995.

Limaiem, A. and ElMaraghy, H.A., "Analytical Solid Modeling Using Dual Kriging," *Proc. 21st ASME Design Automation Conference*, Boston, MA, September 1995.

Limaiem, A., and Trochu, F., "Geometric Algorithms for the Intersection of Curves and Surfaces," *Computer & Graphics*, Vol. 19, No. 3, pp. 391-403, 1995.

Lozano-Perez, T., "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Comput.*, Vol. C-32, pp. 108-120, 1983.

Lozano-Perez, T. and Nesley, M.A., "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," *Communications of the ACM*, 22(10), pp. 560-570, October 1979.

Lozano-Perez, T. and Nesley, M.A., "Automatic Planning of Manipulator Transfer Movements," *IEEE Transactions Syst., Man, Cybern.*, Vol 8MC-11, No. 10, October 1981.

Matheron, G., "The intrinsic random functions and their applications," *Advances in Applied Probability*, pp. 439-468, December 1973.

Matheron, G., "Splines et Krigeage: leur équivalence formelle," *Rapport N-667*, Centre de Géostatistique, Fontainebleau, École des Mines de Paris, 1980.

Marciniak, K., *Geometric Modelling for Numerical Controlled Machining*, Oxford University Press, pp. 128-136, 1990.

Medland, A.J. and Mullineux, G., "Strategies for the Automatic Path Planning of Coordinate Measuring Machines," *24th CIRP International Seminar on Manufacturing Systems*, pp. 209-218, 1992.

Menderos, D.J., Thomas, G., Ratkus, A.B. and Cannon, D., "Off-line Programming of Coordinate Measuring Machines Using a Hand-Held Stylus," *Journal of Manufacturing Systems*, Vol. 13, No. 6, pp. 401-411, 1994.

Menq, C.H., Yau, H.T. and Lai, G.Y., "Automated Precision Measurement of Surface Profile in CAD-Directed Inspection," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 2, pp. 85-92, April 1992.

Menq, C.H., Yau, H.T. and Wong, C.L., "An Intelligent Planning Environment for Automated Dimensional Inspection Using Coordinate Measuring Machines," *Journal of Engineering for Industry*, Vol. 114, pp. 222-230, May 1992.

Merat, F.L., Radack, G.M., Roumina, K., and Ruegsegger, S., "Automated Inspection Planning Within the Rapid Design System," *Proc. IEEE International Conference on Systems Engineering*, pp. 42-48, 1991.

Miller, J.M. and Stockman, G.C., "On the Number of Linear Extensions in a Precedence Graph," *Technical Report*, Michigan State University, 1989.

Montes, P., "Kriging Interpolation of a Bezier Curve," *computer-aided design*, Vol. 23, No. 10, pp. 713-716, 1990.

Morteson, M. *Geometric Modeling*, John Wiley and Sons Ltd., New York, 1985.

Mullenheim, G., "On Determining Start Point for a Surface/Surface Intersection Algorithm," *Computer Aided Geometric Design*, Vol. 8, pp. 401-408, 1991.

Neumann, A., *Geometric Dimensioning and Tolerancing*, Technical Consultants Incorporated, Longboat Key, Florida, 1995.

Nielsson, N.J., *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill Inc, New York. 1971.

Oetjens, T.J., "New Developments in Coordinate Measuring Machines for Sheet Metal," *Autofact '89*, pp. 19-11, 1989.

Pahk, H. and Kim, J., "Development of Computer Integrated System for Error Diagnosis of a CMM Using Calibrated Mechanical Artifacts," *Int. J. Mach. Tools Manufact.*, Vol. 33, No. 6, pp. 773-790, 1993.

Phillips, S.D., Borchardt, B. and Gadrey, G., "Measurement Uncertainty Consideration for Coordinate Measuring Machines," *NISTIR5170*, U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, 1993.

Piegl, L. and Tiller, W., "Curve and Surface Constructions Using Rational B-Splines," *Computer-Aided Design*, Vol. 19, No. 9, pp. 485-498, November 1987.

Piegl, L., "On NURBS: Survey," *IEEE Computer Graphics and Applications*, pp. 51-71, January 1991.

Prabhu, P., Elhence, S., Wang, H. and Wysk, R., "An Operations Network Generator for Computer Aided Process Planning," *Journal of Manufacturing Systems*, Vol. 9, No.4., 1991.

Pratt, M.J. and Geisow, A.D., "Surface/Surface intersection problems in: Gregory," *The Mathematics of Surfaces*, Oxford University Press, Oxford, 1986.

Pujair, L.A.K. and Reddy P.G., "Linear Octree by Volume Intersection," *Computer Vision, Graphics, and Image Processing*, Vol. 45, pp. 371-379, 1989.

Rho, H.M, Geelink, R., van't Erve, A.H. and Kals, H.J.J., "An integrated Cutting Tool Selection and Operation Sequencing Method," *Annals of the CIRP*, Vol. 41, No 1, pp. 517-520, 1992.

Requicha, A.A.G. and Chan, S.C., "Representation of Geometric Features, Tolerances and Attributes in Solid Modelers Based on Constructive Geometry," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 3, pp. 156-166, September 1986.

Rich, E., *Artificial Intelligence*, McGraw-Hill, 1991.

Rivest, R.L. and Vuillemin, J., "On Recognizing Graph Properties from Adjacency Matrices," *Theoritical Computer Science*, Vol. 3 , North-Holland Publishing Company, pp. 371-384, 1976.

Sahoo, K.C. and Menq, C.H., "Localization of 3-D Objects having Complex Sculptured Surfaces Using Tactile Sensing and Surface Description," *Journal of Engineering for Industry*, Vol. 113, pp. 85-92, February 1991.

Sarkar, B. and Menq, C.H., "Scanning Compound Surfaces with no Existing CAD Model Using Touch Probe of a Coordinate Measuring Machine," PED-vol.50, *Intelligent Design and Manufacturing for Prototyping*, pp. 97-112, 1991.

Sostar, A., "Coordinate Measuring Machines in Flexible Manufacturing Systems," *Manufacturing Systems*, Vol. 23, No. 2, pp. 119-128, 1994.

Spatial Technology inc., ACIS Geometric Modeler, Programmers Manual, version 1.5 9/93, issue 1, 1993.

Spyridi, A.J. and Requicha, A.A.G., "Accessibility Analysis for the Automatic Inspection of Mechanical Parts by Coordinate Measuring Machines," *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, OH, pp. 1284-1289, May 1990.

Spyridi, A.J. and Requicha, A.A.G., "Accessibility Analysis for Polyhedral Objects," in S.G. Tzafestas, ed., *Engineering System with Intelligence: Concepts, Tools and Applications*, Dordrecht, Holland, Kluwer Academic Publishers Inc., pp. 317-324, 1991.

Spyridi, A.J. and Requicha, A.A.G., "Automatic Planning for Dimensional Inspection," *Manufacturing Review*, Vol. 6, No. 4, pp. 314-319, December 1993.

Spyridi, A.J. and Requicha, A.A.G., "Automatic Programming of Coordinate Measuring Machines," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, pp. 1107-1112, 1994.

Syslo, M.M, Deo, N. and Kowalick, J.S., *Discrete Optimization Algorithms with Pascal Programs*, Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1983.

Tandler, B., *Applying Geometric Dimensioning and Tolerancing to CMM Operation*, SME, Dearborn, MI, 1994.

Tao, L.G. and Davies, B.J., "Knowledge-Based 2 1/2D Prismatic Component Inspection Planning," Int. *Journal of Advanced Manufacturing Technology*, Vol. 7, Springer-Verlag London, pp. 339-347, 1992.

Trochu, F., "A Contouring Program Based on Dual Kriging Interpolation," *Engineering with Computers*, Vol. 9, pp. 160-177, 1993.

Trochu, F., Hammami, A. and Benoit, Y., "Prediction of Fibre Orientation and Net Shape Definition of Complex Composite Parts," *accepted for publication in the Journal of Composites Manufacturing*, 1995.

Tsui, H.T., Chan, M.H., Chu, K.C. and Kong, S.H., "Orientation Estimation of 3D Surface Patches," Computer Vision, *Graphics and Image Processing*, pp. 112-124, 1990.

Vafaeesefat, A., "Automatic Generation of NC Tool Path Programs for Multi-axis Mould Manufacturing," *Master Thesis*, Ecole Polytechnique, Montreal, 1994.

Walker, I. and Wallis, F., "Application of 3-D Solid Modelling to Coordinate Measuring Inspection," *Int. J. Mach. Tools Manufact.*, Vol. 32, No. 1/2, pp. 195-201, 1992.

Whitley, D., Starkweather, T. and Fuquay, D.A., "Scheduling Problems and Travelling Salesmen : The Genetic Edge Recombination Operator," *Proc. of Genetic Algorithms Conference*, pp. 133-140, 1989.

Wong, C.L., Menq, C.H. and Bailey, R., "Computer-Integrated Dimensional Inspection of Manufactured Objects having Sculptured Surface," *Advanced Manufacturing Engineering*, Vol. 5, pp. 37-44, January 1991.

Wong, E.K. and Fu, K.S., "A Hierarchical Orthogonal Space Approach to Three-Dimensional Path Planning," *IEEE Journal of Robotics and Automation*, Vol RA-2, No. 1, pp.42-53, March 1986.

Yau, H.T. and Menq, C.H., "An Automated Dimensional Inspection Environment for Manufactured Parts Using Coordinate Measuring Machines," *Int. J. of Prod. Research*, Vol. 30, No. 7, pp. 1517-1536, 1992.

Yau, H.T. and Menq, C.H., "Path Planning for Automated Dimensional Inspection Using Coordinate Measuring Machines," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, pp. 1934-1939, April 1991.

Zeid, I., *CAD/CAM Theory and Practice*, McGraw-Hill, Inc., 1990.

APPENDIX A

DIJKISTRA'S ALGORITHM

This algorithm determines the shortest path from a specified node *S* to another specified node *T* in a network *G*. Here, all edge weights are non-negative. *V* is the set of all nodes, and the algorithm as described in (Dijkistra, 1959 and Syslo et al., 1983) is presented in Figure A.1

The interference checking takes place at the loop level:

for every immediate successor *v* of *recent* **if not** *final(v)* **do**

which is replaced by

for every immediate successor *v* of *recent* **if not** *final(v)* **and not** *interference_order1(v)* **and not** *interference_order2(recent,v)* **do**

Remark

The array *pred* keeps track of the immediate predecessor of a node in the shortest path from *S* to *T*. At the end of the iterations the shortest path is given by the following sequence of nodes:

S, pred(pred(...)), ... , pred(pred(T)), pred(T), T

```
INITIALIZATION
        for all v ∈ V do
            begin
                dist(v) = ∞;
                final(v) = false;
                pred(v) = -1;
            end
        dist(S) = 0;
        final(S) = true;
        recent = S;

ITERATION
        while final(T) = false do
            begin
                for every immediate successor v of recent if not final(v) do
                    begin
                      newlabel = dist(recent) + distance(point(recent), point(v));
                      if newlabel < dist(v) then
                          begin
                              dist(v) = newlabel;
                              pred(v) = recent;
                          end
                    end
                    let y be the node with smallest temporary label, which is ≠ ∞;
                    final(y) = true;
                    recent = y;
            end
```

Figure B.1 Dijkistra's Shortest-Path Algorithm

APPENDIX B

BRANCH-AND-BOUND TRAVELLING SALESMAN OPTIMIZATION ALGORITHM

This algorithm (Syslo et al., 1984) is based on a tree search where at each step all possible solutions of the current problem are partitioned into two subsets: those that contain a specific edge *(i,j)* and those that do not. This branching is performed according to some heuristic which reduces the amount of search to be conducted for the optimal solution . After branching, lower bounds of the cost of each of the two subsets are computed. The solution space with the smaller lower-cost bound is chosen for the next search. This process is continued until a Hamiltonian cycle is obtained. Then only those subsets of solutions whose lower bounds are smaller than the current solution need to be searched.

When Applied to the Travelling Salesman Problem, the Branch and Bound algorithm operates reccursively on a graph where the cost of each edge is represented by the euclidian distance of the nodes. The graph is represented by a weight matrix *W*.

The algorithm may be decomposed into three main parts:

1. Reduction of the cost matrix associated with every node in the search tree.
2. Selection of the best edge in the search tree
3. Depth-first exploration of the search tree

```
function REDUCE(A);
    begin
        rvalue = 0;
        for i = 1 to size do
            begin
                rowred(i) = smallest element in ith row;
                if rowred(i) > 0 then
                    begin
                        subtract rowred(i) from every finite element in ith row;
                        rvalue = rvalue + rowred(i);
                    end
            end
        for j = 1 to size do
            begin
                colred(i) = smallest element in jth column;
                if colred(j) > 0 then
                    begin
                        subtract colred(j) from every finite element in jth column;
                        rvalue = rvalue + colred(j);
                    end
            end
            return rvalue;
    end
```

Figure B.1 Reduction of Matrix A

```
procedure BESTEDGE(A, size, r, c, most);
    begin
      most = ∞;
      for i = 1 to size do
        for j = 1 to size do
          if a_ij = 0 then
            begin
              minr = smallest entry in ith row, other than a_ij;
              minc = smallest entry in jth column, other than a_ij;
              total = minr + minc;
              if total > most then
                begin
                  most = total;
                  r = i;
                  c = j;
                end
            end
    end
```

Figure B.2 Selecting the Best Edge (*r*,*c*)

```
procedure EXPLORE(edge, cost, A);
  begin
    cost = cost + REDUCE(A);
    if cost < tweight then
      if edges = n - 2 then
          begin
            add the last two edges;
            tweight = cost;
            record the new solution;
          end
          else
            begin
              apply procedure BESTEDGE to find (r,c) the best edge to split the
              solutions on;
              let most be amount subtracted from row r and column c;
              lowerbound = cost + most;
              prevent cycle;
              newA = A - column c - row r;
              if lowerbound < tweight then
                begin
                  a_rc = ∞;
                  EXPLORE(edges, cost, A);
                  a_rc = 0;
                end
            end
          unreduce A;
  end
```

Figure B.3 Depth-First Exploration of Search Tree

APPENDIX C

DUAL KRIGING APPLICATION EXAMPLES

In this appendix some detailed applications of dual Kriging for curve, surface and solid interpolation are presented.

It is important to note that the parametrization of the surface/solid, i.e. the choice of the direction for each parameter, is very important for the solid (or surface) representation. Furthermore, degeneracy [e.g., when an isoparametric surface or curve is reduced to a point, or an isoparametric surface is reduced to a curve] is useful for the representation of complicated shapes as well as simple shapes, and closely depends on the symmetries in the object.

C.1 CURVE INTERPOLATION

C.1.1 Example 1

We will consider the simple case of interpolating a square curve (*P1 P2 P3 P4 P1*) (see Figure C.1).

The profile chosen for this curve is composed of a linear covariance and a constant drift. The parameters associated with the data points are:

$$s = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$$

The Kriging matrix is in this case (equation 6.9 without the derivatives terms)

$$K_A = \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & 1 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \\ \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & 1 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 1 \\ 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

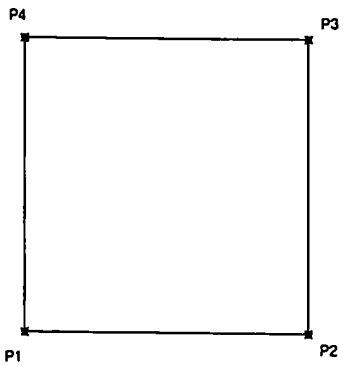


Figure C.1 Curve Interpolation with Position Data Points

The equation of the square is then (equation 6.19)

$$P(t) = [\,|s|\ |s-\tfrac{1}{4}|\ |s-\tfrac{1}{2}|\ |s-\tfrac{3}{4}|\ |s-1|\ 1\,]\,[K_A]^{-1} \begin{bmatrix} P1 \\ P2 \\ P3 \\ P4 \\ P1 \\ 0 \end{bmatrix}$$

or

$$P(s) = \begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} -4(|s| - |s-\frac{1}{4}| - |s-\frac{1}{2}| + |s-\frac{3}{4}|) + 1 \\ -4(|s-\frac{1}{4}| - |s-\frac{1}{2}| - |s-\frac{3}{4}| + |s-1|) + 1 \end{bmatrix}$$

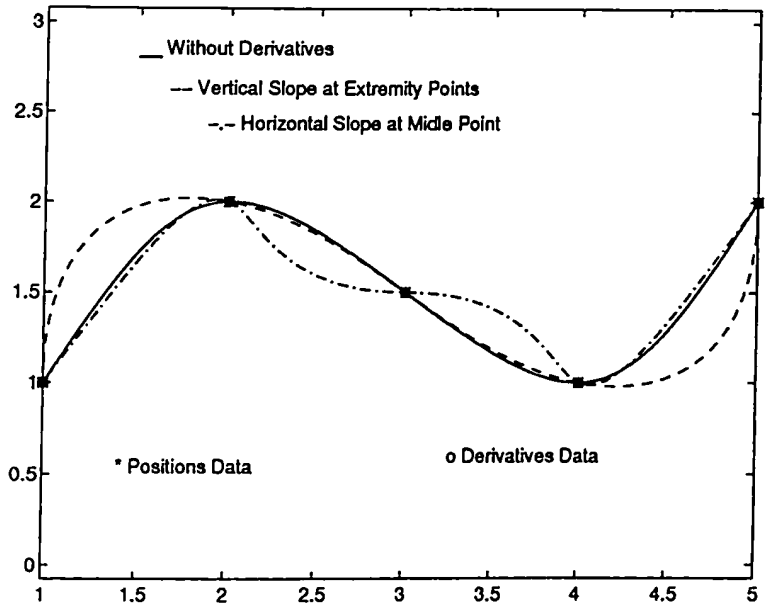


Figure C.2 Curve Interpolation with Position and Derivatives Data Points

C.1.2 Example 2

Figure C.2 presents an example of curve interpolation using parametric Kriging with derivatives data. The continuous line curve was interpolated using only position data (5 points). The dashed curve was interpolated with two vertical slopes at extremity points. Finally, the dash-dot curve was obtained with the data points of the first curve and a horizontal slope specified at the middle of the curve.

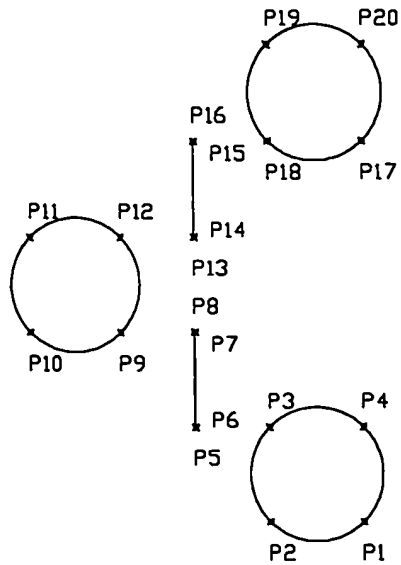


Figure C.3 Data Points in XZ plane

C.2 SURFACE INTERPOLATION

C.2.1 Interpolating a Grid of Data Points

The two examples were generated using the same set of data points (4×5 points) P_1-P_{20} (see Figure C.3). They differ only at the level of Kriging profiles. A trigonometric drift and a linear covariance combined with a trigonometric drift and a linear covariance were used for the first example (Figure C.4), and a trigonometric drift and a linear covariance combined with a constant drift and a linear covariance were used for the second example (Figure C.5).

The Kriging profiles of Figure C.4 are:

Profile A represents the circular shape ($P_1\ P_2\ P_3\ P_4\ P_1$)

Profile B represents the helical shape ($P_1\ P_5\ P_9\ P_{13}\ P_{17}$)

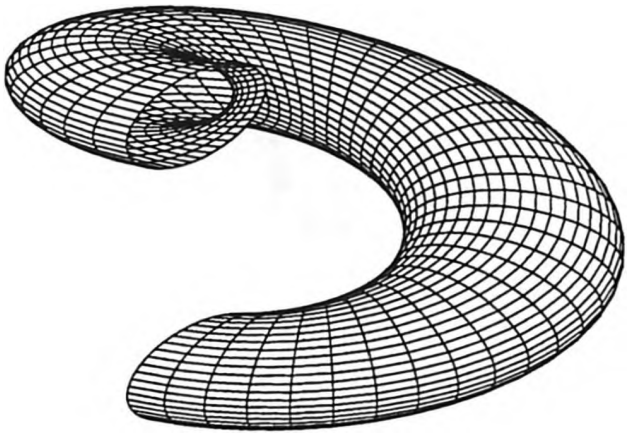


Figure C.4 Example 1 of Surface Interpolation

Note that the example in Figure C.4 could be obtained using only 3 data points for each circular profile.

The Kriging matrices are (equations 6.29 and 6.33)

$$[K_A] = [K_B] = \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & 1 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & 1 & -1 & 0 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 1 & 0 & -1 \\ 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

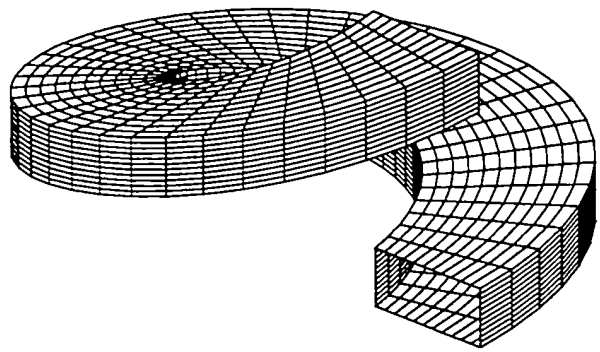


Figure C.5 Example 2 of Surface Interpolation from a Grid of Data Points

Matrix $[M]$ is in this case (6.37)

$$[M] = [K_A]^{-1} \begin{bmatrix} P_1 & P_5 & P_9 & P_{13} & P_{17} & 0 & 0 & 0 \\ P_2 & P_6 & P_{10} & P_{14} & P_{18} & 0 & 0 & 0 \\ P_3 & P_7 & P_{11} & P_{15} & P_{19} & 0 & 0 & 0 \\ P_4 & P_8 & P_{12} & P_{16} & P_{20} & 0 & 0 & 0 \\ P_1 & P_5 & P_9 & P_{13} & P_{17} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} [K_B]^{-1}$$

and (6.38)

$$[k_a(u)]^T = [|u| \ \ |u-\frac{1}{4}| \ \ |u-\frac{1}{2}| \ \ |u-\frac{3}{4}| \ \ |u-1| \ \ 1 \ \ \cos 2\pi u \ \ \sin 2\pi u]$$

$$[k_b(v)]^T = [|v| \ \ |v-\frac{1}{4}| \ \ |v-\frac{1}{2}| \ \ |v-\frac{3}{4}| \ \ |v-1| \ \ 1 \ \ \cos 2\pi v \ \ \sin 2\pi v]$$

As for Figure C.5, the Kriging matrices are (6.29 and 6.33)

$$[K_B] = \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & 1 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & 1 & -1 & 0 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 1 & 0 & -1 \\ 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad [K_A] = \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & 1 \\ \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 \\ \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} & 1 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 1 \\ 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Matrix $[M]$ is in this case (6.37)

$$[M] = [K_A]^{-1} \begin{bmatrix} P_1 & P_5 & P_9 & P_{13} & P_{17} & 0 & 0 & 0 \\ P_2 & P_6 & P_{10} & P_{14} & P_{18} & 0 & 0 & 0 \\ P_3 & P_7 & P_{11} & P_{15} & P_{19} & 0 & 0 & 0 \\ P_4 & P_8 & P_{12} & P_{16} & P_{20} & 0 & 0 & 0 \\ P_1 & P_5 & P_9 & P_{13} & P_{17} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} [K_B]^{-1}$$

and (6.38)

$$[k_a(u)]^T = [|u| \ \ |u-\tfrac{1}{4}| \ \ |u-\tfrac{1}{2}| \ \ |u-\tfrac{3}{4}| \ \ |u-1| \ \ 1]$$

$$[k_b(v)]^T = [|v| \ \ |v-\tfrac{1}{4}| \ \ |v-\tfrac{1}{2}| \ \ |v-\tfrac{3}{4}| \ \ |v-1| \ \ 1 \ \ \cos 2\pi v \ \ \sin 2\pi v]$$

C.2.2 Interpolating a Set of Continuous Curves (Skinning)

The third example shows a surface interpolated using the three curves ($C_1(v)$, $C_2(v)$, $C_3(v)$) shown in Figure C.6 (from top to bottom), one of which being only C^0. The surface obtained is represented in Figure C.6.

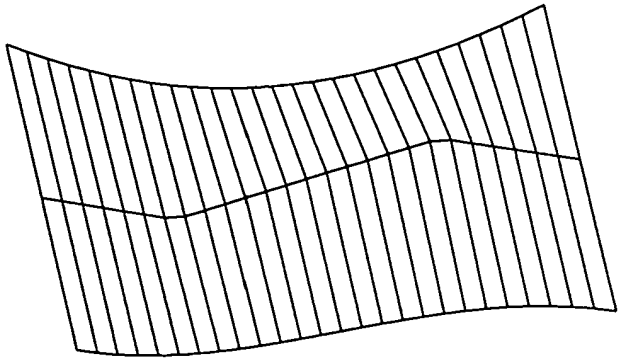


Figure C.6 The Curves to be Skinned

The Kriging profile utilized is a combination of a linear drift with a cubic covariance. The parameters associated with each curve are equidistant, and are $u = \{0, 1/2, 1\}$. The Kriging matrix becomes (equation 6.9):

$$K = \begin{bmatrix} 0 & \frac{1}{8} & 1 & 1 & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} & 1 & \frac{1}{2} \\ 1 & \frac{1}{8} & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 1 & 0 & 0 \end{bmatrix}$$

The equation of the surface is then (6.34)

$$S(u,v) = [\,|u|^3 \; |u-\tfrac{1}{2}|^3 \; |u-1|^3 \; 1 \; u\,][K]^{-1} \begin{bmatrix} C_1(v) \\ C_2(v) \\ C_3(v) \\ 0 \\ 0 \end{bmatrix}$$

where

$$C_1(v) = \begin{bmatrix} x_1(v) \\ y_1(v) \\ z_1(v) \end{bmatrix} = \begin{bmatrix} 1 \\ v \\ v(v-1) \end{bmatrix}, \quad C_2(v) = \begin{bmatrix} x_2(v) \\ y_2(v) \\ z_2(v) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2}(|v| - |v-1| + 1) \\ \frac{1}{4}(|v| - 2|v-\frac{1}{4}| + 2|v-\frac{3}{4}| - |v-1|) \end{bmatrix}$$

$$C_3(v) = \begin{bmatrix} x_3(v) \\ y_3(v) \\ z_3(v) \end{bmatrix} = \begin{bmatrix} -1 \\ v \\ v(v-\frac{1}{2})(v-1) \end{bmatrix}$$

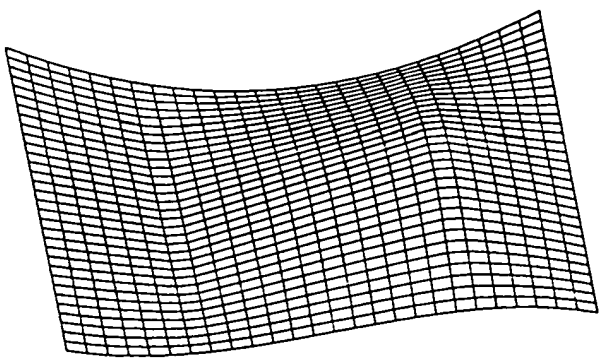


Figure C.7 Example of Surface Obtained by Skinning

C.3 SOLID INTERPOLATION

Two examples are presented. A helical solid (Figure C.8) and a three quarters of a cylinder with a cylindrical hole combined with a conical hole (Figure C.9).

C.3.1 Example 1

The helical solid in Figure C.8 is obtained by considering a linear profile C in addition

to the circular profile A and the linear profile B of a surface. The kriging matrix of profile A is similar to the kriging matrix of profile A of the surface in Figure C.4, and the kriging matrix of profile B is similar to the kriging matrix of profile A of the surface in figure C.5. A degenerated surface is generated by considering the centres (*P21 P22 P23 P24 P25*) of each section defined in Figure C.2.

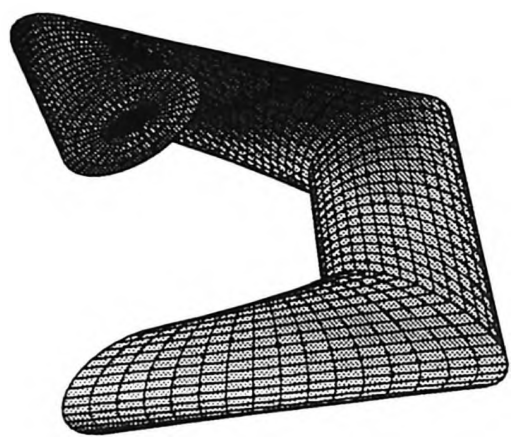


Figure C.8 Example 1 of Solid Interpolation

The linear profile C is defined by a linear covariance and a constant drift. Its kriging matrix is

$$[K_C] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

with

$$[M_1] = [K_A]^{-1}\begin{bmatrix} P_1 & P_5 & P_9 & P_{13} & P_{17} & 0 \\ P_2 & P_6 & P_{10} & P_{14} & P_{18} & 0 \\ P_3 & P_7 & P_{11} & P_{15} & P_{19} & 0 \\ P_4 & P_8 & P_{12} & P_{16} & P_{20} & 0 \\ P_1 & P_5 & P_9 & P_{13} & P_{17} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}[K_B]^{-1}$$

and

$$[M_2] = [K_A]^{-1}\begin{bmatrix} P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & 0 \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}[K_B]^{-1}$$

and

$$[k_a(u)]^T = [|u| \ |u-\frac{1}{4}| \ |u-\frac{1}{2}| \ |u-\frac{3}{4}| \ |u-1| \ 1 \ \cos 2\pi u \ \sin 2\pi u]$$

$$[k_b(v)]^T = [|v| \ |v-\frac{1}{4}| \ |v-\frac{1}{2}| \ |v-\frac{3}{4}| \ |v-1| \ 1]$$

$$[k_c(w)]^T = [|w| \ |w-1| \ 1]$$

C.3.2 Example 2

Thirty six *(4 x 3 x 3)* data points were used, two of the three profiles have a constant drift combined with a linear covariance, and the third is a combination of a trigonometric drift and a linear covariance.

Figure C.9 Example 2 of Solid Interpolation