1994

Collision-free automatic dimensional inspection using coordinate measuring machines

Dong-Keun Park
Iowa State University

Order Number 9518428

Collision-free automatic dimensional inspection using coordinate measuring machines

Park, Dong-Keun, Ph.D.

Iowa State University, 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Collision-free automatic dimensional inspection using coordinate measuring machines

by

Dong-Keun Park

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department: Industrial and Manufacturing Systems Engineering
Major: Industrial Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1994

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Overview

Today, the use of Coordinate Measuring Machines (CMMs) is wide-spread in industrial environments, where the inspection of geometry is important. A CMM can inspect a wide variety of workpieces by creating workpiece specific inspection programs. CMM inspection programs require careful planning for complex workpieces including the consideration of number and location of sample points, path planning, and probe qualification. Changes in design specifications can have a significant impact on CMM inspection programs, often requiring extensive development time for a new program.

In taking point measurements, the operator must teach the CMM both the inspection path through the points and the probe orientation for each point to avoid interference with the workpiece. Whether it is done manually or with a program, the operator must check for possible interference with the workpiece. The main objective is to derive the minimal number of workpiece orientations and the associated probe orientations, and to search the collision-free inspection path through the inspection points.

A setup is characterized by a fixed workpiece orientation with respect to the CMM coordinate system. Since workpiece setup interrupts the inspection process (and can require re-establishment of the datum reference frame), it is useful to touch as many points as possible in a given setup. Furthermore, if we can minimize the number of setups with respect to the workpiece orientation, then this contribution to the total inspection time should be minimized. This problem can be viewed as a minimal clustering problem.

With a fixed probe orientation, the probe axis is always coincident with the Z-axis of CMM. Therefore, the probe can touch the points on the workpiece surface with a given workpiece orientation if the length of probe is long enough to reach the points and there is no interference with the workpiece. If a probe has a multiple degree of freedoms, then the probe may be able to reach more points in a given setup.

The inspection path of a CMM on a part surface makes a significant contribution to the overall inspection time. Current practice shows little consideration of the total inspection path. Typically, the path is determined arbitrarily by an operator during development of a measurement program. Choices made by the operator usually consider the geometry of individual features without looking at the total path. In general, the inspection path is easily generated for relatively simple features that require few sample points. However, for more complex features or when the number of sample points are increased to ensure accuracy, the inspection time will be penalized as sample size increases. To reduce the inspection time, we need to provide the shortest overall travel path. One recognizes this problem immediately as the traveling salesman problem.

This research presents methodologies to solve the problem of determining workpiece and probe orientation for CMM inspection planning based on workpiece and probe geometry. The methodologies to obtain the safe and locally shortest path with respect to the workpiece geometry is also presented.

1.2 Research objectives

The primary objective of this research is to develop a set of algorithms for the automatic dimensional inspection of workpiece surface using a CMM. This research consists of the following three objectives.

1. A general methodology is needed for determining the accessibility of a probe to a workpiece.

2. The second objective is to determine the minimum number of workpiece setups, which is a minimal clustering problem, and to find the associated probe orientations for each workpiece setup.

3. The third objective is to generate a collision-free path through the inspection points with the shortest distance.

1.3 Organization of thesis

This thesis is organized into six chapters. In Chapter 1, we discuss existing problems in CMM inspection planning and suggest possible approaches to solve these problems. Chapter 2 gives a review of related research work in the area of accessibility analysis and path generation. Chapter 3 presents method for constructing the visibility map (VMAP) for an inspection point and the minimal clustering problem to find the optimal workpiece orientations and the related probe orientations. In Chapter 4, the swept volume of the probe model is introduced for collision detection and the collision-free path is generated by modifying the collide path segments.

Chapter 5 demonstrates the computer simulation results using the several test blocks. Finally, the findings and significance of this research are summarized in Chapter 6 and implications for future research are presented.

CHAPTER 2. LITERATURE REVIEW

2.1 Accessibility analysis

The notion of feature accessibility and a technique for computing accessibility cones was first proposed by Spyridi and Requicha [46]. Accessibility cones were used to compute the angle in which the probe can approach the workpiece. With this method, the accessibility cone is divided into two portions: the local accessibility cone (LAC) and the global accessibility cone (GAC). The LAC is the accessibility cone corresponding to a specific feature, while the GAC is a complete accessibility cone which considers all other features. Due to the computation complexity of the applied algorithm (called a Minkowski algorithm), this method cannot generate accessibility cones for a general surface.

Another approach to perform accessibility analysis is to utilize a visibility map (VMAP). Hilbert's [17] approach maps a surface onto the unit sphere by calculating the normal vector at each point on the surface. These vectors will fall within a unit sphere. The union of these vectors is defined as a Gaussian Map (GMAP) for a given surface. Similarly, a VMAP for a point on the workpiece surface is that portion of the unit sphere surface defined by a set of unit length vectors with common origin at the inspection point. The set of vectors is constrained to those that do not intersect the workpiece geometry. Figure 2.1 illustrates the relationship

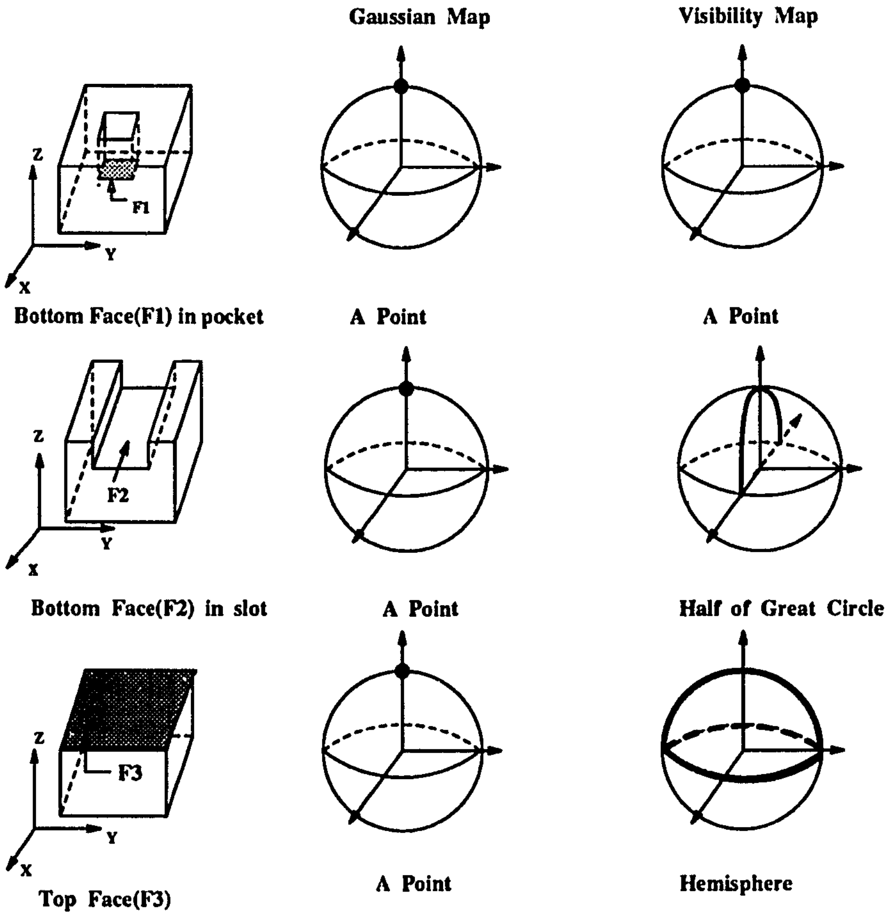


Figure 2.1: Relationship between GMAP and VMAP

between GMAP and VMAP for an entire surface. For the bottom face of pocket, the GMAP and VMAP both represent by a point on a unit sphere. However, the slot has different representations for the bottom face. The GMAP of the bottom face F_2 is a point, while the VMAP is the great half circle on X-plane. For the flat face F_3, the GMAP is also a point, but the VMAP is represented by a hemisphere.

A similar problem can be found in the determination of tool-approach direction for NC machining. A machine tool and the CMM probe share a similar goal of making contact with the workpiece without touching the obstacles. Woo and Turkovich [55], and Chen and Woo [8] described the construction of the VMAP for the geometry of a cutter tool and the workpiece and used them to find the workpiece setup as the common intersection of VMAPs. For instance, the axis of a ball-end mill can deviate up to 90 degrees on each side from the surface normal (i.e., GMAP). In this case, the VMAP of a point is a hemisphere and the VMAP of a surface is defined as the intersection of hemispheres associated with all points in a given GMAP. It should be noted that the VMAP constructed directly from the GMAP represents the local visibility of the corresponding surface. Tseng and Joshi [50] determine the bound of the tool approach angle for machining Bézier curves and surfaces. They presented the algorithm based on the subdivision to analyze Bézier curves and surfaces to determine feasible tool-approach directions. However, they only considered a three axis NC machine.

Haghpassand and Oliver [12] suggested a method to find the optimal workpiece orientation for a three-axis milling operation, in which the orientation is optimized such that the angle between the nominal surface (i.e., design specification) normals and the milling tool axis is minimized. They also applied the spherical geometry to determine whether the workpiece can be milled completely in a single set-up on a three-axis milling machine. Tang et al. [48] proposed a method for finding the optimal workpiece orientation for a 4- and 5-axis machining. They defined the optimal workpiece orientation as the maximum VMAP intersection area in which the tool can approach the maximum number of surfaces that can be machined at a single setup.

2.2 Collision-free path generation

Most CMMs can be viewed as a Cartesian robot with the probe tip acting as an end-effector. Thus the issue of a collision-free path for a CMM is similar to that of robot manipulators. A great deal of research has been done in collision-free path planning for robot manipulators in a known environment. Boyse [3] classified interference detection methods for solid objects as either static and dynamic. Cameron [5] proposed three different collision detection methods; multiple interference checking, swept volume, and four dimensional interference checking.

Udupa [51] first formulated the obstacle avoidance problem in terms of an obstacle transformation that treats the moving object as a point. Generalization of these obstacle transformation techniques and a review of related work can be found in [31]. To simplify the collision detection and avoidance problem, Configuration Space (C-Space) is widely used to perform path planning for robot manipulators [31]. In C-Space, obstacles are approximated by a simple polyhedra and the trajectory of a reference point on the moving object is only considered instead of the complete object so that interference is easily detected. Lozano-Perez and Wesley [30] show that the shortest-path for a polygon amidst polygonal obstacles can be solved using Dijkstra's shortest path algorithm applied to a certain visibility graph.

The collision-free inspection path for CMMs can be approached in a similar manner. Yau and Menq [56] performed collision-free path planning for CMMs by first creating an initial path without interference checking and then modifying the path by checking the interference between the probe and the part surface on individual path segments. Actually, the probe has three different components: probe head, probe, and stylus. Thus, for correctness, the interference check between a part feature and each component of the probe must be done before searching the inspection path. When a collision occurs, then the inspection path is heuristically modified, but the travel length of the path is not considered.

Finally, Lee et al. [26] suggested the optimal probe path for a sculptured surface. The initial probe path was determined by a traveling salesman algorithm based on the Euclidean distance between two points. For each path segment, a new guided point was introduced to avoid the collision of the probe tip with the workpiece along the path. However, they did not consider the probe orientation and its dimension.

CHAPTER 3. ACCESSIBILITY ANALYSIS

3.1 Overview

To avoid probe collision, workpiece setup and probe orientation must be established in the inspection plan. Figure 3.1 shows a sequence of point measurements using a probe, where the probe can touch the points with and without interference with respect to the geometry of the workpiece.

To determine a workpiece setup that is collision-free, the accessibility of the probe becomes an important factor. Let us assume that the probe is abstracted as a straight line and has a fixed orientation. Based on the geometry of the probe and the workpiece, accessibility is defined by the bounded space (which we represent as a *Visibility Map* (VMAP)) in which the probe can access a target point without

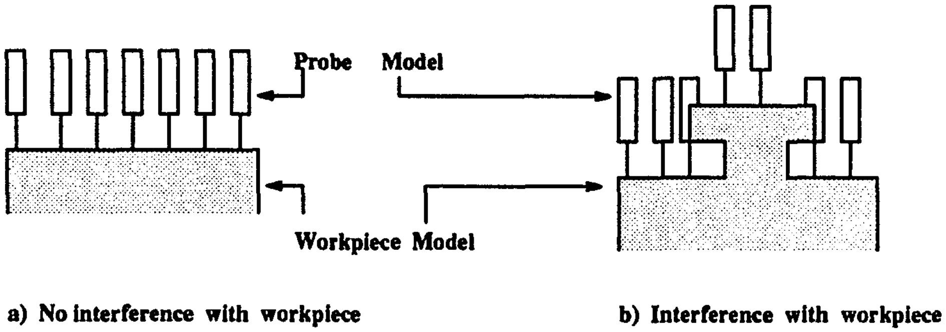


Figure 3.1: Probe interference with respect to the geometry of workpiece

interference with the workpiece.

A VMAP is defined as a set of unit length vectors representing points on a unit sphere visible to a sample point. The mapping of the unit vectors on a unit sphere can be represented by a *spherical cap*, which is a subset of the surface of the sphere. A spherical cap cannot be more than a hemisphere since the viewing direction cannot pass below the tangent plane at an inspection point. This is known as the *visibility constraint* [8].

First we present a new generalized method for constructing a VMAP for a given point on the workpiece surface. Starting with an arbitrary small hemisphere as an initial VMAP, the radius of the hemisphere is increased iteratively using a fixed step size. The choice of the step size depends on the desired level of accuracy.

We also present a modified algorithm to generate a VMAP for a given point based on a discretized approximation of the workpiece surface. Specifying the visible vertices, we can construct a polyhedral cone based on the visible vertices and the inspection point. A VMAP is obtained from the intersection between the polyhedral cone and a unit sphere. Therefore, the number of steps to generate a VMAP becomes a function of the number of visible vertices.

If there is a common intersection among VMAPs, the corresponding points are accessible by a probe in a single workpiece setup. Assuming a fixed probe orientation (i.e., one setup per probe orientation), the number of workpiece setups to inspect all the points can be determined by the number of VMAP intersections. We describe an algorithm to determine the minimum number of VMAP intersections (called a *minimal clustering problem* [46, 55]) and, at the same time, minimize the travel distance through the sample points. Unfortunately, the minimal clustering problem is known to be NP-complete [46]. In this study, we apply the simulated annealing algorithm to obtain a near optimal solution for this problem. The shortest path for a set of sample points within each set-up can be viewed as a traveling salesman problem which is also known to be an NP-complete problem. Again, simulated annealing is applied to obtain the shortest travel distance. To achieve these two goals simultaneously, we present a *multi-echelon optimization* method. The constrained workpiece orientation is discussed in the last section.

3.2 Visibility map (VMAP) generation

3.2.1 Visibility

Visibility is an important concept in computational geometry. In general, two points are visible to each other if they can be joined by a line segment (i.e., a visible ray) that does not intersect the geometry. Suppose p is a point on a workpiece surface and q is the probe tip. A probe tip q can reach a point p via a straight line if p and q are visible to each other. For a point to be visible by a probe, its orientations are bounded by a set of visible rays from the point. This set of rays forms the visibility cone [42]. As shown in Figure 3.2, the probe axis could not deviate from the surface normals by more than 90°. This was defined by Chen and Woo [8] as the visibility constraint.

Let $v(p)$ be a set of visible rays emanating from a point p, l be the ray with the origin p and W be the workpiece model. Then the set of visible rays for a point is given by

$$v(p) = \{l \mid \angle(l, N_p) \leq \frac{\pi}{2} \quad and \quad l \cap W = \emptyset\} \tag{3.1}$$

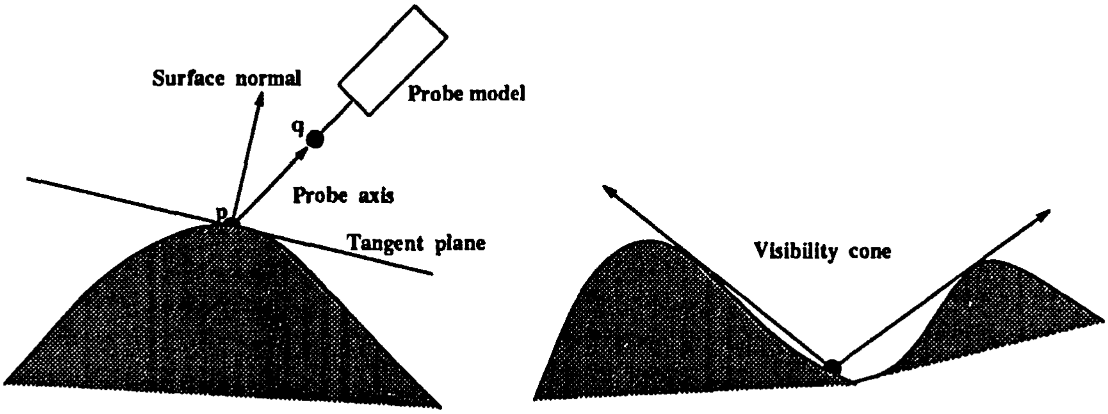


Figure 3.2: Visibility for a point on the workpiece surface

where N_p is the unit normal vector at a point p. A union of the set of visible rays, $v(p)$, is denoted as a visibility cone which is represented by

$$V\_cone = \{l \quad | \quad l \in v(p)\} \tag{3.2}$$

Any direction on or inside the visibility cone could be a potential probe orientation. Spyridi and Requicha [46] use the local and global accessibility cone to compute the accessibility of surface features in planning the dimensional inspection of workpieces by Coordinate Measuring Machines (CMM). The local accessibility cone (LAC) is the accessibility cone corresponding to a specific feature, while the global accessibility cone (GAC) is the accessibility cone which considers all other features. Due to the complexity of the applied algorithm (called the Minkowski algorithm), this method cannot generate accessibility cones for a general surface. They only described how to compute LACs and GACs for planar and quadric surfaces. For more general surfaces, no exact algorithm is known. In this research, we propose a VMAP to determine the accessibility of the probe to each sample point on any kind of surface.

3.2.2 Spherical geometry

By definition, a visibility map is a portion of the surface of a sphere. Therefore, a spherical algorithm is applied to derive the VMAP using two dimensional spherical geometry. A unit sphere, S^2, is defined by a set of points such that

$$S^2 = \{\mathbf{p} \in E^3 \quad | \quad |\mathbf{p}| = 1\} \tag{3.3}$$

Terminology associated with spherical geometry is described in the computational geometry literature (e.g., Preparata and Shamos [40]) and in related research (e.g., Chen and Woo [8] and Haghpassand and Oliver [12]). The definitions of the objects and important concepts are briefly reviewed.

- Points: A point p in S^2 is a unit vector in E^3.
- Line segment: The line segment joining two distinct points p and q in S^2, denoted by $\overline{pq}$, is the shorter of the two arcs in the great circle containing p and q.
- Hemisphere: The surface of the sphere is partitioned into two hemispheres by a plane which contains the origin.
- Spherical polygon and spherical cap: A spherical polygon is a closed path on the surface of the sphere of connected ordered line segments $\overline{ab}, \overline{bc}, \cdots, \overline{pq}, \overline{qa}$ which do not cut across themselves. A closed continuous path on the sphere is called a spherical cap.
- Gaussian map (GMAP): A Gaussian map is the intersection of outward normals of a surface with a unit sphere. The set of the unit normal vectors on a surface

is called a Gaussian image of the surface, and is denoted by $GI(F)$. A GMAP of the surface F is represented by

$$GMAP(F) = \{q \quad | \quad q = N_p \cap S \quad for \quad N_p \in GI(F)\}$$

where S is a unit sphere and q is an intersection point on a unit sphere.

In the next section, we show how to construct the visibility map.

3.2.3 Construction of VMAP

By the visibility constraint, the probe axis cannot deviate more than 90^o. Thus, the accessible area to the point could be represented by some portion of a hemisphere depending on the geometry of the workpiece. For example, the collection of the visible rays to a sample point on a half-plane are simply approximated by the hemisphere as shown in Figure 3.3. For any point within a slot (also shown in Figure 3.3), the visible rays are limited by the obstacles around the sample point and the accessibility area must be less than the hemisphere.

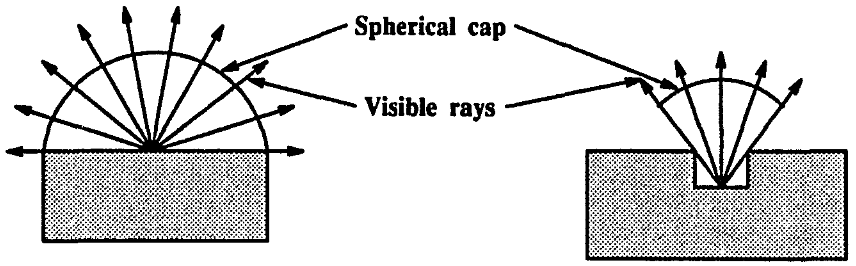


Figure 3.3: Visible rays to the sample point

A VMAP can be found by applying the hemisphere with the center as the coordinate of the inspection point. The base of the hemisphere is located such that the tangent plane is at the inspection point.

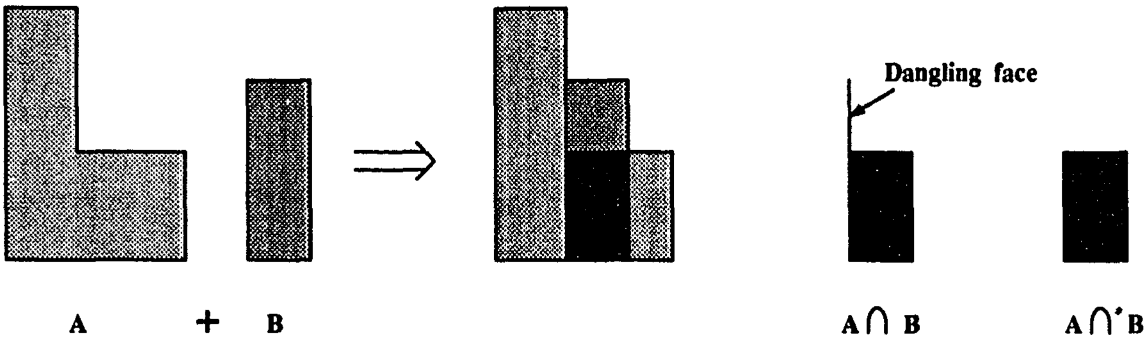


Figure 3.4: The regularized Boolean intersection of two objects

In this study, the relationship between the objects is represented by the regularized Boolean operation (Hoffman [18] and Requicha [41]). The operations are the regularized union, denoted $\cup^*$: regularized intersection, denoted $\cap^*$: and regularized difference, denoted $-^*$. They differ from the conventional (Boolean) set operations in that the result is the closure of the operation on the interior of the two objects (see mathematical definition in Appendix A), and they are used to eliminate *dangling* objects from the result of operation. For example as shown in Figure 3.4, they are not algebraically closed under the conventional set operation since there is a dangling face, but they are closed under the so-called regularized Boolean operations.

3.2.3.1 General approach For an initial hemisphere H_0 with arbitrary small radius r_0, a VMAP will be a hemisphere if the following condition is satisfied

$$H_0 \cap^* W = \emptyset \tag{3.4}$$

When we increase the radius of the hemisphere iteratively, equation 3.4 may not be satisfied because of the geometry of the workpiece (see Figure 3.5). We can obtain a spherical cap by subtracting the workpiece model from the generated hemisphere H_i

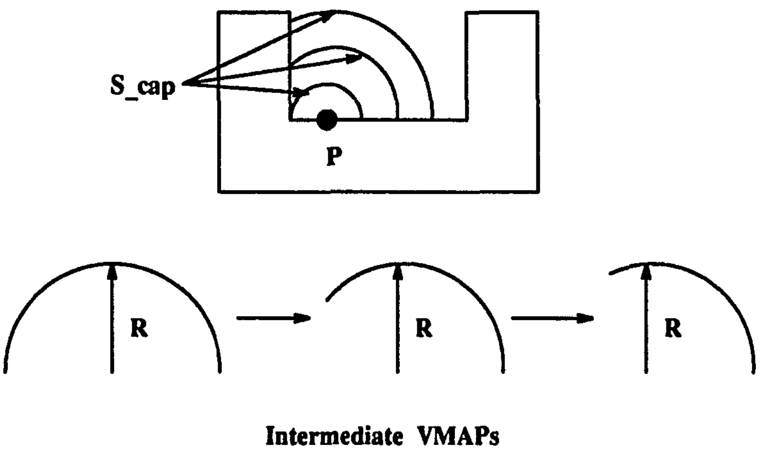


Figure 3.5: Generation of intermediate VMAPs using the spherical caps

such that

$$S\_cap = H_i -^* W \tag{3.5}$$

where S_cap means the spherical cap.

To update the VMAP, we need to generate the intermediate VMAP using the spherical cap. As shown in Figure 3.5, an intermediate VMAP can be generated by offsetting the spherical cap along the normal by an offset distance which is the difference between the radius of the smallest enclosing sphere (or simply, sphere) and the current radius of hemisphere. Offset surfaces are expanded or contracted versions of an original object (Rossignal and Requicha [43]). We can construct the offset surface by displacing each point on the spherical cap by a distance r along the unit normal at each point.

The offset surface, $S\_cap \parallel r$, becomes the current VMAP. The VMAP is updated by computing the intersection between the previous and the current VMAPs such

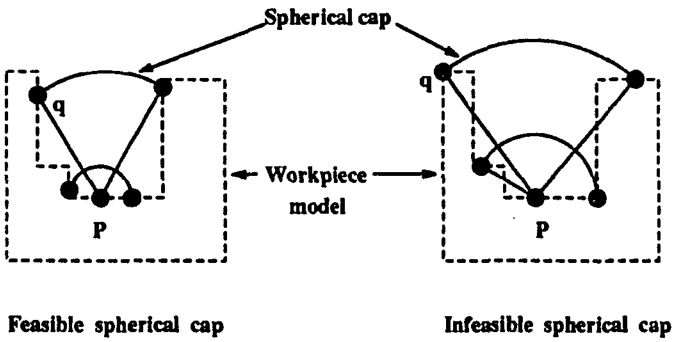


Figure 3.6: Feasibility test of spherical caps

that

$$VMAP_p = VMAP_p \cap^* VMAP_c \tag{3.6}$$

where $VMAP_p$ is the previous VMAP and $VMAP_c$ is the current VMAP. We increase the radius of hemisphere iteratively until some stopping conditions are met.

3.2.3.2 Feasibility test The generated spherical cap represents the visible area of a inspection point P if the visible condition is satisfied. That is, if any point q on the spherical cap can reach a point P via a straight line, the generated spherical cap is feasible. The feasibility of a spherical cap is easily checked by using a spherical cone as shown in Figure 3.6. The boundary of the spherical cap is used to generate the spherical cone. If we connect the straight lines from the point to the boundary of the spherical cap, it becomes the spherical cone as shown in Figure 3.7. If the spherical cone does not intersect with the workpiece, then the spherical cap is said to be feasible. Figure 3.6 illustrates feasible and infeasible spherical caps.

The spherical cone is easily constructed using the boundary of the spherical cap. Let $L(p, q)$ be the half-line from the inspection point p to any point q on the

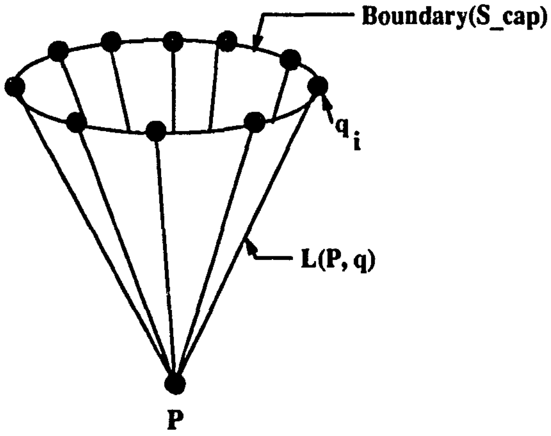


Figure 3.7: Construction of Spherical cone

boundary curve of the spherical cap. If we collect an infinite number of half-lines, l_i, with the common end point **P**, it results in the spherical cone as shown in Figure 3.7. Therefore, the spherical cone is represented by

$$S\_cone(P) = \{l \quad | \quad l = L(P, q_i), \quad q_i \in Boundary(S\_cap)\}$$

The spherical cap is feasible if

$$S\_cone(p) \cap^* W = \emptyset \tag{3.7}$$

Otherwise, the spherical cap is infeasible.

3.2.3.3 Update VMAP We need to update the VMAP continuously until the stopping condition is met. There are two possible states: the feasible state and the infeasible state. If the spherical cone does not intersect with the workpiece, then the cone is in the feasible state.

In the feasible state, we offset the generated spherical cap by the distance r which is the difference between the radius of a sphere and hemisphere H_i. The offsetting spherical cap ($S\_cap \parallel r$) becomes the current VMAP (called a $VMAP_C$). Since the size of VMAP is continuously decreased in the feasible state, we just update the VMAP as the $VMAP_C$. In the feasible state we return to the first step by increasing the radius of the hemisphere.

If the generated spherical cone intersects with the workpiece, the cone is in the infeasible state. We must keep reducing the radius of the hemisphere and do a feasible test until the cone satisfies the feasibility condition. The step size is reduced by a half (i.e., a binary search). If the spherical cap satisfies the feasibility condition, then we update the VMAP at this point and the step size returns to an initial step size, Δ. We continuously increase the radius of the hemisphere and update the VMAP by surface-surface intersection between $VMAP_P$ and $VMAP_C$, but the feasibility test is not performed until the status turns into the feasible state. If $VMAP_C$ becomes the subset of $VMAP_P$, then the status turns into the feasible state.

3.2.3.4 Example As shown in Figure 3.8, the hemisphere $H(1)$ intersects the workpiece. Subtracting the workpiece from the hemisphere $H(1)$, we obtain the spherical cap. The spherical cone for the feasibility test is easily generated by connecting the lines from the point to the boundary of spherical cap which is represented by $S(1)$. Since $S(1)$ has no interference with the workpiece, the offset spherical cap is generated. This is an initial VMAP.

However, the generated spherical cone $S(j+1)$ interferes with the workpiece and therefore is in the infeasible state. We must reduce the radius of the hemisphere and generate the spherical cone again to check for feasibility. The spherical cone $S(m+1)$ is feasible so the offset spherical cap is generated and VMAP is updated. From this

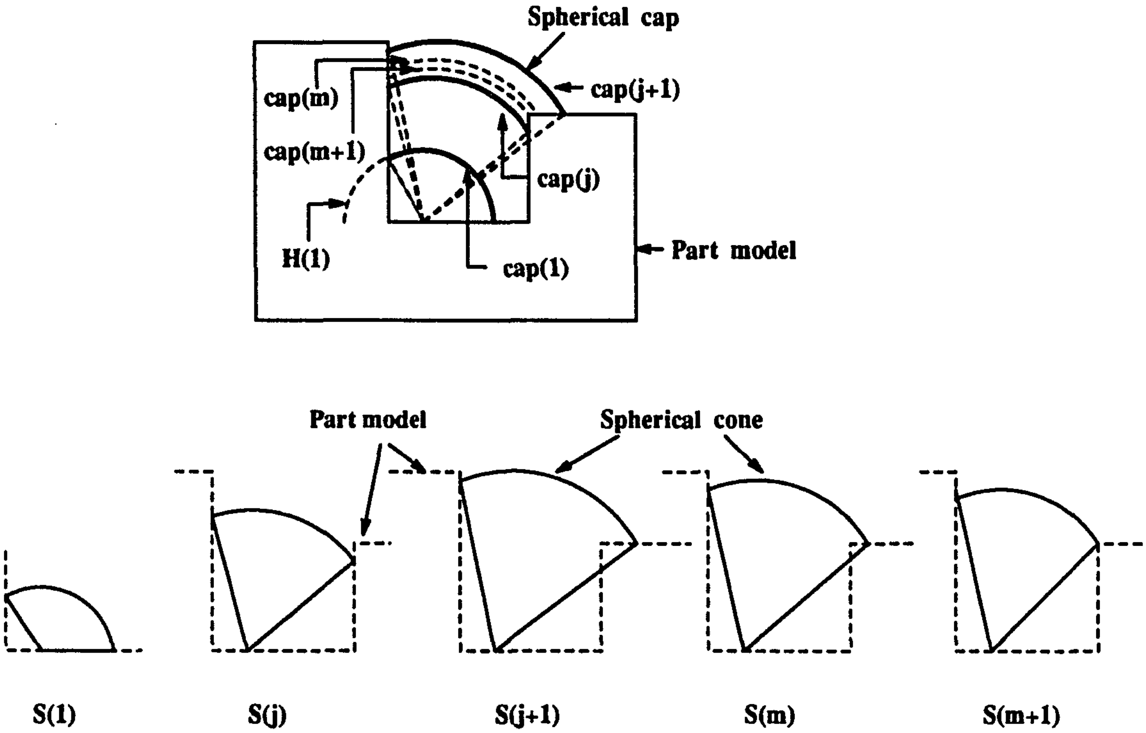


Figure 3.8: Construction of VMAP on 2-D space and the spherical cones

point, we return to increase the radius of hemisphere with the initial step size and try to update the VMAP continuously.

3.2.4 General algorithm

This section presents a general algorithm to generate the VMAP for each point on any general surface. This algorithm was implemented in the C language using the Shapes geometric computing system [45] on a Silicon graphics workstation. In what follows, we summarize the procedure for generating a VMAP.

1. Initialization

   - Build the workpiece model (W).
   - Choose the sample points on the workpiece surface.
   - Determine the smallest enclosing sphere (*SPHERE*) with center at the sample point with the radius R.
   - Set the update condition as the feasible state.
   - Set the initial radius of hemisphere, $r = r_0$.
   - Set the initial step size $\Delta = \Delta_0$.

2. VMAP construction

   LOOP 1:

   - step 1: Generate a hemisphere (H_i) with the radius r.
   - step 2: Generate a spherical cap:

$$S\_cap = H_i -^* W \tag{3.8}$$

- step 3: Check the update condition
  - If it is a feasible state, then go to step 4.
  - If it is an infeasible state, then go to step 5.
- step 4: Feasibility check
  - If $S\_cone \cap^* W = \emptyset$(i.e., feasible), then
    * $r = r + \Delta$
    * Go to step 5
  - Otherwise (i.e., the cone is in infeasible state),

    LOOP 2:

    (a) $r = r - \Delta$ where $\Delta = \frac{\Delta}{2}$

    (b) Generate a hemisphere (H_i) with the radius r.

    (c) Generate a spherical cap.

    (d) Check for feasibility.
      * if $S\_cone \cap^* W = \emptyset$,

        $\Delta = \Delta_0$

        Go to step 5
      * Otherwise, Go to (a).
- step 5: Update VMAP
  - $VMAP_c = S\_cap \parallel dist$

    where $S\_cap \parallel dist$ represents the offset spherical cap and $dist = R - r$.
  - If feasible state: $VMAP_p = VMAP_c$
  - If infeasible state: $VMAP_p = VMAP_p \cap^* VMAP_c$

- step 6: If $VMAP_C \subset VMAP_P$, then the cone is in the feasible state.
- step 7: Termination check.
  - If the stopping condition is met, then end.
  - Otherwise, $r = r + \Delta$ and return to step 1.

3. Stopping condition:
   - *dist* is less than the arbitrary small value ϵ,

     where $dist = R - r$.

3.2.5 Test block

The test block shown in Figure 3.9 has a rectangular pocket within a rectangular box. If we choose an inspection point on the bottom of the pocket, the intermediate VMAPs are shown in Figure 3.10. The initial VMAP in this figure is a hemisphere itself, but the area of VMAP is diminished as the process iterates.

3.2.6 Modified VMAP generation

The general algorithm presented in the previous section can generate the VMAP for each point on any surface, but the computation time is large since it could generate a large number of iterations and in each generation a surface-surface intersection (which is computationally intensive) between the offset spherical cap and a sphere is calculated. We can increase the efficiency of the algorithm by minimizing the total number of intersection calculations (i.e., minimize the number of iterations). This can be done by using the *visible vertices* from the part model to form a boundary for the visibility cone. A visible vertex is defined as a vertex visible from the inspection point.

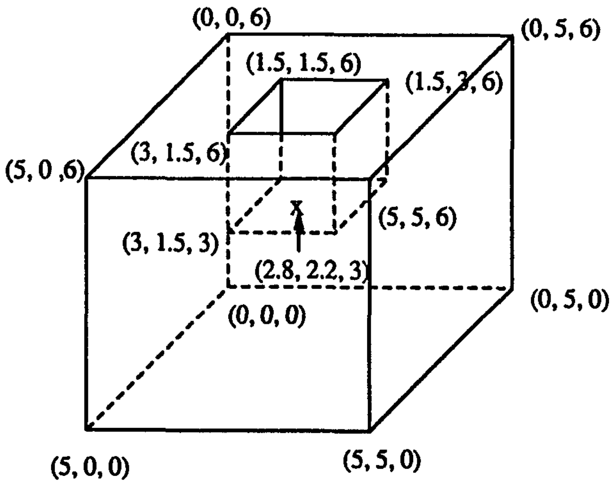


Figure 3.9: Test block

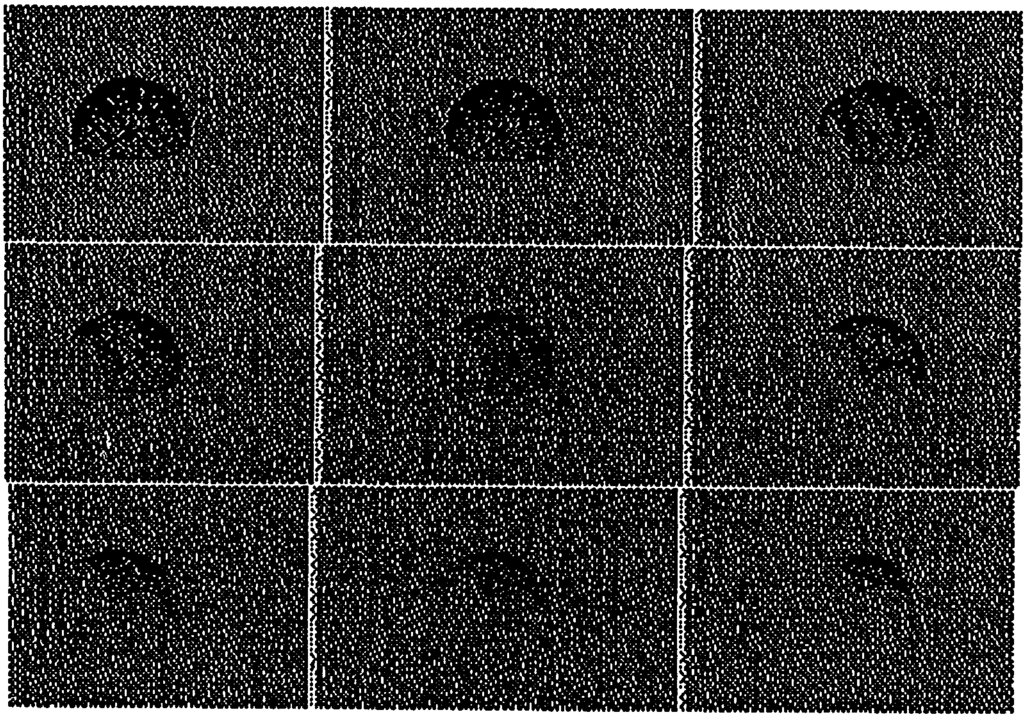


Figure 3.10: Development of VMAPs

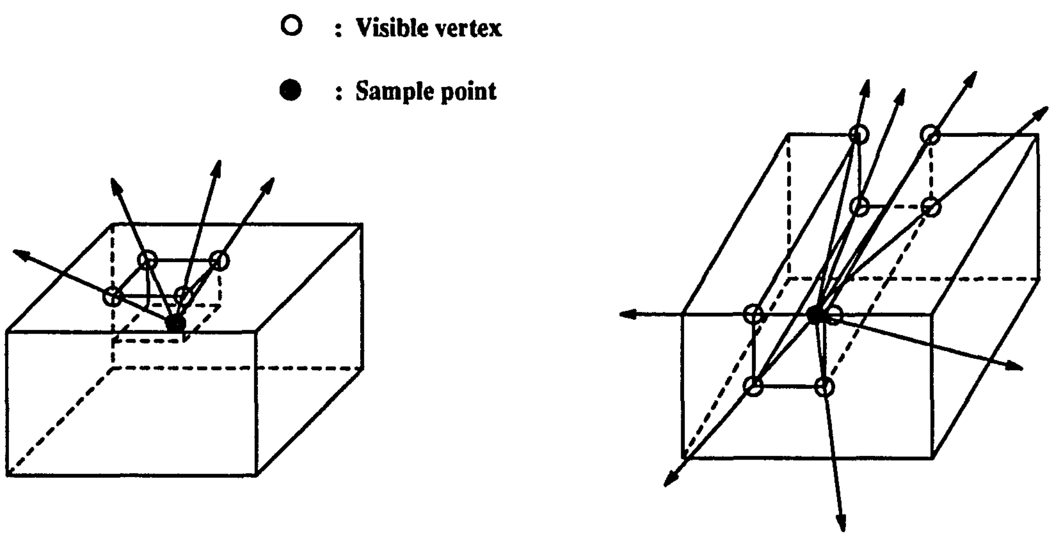


Figure 3.11: Selection of visible vertices from the polyhedral parts

For a polyhedral part model, the visible vertices are easily determined by checking the intersection between the part model and the rays, which are the connecting lines from the inspection point through the vertices. However, we need to discretize the workpiece surface to determine the visible vertices for the general workpiece. After discretizing the workpiece surface, the visible vertices are determined in the same way as the polyhedral part model. For example as shown in Figure 3.11, there are just four visible vertices for the rectangular pocket, while the number of visible vertices are eight for the slot.

Let l_i be the ray with origin at the inspection point p that contains the i^{th} visible vertex. The spherical polygon (SP) on a unit sphere can be represented by the relationship between the unit sphere and the set of lines. The vertices of SP are determined by the intersection between the unit sphere and the set of lines. That is,

$$p_i = l_i \cap^* S$$

where p_i is the vertex of SP and S is the unit sphere. The line segment, denoted by $e_{ij} = \overline{p_i p_j}$, between two adjacent vertices is obtained from the intersection between the unit sphere and the half-space containing two visible vertices and the origin point. The resulting spherical polygon on a unit sphere represents a VMAP of a point on a surface. Therefore, the number of iteration to compute the VMAP becomes a function of the number of visible vertices. The computation of the VMAP for the general surface is more complicated, but the process is the same as the polyhedral part.

To generate the spherical polygon, we need to introduce a tetrahedron, TH, which can be used to construct the polyhedral cone, PC, based on the visible vertices and the inspection point. Each tetrahedron contains three visible vertices and the inspection point as an apex of the tetrahedron. Figure 3.12 shows the generation of the tetrahedra and the polyhedral cone which is the union of the tetrahedra, TH_1 and TH_2.

A tetrahedron formed by taking a collection of four planar triangular pieces and joining them along their edges, is defined as *piecewise flat surface* [34]. To represent the tetrahedron, we describe each face separately and keep track of which edges are adjoining. This kind of representation is called an *atlas*. Figure 3.13 shows the atlas of a tetrahedron.

Assuming that there are N visible vertices, then there are (N-2) non-intersecting tetrahedra. The union of these tetrahedra forms the polyhedral cone. The spherical polygon on a unit sphere is simply generated by the intersection between the unit sphere and the polyhedral cone.

The modified algorithm is given as follows;

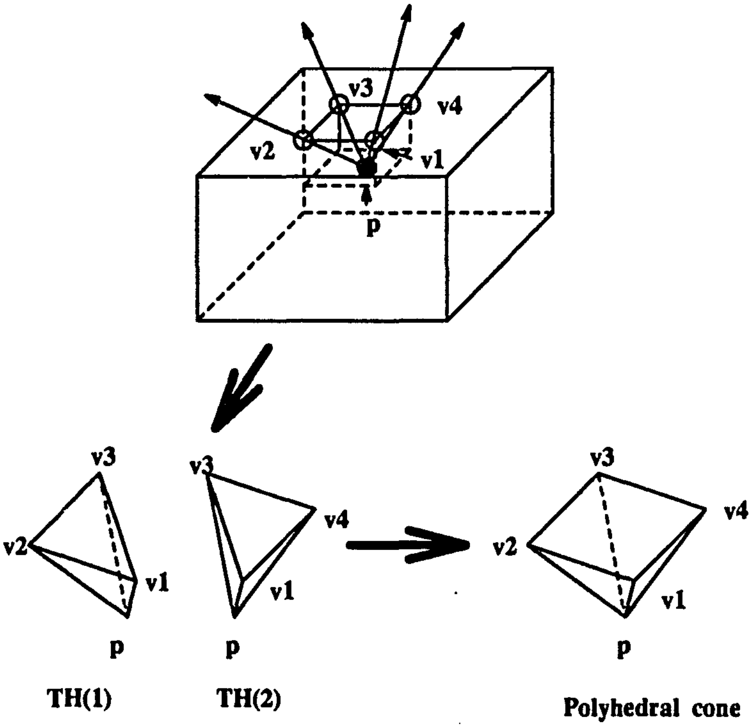


Figure 3.12: Generation of tetrahedra and polyhedral cone for a pocket

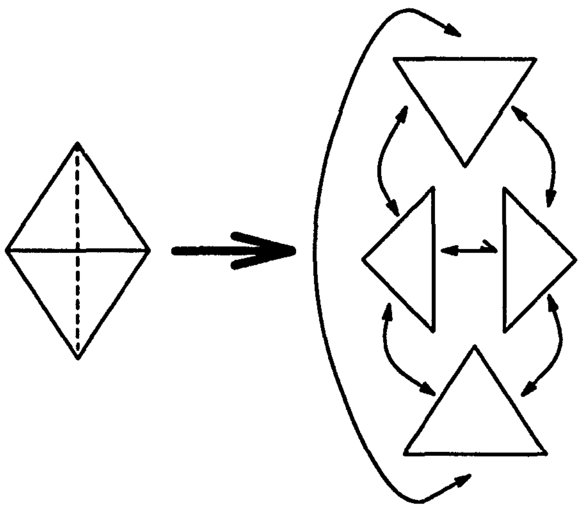


Figure 3.13: Topological atlas of a tetrahedron

1. Initialization

   - Discretize the workpiece model (W) to create all vertices.
   - Set the unit sphere (S).

2. Determine the visible vertices

   - Generate the rays from the inspection point to each vertex.
   - Check for the intersection between the workpiece and rays.
     - If it is a point, then the corresponding vertex is visible.
     - Otherwise, the vertex is invisible.

3. Generate VMAP

   - Generate the tetrahedra, $TH(i), \quad i = 1, \cdots, N-2$.

     where N is the number of the visible vertices.
   - Generate the polyhedral cone.

$$PC = \cup \quad TH(i), \quad i = 1, \cdots, N-2$$

   - Generate the VMAP

$$VMAP = PC \cap^{*} S$$

Since there is no surface-surface intersection, the computational effort to generate the VMAP is greatly reduced. The number of iterations in the modified algorithm is proportional to the number of vertices on a workpiece. Although a general workpiece surface may have a large number of vertices by discretization of the surface, the number of vertices could be adjusted by the relationship between the level of accuracy and the computation time.

3.2.7 Relationship between VMAPs: Adjacency matrix

The relationship between VMAPs are represented by the adjacency matrix in which the element of matrix is called a indicative function $I(i,j)$, where $I(i,j) = 1$ indicates that VMAP(i) and VMAP(j) intersect each other and $I(i,j) = 0$ if there is no intersection. This matrix is used to check the inter-relationship between VMAPs and apply a multi-echelon optimization method which is described in the next section. Figure 3.14 shows the 2D intersection of the VMAPs and Table 3.1 shows the adjacency matrix of this figure.

This adjacency matrix has some properties. The notation and the properties of the matrix are stated as follows:

- Notation
  - V_i: i^{th} VMAP
  - **A**: Adjacency matrix
  - n_i: i^{th} node which represents the i^{th} VMAP
  - Indicative function: $I(i,j)$

$$\begin{aligned} I(i,j) &= 1 \quad if \quad V_i \cap V_j \neq \emptyset \\ &= 0 \quad if \quad V_i \cap V_j = \emptyset \end{aligned}$$

- Properties of matrix **A**
  - Square matrix
  - Symmetric
  - If $I(i,j) = 1$, then (i,j) pair of VMAPs has a common intersection area

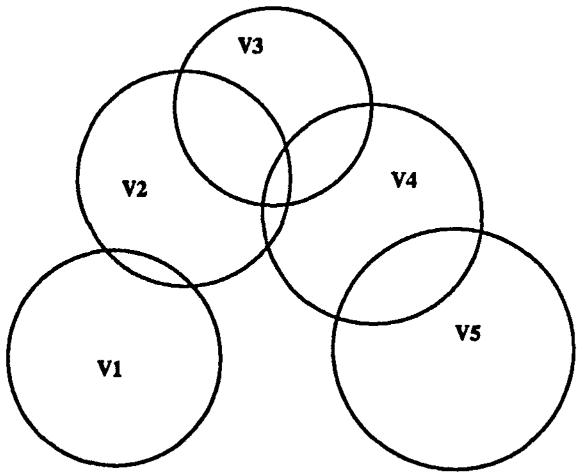


Figure 3.14: Intersection of VMAPs

Table 3.1: Adjacency matrix

Nodes	1	2	3	4	5
1	1	1	0	0	0
2	1	1	1	1	0
3	0	1	1	1	0
4	0	1	1	1	1
5	0	0	0	1	1

- If $I(i,j) = 0$, then V_i and V_j are disconnected from each other
- If all elements in the upper triangle matrix are 1, then there is a unique intersection area. There is one workpiece set-up.
- If all elements in the upper triangle except a diagonal are 0, then there are no intersection pairs. It means that there are n different workpiece set-ups.

3.3 Minimal clustering problem

3.3.1 Formulation of objective function

Workpiece orientation is an important consideration in developing an inspection process for a Coordinate Measuring Machine (CMM). Ideally, the inspection process should be performed in one setup. However, portions of the workpiece may not be accessible by a measurement probe (either contact or non-contact) in a single set-up due to the kinematic limitations of the measuring machine and the geometry of the workpiece. Refixturing of the workpiece may be required to perform the inspection process for the entire workpiece.

In general, since workpiece set-up is a time-consuming and labor-intensive process, setup costs are greatly influenced by the choice of workpiece orientation. To inspect all of the given points at a minimum cost, the number of workpiece orientations should be minimized. This problem can be formulated as a minimal clustering problem [46, 55], where a cluster is defined as a group of VMAPs with non-empty intersection.

For example, the VMAPs for ten points are given in Figure 3.15. Dominant clusters C_1, C_2, and C_3 are identified in Figure 3.16 for a 3-axis machine (i.e., three setups are necessary). Combined with the minimal clustering problem, the inspection path through the points within each cluster must be minimized to reduce the inspection cost.

The objectives are to solve the minimal clustering problem simultaneously with the travel path for the probe. In general, the setup time is much longer than the travel time per unit distance, albeit time for the workpiece setup and the probe travel

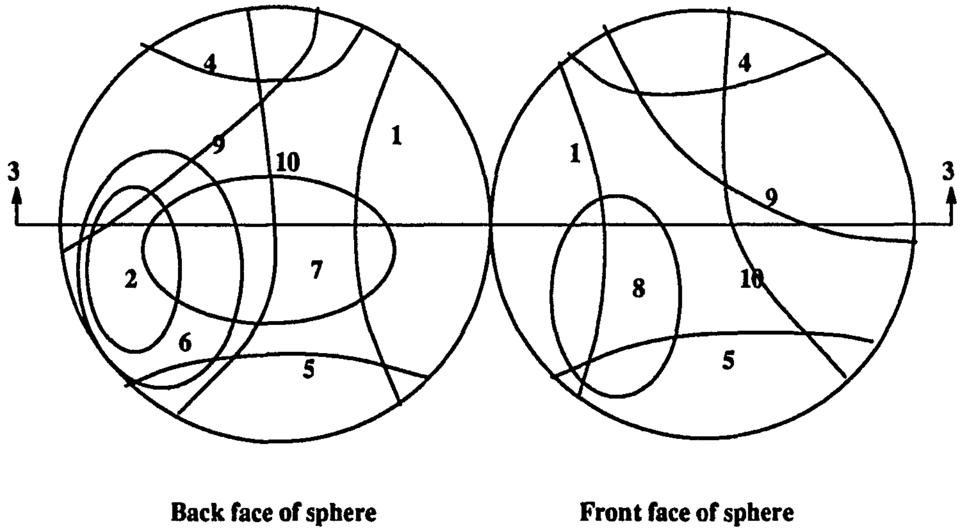


Figure 3.15: VMAPs for ten sample points

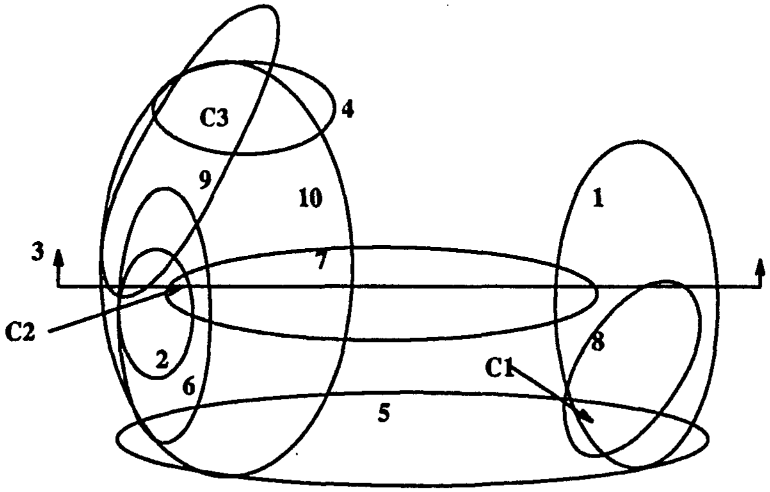


Figure 3.16: Clustering for 3-axis inspection machine

time per unit distance will varied based upon the inspection process. Therefore, the primary objective is to determine the number of setups by clustering the VMAPs. The secondary objective is to determine the shortest path within cluster.

These multiple objectives can be achieved through a multi-echelon optimization method: the first echelon is to determine the minimum number of clusters, s, the second echelon finds the optimal configuration of cluster for a given s, and the last echelon is to compute the shortest path within the cluster using the traveling salesman problem technique. The objective function for this problem can be formulated as

$$Min \qquad Z = C_s * N_s + C_t * D_t + C_p \tag{3.9}$$

where

- C_s = Setup cost
- N_s = Number of setup orientations
- C_t = Travel cost per unit distance
- D_t = Total travel distance
- C_p = Penalty cost

We assume that the setup cost and the travel cost per unit distance are fixed and given by the operator. The penalty cost is added into the objective function when the candidate configuration becomes infeasible.

Given a set of inspection points on a workpiece surface, corresponding VMAPs for the inspection points, Euclidean distance between the inspection points, and VMAP intersections, we can evaluate the objective function. Based on this information, the problem becomes: Given a collection of m VMAPs $\{V_1, \cdots, V_m\}$, perturb these members into $s \leq m$ clusters, such that each member V_i belongs to at least one of the clusters, and the number of clusters and the travel distance are minimal. The perturbation process could be repeated until the optimal solution is obtained.

Let m be the number of VMAPs and s the number of clusters. The total sum of the cluster size must be the same as the number of VMAPs, that is,

$$\sum_{i=1}^{s} n_i = m$$

where n_i is the size of cluster i. The first step of the optimization process is to minimize the value s (i.e., the number of clusters). The number of ways to assign each member V_i to one of the clusters, will be at most

$$\sum_{s=1}^{m} \frac{m^{s-1}}{s!} \tag{3.10}$$

For the given number of clusters with the size $n_i, i = 1, \cdots, s$, there are $m!/\Pi_{i=1}^{s}(n_i!)$ possible candidates. Therefore, in the worst case we can get

$$\sum_{s=1}^{m} \frac{m^{s-1}}{s!} \frac{m!}{\Pi_{j=1}^{s}(n_j!)} \tag{3.11}$$

possible candidates to evaluate the objective function. To obtain the optimal solution of the multiple-objective function, we propose a multi-echelon optimization method in which each objective can be evaluated using a simulated annealing algorithm.

3.3.2 Simulated annealing algorithm

The class of heuristics presented here use simulated annealing (SA) to find the minimum number of workpiece orientations and to determine the shortest travel distance. SA originated from an analogy with the physical planning process of finding low energy states of a solid in a heat bath (Metropolis et al. [33]). Pincus [39] developed an algorithm based on this analogy for solving discretizations of continuous global optimization problems. Most of the other applications to date have been to discrete combinatorial optimization problems (e.g., Kirkpatrick, Gelatt and Vecci [20], and Aaarts and Van Laarhoven [1]).

In essence, SA is an approach that attempts to avoid local optima by allowing an occasional *uphill* move with a probabilistic acceptance criterion. In the course of the SA process, the probability of acceptance descends slowly towards zero. These deteriorations make it possible to move away from local optima and explore other regions through a set of permissible solutions called neighborhoods. The set of solutions that can be obtained from a current solution is called the neighborhood solution. The neighborhood can be reached by random perturbations from the current state.

After selecting a candidate neighbor and determining a likelihood P from a $U(1,0)$ distribution, the acceptance probability (P_A) is computed as

$$P_A = \left\{ \begin{array}{ll} exp[-(Z^{ns} - Z^{cs})/T] & if \quad Z^{ns} > Z^{cs}, \\ 1 & \text{otherwise} \end{array} \right\}$$

where Z^{ns} is the value of the objective function at the candidate neighbor solution, Z^{cs} is the value of the objective function at the current solution and T is the *temperature* (i.e., normalization constant) used for this test [1].

If the perturbation results in a lower objective value (i.e., $Z^{ns} \leq Z^{cs}$), then the process is continued with the candidate neighbor. Otherwise, the walk proceeds to the candidate neighbor if and only if $P \leq P_A$. This rule for accepting new state is referred to as the Metropolis criterion [1, 20, 33]. Repetition of this step continues until the steady state is achieved. At that point, the temperature is lowered and the procedure repeated.

The temperature is slowly lowered with a long time spent at temperatures near the freezing point. This process is referred to as annealing [1, 20]. The period of time at each temperature must be sufficiently long to allow a steady state to be realized. This annealing process has two additional parameters, a cooling ratio α and a chain length L. A chain here is defined as a sidewalk during which the temperature T is held constant. The temperature T decreases after every L steps of the walk and the procedure is repeated until the system freezes. At each temperature, the annealing schedule must allow the simulation to proceed long enough for the system to reach steady state. The temperature T is updated according to

$$T = T * \alpha \quad with \quad 0 < \alpha < 1$$

which was used by Kirkpatrik et al. [20].

SA is considered to be a heuristic and therefore it does not guarantee to find the optimal solution, but Kirkpatrik et al. [20] argue that taking controlled uphill moves allows one to break away from solutions leading to local optima and hence increases the likelihood of obtaining a higher quality solution.

3.3.3 Multi-echelon optimization method

The objective of this section is to present the multi-echelon optimization method to get the optimal solution for the given objective function in which there are two goals to achieve at the same time. One of the goals is to minimize the number of clusters and the other is to minimize the probe travel distance through the sample points. These multiple objectives can be achieved simultaneously by applying the multiple-echelon SA algorithms. Let S_C be a set of cluster configuration and S_t be defined as the shortest path for a given S_C. The basic algorithm for this nested procedure is as follows:

- Step 1: Initialize $m = s$ (i.e., one VMAP per cluster).
- Step 2: Compute an initial solution for S_C and S_t.
- Step 3: Find the optimal S_C^* using SA algorithm.
  - Step 3.1: Find the shortest travel distance, S_t^*, for each cluster using SA algorithm.
- Step 4: Return S_C^* and S_t^*.
- Step 5: If the stopping criterion is met, stop.
  Otherwise, decrement s by 1 and go to Step 2.

The SA algorithm to get the optimal solution for the multi-echelon optimization method is schematically shown in Figure 3.17.

There are three different echelons in the algorithm. The first echelon represents the number of clusters, s, and uses simple enumeration starting from $s = m$ clusters (i.e., one VMAP per cluster). The second echelon investigates the optimal set of cluster configuration, S_C^*, using the simulated annealing algorithm for the s clusters from the first echelon. The last echelon finds the shortest path S_t^* for a given S_C^* passed by the second echelon. Another simulated annealing algorithm for the third echelon is applied to get the shortest path.

The procedures are nested so that the lower echelon passes its solution to the higher solution. That is, the optimal solution S_t^* from the third echelon becomes the value to be considered in selecting the optimal S_C^* in the second echelon.

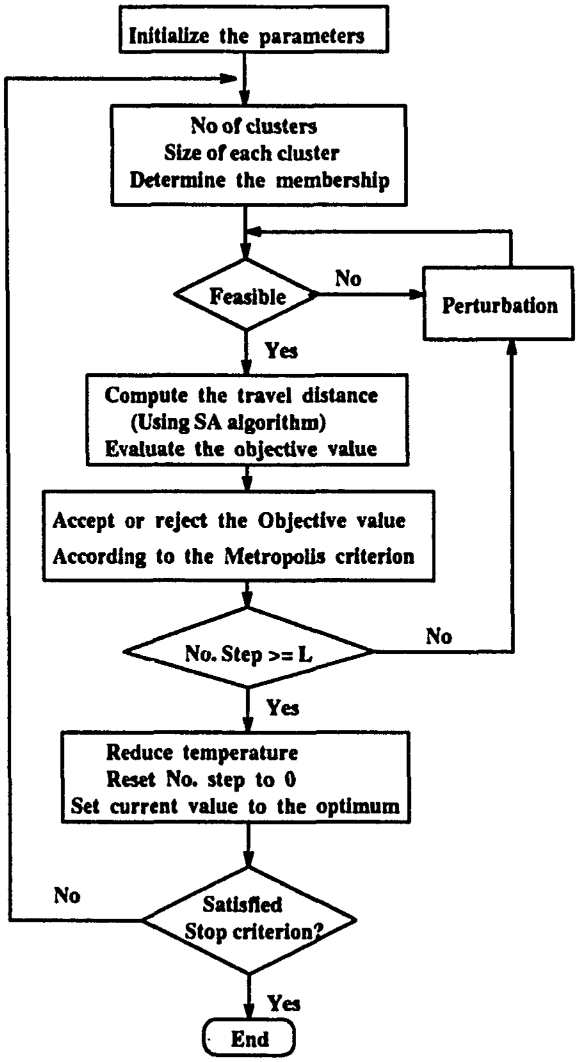


Figure 3.17: Multi-echelon simulated annealing algorithm

3.3.3.1 Optimal set of cluster configuration For a given number of clusters, s, we need to find the optimal set of cluster configuration, S_c^*, using simulated annealing algorithm. The application of SA algorithm must involve the definition of a solution, a cost function, an annealing schedule, and a generation mechanism. The generation mechanism defines a neighborhood for each solution, consisting of all solutions that can be reached from the current solution in a single transition.

In this problem, we are given a set $V = (V_1, V_2, \cdots, V_m)$ and are asked to find a minimal set of partition $S_c = \{C_1, C_2, \cdots, C_s\}$ in which a subset C_i has to be contained the number of members from a set V and $V = C_1 \cup C_2 \cup \cdots \cup C_s$.

For our annealing scheme, a solution will be any partition of the set V. Two different partitions will be neighbors if one can be obtained from the the other by moving a single member from one of its sets to the other.

An initial solution S_c for a given s is obtained by assigning a single member V_i to each cluster C_i. The random walk through the neighbors starts at the initial solution, then rearrange it until an accepted solution S_c^* is found. S_c^* then becomes the starting point for further rearrangement.

If the selected solution S_c is infeasible, then it is perturbed until feasibility is obtained. Feasibility is easily checked by inspecting the adjacency matrix which contains the information of intersection relationship between VMAPs. S_c is perturbed by selecting one of the clusters according to a random walk. A randomly chosen member V_j is taken out from the selecting cluster C_i and transferred to the other cluster C_k. C_k is also determined by a random walk. Therefore each perturbation requires three random numbers from uniform distribution.

If the new solution S_c is feasible, then the next optimization process for getting the shortest path within each cluster $C_i, i = 1, \cdots, s$, must be performed. The next section describes how to get the shortest path from the given number of sample points.

The cost function is defined as the sum of the setup cost and the travel cost. The evaluation of the cost function is based on the value of the cost function for the candidate neighbor. If the objective value for the neighbor, Z^{ns}, is lower than one for the current state, Z^{cs}, i.e., $Z^{ns} \leq Z^{cs}$, then the new solution becomes the current optimal solution. The higher values will be accepted with a probability based on the current solution. That is, a candidate solution is accepted or rejected according to the Metropolis criterion [33];

- If $\Delta Z \leq 0$, then accept the new solution; $Z^{cs} = Z^{ns}$.
- Otherwise, accept the new solution with accepting probability;

$$P_A = exp(-\Delta Z/T)$$

  where $\Delta Z = Z^{ns} - Z^{cs}$.

The probability that an uphill move of size ΔZ will be accepted diminishes as the temperature declines, and, for a fixed temperature T, small uphill moves have a higher probability of acceptance than large ones.

The annealing schedule controls the rate at which the temperature T decreases and therefore the acceptance probability of higher solutions. There are several parameters of schedule to be specified;

- Temperature function
  - Maximum attempts at a temperature (Chain length): L

- Number of better solutions at a temperature: N_A
- Temperature: T
- Cooling ratio: α

• Stopping criterion

- Number of successive temperature changes without a better solution: N_B
- Number of chains: N_C

The temperature T is decreased whenever L attempts have been made or N_A solutions are obtained. The temperature function is given by

$$T = T * \alpha, \quad 0 < \alpha < 1$$

where the cooling ratio is in general given by $\alpha = 0.9$ (typically between 0.9 and 0.99) [20].

This process is continued until the stopping criterion is satisfied. If the acceptance of the candidate solution is not occurred at the number of consecutive temperatures, N_B, the system is considered "frozen" and annealing stops. And also the system is stopped whenever the process is reached to the number of chains, N_C.

Although the search for adequate annealing schedules has been addressed in many papers [1, 13, 20], they must be determined experimentally for a specific problem.

Figure 3.18 shows the trend of SA convergence to find the near-optimal solution. The X-axis indicates the number of perturbations and the Y-axis indicates the number of possible workpiece orientations. The graph shows that the number of perturbations to reach at the near optimal solution could be increased whenever the sample size is

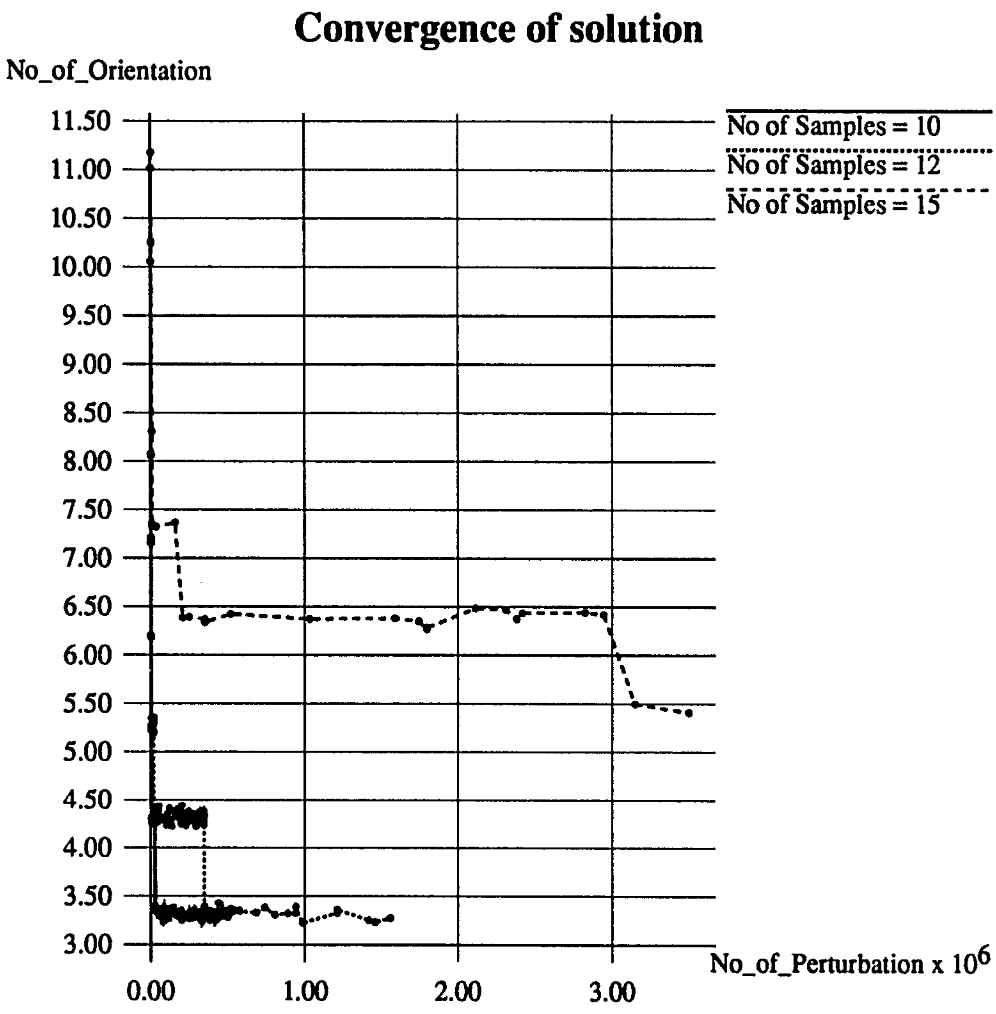


Figure 3.18: SA convergence toward the optimality

Table 3.2: Average number of perturbation

No. of Sampie	S.A.	Enumeration
n = 10	0.7e+06	1.7e+09
n = 12	1.56e+06	1.4e+11
n = 15	3.5e+06	2.24e+13

increased. For the case of n=10, the number of perturbations is less than $0.7 * 10^6$ when the optimal solution is obtained. The number of perturbations is increased to approximately $1.56 * 10^6$ when the sample size becomes twelve. As the sample size becomes n=15, the number of perturbation jumps to $3.5 * 10^6$.

The computational efforts for the SA algorithm and the simple enumeration are given in Table 3.2. It shows that the multi-echelon optimization method can approach the near optimal solution quickly and find an equivalent solution with respect to travel distance. The final output will be the minimum number of workpiece setups and the shortest path. This inspection path becomes the initial inspection path for the collision-free path generation process .

3.3.3.2 Shortest path generation The lower level of this optimization process is to generate the shortest path within the feasible configuration of the workpiece setup. This problem is the same as the "traveling salesman" problem [10, 27, 28] which is a well-known optimization problem to construct the shortest tour of a prescribed list of N sample points. The basic requirements of this problem are that the path must be as short as possible and the path must be a tour. That is, each point is to be visited only once, and the path is to be made as short as possible.

Let S be the set of all edges (i.e., $\frac{N(N-1)}{2}$ edges between the N sample points) and let T be an subset of S that forms a tour. In the traveling salesman problem, we have to find a subset T that forms a tour and has minimum length from the set S. The SA algorithm is again applied to get the solution of this problem. The application of an SA algorithm presupposes the definition of solutions, a cost function and a generation mechanism.

Each solution T of the problem is defined as a permutation of the sequence of points, interpreted as the order in which the sample points are visited. The total number of different solutions is $\frac{1}{2}(N-1)!$.

The cost function is just the total length of travel

$$D_T = \sum_{k=1}^{s} \sum_{i \neq j}^{n_k} D_{ij}$$

where D_{ij} is the Euclidean distance between two points i and j, and n_k is the size of the k^{th} cluster.

A generation mechanism in this problem is defined as a replacement of k edges from T with k edges from $S - T$, such that the resulting solution T' is feasible. This is repeated as long as such group can be found. This mechanism was applied to the traveling salesman problem by Croes [10], with $k = 2$, and by Lin [27], with k=3. Our basic algorithm attempts to transform T into T' by exchanging $k = 2$ or 3 edges between T and $S - T$ randomly. Figure 3.19 illustrates the situations for $k = 2$ and $k = 3$. Notice that x_i and y_i share an endpoint and so do y_i and $x_{i+1} (x_{k+1} = x_1)$. Iterative improvement algorithms based on this generation mechanism have been shown to be quite effective for the traveling salesman problem (Lin and Kernigham [28]).

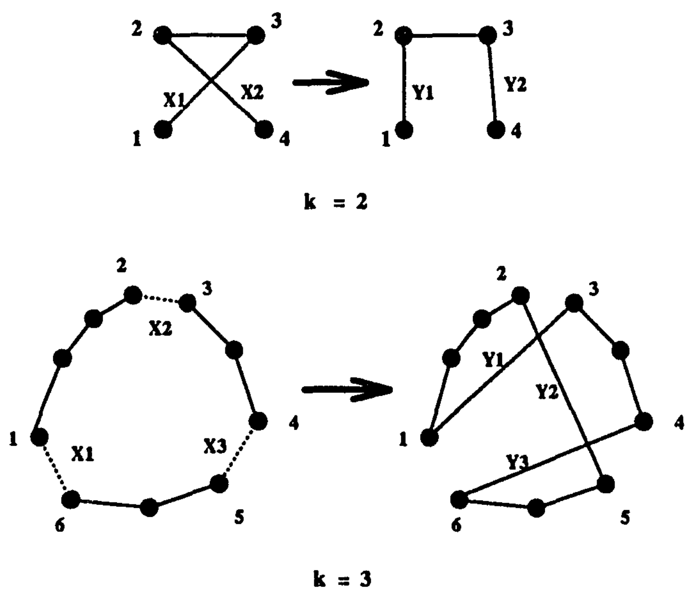


Figure 3.19: Generation mechanism

3.4 Constrained workpiece orientation

3.4.1 Selection of workpiece orientation

While the multi-echelon SA procedure provides the minimum number of workpiece orientations with the shortest travel distance, the final orientation to setup the workpiece on a CMM has yet to be determined. By choosing one direction vector within the common intersection area of VMAPs, which is called a clustering area, a feasible orientation can be obtained. There are at least two ways to select the workpiece orientation:

1. Random selection (Unconstrained selection).

2. Limited selection (Constrained selection).

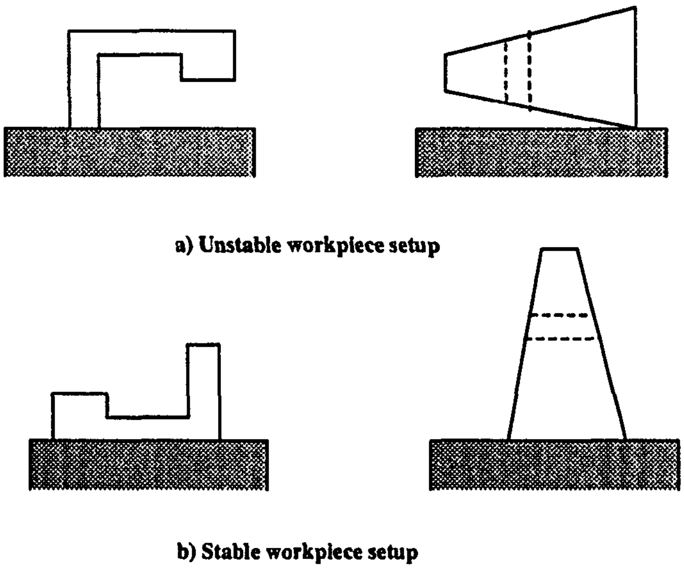


Figure 3.20: Stable and unstable workpiece setups

In the first approach, any direction vector within the clustering area is feasible, but it may cause the high setup cost if the workpiece requires special fixturing. For example, as shown in Figure 3.20 a, the workpiece cannot support itself without fixturing equipment. Since this approach will generate the setup orientation arbitrary, the proper fixturing equipment must be supplied to position the workpiece on a CMM safely so that the setup cost becomes higher and inspection time increases.

Using the second approach, in which the workpiece orientation is chosen manually with respect to the geometry of workpiece, the workpiece can be positioned safely on a CMM without using fixturing equipment if and only if there is a flat plane on the workpiece. With this approach, the inspection process becomes more efficient because of easy and safe setup of the workpiece on CMM. Figure 3.20 b shows the stable workpiece setup without using fixturing equipment. The workpiece orientation is chosen from one of the normal direction vectors of the flat planes on the workpiece.

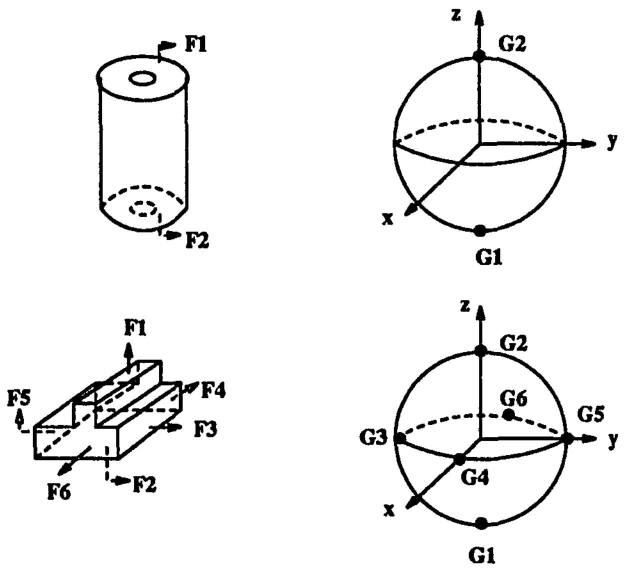


Figure 3.21: Reference planes and spherical representation of RPs

The requirements of the flat plane to be the workpiece orientation will discuss in the next section.

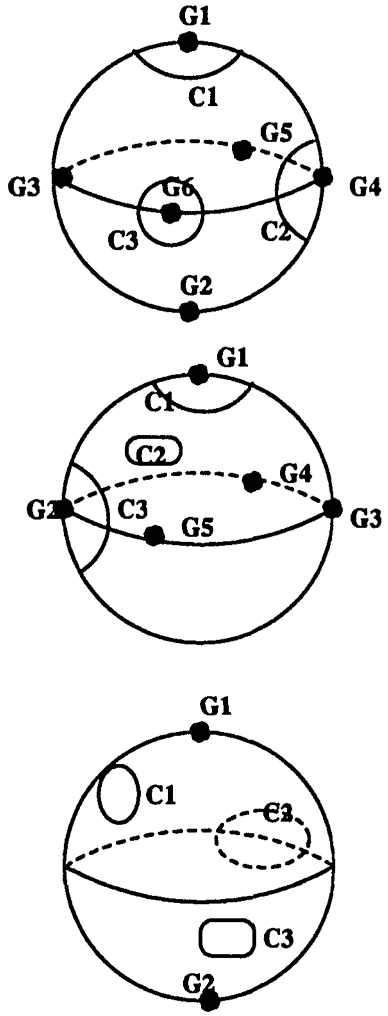
3.4.2 Reference plane and workpiece orientation

Let's introduce the concept of a reference plane (RP). A reference plane (RP) is the open and flat surface on the workpiece such that the workpiece can be setup safely without using fixturing equipments. A workpiece is oriented along the normal direction of the RP, in which we can position the workpiece firmly without using fixturing equipment.

In spherical representation, the inward normal of RP is mapped onto a unit sphere as a point, which is called the Gaussian Map (GMAP) of RP [8, 17]. We define G_i as the GMAP of the i^{th} RP. Figure 3.21 shows RPs and their GMAPs on a unit sphere in which the coordinate system of a unit sphere comes from the datum reference frame of the workpiece. Let D_i be the direction vector of G_i. The direction vector, D_i, represents the possible workpiece orientation with respect to the geometry of workpiece since the workpiece locates on a CMM safely with this direction vector. Therefore, the workpiece orientation can be selected from one of the direction vectors. For example, in Figure 3.21 the cylindrical workpiece has two possible orientations, while the prismatic workpiece has six.

We should choose one or more direction vectors to inspect all of the points. If G_i is located within the specific clustering area C_j (i.e., $G_i \in C_j$), then the corresponding points within C_j can be inspected by the fixed probe along the workpiece orientation D_i. However, if one of the clustering area does not hold any G_i, then the separate workpiece orientation must be generated for inspecting the points within the corresponding cluster.

There are three possibilities of relationship between the location of G_i and the distribution of clustering areas as shown in Figure 3.22. Figure 3.22 a indicates that all G_i are located within the specific clustering area, respectively. In this case, we can choose the minimum number of direction vectors to cover all of the points without using fixturing equipments, i.e., three direction vectors become the minimum number of workpiece orientations. This is the ideal case to find the workpiece orientation. The second case as shown in Figure 3.22 b has a clustering area which cannot have a connection with any GMAP of RP. For inspecting this clustering area using the fixed probe, we need to setup the workpiece with the fixturing equipment such that the direction vector of workpiece orientation should be matched with one of the direction vectors of the corresponding clustering area. The clustering areas in the third case are



a) All clusters hold Gi

Do not need the fixturing equipment

b) One cluster could not hold Gi

One setup with the fixturing equipment

Two setups without using fixturing equipment

c) All clusters do not hold Gi

Three setups with the fixturing equipments.

NOTE :

Gi : GMAP of RP

Ci : Cluster

Figure 3.22: Relationship between G_i and clustering areas

completely mismatched with the G_is. It means that the points cannot be accessed by the fixed probe if the constrained workpiece orientations are applied. That is, RPs are not applicable to setup the workpiece on a CMM. In this example, three random setups with the different fixturing equipments are required to inspect all of the points.

As shown in the above examples, the constrained workpiece orientation may fail to provide the proper workpiece orientations to inspect all of the points since the selection of workpiece orientation is limited by the geometry of the workpiece. To utilize the constrained workpiece orientation related with the RPs on a workpiece, the probe must be rotatable so that the probe can approach points by tilting the axis of the probe. The next section describes the probe orientations for a constrained workpiece orientation.

3.4.3 Probe orientation

The probe head available in this study is a PH9A by Renishaw. It provides two rotational degrees of freedom about two orthogonal axis and has a rotation range of $0^\circ \sim 105^\circ$ on the A-axis (called a pitching angle) and $-180^\circ \sim 180^\circ$ on the B-orientation (called a rolling angle). Practically, however, the pitching angle must be limited to the range of $0^\circ \sim 90^\circ$ since the bottom of the workpiece is not accessible by the probe. Figure 3.23 illustrates the probe abstract and its two rotational degrees of freedom.

Adding two rotational degrees of freedom of the probe to the inspection machine the probe accessibility with a given workpiece orientation could be represented by the hemisphere as shown in Figure 3.24. With the variable probe angle the probe

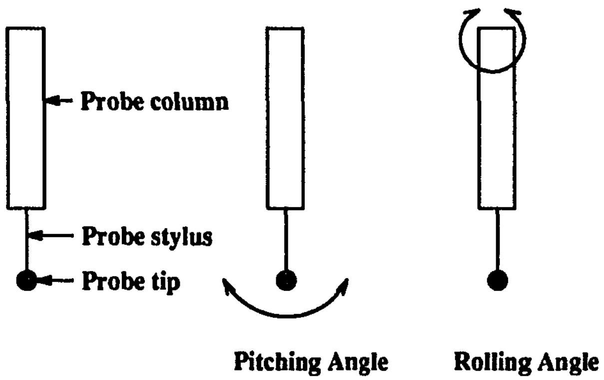


Figure 3.23: Probe abstraction and its rotational capability

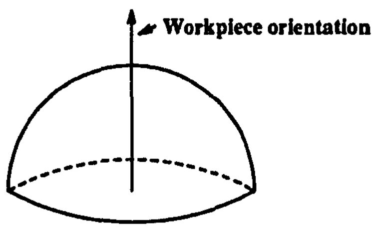


Figure 3.24: Range of probe orientation

can access points within the nonintersecting clustering area C_j (i.e., $D_i \cap C_j = \emptyset$) without changing the workpiece orientation. Therefore, we can possibly reduce the number of workpiece orientations by adding a few probe orientations. Although the probe rotation will cause additional time (qualification time, probe clearance time and probe rotation time), it is insignificant compared with the time for refixturing the workpiece.

The problem again is to minimize the number of constrained workpiece orientations to inspect all of the points with the number of probe orientations. The probe orientation A_j is determined by the line between the origin and the center of the j^{th} clustering area (as shown in Figure 3.24). If the angle between the direction vector D_i and the probe orientation A_j is less than the pitching angle, i.e., $\angle(D_i, A_j) \leq 90^{\circ}$, then this probe orientation is said to be feasible with respect to the direction vector D_i. That is, the points within the clustering area C_j can inspect by the probe if the probe orientation is feasible.

Thus, the problem is to find the direction vector D_i which has the maximum number of feasible probe orientations such that

$$Maximize \quad \#\{(i,j) : \angle(D_i, A_j) \leq 90^{\circ}, i = 1, \cdots, n, j = 1, \cdots, s\} \tag{3.12}$$

where $\#\{E\}$ is the cardinality of the set E.

For example, in Figure 3.26 we have four direction vectors and three clustering areas in which all direction vectors are not matched with any clustering area. Without additional degrees of freedom for the probe, the workpiece must be positioned on a CMM using a fixture. Now we determine the probe orientation by connecting the line between the center of clustering area and the origin of a unit sphere. If the direction vector D_i intersects with the clustering area C_j, then the probe orientation

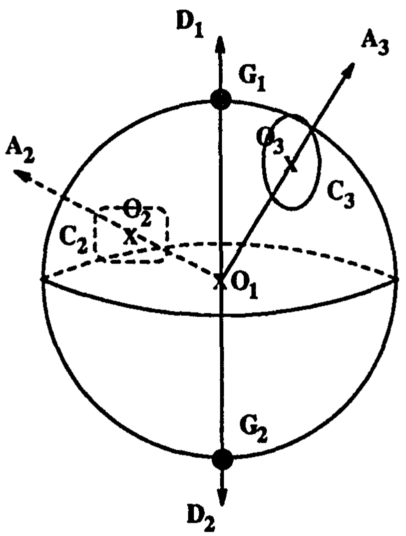


Figure 3.25: Determination of probe orientation

A_j changes to the direction vector (i.e., $A_j = D_i$). Using the equation 3.12, the maximum number of feasible probe orientations for each direction vector could be determined. Heuristically, we can find the combination of direction vectors D_1 and D_2 as the workpiece orientations by adding the number of probe orientations to inspect all of the points (or the combination of D_3 and D_4 is the other candidate). With the direction vector D_1 as the workpiece orientation, two probe orientations A_1 and A_2 are required to inspect the points on a upper hemisphere. For the points on a lower hemisphere, we position the workpiece along the direction vector D_2 and assign the probe orientation A_3 to inspect the points within C_3.

As shown in the above example, the constrained workpiece orientation is a function of the reference planes on a workpiece and the probe orientation for each clustering area. With this approach, we can determine the workpiece orientations with respect to the geometry of the workpiece. Related with given direction vectors, the

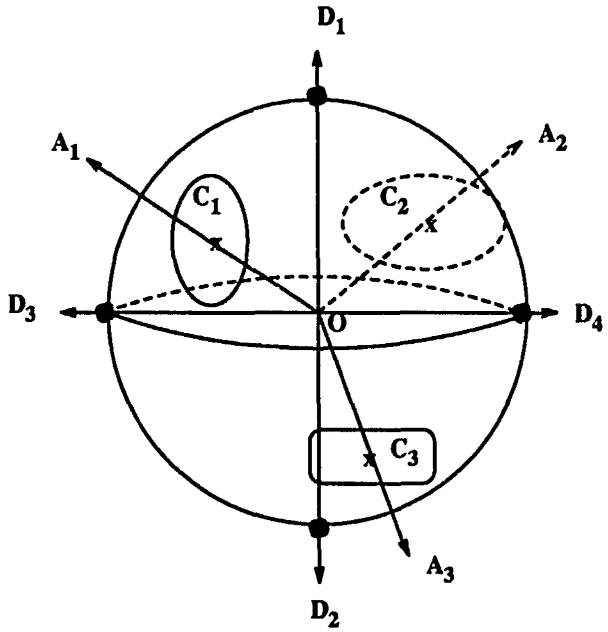


Figure 3.26: Combination of direction vectors and probe angles

combination of the workpiece orientation and the probe orientation to inspect all of the points could be selected heuristically.

CHAPTER 4. COLLISION-FREE PATH GENERATION

4.1 Path generation method

The current path generation method is shown in Figure 4.1, where the probe tip trajectory is divided into two major paths: the approach/retract path and the drive path.

In the approach/retract path, the probe approaches the inspection point from a prehit point and retracts to the prehit point after taking a point measurement. The prehit point lies on the normal direction vector of the inspection point since measurement data taken by the probe along this direction can reduce the measurement inaccuracy. Its distance from the inspection point is arbitrary, but it is in general set at ≥ 0.5 inch.

The drive path is defined as a probe trajectory between two prehit points. If the drive path has no intersection with the workpiece, then we obtain a simple straight line trajectory. However, the drive path must be modified if a collision is detected along the path segment. The inspection path between two adjacent points and its detour for a general workpiece surface is shown in Figure 4.1. Let us define the notation.

- t_{ij}: Probe trajectory between p'_i to p'_j $(= \overline{p'_i p'_j})$

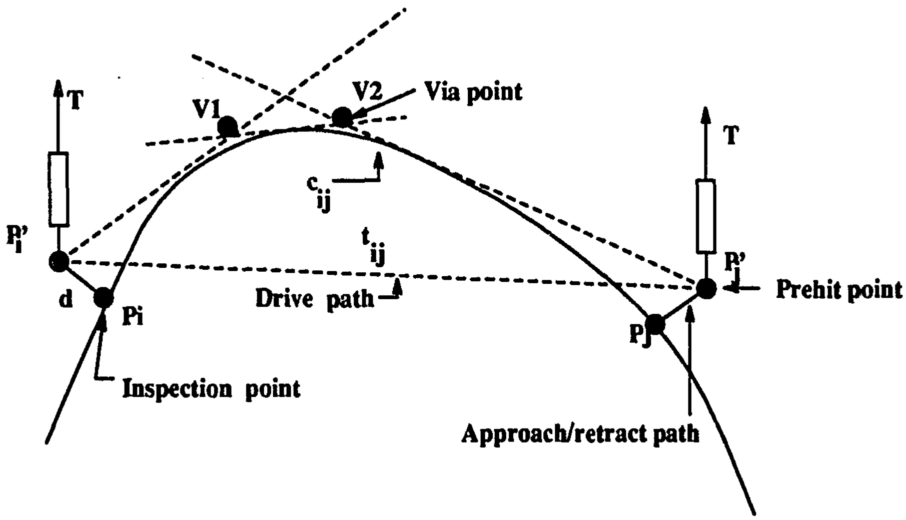


Figure 4.1: Path generation method to take point measurements

- c_{ij}: Path between p_i to p_j $(= \widetilde{p_i p_j}$
- p_i: Inspection point
- $p'_\cdot$: Prehit point with respect to the inspection point p.
- d: Safety distance = Approach distance = Retract distance
- T: Probe axis
- P: Path direction vector
- V.: via point

By determining the probe orientation (as described in Chapter 3) and therefore the approach/retract path a priori, we need only consider the drive path. In addition, the probe orientation information is used to group points that can be measured by the same probe angle. By partitioning the inspection points based on the probe

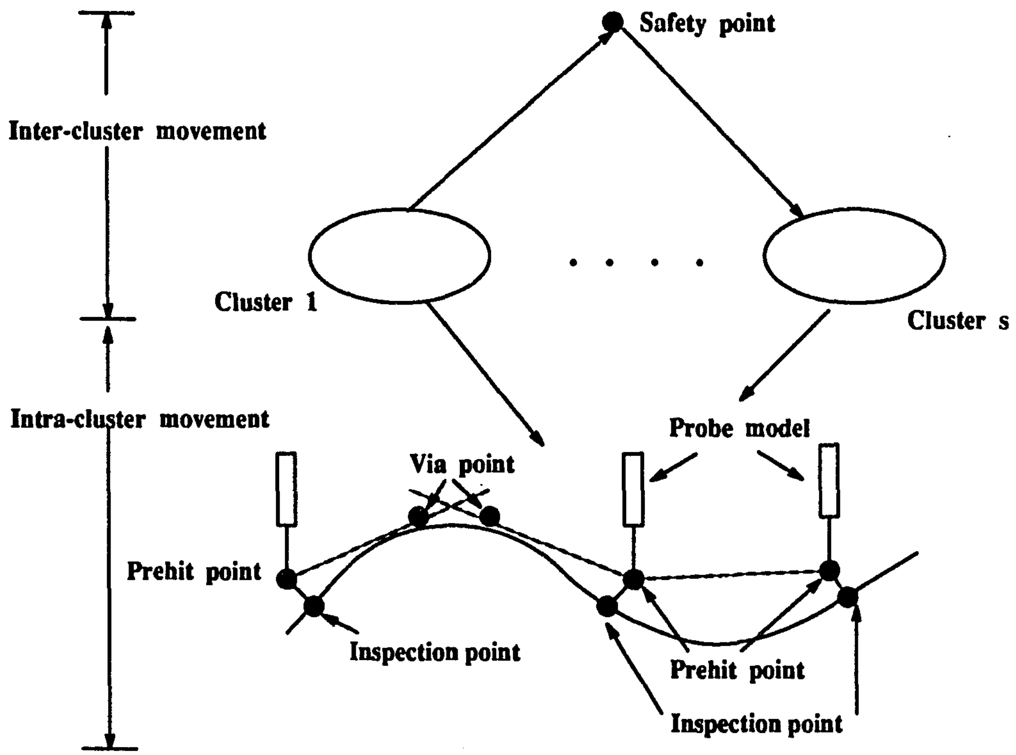


Figure 4.2: sub-path generation methods: intra- and inter-cluster

orientation information, the path generation method can be divided into the following two sub-path generation methods (see Figure 4.2).

- Intra-cluster: Find a safe and locally shortest path to traverse within each cluster.

- Inter-cluster: Find a sequence of clusters and specify a safety point.

An intra-cluster path generation will provide the safe and locally shortest path through the points within the cluster, in which the probe orientation must be maintained the same angle. The drive path within a cluster can be modified if a collision is detected along the path segment. When considering a probe movement on the workpiece surface, there is an infinite number of paths between two points, p_i and p_j. In this research, we consider only the path (c_{ij}) between two points which lie within a unique plane. This plane is defined by two vectors, the direction vector of the drive path (P) and the probe orientation (T). The cross product of P with the probe orientation vector T results in a vector H which defines the plane. The path is generated by the intersection between the workpiece surface and the plane H (i.e., $c_{ij} = H \cap^* W$). The detour between two points is derived within this plane.

An inter-cluster path generation will generate the sequence of clusters to be visited by a probe and specify a safety point for changing the probe orientation without interference with the workpiece. After completing the point measurements for the cluster of points, the probe has to be retracted to a safety point that clears the workpiece to ensure the probe does not collide with the workpiece during the rotation. The safety point is located slightly higher than the height of the workpiece. The probe will return to a safety point in the vertical direction and change the probe orientation for the next cluster of points. This path is called a safety path. The inspection time increases not only for the rotation, but also for the time needed for the probe to move to a safety point. It could be compensated by partitioning the inspection points based on the probe orientation such that the number of necessary rotations is minimized.

Let the home position of the system be the initial start point. The closest point among a set of inspection points from the start point is selected to be a first inspection point. The corresponding cluster becomes the first group to be inspected. The next start point is the safety point which is the position retracted from the last point in that cluster. The next cluster is selected in the same way. That is, the sequence of

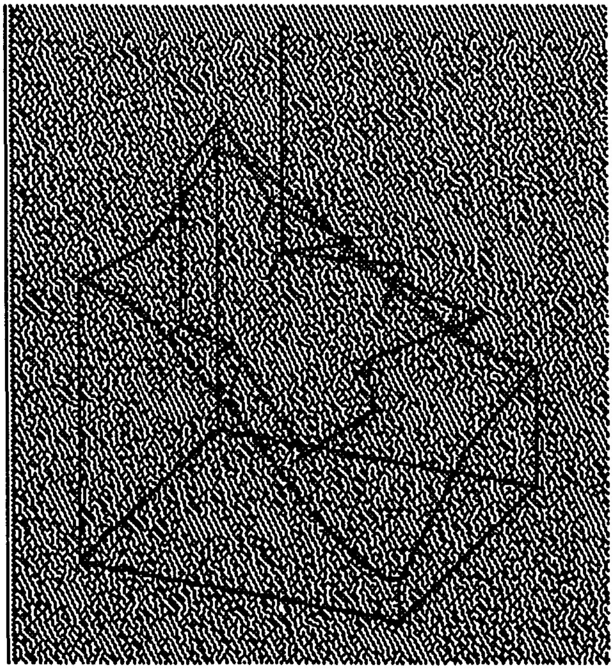


Figure 4.3: Probe inspection path on the boundary surface

clusters to be visited by the probe is iteratively determined by the shortest distance from the start point.

Figure 4.3 illustrates the inspection path on the boundary surface generated. It shows the shortest travel path through the inspection points within a selected cluster. The probe starts at the safety point and then approaches the prehit point. Given the approach distance (d), the probe approaches the point from the prehit point along the normal direction vector of the inspection point and retracts to the prehit point. The next cluster is one that has the closest point from the safety point.

Obstacles may exist when the probe moves from a prehit point to the next prehit point. That is, the probe model may collide with portions of the workpiece during travel along the drive path. For each segment of the drive path, the collide path segment must be verified using a collision detection method. The swept volume of the probe model is used to check the interference with the workpiece during its movement. If there is a collision, we should generate a safe path around the object to avoid the collision. A collision avoidance procedure is proposed to generate the collision-free path with the shortest travel distance. In the next section we present a collision detection method to determine the collide path segments and a collision avoidance method to generate the safe and locally shortest path for a collide path segment.

4.2 Collision detection

A collision between objects occurs when they attempt to occupy the same space. Several approaches have been studied for three-dimensional collision detection. In general, collision detection has two broad cases; static interference and dynamic collision detection [3]. This study will focus its attention on dynamic collision detection which is interference detection of moving objects traversing specified paths.

There are many different algorithms used for dynamic collision detection. One algorithm relies on the use of repetitive static interference checking throughout a given trajectory. This algorithm is called the multiple interference detection method which was proposed by Boyse [3]. Although this approach is very general, it creates an intensive computational load. Another algorithm develops the swept volume of a moving object over its trajectory to find the intersection of the swept volume with the obstacles [5, 11, 54]. Cameron [5] developed a four-dimensional collision detection method in which the object S is extruded in space. Objects collide if and only if their extrusions intersect. This technique is a combination of both time domain and swept volume techniques. It is useful to detect the collision between the multiple moving objects within the same environment. Since we only consider a probe as a moving object, the swept volume of a probe is used to detect the collision with the fixed obstacles in this research.

Once the inspection path is given, the procedure is activated to detect collisions for each individual path segment. The collision detection process requires the following steps.

1. Find the swept volume of the probe model along a proposed inspection path segment (Sweeping operation).
2. Determine any overlap between the swept volume, the part model and the fixture model (Hierarchical collision detection).
   - step 1: collision detection against probe tip.
   - step 2: collision detection against probe stylus.
   - step 3: collision detection against probe column.

The problem of dynamic collision detection is now a problem of determining static interference between the swept volume of the probe model and the workpiece. One of two possible results will be produced by the intersection operator. Either a null set (which indicates no interference) or another solid object (which is the interference object) will result.

Actually, a probe consists of three components: probe tip, probe stylus, and probe column. Each one has a different swept volume. In collision detection, the swept volume of each component is tested against the workpiece step-by-step. This is called a hierarchical procedure to detect the collision. At the first level of this procedure is an interference test between the workpiece and the swept volume of probe tip. If interference is detected, then this path segment is definitely modified to avoid the collision. If there is no interference, then the procedure moves up to the second level. At the second level, interference is checked against the probe stylus. The third level is activated if the second level does not indicates a collision. The third level involves the interference checking against probe column. If one of the probe components collides with the workpiece, the corresponding path must be modified according to the collision avoidance procedure.

4.2.1 Sweeping operation

Sweeping operations are useful in engineering applications such as collision detection of moving objects in space or simulations of material removal due to a machining operation. In collision detection, a moving object collides with a fixed obstacle if the swept volume due to the motion of a moving object intersects with the part model.

Conceptually, the swept volume of a moving object S, called the generator, is the set of points encountered by the object during its trajectory. In general, a moving generator defines another geometric entity of one higher degree of freedom. If the trajectory is continuous, and is parameterized by $t \in [0,1]$, then the swept volume $SV(S)$ is defined as

$$SV(S) = \cup S_t, \quad t \in [0,1] \tag{4.1}$$

where S_t denotes the instance of the generator at t. However, in the application of collision detection for the inspection path using the contact probe, only the overall interference between a moving object and a static environment is of concern. In this case only the total swept volume is required rather than the object at any particular time instance. Therefore, rather than searching for intersections of two objects for many sequential positions of the moving object, we compute instead the intersection of the stationary object and the volume defined by the moving object.

There are three different types of sweeps; linear, nonlinear, and hybrid sweeps. Since we assume that the probe movement is along a straight line and has no rotational movement on the path, the linear (translation) sweep is of concern in this study. Relative to the direction of the path the moving object does not change its orientation. Thus the swept volume of the moving object is bounded by relatively simple cylindrical surface. We will generate the swept volume of each probe component separately and perform the collision detection procedure for each swept volume.

4.2.2 Swept volume of probe model

The swept volume of probe model is shown in Figure 4.4. The sweeping object in this figure is a cross section of the probe model instead of a whole body of probe model since the probe model does not interfere with the workpiece at the starting and the final positions based on accessibility analysis.

The collision detection is accomplished by the intersection checking between the boundary surfaces of the swept volume and the workpiece. One cylinder and three parallel planes from the swept volume of probe model are considered. The sweeping operation of the probe tip generates a cylindrical shape and becomes the first generator S_1. The radius of generator is the same as the radius of probe tip and

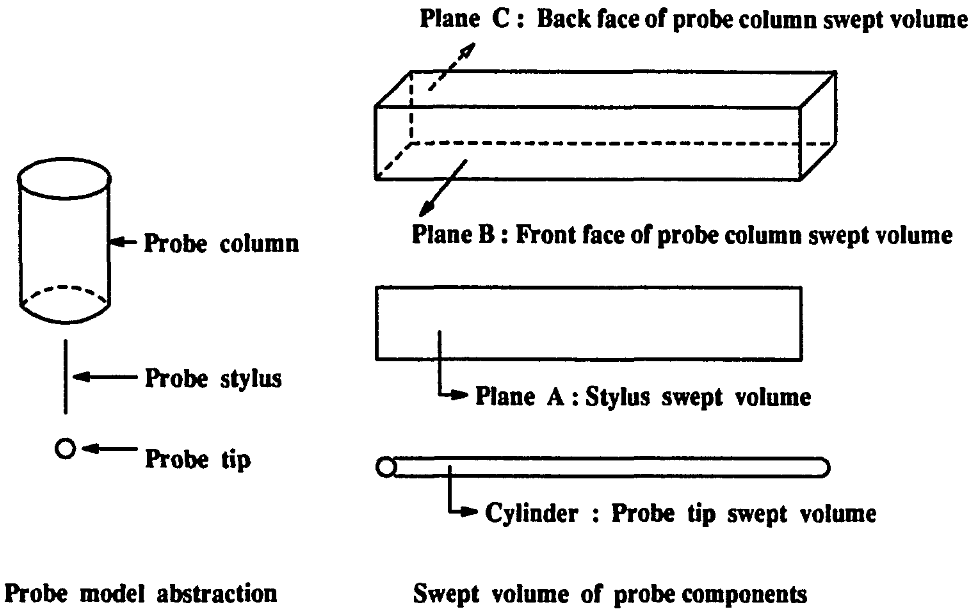


Figure 4.4: Simplification of probe swept volume

the direction of probe trajectory is determined by the difference between two prehit points. The second generator S_2 is the swept volume of probe stylus. To simplify the problem, the probe stylus is abstracted as a straight line. The swept volume S_2 is then a flat plane. For generation of third generator S_3, both edges of the cross section of the probe column are swept along the probe trajectory. There are two generators S_{3b} and S_{3f} in which S_{3b} indicates the back face of the probe column swept volume and S_{3f} indicates the front face of the probe column swept volume. These generators are applied to detect the interference with workpiece starting from the first generator to the third generator.

4.2.3 Hierarchical procedure of collision detection

The basic goal of this procedure is to detect the collide path segments by sequentially checking interference between the workpiece and each swept volume of probe components.

The first level of procedure checks for interference against the swept volume of the probe tip for each path segment. If interference is detected, the corresponding path segment is defined as the collide path and must be modified according to the collision avoidance method. If there is no interference, the second level is activated to check interference against the swept volume of probe stylus. If a collision is not detected, then the interference checking against probe column as the third level is performed. The path is a collision-free if there is no collision at all. The hierarchical procedure of collision detection is shown in Figure 4.5.

4.2.3.1 Collision detection against the swept volume of probe tip

Since the probe tip is a sphere with a certain diameter, the swept volume of probe tip forms the cylindrical shape. Thus the problem of collision detection is detecting intersection between the workpiece model and the swept volume defined by the probe tip. However, we can simplify the test by shrinking the probe tip to a point, and creating a new version of the workpiece model that is grown by an amount corresponding to the tip radius. The swept volume in this case is just a trajectory line along the path segment, while the workpiece surface is enlarged by the amount equal to the tip radius normal to the surface. This enlarged workpiece surface is called the grown workpiece surface [31, 30]. Thus we can find the intersection between the workpiece surface and a line traversed by the tip if there is interference between the

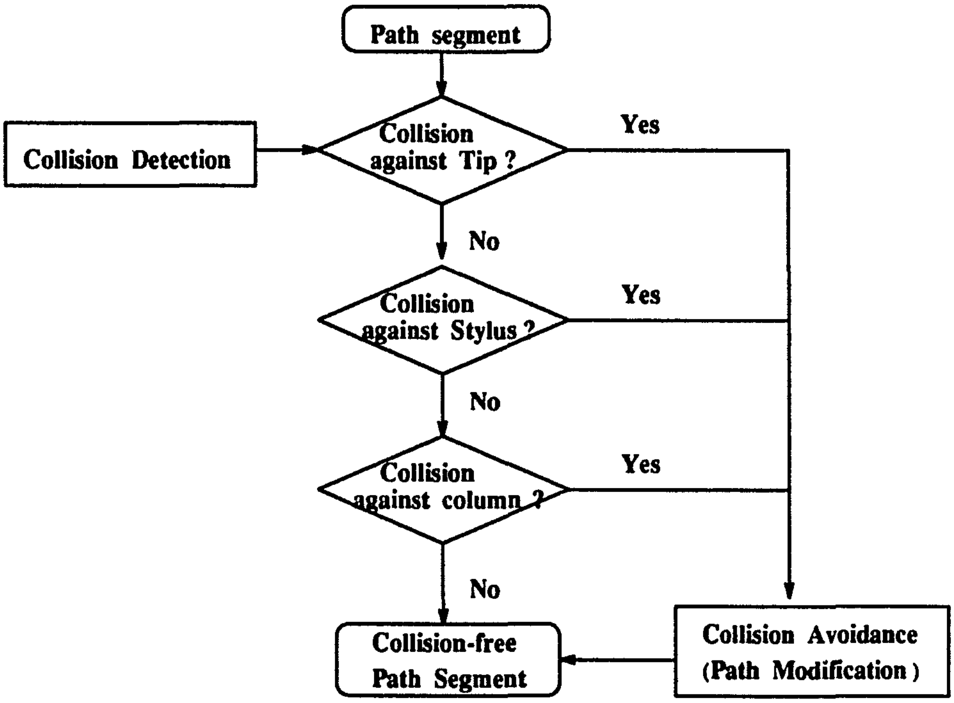


Figure 4.5: Path verification using hierarchical collision detection

workpiece surface and a probe tip. The task of detecting intersections between a line and the workpiece model is relatively simple to perform.

Consider the intersection of a straight-line segment with a sculptured surface. Let points on the surface be $\mathbf{p(u, w)}$. Using algebraic form, $\mathbf{p}(u, w)$ can be written as

$$\mathbf{p}(u,w) = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{c}^{ij} u^i w^j \qquad u, w \in [0,1] \tag{4.2}$$

This equation can be expressed in matrix form as

$$\mathbf{p}(u,w) = \mathbf{U}^T [\mathrm{C}] \mathbf{W} \tag{4.3}$$

where $\mathbf{U} = [u^m u^{m-1} \cdots u1]$, $\mathbf{V} = [v^n v^{n-1} \cdots v1]$, and $\mathbf{C}$ is the corresponding coefficient matrix in algebraic form. When the workpiece surface is enlarged by the amount of tip radius, the grown workpiece surface becomes

$$\mathbf{p}'(u,w) = \mathbf{p}(u,w) + r\mathbf{n}(u,w) \tag{4.4}$$

where r is the probe tip radius and the unit normal $\mathbf{n}(u, w)$ is determined by

$$\mathbf{n} = \frac{\mathbf{p}^u \times \mathbf{p}^w}{|\mathbf{p}^u \times \mathbf{p}^w|} \tag{4.5}$$

where $\mathbf{p}^u$ and $\mathbf{p}^w$ are derivatives along the u and w directions, respectively.

Points on a line can be represented as

$$\mathbf{q}(t) = \mathbf{a} + \mathbf{b}t \qquad t \in [0,1] \tag{4.6}$$

where $\mathbf{a}$ and $\mathbf{b}$ are the endpoints of the line. Therefore, points of intersection occur when simultaneous sets of u, v, t satisfy

$$\mathbf{p}(u,w) + r\mathbf{n}(u,w) - \mathbf{q}(t) = 0 \tag{4.7}$$

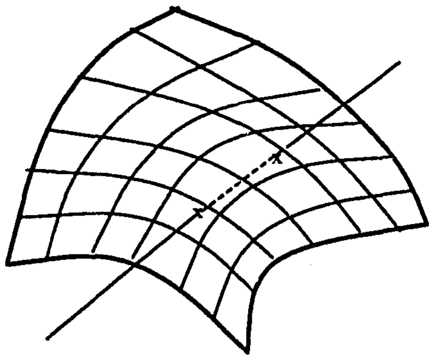


Figure 4.6: Intersection of a straight line and a general surface

Note that this set of equations represents three simultaneous, nonlinear equations in three unknowns. The solution is easily obtained using numerical analysis methods. Figure 4.6 illustrates an intersection between a straight line and a workpiece surface.

If intersection is detected in this level, the corresponding path segment is defined as a collide path and must be modified according to the collision avoidance method. Otherwise, the next level of collision detection procedure attempts to check interference for the same path segment.

4.2.3.2 Collision detection against the swept volume of probe stylus

The probe stylus has a cylindrical shape. Its radius is in general smaller than the radius of probe tip. For simplicity, it is abstracted as a half-line. Thus the swept volume of probe stylus forms a face when a CMM moves in a path segment.

Since the probe tip trajectory has no intersection with the workpiece (if the probe tip collides with the workpiece along the path segment, then this level of collision detection is not considered), a collision would be happen if only if there is

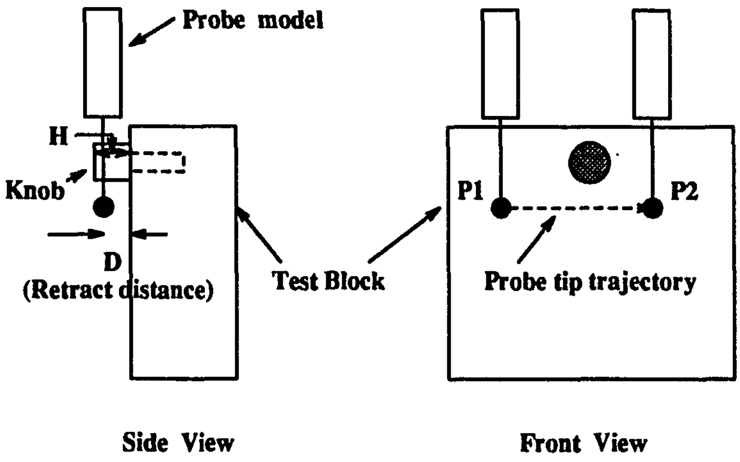


Figure 4.7: Interference between the probe stylus and a knob on the block

any obstacle within the area swept by the probe stylus. Figure 4.7 illustrates the possible interference with the probe stylus. There is a knob on the upper part of block. Along the path segment between the first point P_1 and the second point P_2, the probe tip can travel without interference. However, the probe stylus may collide with a knob through its trajectory if the distance (D) between the prehit point and the inspection point is less than the height (H) of a knob head.

A face swept by the probe stylus along its trajectory is bounded by two prehit points $\mathbf{P}_1$ and $\mathbf{P}_2$, and the probe axis $\mathbf{T}$ as shown in Figure 4.8. Assume that the point $\mathbf{P}_1$ defines $u = 0$ and $v = 0$ and the vectors $(\mathbf{P}_2 - \mathbf{P}_1)$ and $\mathbf{T}$ defines the u and v directions respectively. The position vector of any point on the face can be written by

$$\mathbf{P}(u,v) = \mathbf{P}_1 + u(\mathbf{P}_2 - \mathbf{P}_1) + v \cdot L_1 \cdot \mathbf{T}, \qquad u,v \in [0,1] \tag{4.8}$$

where L_1 is the length of the half-line and this length must be longer than the whole length of the touch trigger probe.

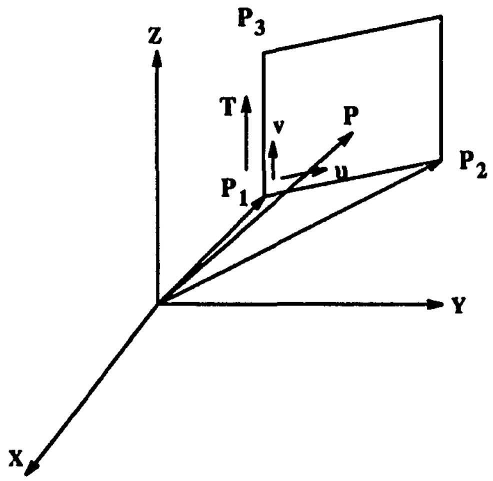


Figure 4.8: A face swept by the probe stylus

The geometry of the swept volume, $S_2 = SV(stylus)$, is easily generated by using the sweeping operation in a solid modeler (Shapes [45]). A collision is detected if the workpiece intersects with the face to be swept by the probe stylus (i.e., $S_2 \cap W \neq \emptyset$). If there is no collision, then the test proceeds to the next level to check for interference with the swept volume of the probe column for the same path segment.

4.2.3.3 Collision detection against the swept volume of probe column

Finally, the collision detection is attempted against the swept volume of probe column on the specified path segment. The probe column has a cylindrical shape so that its swept volume is modeled by a bounding box as shown in Figure 4.9.

The two edges, e_1 and e_2, of the cross-section generate the face respectively by the sweeping operation. One swept area, $SV(e_1)$, is a front face generate by sweeping the edge e_1 and another swept area, $SV(e_2)$, is a back face swept by the edge e_2 along its trajectory. The three independent vectors necessary to define faces are generated

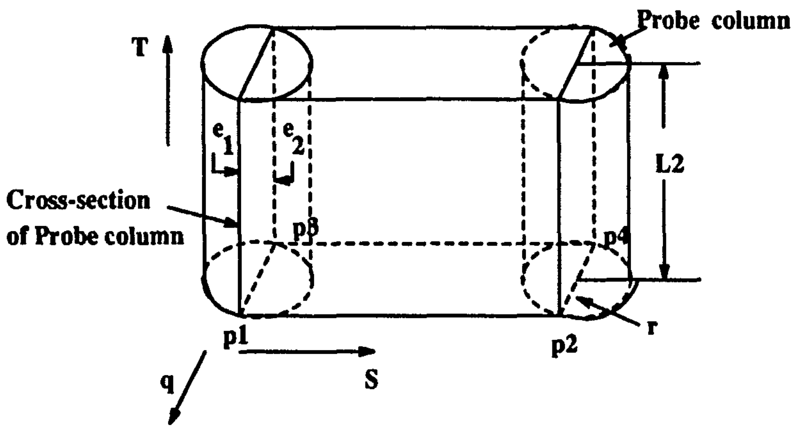


Figure 4.9: Swept volume of probe column

from the direction of the probe axis T and the direction of probe path. The probe path S is calculated as the difference between two prehit points. The cross product of T and S results in a vector q which defines the faces swept by the edges.

The intersection test for the probe column is performed by checking the interference between the swept volumes and the workpiece. Interference exists if

$$SV(e_1) \cap W \neq \emptyset$$

or

$$SV(e_2) \cap W \neq \emptyset$$

If the collision is not detected through the hierarchical collision detection process, then the corresponding path segment is defined as a collision-free path segment.

4.3 Collision avoidance

Avoiding collisions when operating a moving object using a computer program is an important element of path planning. Many different approaches to path planning have been proposed. Previous approaches can be categorized into *potential field, cell decomposition*, and *roadmap* methods [25].

Potential field methods [15, 21, 49] generally employ positive potential fields around obstacles and negative potential fields at the goal position. A path between the start and goal position is constructed by tracking the negated gradient of the total potential. This method will often lead the path to some local minimum from which it cannot escape and therefore cause to find the optimal path. Cell decomposition methods [16, 19, 32] are based on decomposing the tool's free space into simple regions, called cells. The adjacency of these cells is then represented in a connectivity graph which is searched for a path. The outcome of the search is a sequence of cells called a channel. A collision-free path can be obtained from this sequence. The problem is that all the cells must be constructed before a path can be founded; the number of cells tends to grow exponentially with respect to the workpiece geometry. This represents a large pre-processing computation time to obtain a path.

Roadmap approaches [31] attempt to compute the connectivity of the free space in the form of a network of one-dimensional curves, called the *roadmap* [25]. Path planning is reduced to connecting the start and goal position to the roadmap and searching the network for the optimal path. The constructed path is the concatenation of three subpaths: a subpath connecting the start position to the roadmap, a subpath contained in the roadmap, and a subpath connecting the roadmap to the goal position.

The *visibility graph (Vgraph)* method [30] is one of the roadmap approaches. The Vgraph is the non-directed graph whose nodes are the start and goal position and all the obstacle vertices. The *links* of the Vgraph are all the straight line segments connecting two nodes that do not intersect any of the obstacles. The resulting path is a polygonal line connecting the start to goal position through the obstacle vertices. The Vgraph is limited in representation of polygonal obstacles, whereas most obstacles have curved boundaries. The Vgraph can be improved by including links that are tangents to the obstacles resulting in the *tangent graph* [29]. This method can be extended into the higher dimension of geometry by breaking obstacle edges into short segments and using these segments as nodes on the graph [35].

In this research, the Vgraph method with the tangent graph is used to create a safe path around an obstacle. For the collide path segment, the *interference object* is generated by the intersection between the swept volume of the probe and the workpiece. The tangent graph is constructed from the boundary of this interference object and two prehit points (i.e., the start and goal position). A collision free path is obtained from the tangent graph.

4.3.1 Construction of tangent graph

In the tangent graph, the nodes correspond to tangent points on the boundary of the interference object and the edges represent a set of collision-free tangent lines, see Figure 4.10. To find the safe and locally shortest path around the interference object, we propose to generate the *three tangent lines* connecting the start and goal points.

The first tangent line T_1 is the connection line between the start point p_1 and the tangent point t_1, and the second tangent line T_2 is one between the tangent point t_2 and the goal point p_2. Using the bisection method, we decide the third tangent line T_3 such that the distance between two prehit points p_1 and p_2 is minimum.

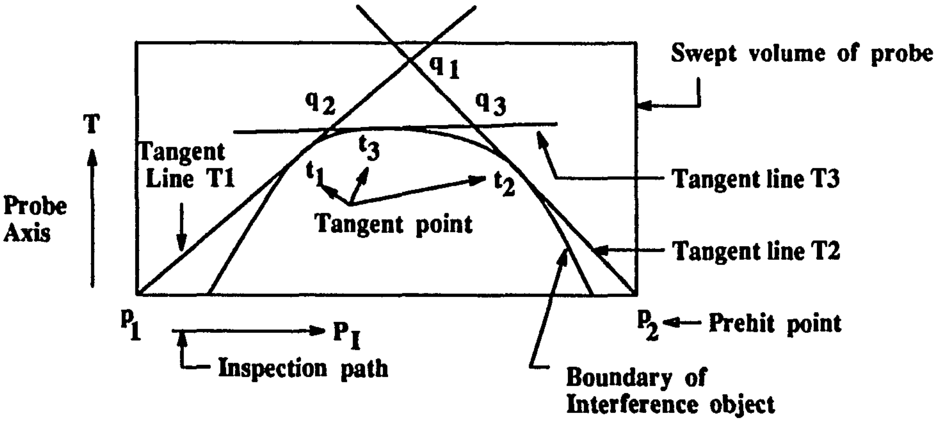


Figure 4.10: Generation of tangent graph

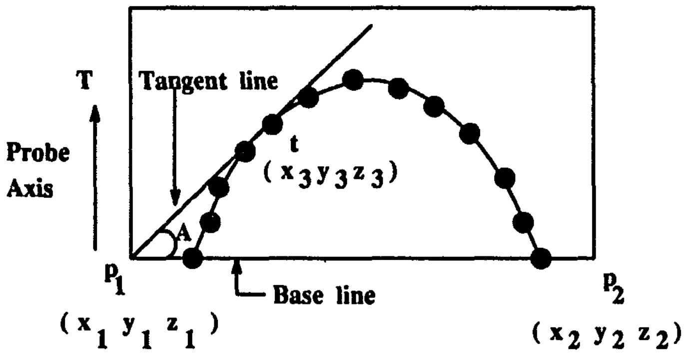


Figure 4.11: Find the tangent line from the start point

In Figure 4.10, the distance between p_1 and p_2 through the intersection point q_1 is definitely longer than the distance between p_1 and p_2 through the via points q_2 and q_3.

We define the boundary of the interference object as the set of points as shown in Figure 4.11. The tangent points are located by considering the set of lines from the prehit point p_1 or p_2 to each point in the boundary. The line with the greatest angle from the base line will be the tangent line. The base line is determined by the direction vector of the inspection path.

Let $L(p,q)$ be a straight line between two points p and q. Consider the two lines $L(p_1,p_2)$ and $L(p_1,t)$. The angle between them, A, is given by

$$A = cos^{-1}\left(\frac{[a_1 \quad a_2 \quad a_3]\cdot[b_1 \quad b_2 \quad b_3]}{|a_1 \quad a_2 \quad a_3||b_1 \quad b_2 \quad b_3|}\right) \tag{4.9}$$

where

$$[a_1 \quad a_2 \quad a_3] = [p_{2x} - p_{1x} \quad p_{2y} - p_{1y} \quad p_{2z} - p_{1z}]$$

$$[b_1 \quad b_2 \quad b_3] = [t_x - p_{1x} \quad t_y - p_{1y} \quad t_z - p_{1z}]$$

Therefore, the tangent line is determined by choosing the line which meets the following condition

$$\max_{a\in A}\{a = \angle(L(p_1,p_2), L(p_i,e_j))\}, \quad i = 1,2 \quad and \quad j = 1,\cdots,n \tag{4.10}$$

where e_j is the j^{th} point on the boundary.

To determine the third tangent line, we apply the bisection method. The objective function is (See Figure 4.10).

$$Minimize \qquad Z = Dist(p_1,q_2) + Dist(q_2,q_3) + Dist(q_3,p_2) \tag{4.11}$$

where $Dist()$ is an Euclidean distance function. For the given interval $s_0 = [t_1, q_1]$, the objective function is evaluated at the midpoint $q_2 = \frac{q_1 - t_1}{2}$. Starting at the point q_2, the tangent line T_3 is generated by choosing the line which has the greatest angle, B, with the first tangent line T_1 (i.e., $\max_{b\in B}\{b = \angle(T_1, L(q_2,e_j)), j = 1,\cdots,n\}$), as shown in Figure 4.12. The tangent point t_3 and the via point q_3 are obtained from the tangent line T_3. If the value of objective function at the point q_2 is less than the initial objective function, then the midpoint is moved to the upper limit

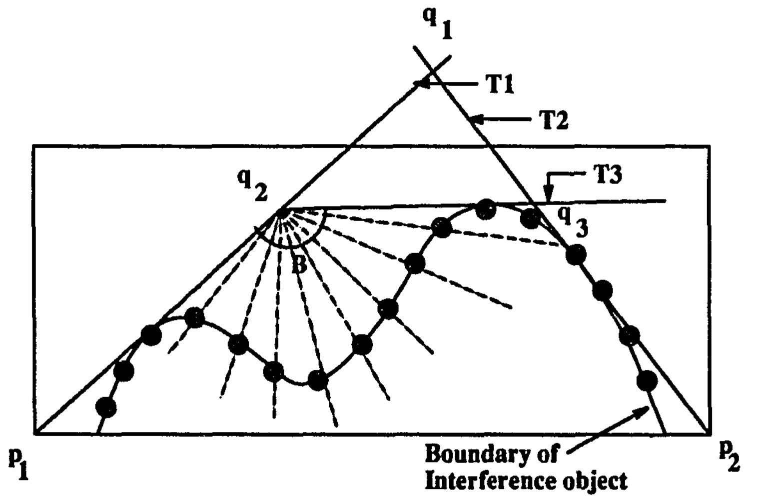


Figure 4.12: Generation of the third tangent line using simulation method

(i.e., $q_1 = q_2$). Otherwise, the midpoint becomes the lower limit (i.e., $t_1 = q_2$). The iteration continues until the interval s_i becomes smaller than the specified tolerance value ϵ. The tangent line T_3 is the connection line between two via points q_2 and q_3 through the tangent point t_3 on the boundary.

The tangent graph consists of the tangent lines T_1, T_2 and T_3 as the edges of graph, and the nodes including the two prehit points, two tangent points and two via points. The collision-free path for the collide path segment is obtained from this tangent graph.

4.3.2 Heuristic methods to avoid collision

The basic procedure of the collision avoidance in this research is dependent on the information about the detected collisions. From the results of collision detection, we propose the heuristic collision avoidance procedure in terms of probe components. Figure 4.13 shows the process of path modification using the heuristic

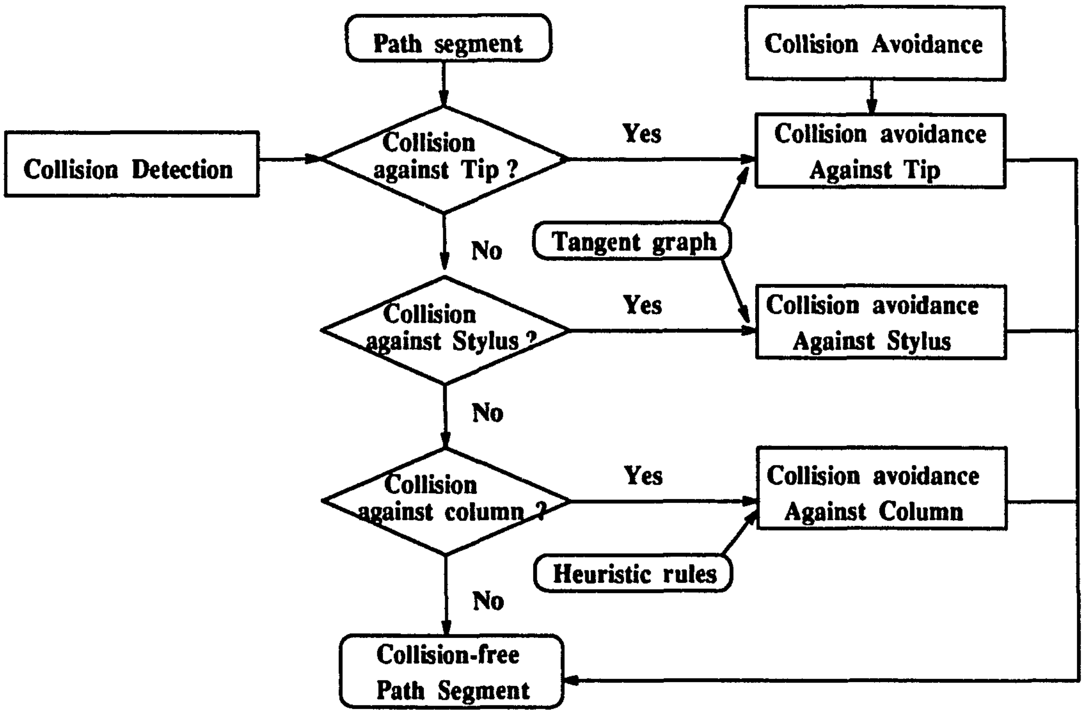


Figure 4.13: Path modification using the heuristic collision avoidance according to the results of the hierarchical collision detection

collision avoidance method for the collide path segment which is resulted according to the hierarchical collision detection. For the collide path segments by the probe tip or probe stylus, the collision-free path is obtained from the tangent graph. If interference is detected against the probe column, the collide path segment is modified according to some heuristic rules.

4.3.2.1 Collision avoidance for the probe tip If interference is detected against the trajectory of the probe tip between two prehit points, then the path must be modified so that it becomes the safe and locally shortest path around obstacles.

In path generation of a probe tip with geometric size, its *configuration space*

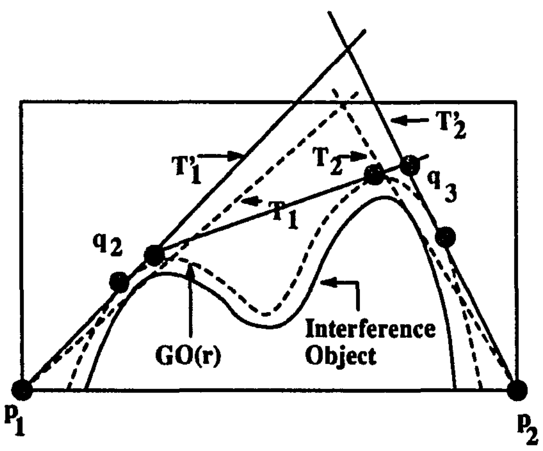


Figure 4.14: Relationship between the growing obstacle and the interference object

(C-space) [30, 31, 51] is usually computed so that it can be processed as a point. The C-space of a probe tip with a radius r can be computed by growing the workpiece surface a radius r. The result of growing workpiece by r will be indicated by $GW(r)$, i.e., the Growing Workpiece by r. As shown in Figure 4.14, the tangent point is located on the boundary of $GW(r)$ instead of the boundary of interference object.

Therefore, the tangent graph is constructed with respect to $GW(r)$, where its boundary is discretized as the set of points, $e_j, j = 1, \cdots, n$. The safe and locally shortest path around the obstacle is obtained from the tangent graph. The modified path is then the path from p_1 to p_2 through the via-points q_2 and q_3.

Notice that the interference object is generated by the intersection between the obstacle and the imaginary surface S_b. The imaginary surface S_b is bounded by the probe axis and the probe trajectory between two points p_1 and p_2.

4.3.2.2 Collision avoidance for the probe stylus The interference of the probe stylus could only happen with a protrusion on the workpiece along the path

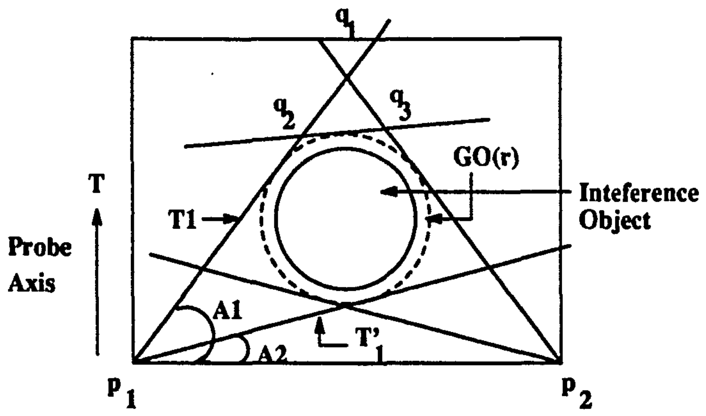


Figure 4.15: Interference object created from the sweeping operation of probe stylus

segment. Therefore, the interference object created from the intersection between the swept volume of probe stylus and the obstacle will be a closed curve as shown in Figure 4.15.

The tangent graph is constructed to find the safe and locally shortest path around the obstacle. We can generate two possible tangent lines T_1 and T_1' from the start point p_1. Intuitively, the line T_1' is not feasible since the probe stylus still collides with the obstacle along this path. Based on the construction rule of tangent graph, the tangent line T_1 is feasible since it has the greatest angle with the base line. That is, T_1 is the line which has the greatest angle such as

$$A_1 = \max_{a \in A}\{a = \angle(L(p_1 p_2), L(p_1, e_j))\} \quad j = 1, \cdots n$$

where e_j is the j^{th} point on the boundary of $GW(r)$. The third tangent line T_3 is also determined by the bisection method. The modified path is the connection line between p_1 and p_2 through the via points q_2 and q_3.

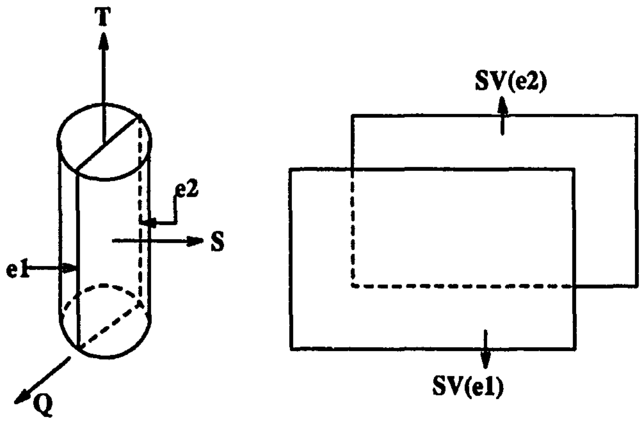


Figure 4.16: Swept volume of probe column

4.3.2.3 Collision avoidance for the probe column The swept volume depends on the probe column's cross section perpendicular to that angle. The cross section can be broken into two components, that on the left of the direction of the movement and that on the right. Figure 4.16 illustrates the cross section of probe column and the swept volume $SV(e_1)$ and $SV(e_2)$.

The cross product of the probe axis **T** and the path direction vector **S** results in a vector **q** which defines the swept volumes $SV(e_1)$ and $SV(e_2)$. That is, the swept volume $SV(e_1)$ is defined by the vector **q** and the point $p_r = p_0 + r_1 * \mathbf{q}$, while $SV(e_2)$ is defined by the vector $-\mathbf{q}$ and the point $p_l = p_0 + r_1 * (-\mathbf{q})$ where r_1 is the radius of probe column.

If interference is detected against the swept volume of probe column, we can consider two possibilities;

$$SV(e_1) \cap GW(r) \neq \emptyset \tag{4.12}$$

or

$$SV(e_2) \cap GW(r) \neq \emptyset \tag{4.13}$$

Based on the previous test, the obstacle cannot intersect with both $SV(e_1)$ and $SV(e_2)$ simultaneously.

Therefore, we can determine the collision avoidance heuristically with respect to the condition of intersection. If $SV(e_1)$ intersects with the obstacle (i.e., $SV(e_1) \cap GW(r) \neq \emptyset$) between the start and goal points p_1 and p_2, then we create the via points on the left side of each point along the vector -**q** slightly more than the radius of probe column such as

$$p_i^l = p_i + (r_1 + \epsilon) * (-\mathbf{q}), \quad i = 1 \quad and \quad 2 \tag{4.14}$$

For interference with $SV(e_2)$, the via points are created on the right side of each point along the vector **q** slightly more than the radius of probe column such as

$$p_i^r = p_i + (r_1 + \epsilon) * \mathbf{q}, \quad i = 1 \quad and \quad 3 \tag{4.15}$$

In Figure 4.17, the solid line is the collide path and the dotted line is the collision-free path between two prehit points through the via points.

The collide path segment related with the probe column can be modified by generating the via points according to the heuristic methods. This modified path is the safe and locally shortest path around the obstacle.

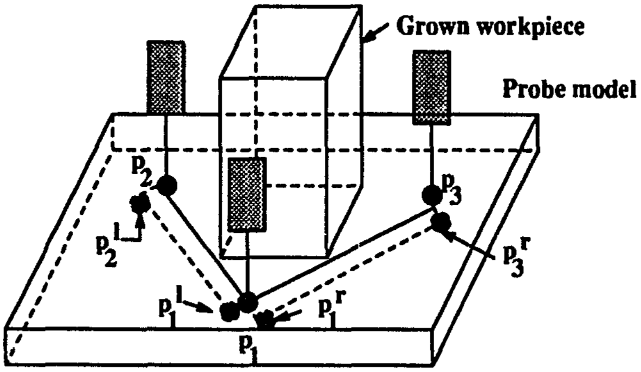


Figure 4.17: Generation of collision-free path in case of collision against the probe column

CHAPTER 5. COMPUTER SIMULATIONS

This chapter demonstrates the algorithms discussed earlier through computer simulation. The simulation was implemented in the C language using the SHAPES geometric computing system on a Silicon Graphics workstation. In this simulation, the dimension of the probe is given in Figure 5.1, where the probe tip diameter is still considered to be negligible.

5.1 Inspection path planning procedure

The inspection path planning procedure described in this research was implemented using the following sequence.

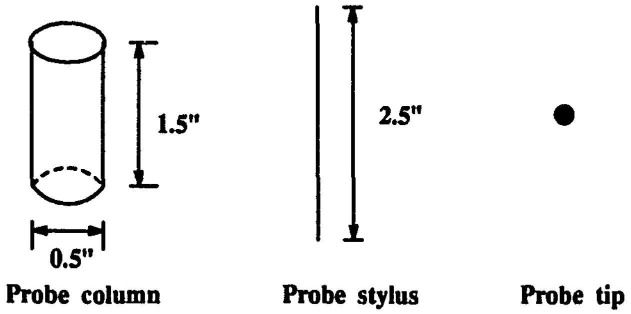


Figure 5.1: Dimensions of the probe components

- Step 1: Input a part model and a set of inspection points.

  - $P = \{p_1, p_2, \cdots, p_m\}$

- Step 2: Generate the VMAP using the method in section 3.2.4.

  - Set the initial radius of hemisphere, $r = 0.1$.
  - Set the step size, $\Delta = 0.5$.
  - $V = \{V_1, V_2, \cdots, V_m\}$.

- Step 3: Generate the adjacency matrix (see section 3.2.7).

- Step 4: Use the multi-echelon SA algorithm (see section 3.3.3) to find the minimal number of clusters and the shortest path.

  - Set the setup cost and the travel cost, $C_s = 120$ sec and $C_t = 1$ sec/inch.
  - Set the annealing schedule.

    $T = 100,\quad \alpha = 0.95,\quad L = 10000,\quad N_A = 5,\quad N_B = 3$ and $N_C = 3$
  - Generate a set of cluster configurations.

    $S_c = \{c_1, c_2, \cdots, c_s\}$
  - Generate the shortest path within each cluster.

    $S_t = \{T_1, T_2, \cdots, T_s\}$

- Step 5: Set the probe orientation as the normal to the surface of the sphere at the centroid of the clustering area.

  - $A = \{A_1, A_2, \cdots, A_s\}$

- Step 6: Set the constrained workpiece orientation and the related probe orientations using the method in section 3.4.3.
  - $D_1 = \{A_{11}, \cdots, A_{1i}\}$
  - $D_2 = \{A_{21}, \cdots, A_{2j}\}$
- Step 7: Set the initial inspection path by connecting the partial paths, T_i, according to the method described in section 4.1
- Step 8: Apply the hierarchical collision detection method (see section 4.2) for each path segment.
- Step 9: Apply the collision avoidance method (see section 4.3) to generate the collision-free path.

5.2 Examples

We have tested the procedure on four different part models. A set of sample data for each part model is given in the appendices. The sample points for each part model surface were generated randomly using a uniform distribution. To generate random points on the surface, we use a cutting plane to generate 2D cross-sections for the surface. For a planar surface, we choose the plane perpendicular with the workpiece surface. For a general surface, we choose one plane from XY, YZ, and XZ planes depending on the surface orientation. The random points are uniformly chosen from the 2D intersection curve between the cutting plane and the workpiece surface, where the center of plane is translated to the centroid of surface and is to be rotated about the axis defined by the normal vector to the workpiece surface. The

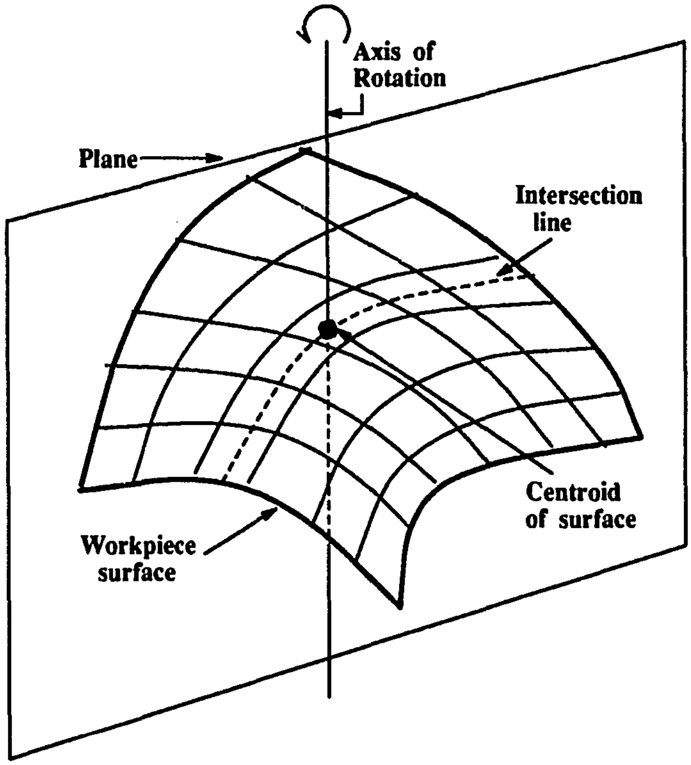


Figure 5.2: Selection of sample points along the intersection line between the workpiece surface and the plane

rotation angle between 0^o to 360^o is treated as a random number from a uniform distribution. Figure 5.2 illustrates the sampling procedure.

The first part model has a rectangular pocket as shown in Figure 5.3 (see Appendix B). For this case, our procedure selected three probe orientations to inspect all the sample points; probe orientation **A1** for the top surface and the pocket, **A2** for the side surface, and **A3** for the other side surface. Figure 5.3 shows the final inspection path for the first part model. The probe is started at the home position and approachs the closest point. The solid lines are the inspection path within a cluster and the dotted lines are the movement between clusters.

The second part model has a rectangular slot as shown in Figure 5.4 (see Appendix C). The three probe orientations selected by our procedure are **A1** for the top surface and the slot, **A2** for the side surface, and **A3** for the other side surface.

A part model with a cylindrical hole is chosen for the third case as shown in Figure 5.5 (see Appendix D). It also requires three probe orientations to inspect all the sample points (see Figure 5.5).

The fourth part model has a Bézier surface. The control points of the surface are given in Appendix E along with the sample points. As shown in Figure 5.6, there are two probe orientations; one for the Bézier surface and one side surface, and one for the other side surface.

The inspection path is classified into two parts; inter-cluster movement and intra-cluster movement. The inter-cluster movement contains the movement between the clusters and the intra-cluster movement contains the movement within the cluster. Table 5.1 shows the probe travel distance through the points. It includes the travel distance for the inter-cluster movement, the travel distance for the intra-cluster movement, and the total travel distance.

From Table 5.1, the travel distance for the intra-cluster movement is monotonically increasing as the sample size increases. However, the travel distance between clusters varies with respect to the workpiece geometry and the number of sample points. The data shows that the travel distance between clusters is in general decreasing as the sample size increases. This is due to the possible reduction of the distance between clusters as the sample size increases.

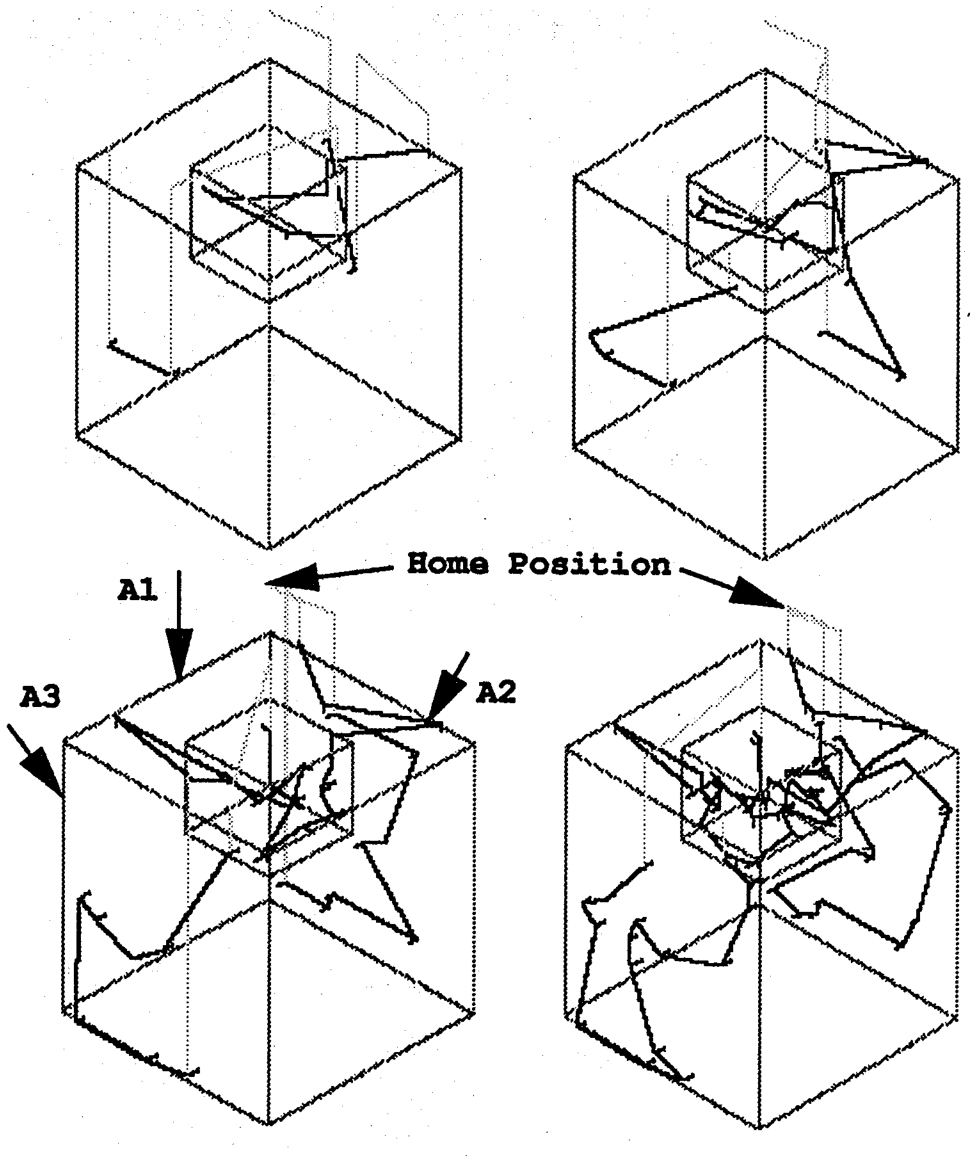


Figure 5.3: Inspection path generation for part model 1 (Sample size: n=10, n=20, n=40, n=80)

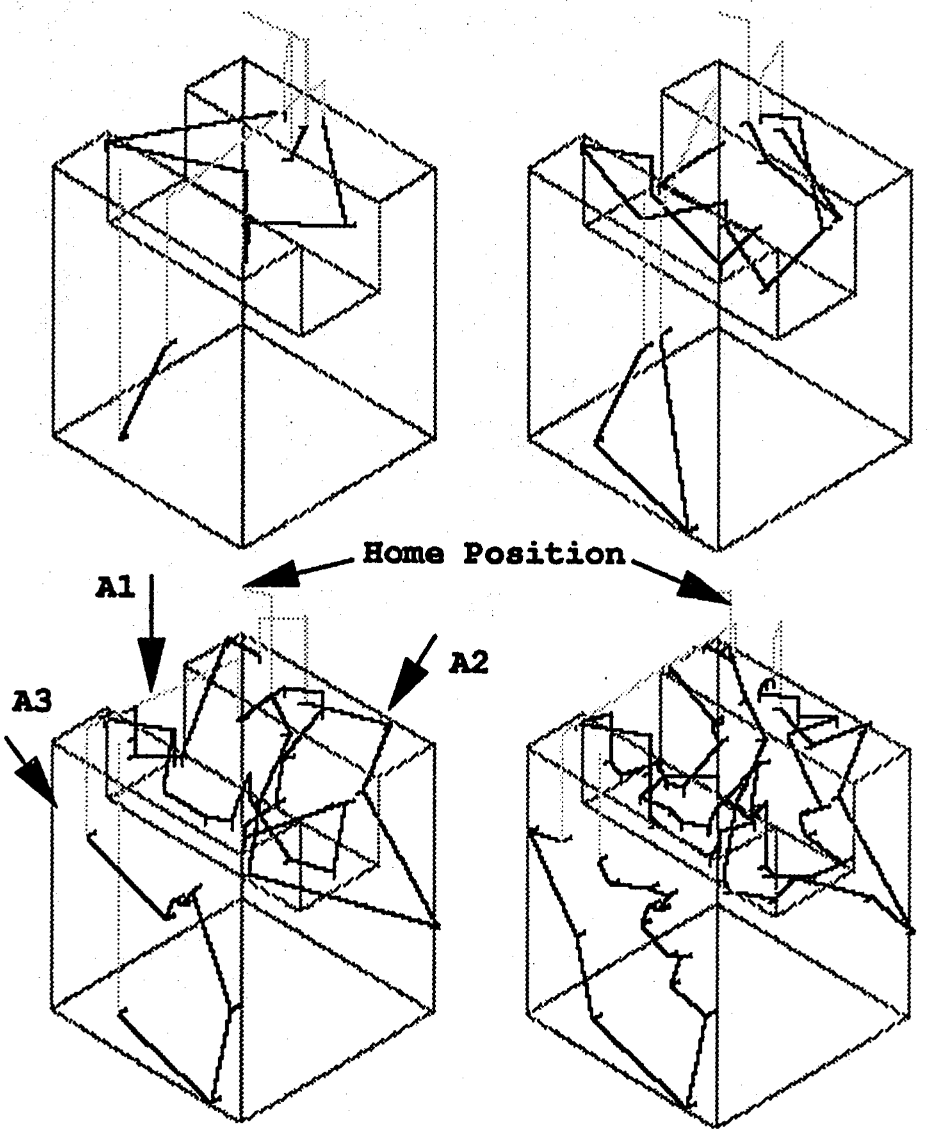


Figure 5.4: Inspection path generation for part model 2 (Sample size: n=10, n=20, n=40, n=80)

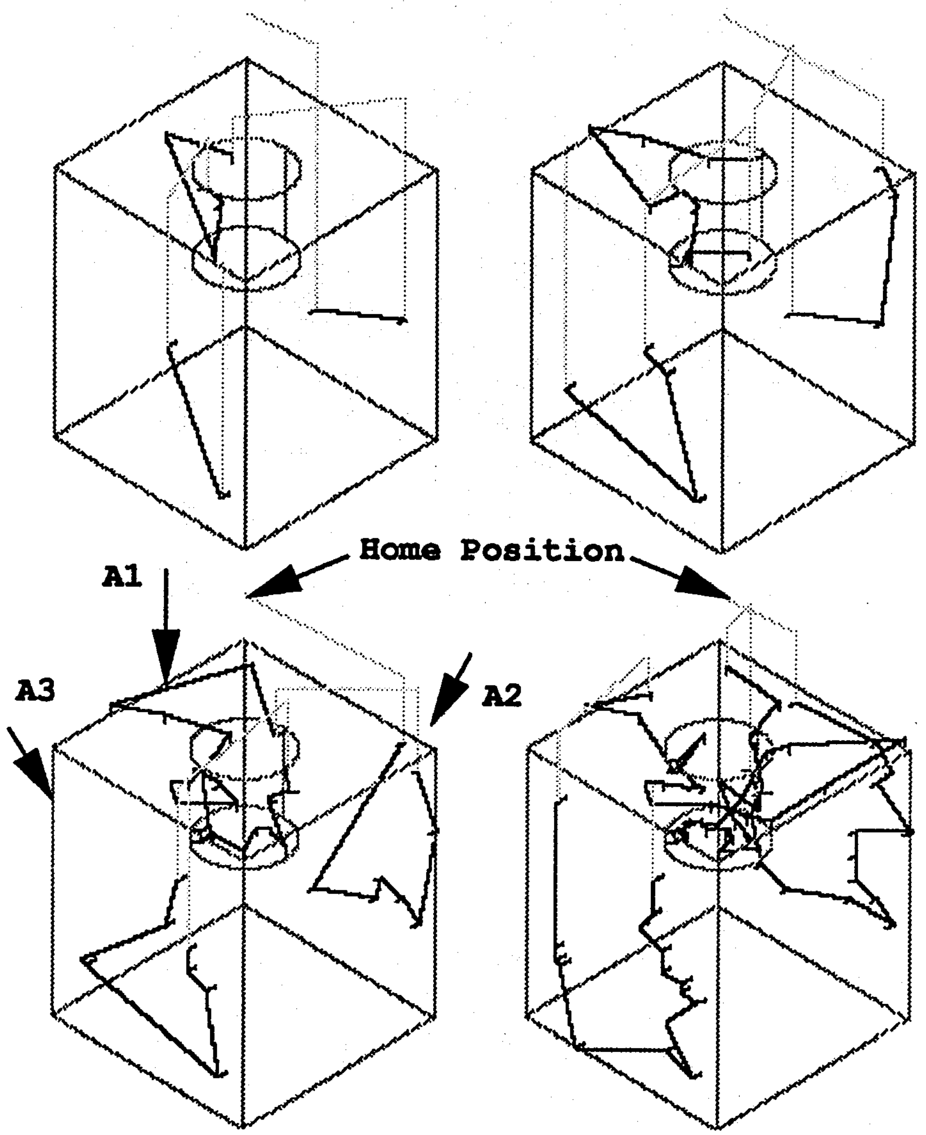


Figure 5.5: Inspection path generation for part model 3 (Sample size: n=10, n=20, n=40, n=80)

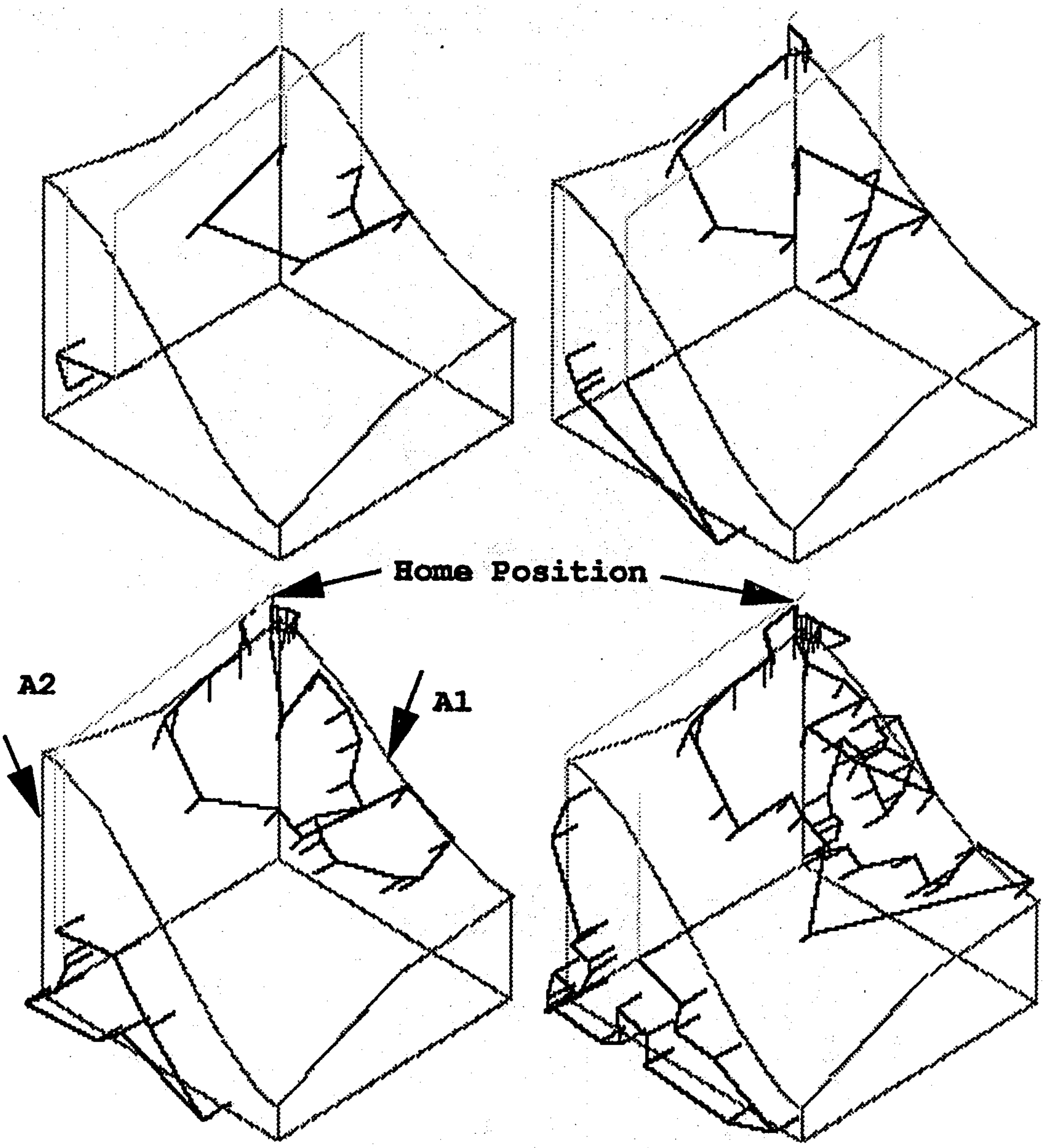


Figure 5.6: Inspection path generation for part model 4 (Sample size: n=10, n=20, n=40, n=80)

Since the optimal workpiece orientations can be determined by the setup cost and the travel distance, it will be affected by the amount of setup cost and the travel distance. Referring to equation 3.9 (see section 3.3.1), the objective function in terms of the setup cost and the travel cost for a fixed probe orientation is given by

$$Z_1 = N_s * C_s + \sum_{i=1}^{N_s} D_i * C_t \tag{5.1}$$

where

- C_s: Setup cost
- N_s: Minimum number of setups
- D_i: Travel distance within i^{th} cluster
- C_t: Travel cost / unit distance

and the penalty cost is excluded since it is a feasible solution. The travel distance for the inter-cluster movement is not included in equation 5.1.

When we use a flexible probe (which has two DOFs) instead of the fixed probe, only one setup is required so the objective function is reduced to

$$Z_2 = C_s + D_T * C_t \tag{5.2}$$

where D_T is the total travel distance (i.e., the sum of the travel distances for the inter- and intra-cluster movement).

We arbitrarily assign the probe travel cost per unit distance a value of 1 sec/inch and vary the setup cost depending on the part model. Table 5.2 shows the variation of objective values, Z_1 and Z_2, by changing the setup cost for part models 1 and

4. For part model 1, the inspection time is dominated by the setup time. As would be expected, the inspection time with a single setup (Z_2) is much shorter than the inspection time with a multi-setup (Z_1). The effect of setup time is diminished for a part model with larger dimensions as in case 4. For the small amount of setup time (e.g., $C_S = 30$ sec), the inspection time for a single probe (Z_2) becomes higher because of the travel time between clusters. Note that the differences in Z_1 and Z_2 for different setup times is simply the total setup time difference (i.e., the inspection paths have not changed). For case 4 there is a reversal for $C_S = 30$ sec where Z_1 is less than Z_2. This is due to the setup time being less than the travel time between clusters.

Table 5.1: Comparison of the probe travel distance for the inter- and intra-cluster movement

Test block	Sample size	Inter-cluster	Intra-cluster	Total
case 1	m=10	26.1271	10.4251	36.5522
	m=20	22.9369	23.3615	46.2984
	m=40	25.454	42.8634	68.3174
	m=80	16.6874	62.5425	79.2299
case 2	m=10	23.2844	14.0648	37.3492
	m=20	18.1234	31.0273	49.1507
	m=40	19.8027	47.1773	66.98
	m=80	15.2956	64.9997	80.2953
case 3	m=10	29.1209	12.0104	41.1313
	m=20	28.9627	24.5141	53.4768
	m=40	21.8412	43.5399	65.3811
	m=80	16.2653	59.7396	76.0049
case 4	m=10	73.1467	44.1726	117.3193
	m=20	59.4436	100.4994	159.943
	m=40	49.2714	137.5552	186.8266
	m=80	38.4269	212.2024	250.6293

Table 5.2: Variation of objective values with respect to the setup cost

		Cs = 120 sec		Cs = 60 sec		Cs = 30 sec	
		Z1	Z2	Z1	Z2	Z1	Z2
case 1	m=10	370.4251	156.5522	190.4251	96.5522	100.4251	66.5522
	m=20	383.3615	166.2984	203.3615	106.2984	113.3615	76.2984
	m=40	402.8634	188.3174	222.8634	128.3174	132.8634	98.3174
	m=80	422.5425	199.2299	242.5425	139.2299	152.5425	109.2299
case 4	m=10	284.1726	237.3193	164.1726	177.3193	104.1726	147.3193
	m=20	340.4994	279.943	220.4994	219.943	160.4994	189.943
	m=40	377.5552	306.8266	257.5552	246.8266	197.5552	216.8266
	m=80	452.2024	370.6293	332.2024	310.6293	272.2024	280.6293

CHAPTER 6. CONCLUSIONS

This research presents an automated dimensional inspection process for CMMs using collision detection and modification of inspection path based on accessibility analysis. The main contributions of this research are; 1) it provides interference-free contact of the probe model on the surface at the results of accessibility analysis, 2) it minimizes the number of workpiece orientations and the related probe orientations by clustering the inspection points based on the information of VMAPs, 3) it detects collision of the trajectories of the probe model, and 4) it modifies the collide path segment to avoid the collision.

The set of algorithms discussed in this research can be used to generate the inspection path planning for a general workpiece surface. The accessibility analysis and the collision-free path generation have been tested on a range of the workpiece geometries and have been found to be successful. Application of this procedure during the inspection planning phase can help to reduce the human interface to determine the probe orientations and the sequence of inspection points to be visited by the probe without interference with the workpiece geometry. The efficiency of the inspection process has been enhanced in two areas; 1) the inspection time can be reduced by finding the minimal number of the workpiece orientations and the probe orientations, and by computing the shortest path through the points. 2) the generated path is the safe path, because each path segment is verified respectively using the hierarchical collision detection method and then the collide path segment is modified according to the heuristic collision avoidance method.

We have assumed that the probe can be abstracted as a straight line to compute a VMAP for a point. In certain cases, the points on the surface may be accessible to a straight line but not to a real probe. An additional verification step is needed to compensate for the dimension of a real probe.

For computation of the shortest travel distance for minimal clustering problem, we simplified the problem by considering only the Euclidean distance between two points instead of the distance along the surface. Therefore, the shortest path from the TSP algorithm may not be the shortest travel path.

To generate the collision avoidance path for a collide path segment, we only considered the path on the plane which is specified by the two vectors; the probe orientation and the probe path. Since there is an infinite number of paths between two points, the final path may not be the shortest path. We only compute a locally shortest path between two points on the plane.

More research needs to be done to make these algorithms practical for industrial applications. First, a more general method to generate a VMAP with respect to the geometry of the workpiece and the probe must be investigated. Second, the shortest inspection path must be computed based on the three-dimensional distance between two points on the workpiece surface. Finally the heuristic method to avoid a collision needs more exploration.

BIBLIOGRAPHY

[1] Aarts, E.H.L. and van Laarhoven, P.J.M.: Statistical cooling: A general approach to combinatorial optimization problems, *Philips Journal of Research*, Vol. 40, No.4, pp.193-226, 1985.

[2] Aziz, N.M. and Bhat, S.: Bézier surface/surface intersection, *IEEE Computer Graphics and Applications*, pp.50-58, 1990.

[3] Boyse, J.W.: Interference detection among solids and surfaces, *Communications of the ACM*, Vol.22, No.1, pp.3-9, 1979.

[4] Brooks, R.: Solving the find-path problem by good representation of free space, *IEEE Transactions on System, Man, and Cybernetics*, Vol.SMC-13, No.3, pp.190-197, 1983.

[5] Cameron, S.: A study of the clash detection problem in robotics, *IEEE International Conference on Robotics and Automation*, Vol.1, pp.488-493, 1985.

[6] Cameron, S.: Collision detection by four-dimensional intersection testing, *IEEE Transaction on Robotics and Automation*, Vol.6, No.3, pp.291-302, 1990.

[7] Canny, J.: On detecting collisions between moving polyhedra, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.8, pp.200-209, 1986.

[8] Chen, L. L. and Woo, T. C.: Computational geometry on the sphere with application to automated machining, *ASME Advances in Design Automation*, DE-Vol. 23-1, pp.165-174, 1990.

[9] Cohen, E., Lyche, T., and Riesenfeld, R.: Discrete B-spline and subdivision techniques in computer aided geometric design and computer graphics, *Computer Graphics and Image Processing*, Vol.14, No.2, pp.87-111, 1988.

[10] Croes, A.: A method for solving traveling salesman problems, *Operations Research*, Vol. 5, pp.791-812, 1958.

[11] Ganter, M.A. and Uicker, J. J., Jr.: Dynamic collision detection using swept solids, *ASME Journal of Mechanisms, Transmissions and Automation*, Vol.108, No.4, 1986.

[12] Haghpassand, K. and Oliver, J.: Computational geometry for optimal workpiece orientation, *Advanced in Design Automation*, Vol.2, pp.169-175, 1991.

[13] Hajek, B.: Cooling schedules for optimal annealing, *Mathematics of Operations Research*, Vol.13, No.2, pp.311-329, 1988.

[14] Hanna, S. L., Abel, J. F., and Greenberg, D. P.: Intersection of parametric surfaces by means of look-up tables, *IEEE Computer Graphics and Applications*, Vol.3, No.7, pp.39-48, 1983.

[15] Hawang, Y. K. and Ahuja, N.: A potential field approach to path planning, *IEEE Transactions on Robotics and Automation*, Vol.8, No.1, pp.23-32, 1992.

[16] Hayward, V.: Fast collision detection scheme by recursive decomposition of a manipulator workspace, *IEEE International Conference on Robotics and Automation*, Vol.2, pp.1044-1049, 1986.

[17] Hilbert, D. and Cohen-Vossen, S.: *Geometry and the imagination*, Chelsea Publishing Company, New York, 1952.

[18] Hoffman, C. M.: *Geometric and solid modeling: An introduction*, Morgan Kaufmann Publishers Inc., San Mateo, Calif., 1989.

[19] Jun, S. T. and Shin, K. G.: Shortest path planning in discretized workspaces using dominance relation, *IEEE Transactions on Robotics and Automation*, Vol.7, No.3, pp.342-350, 1991.

[20] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P.: Optimization by simulated annealing, *Science*, Vol. 220, pp.671-679, 1983.

[21] Koditschek, D. E.: Robot planning and control via potential functions, in *The Robotics Review 1*, edited by Khatib, O., Craig, J. J. and Lozano-Pérez, T., MIT Press, Cambridge, Mass., 1989.

[22] Kuratowski, K. and Mostowski, A.: *Set theory*, North-Holland, Amsterdam, 1976.

[23] Lane, J. M. and Riesenfeld, R. F.: A theoretical development for the computer generation of piecewise polynomial surfaces, *Pattern Analysis and Machine Intelligence*, Vol.2, No.1, pp.35-46, 1980.

[24] Lasser, D.: Intersection of parametric surfaces in the Bernstein-Bézier representation, *Computer-Aided Design*, Vol.18, No.4, pp.162-192, 1988.

[25] Latombe, J. C.: *Robot motion planning*, Kluwer Academic Publishers, Boston, 1991.

[26] Lee, J. W., Cho, J. H., and Kim, K.: Optimal probe path generation for sculptured surface in the coordinate measuring machine, *Transactions of the North American Manufacturing Research Institution of SME*, Vol.18, No.4, pp.162-192, 1994.

[27] Lin, S.: Computer solutions of the traveling salesman problem, *The Bell System Technical Journal* , Vol.44, pp.2245-2269, 1965.

[28] Lin, S. and Kernighan, B. W.: An effective heuristic algorithms for the traveling salesman problem, *Operations Research* , Vol.21, pp.498-516, 1973.

[29] Liu, Y. H. and Arimoto, S.: Proposal of tangent graph and extended tangent graph for path planning of mobile robots, *IEEE International Conference on Robotics and Automation*, pp.312-317, 1991.

[30] Lozano-Pérez, T. and Wesley, M. A.: An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM*, Vol.22, No.10, pp.560-570, 1979.

[31] Lozano-Pérez, T.: Spatial planning : A configuration space approach, *IEEE Transactions on Computers*, Vol. C-32, No.2, pp.108-120, 1983.

[32] Lozano-Pérez, T.: A simple motion-planning for general robot manipulators, *IEEE Transactions of Robotics and Automation*, Vol. RA-3, No.3, pp224-238, 1987.

[33] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E.: Equation of state calculations by fast computing machines, *The Journal of Chemical Physics*, Vol.21. pp1087-1090, 1953.

[34] Mortenson, M. E.: *Geometric modeling*, John Wiley & Sons, Inc., New York, 1985.

[35] Papadimitriou, C. M.: An algorithm for shortest path motion in three dimension, *Information Processing Letters*, Vol.20, pp259-263, 1985.

[36] Park, H. D. and Mitchell, O. R.: CAD based planning and execution of inspection, *IEEE Computer Vision and Pattern Recognition*, pp.858-863, 1988.

[37] Paul, R.: Manipulator Cartesian path control, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-9, No.1, pp.702-711, 1979.

[38] Peng, Q. S.: An algorithm for finding the intersection lines between two B-spline surfaces, *Computer-Aided Design*, Vol.16, No.4, pp.191-196, 1984.

[39] Pincus, M.: A Monte-Carlo method for the approximate solution of certain types of constrained optimization problems, *Operations Research*, Vol.18, pp.1225-1228, 1970.

[40] Preparata, F. D. and Shamos, M. L.: *Computation geometry, An introduction*, Springer-Verlag, New York, 1985.

[41] Requicha, A. A. G.: Representation for rigid solids: Theory, methods, and systems, *Computing Surveys*, Vol.12, No.4, pp.437-465, 1980.

[42] Rockafellar, R. T.: *Convex analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[43] Rossignac, J. R. and Requicha, A. A. G.: Offsetting operations in solid modeling, *Computer Aided Geometric Design*, Vol.3, pp.129-148, 1985.

[44] Shaffer, A. C. and Herb, G. M.: A real-time robot arm collision avoidance system, *IEEE Transactions on Robotics and Automation*, Vol.8, No.2, pp.149-160, 1992.

[45] *Shapes Reference Manual: A system for computing with geometric objects*, Version 2.0.8, XOX Corporation, Minneapolis, Minnesota, 1993.

[46] Spyridy, A. J. and Requicha, A. A. G.: Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines, *IEEE International Conference on Robotics and Automation*, Vol.2, pp.1284-1289, 1990.

[47] Su, C. J. and Mukerjee, A: Automated machinability checking for CAD/CAM, *IEEE Journal of Robotics and Automation*, October, pp.691-698, 1991.

[48] Tang, K., Woo, T., and Gan, J.: Maximum intersection of spherical polygons and workpieces orientation for 4- and 5-axis machining, *Journal of Mechanical Design* Vol.114, pp.477-485, 1992.

[49] Tilove, R. B.: Local obstacle avoidance for mobile robots based on the method of artificial potentials, *IEEE International Conference on Robotics and Automation*, Vol.1, pp.566-571, 1990.

[50] Tseng, Y. J. and Joshi, S.: Determining feasible tool-approach directions for machining Bézier curve and surface, *Computer Aided Design*, Vol.23, No.5, pp.367-379, 1991.

[51] Udupa, S. M.: Collision detection and avoidance in computer controlled manipulators, *5th International Joint Conference on Artificial Intelligence* , pp.737-748, 1977.

[52] Walker, I. and Wallis, A. F.: Application of 3-D solid modeling to coordinate measuring inspection, *Int. J. Mach. Tools Manufact* , Vol.32, No.12, pp.195-201, 1992.

[53] Wang, D. and Haman, Y.: Optimal trajectory planning of manipulators with collision detection and avoidance, *The International Journal of Robotics Research*, Vol.11, No.5, pp.460-468, 1992.

[54] Wang, W. P. and Wang, K. K.: Geometric modeling for swept volume of moving solids, *IEEE Computer Graphics and Applications*, Vol.6, No.12, pp.8-17, 1986.

[55] Woo, T. and Turkovich, B. F.: Visibility map and its application to numerical control, *Annals of the CIRP*, Vol.39, No.1, pp.451-454, 1990.

[56] Yau, H. T. and Menq, C. H.: Path planning for automated inspection using coordinate measuring machines, *IEEE International Conference on Robotics and Automation*, Vol.3, pp.1934-1939, 1991.

ACKNOWLEDGEMENTS

I would like to express my gratitude to many persons who assisted me in this research and in the completion of my Ph.D program. Dr. John Jackman, my major professor, deserves special recognition for his guidance, patience, and encouragement throughout the course of this research. Without his assistance and consideration, this research never would have been accomplished.

I also would like to thank my committee members: Dr. Jim Oliver, Dr. Stephen Vardeman, Dr. Robert Stephenson, Dr. Raymond Cheung, all of whom provided valid criticisms and comments concerning my final dissertation.

Last, but most importantly, I want to acknowledge the members of family. First, I'd like to thank may parents for providing me with a lot of valuable encouragement and support throughout.

Next, I'd like to thank my sons, Jae-Hyeon and Jae-Young, for never complaint about having a father who was frequently preoccupied. I can't say enough to thank my wife, Hwa-Sil. She was always there to give me what I needed most, care, forgiveness, love. She was endured so much and asked so little. I do always love her and dedicate this work.

To all of these, other family members, and other friends, I am most humbly grateful.

APPENDIX A. REGULARIZED SET OPERATION

Let W be a set and T a topology on W, that is, the collection of all open subset of W. In the topological space (W, T) a subset X of W is a (closed) regular set if it equals the closure of its interior, that is,

$$X = kiX$$

where k and i denotes, respectively, closure and interior. The regularized set union ($\cup^*$), interior ($\cap^*$), difference ($-^*$), and complement (c^*) of two subsets X and Y of W are defined as

$$\begin{aligned} X \cup^* Y &= ki(X \cup Y) \\ X \cap^* Y &= ki(X \cap Y) \\ X - {*}Y &= ki(X - Y) \\ c^* X &= ki(cX) \end{aligned}$$

where c denotes the usual complement with respect to W.

APPENDIX B. CASE 1

Dimensions of part model

- Pocket: $2 \times 2 \times 2$ inches
- Part model: $5 \times 5 \times 6$ inches

Table B.1: Sample size = 10

NO	X	Y	Z	NO	X	Y	Z
1	0.0	2.1047	2.1937	6	3.2279	1.5	5.4103
2	0.0	1.4047	4.7691	7	1.8814	2.3065	4.0
3	5.0	2.6847	2.8175	8	1.8061	3.5	5.4871
4	5.0	1.0385	2.6186	9	1.5	3.4351	4.5186
5	0.1584	4.3221	6.0	10	3.5	2.2340	5.5787

Table B.2: Sample size = 20

NO	X	Y	Z	NO	X	Y	Z
1	0.0	2.1047	2.1937	11	0.0	3.4351	0.7325
2	0.0	1.4047	4.7691	12	0.0	1.3967	0.7114
3	5.0	2.6847	2.8175	13	5.0	0.6375	2.9278
4	5.0	1.0385	2.6186	14	5.0	4.2661	5.7585
5	0.1584	4.3221	6.0	15	3.2279	1.5	5.4103
6	1.1412	2.4941	6.0	16	3.4766	1.5	5.2218
7	1.8814	2.3065	4.0	17	1.8061	3.5	5.4871
8	1.5119	3.1075	4.0	18	2.4975	3.5	5.8905
9	1.5	3.4351	4.5186	19	3.5	2.2340	5.5787
10	1.5	2.3125	4.1884	20	3.5	3.4670	5.7849

Table B.3: Sample size = 40

NO	X	Y	Z	NO	X	Y	Z
1	0.0	2.1047	2.1937	21	0.0	0.1855	0.3572
2	0.0	1.4047	4.7691	22	0.0	1.2812	0.3942
3	0.0	3.4351	0.7325	23	0.0	3.4225	4.8820
4	0.0	1.3967	0.7114	24	0.0	2.6168	2.5545
5	5.0	2.6847	2.8175	25	5.0	1.9523	2.2057
6	5.0	1.0385	2.6186	26	5.0	2.2562	0.1125
7	5.0	0.6375	2.9278	27	5.0	3.2739	0.3104
8	5.0	4.2661	5.7585	28	5.0	0.5169	0.1070
9	0.1584	4.3221	6.0	29	0.0362	0.7034	6.0
10	1.1412	2.4941	6.0	30	4.0122	0.2952	6.0
11	3.2279	1.5	5.4103	31	1.6990	1.5	5.4796
12	3.4766	1.5	5.2218	32	2.6909	1.5	4.8237
13	1.8814	2.3065	4.0	33	3.1764	3.1357	4.0
14	1.5119	3.1075	4.0	34	1.6807	1.7166	4.0
15		3.5	5.4871	35	1.8061	3.5	5.1428
16	2.4975	3.5	5.8905	36	2.4611	3.5	4.5159
17	1.5	3.4351	4.5186	37	1.5	3.0504	4.9252
18	1.5	2.3125	4.1884	38	1.5	2.9742	4.7713
19	3.5	2.2340	5.5787	39	3.5	3.1329	4.1420
20	3.5	3.4670	5.7849	40	3.5	3.0289	5.3428

Table B.4: Sample size = 80

NO	X	Y	Z	NO	X	Y	Z
1	0.0	2.1047	2.1937	41	0.0	1.3081	3.8841
2	0.0	1.4047	4.7691	42	0.0	3.7689	1.5043
3	0.0	3.4351	0.7325	43	0.0	4.5583	4.4974
4	0.0	1.3967	0.7114	44	0.0	0.8337	0.1294
5	0.0	0.1855	0.3572	45	0.0	1.8165	4.6544
6	0.0	1.2812	0.3942	46	0.0	2.6606	2.3449
7	0.0	3.4225	4.8820	47	0.0	4.5712	4.3679
8	0.0	2.6168	2.5545	48	0.0	2.4772	4.0292
9	5.0	2.6847	2.8175	49	5.0	2.2182	4.5469
10	5.0	1.0385	2.6186	50	5.0	4.2339	3.3758
11	5.0	0.6375	2.9278	51	5.0	1.8434	2.7124
12	5.0	4.2661	5.7585	52	5.0	4.8877	4.9501
13	5.0	1.9523	2.2057	53	5.0	2.0247	3.2130
14	5.0	2.2562	0.1125	54	5.0	1.3095	3.3583
15	5.0	3.2739	0.3104	55	5.0	2.5533	0.5461
16	5.0	0.5169	0.1070	56	5.0	4.9613	5.5244
17	0.1584	4.3221	6.0	57	3.2667	4.9901	6.0
18	1.1412	2.4941	6.0	58	0.4923	3.5443	6.0
19	0.0362	0.7034	6.0	59	1.5593	4.3531	6.0
20	4.0122	0.2952	6.0	60	1.1132	2.2489	6.0
21	3.2279	1.5	5.4103	61	1.7594	1.5	5.3600
22	3.4766	1.5	5.2218	62	2.9686	1.5	4.6428
23	1.6990	1.5	5.4796	63	2.1598	1.5	4.3719
24	2.6909	1.5	4.8237	64	2.7792	1.5	5.0879
25	1.8814	2.3065	4.0	65	1.8026	1.8980	4.0
26	1.5119	3.1075	4.0	66	2.3148	2.1460	4.0
27	3.1764	3.1357	4.0	67	3.4414	2.5126	4.0
28	1.6807	1.7166	4.0	68	3.4892	2.8372	4.0
29	1.8061	3.5	5.4871	69	2.6940	3.5	5.3002
30	2.4975	3.5	5.8905	70	3.3840	3.5	5.8248
31	2.7036	3.5	5.1428	71	1.7160	3.5	5.5793
32	2.4611	3.5	4.5159	72	1.5871	3.5	4.3374
33	1.5	3.4351	4.5186	73	1.5	2.3052	4.9881
34	1.5	2.3125	4.1884	74	1.5	2.5882	5.1749
35	1.5	3.0504	4.9252	75	1.5	3.1964	5.4863
36	1.5	2.9742	4.7713	76	1.5	2.7680	4.7836
37	3.5	2.2340	5.5787	77	3.5	1.5949	4.9567
38	3.5	3.4670	5.7849	78	3.5	2.2119	4.9239
39	3.5	3.1329	4.1420	79	3.5	1.8124	5.2021
40	3.5	3.0289	5.3428	80	3.5	1.8106	4.4029

APPENDIX C. CASE 2

Dimensions of part model

- Slot: $2 \times 6 \times 2$ inches
- Part model: $5 \times 5 \times 6$ inches

Table C.1: Sample sixe = 10

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.4595	5.0875	6	5.0	1.9687	0.9572
2	0.0	1.0415	4.1760	7	5.0	3.2665	3.7614
3	0.9282	1.9696	6.0000	8	3.9578	0.3927	6.0000
4	0.5134	2.6089	6.0000	9	1.5	4.4085	5.2235
5	2.7843	2.8737	4.0	10	3.5	3.4855	5.6845

Table C.2: Sample size = 20

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.4595	5.0875	11	5.0	1.9687	0.9572
2	0.0	1.0415	4.1760	12	5.0	3.2665	3.7614
3	0.0	0.5564	4.7682	13	5.0	3.7192	4.3198
4	0.0	2.9708	3.8659	14	5.0	4.4221	0.2326
5	0.9282	1.9696	6.0000	15	4.7734	2.7847	6.0000
6	0.5134	2.6089	6.0000	16	3.9578	0.3927	6.0000
7	1.5	4.4085	5.2235	17	2.7843	2.8737	4.0
8	1.5	1.5995	5.6609	18	1.8249	0.2612	4.0
9	1.5	2.5636	4.2865	19	3.5	4.5609	4.8094
10	3.5	3.4855	5.6845	20	3.5	1.5563	5.3820

Table C.3: Sample size = 40

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.4595	5.0875	21	5.0	1.9687	0.9572
2	0.0	1.0415	4.1760	22	5.0	3.2665	3.7614
3	0.0	0.5564	4.7682	23	5.0	3.7192	4.3198
4	0.0	2.9708	3.8659	24	5.0	4.4221	0.2326
5	0.0	4.9018	1.7770	25	5.0	4.9449	2.5715
6	0.0	0.0135	0.4621	26	5.0	1.1396	4.5342
7	0.0	3.6623	5.6826	27	5.0	3.9224	4.7582
8	0.0	0.2862	2.2957	28	5.0	3.4636	4.2963
9	0.9282	1.9696	6.0000	29	4.7734	2.7847	6.0000
10	0.5134	2.6089	6.0000	30	3.9795	0.2766	6.0000
11	0.9028	2.9613	6.0000	31	3.9578	0.3927	6.0000
12	0.5329	0.1778	6.0000	32	4.8266	3.8783	6.0000
13	0.5077	0.9208	6.0000	33	4.5787	4.3019	6.0000
14	1.5	4.4085	5.2235	34	2.7843	2.8737	4.0
15	1.5	1.5995	5.6609	35	1.8249	0.2612	4.0
16	1.5	2.5636	4.2865	36	2.5404	2.6154	4.0
17	1.5	2.7945	5.3952	37	2.3687	4.7798	4.0
18	1.5	3.8158	4.7952	38	3.5	1.5563	5.3820
19	3.5	3.4855	5.6845	39	3.5	0.5447	5.1424
20	3.5	4.5609	4.8094	40	3.5	0.5313	5.5343

Table C.4: Sample size = 80

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.4595	5.0875	41	0.0	1.3020	0.6651
2	0.0	1.0415	4.1760	42	0.0	2.2970	4.6807
3	0.0	0.5564	4.7682	43	0.0	1.8562	1.4413
4	0.0	2.9708	3.8659	44	0.0	0.1912	0.2965
5	0.0	4.9018	1.7770	45	0.0	1.9116	4.7787
6	0.0	0.0135	0.4621	46	0.0	0.1452	5.7898
7	0.0	3.6623	5.6826	47	0.0	3.7707	5.8809
8	0.0	0.2862	2.2957	48	0.0	3.8696	2.0227
9	5.0	1.9687	0.9572	49	5.0	1.5488	2.6080
10	5.0	3.2665	3.7614	50	5.0	2.4088	4.2293
11	5.0	3.7192	4.3198	51	5.0	3.4343	4.6724
12	5.0	4.4221	0.2326	52	5.0	3.9793	2.7364
13	5.0	4.9449	2.5715	53	5.0	2.1136	4.5852
14	5.0	1.1396	4.5342	54	5.0	0.2586	4.3483
15	5.0	3.9224	4.7582	55	5.0	4.2854	3.5271
16	5.0	3.4636	4.2963	56	5.0	3.9378	3.3982
17	0.9282	1.9696	6.0000	57	0.4510	3.5587	6.0000
18	0.5134	2.6089	6.0000	58	0.4193	1.8785	6.0000
19	0.9028	2.9613	6.0000	59	1.4702	0.2007	6.0000
20	0.5329	0.1778	6.0000	60	1.4604	3.3145	6.0000
21	0.5077	0.9208	6.0000	61	0.5430	1.8289	6.0000
22	4.7734	2.7847	6.0000	62	4.0275	4.0515	6.0000
23	3.9795	0.2766	6.0000	63	4.0733	2.6943	6.0000
24	3.9578	0.3927	6.0000	64	3.6828	3.9634	6.0000
25	4.8266	3.8783	6.0000	65	4.5027	4.3337	6.0000
26	4.5787	4.3019	6.0000	66	3.8452	4.7270	6.0000
27	4.6913	3.4141	6.0000	67	1.5	3.8158	4.7952
28	1.5	4.4085	5.2235	68	1.5	4.0308	4.4268
29	1.5	1.5995	5.6609	69	1.5	4.9555	4.2481
30	1.5	2.5636	4.2865	70	1.5	1.6368	5.7679
31	1.5	2.7945	5.3952	71	1.5	1.7362	5.3169
32	2.7843	2.8737	4.0	72	2.0217	1.1281	4.0
33	1.8249	0.2612	4.0	73	1.8706	2.7908	4.0
34	2.5404	2.6154	4.0	74	2.5559	0.0532	4.0
35	2.3687	4.7798	4.0	75	3.2508	3.0702	4.0
36	2.1167	0.7639	4.0	76	3.5	0.5313	5.5343
37	3.5	3.4855	5.6845	77	3.5	1.5817	4.3842
38	3.5	4.5609	4.8094	78	3.5	3.3277	5.7855
39	3.5	1.5563	5.3820	79	3.5	4.7209	4.6973
40	3.5	0.5447	5.1424	80	3.5	1.7311	4.3226

APPENDIX D. CASE 3

Dimensions of part model

- Hole: Radius = 1 inch, Depth = 2 inches
- Part model: 5 × 5 × 6 inches

Table D.1: Sample size = 10

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.6352	1.0866	6	5.0	4.5696	1.0238
2	0.0	3.9697	2.0726	7	5.0	3.1917	3.7045
3	2.5586	2.1950	6.0	8	3.458	2.214	5.858
4	2.9578	0.9062	6.0	9	3.484	2.681	5.675
5	3.038	2.239	4.000	10	2.895	2.006	4.000

Table D.2: Sample size = 20

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.6352	1.0866	11	5.0	4.5696	1.0238
2	0.0	3.9697	2.0726	12	5.0	3.1917	3.7045
3	0.0	3.9522	5.5191	13	5.0	1.1200	1.8180
4	0.0	4.3365	5.1053	14	5.0	3.7657	3.4828
5	2.5586	2.1950	6.0	15	3.458	2.214	5.858
6	2.9578	0.9062	6.0	16	3.484	2.681	5.675
7	4.3867	2.5041	6.0	17	1.500	2.500	5.866
8	3.4555	0.0060	6.0	18	3.500	2.491	4.465
9	3.038	2.239	4.000	19	3.366	2.000	4.070
10	2.895	2.006	4.000	20	2.124	2.840	4.000

Table D.3: Sample size = 40

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.6352	1.0866	21	5.0	4.5696	1.0238
2	0.0	3.9697	2.0726	22	5.0	3.1917	3.7045
3	0.0	3.9522	5.5191	23	5.0	1.1200	1.8180
4	0.0	4.3365	5.1053	24	5.0	3.7657	3.4828
5	0.0	3.2995	1.7338	25	5.0	0.9977	1.8548
6	0.0	4.3879	1.7322	26	5.0	3.7620	2.9466
7	0.0	3.4324	2.2497	27	5.0	3.5287	4.9197
8	0.0	4.8208	4.0292	28	5.0	4.2699	2.7860
9	2.5586	2.1950	6.0	29	4.1544	2.7386	6.0
10	2.9578	0.9062	6.0	30	3.9999	3.7985	6.0
11	4.3867	2.5041	6.0	31	4.8064	3.1262	6.0
12	3.4555	0.0060	6.0	32	0.6984	0.8922	6.0
13	3.458	2.214	5.858	33	2.885	1.577	5.236
14	3.484	2.681	5.675	34	2.691	3.482	5.381
15	1.500	2.500	5.866	35	3.246	1.834	4.198
16	3.500	2.491	4.465	36	1.500	2.500	5.230
17	3.366	2.000	4.070	37	1.966	3.345	5.174
18	3.038	2.239	4.000	38	2.326	2.638	4.000
19	2.895	2.006	4.000	39	2.982	2.968	4.000
20	2.124	2.840	4.000	40	2.358	3.404	4.000

Table D.4: Sample size = 80

NO	X	Y	Z	NO	X	Y	Z
1	0.0	1.6352	1.0866	41	0.0	0.6117	3.3076
2	0.0	3.9697	2.0726	42	0.0	1.8124	5.4047
3	0.0	3.9522	5.5191	43	0.0	3.4006	2.7479
4	0.0	4.3365	5.1053	44	0.0	0.8941	2.9020
5	0.0	3.2995	1.7338	45	0.0	3.4319	3.3212
6	0.0	4.3879	1.7322	46	0.0	3.9672	4.6646
7	0.0	3.4324	2.2497	47	0.0	0.9465	1.2953
8	0.0	4.8208	4.0292	48	0.0	0.7125	2.3194
9	5.0	4.5696	1.0238	49	5.0	4.3280	3.0381
10	5.0	3.1917	3.7045	50	5.0	1.5125	0.1572
11	5.0	1.1200	1.8180	51	5.0	3.4861	4.1484
12	5.0	3.7657	3.4828	52	5.0	3.8784	1.2941
13	5.0	0.9977	1.8548	53	5.0	0.9777	2.1161
14	5.0	3.7620	2.9466	54	5.0	4.6548	2.7319
15	5.0	3.5287	4.9197	55	5.0	3.8756	1.9175
16	5.0	4.2699	2.7860	56	5.0	1.0205	5.3936
17	2.5586	2.1950	6.0	57	2.3167	0.5203	6.0
18	2.9578	0.9062	6.0	58	0.0360	4.8670	6.0
19	4.3867	2.5041	6.0	59	3.6014	4.9638	6.0
20	3.4555	0.0060	6.0	60	3.8136	4.5471	6.0
21	4.1544	2.7386	6.0	61	0.8069	2.4134	6.0
22	3.9999	3.7985	6.0	62	2.1284	4.8832	6.0
23	4.8064	3.1262	6.0	63	2.1166	3.3661	6.0
24	0.6984	0.8922	6.0	64	1.7024	3.5101	6.0
25	3.458	2.214	5.858	65	3.395	2.053	4.353
26	3.484	2.681	5.675	66	1.500	2.500	4.450
27	1.500	2.500	5.866	67	3.406	2.923	5.656
28	3.500	2.491	4.465	68	1.912	3.309	5.131
29	3.366	2.000	4.070	69	3.012	1.641	5.570
30	2.885	1.577	5.236	70	3.245	3.167	4.898
31	2.691	3.482	5.381	71	1.793	1.793	4.792
32	3.246	1.834	4.198	72	3.497	2.421	4.622
33	1.500	2.500	5.230	73	3.497	2.583	4.848
34	1.966	3.345	5.174	74	1.500	2.500	5.687
35	3.038	2.239	4.000	75	2.083	2.518	4.000
36	2.895	2.006	4.000	76	2.481	2.680	4.000
37	2.124	2.840	4.000	77	2.452	2.827	4.000
38	2.326	2.638	4.000	78	2.478	2.055	4.000
39	2.982	2.968	4.000	79	2.606	2.307	4.000
40	2.358	3.404	4.000	80	2.221	2.980	4.000

APPENDIX E. CASE 4

Dimensions of part model

- Part model: $20 \times 20 \times 25$ inches
- Control points of Bézier surface (Table E.1)

Table E.1: Control points

-5.0	-15.0	5.0	0.0	-15.0	5.0	5.0	-15.0	0.0
10.0	-15.0	5.0	15.0	-15.0	5.0			
-5.0	-10.0	5.0	0.0	-10.0	6.0	5.0	-10.0	0.0
10.0	-10.0	10.0	15.0	-10.0	5.0			
-5.0	-5.0	-2.0	0.0	-5.0	-2.0	5.0	-5.0	-2.0
10.0	-5.0	7.0	15.0	-5.0	0.0			
-5.0	0.0	-5.0	0.0	0.0	8.0	5.0	0.0	-5.0
10.0	0.0	-7.0	15.0	0.0	-10.0			
-5.0	5.0	-5.0	0.0	5.0	-5.0	5.0	5.0	-5.0
10.0	5.0	-7.0	15.0	5.0	-10.0			

Table E.2: Sample size = 10

NO	X	Y	Z	NO	X	Y	Z
1	0.000	5.120	10.269	6	10.751	11.754	11.851
2	0.000	5.240	5.430	7	1.154	0.754	17.382
3	0.000	4.215	6.909	8	0.496	1.840	17.190
4	20.000	8.260	8.253	9	1.005	0.954	17.371
5	20.000	3.250	7.410	10	20.000	4.139	5.244

Table E.3: Sample size = 20

NO	X	Y	Z	NO	X	Y	Z
1	0.000	5.120	10.269	11	20.000	8.260	8.253
2	0.000	5.240	5.430	12	20.000	3.250	7.410
3	0.000	4.215	6.909	13	20.000	4.139	5.244
4	0.000	3.041	0.445	14	20.000	15.875	0.376
5	0.000	1.779	1.315	15	20.000	3.924	5.844
6	3.004	12.793	11.634	16	11.100	9.995	13.007
7	0.694	1.299	17.330	17	0.547	0.035	17.479
8	2.216	1.590	17.067	18	14.059	3.563	16.653
9	7.474	1.690	15.964	19	1.111	1.898	17.141
10	15.493	7.768	14.922	20	6.305	6.068	14.803

Table E.4: Sample size = 40

NO	X	Y	Z	NO	X	Y	Z
1	0.000	5.120	10.269	21	20.000	8.260	8.253
2	0.000	5.240	5.430	22	20.000	3.250	7.410
3	0.000	4.215	6.909	23	20.000	4.139	5.244
4	0.000	3.041	0.445	24	20.000	15.875	0.376
5	0.000	1.779	1.315	25	20.000	3.924	5.844
6	0.000	13.015	7.060	26	20.000	0.383	0.726
7	0.000	8.919	2.084	27	20.000	3.211	3.092
8	0.000	10.464	2.974	28	20.000	11.232	4.525
9	0.000	4.056	11.874	29	20.000	6.706	1.282
10	0.000	1.245	2.874	30	20.000	8.972	3.005
11	3.004	12.793	11.634	31	11.699	2.726	16.140
12	0.694	1.299	17.330	32	2.962	0.248	17.037
13	2.216	1.590	17.067	33	2.233	4.789	15.754
14	7.474	1.690	15.964	34	4.847	1.780	16.430
15	15.493	7.768	14.922	35	10.751	11.754	11.851
16	11.100	9.995	13.007	36	1.154	0.754	17.382
17	0.547	0.035	17.479	37	0.496	1.840	17.190
18	14.059	3.563	16.653	38	1.005	0.954	17.371
19	1.111	1.898	17.141	39	1.754	2.103	17.004
20	6.305	6.068	14.803	40	3.286	0.885	16.936

Table E.5: Sample size = 80

NO	X	Y	Z	NO	X	Y	Z
1	0.000	5.120	10.269	41	20.000	8.260	8.253
2	0.000	5.240	5.430	42	20.000	3.250	7.410
3	0.000	4.215	6.909	43	20.000	4.139	5.244
4	0.000	3.041	0.445	44	20.000	15.875	0.376
5	0.000	1.779	1.315	45	20.000	3.924	5.844
6	0.000	13.015	7.060	46	20.000	0.383	0.726
7	0.000	8.919	2.084	47	20.000	3.211	3.092
8	0.000	10.464	2.974	48	20.000	11.232	4.525
9	0.000	4.056	11.874	49	20.000	6.706	1.282
10	0.000	1.245	2.874	50	20.000	8.972	3.005
11	0.000	3.036	8.926	51	20.000	2.089	16.457
12	0.000	6.154	12.944	52	20.000	2.959	2.897
13	0.000	1.842	16.691	53	20.000	0.002	0.004
14	0.000	6.716	7.898	54	20.000	0.852	3.411
15	0.000	0.235	0.198	55	20.000	6.276	3.640
16	0.000	0.252	4.510	56	20.000	0.880	13.182
17	0.000	4.885	10.195	57	20.000	19.631	1.497
18	0.000	0.675	3.076	58	20.000	11.925	6.760
19	0.000	0.173	0.636	59	20.000	2.355	6.267
20	0.000	0.806	1.384	60	20.000	0.966	0.890
21	0.000	1.554	9.104	61	20.000	8.946	0.952
22	0.000	5.145	10.865	62	20.000	0.329	1.232
23	0.000	11.201	4.440	63	20.000	14.685	4.353
24	0.000	7.467	10.401	64	20.000	8.755	0.688
25	0.000	7.743	3.649	65	20.000	4.799	0.491
26	3.004	12.793	11.634	66	1.154	0.754	17.382
27	0.694	1.299	17.330	67	0.496	1.840	17.190
28	2.216	1.590	17.067	68	1.005	0.954	17.371
29	7.474	1.690	15.964	69	1.754	2.103	17.004
30	15.493	7.768	14.922	70	3.286	0.885	16.936
31	11.100	9.995	13.007	71	1.250	19.269	7.872
32	0.547	0.035	17.479	72	14.504	14.358	8.875
33	14.059	3.563	16.653	73	15.429	8.980	13.919
34	1.111	1.898	17.141	74	5.394	9.159	13.452
35	6.305	6.068	14.803	75	0.552	0.069	17.479
36	11.699	2.726	16.140	76	2.047	6.172	14.974
37	2.962	0.248	17.037	77	4.141	1.682	16.614
38	2.233	4.789	15.754	78	0.778	0.976	17.388
39	4.847	1.780	16.430	79	2.998	3.040	16.470
40	10.751	11.754	11.851	80	10.964	8.895	13.688