AUTOMATIC COMPENSATING CLEANUP OPERATION

by

Xiaohong Zhu

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

The University of Utah

December 1998

ABSTRACT

Today's part geometries are becoming ever more complex and require more accurate tool path to manufacture. Machining process efficiency is also a major consideration for designers as well as manufacturing engineers. Although the current advanced CAD/CAM systems have greatly improved the efficiency and accuracy of machining with the introduction of Numerically Controlled (NC) machining, excessive material may still be left on the finished part due to machining constraints, including the inaccessibility of the designed part geometry with respect the cutter, machine motion constraints like ramp angles, specific cutting patterns, etc. Polishing operations such as grinding and hand finishing are quite time consuming and expensive and may damage the surface of the part or introduce inaccuracies because of human errors. Although most of the existing machining approaches attempt to reduce such excessive restmaterials by modifying NC tool paths, none of them is satisfactory. They can be time consuming, error prone, computationally intensive, too complicated to implement, and limited to certain problem domains. A compensating cleanup tool path will be developed in this research to automatically remove these excessive material from the finish part. This method greatly reduces the burden of hand finishing and polishing and also reduces the error and complexities introduced in manually generating cleanup tool paths in the shop floor. More important, the tool path generated by this method will reduce the machining time and increase tool life compared with optimized tool path which left no excessive material behind.

CONTENTS

CHAPTERS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Richard Riesenfeld for introducing me to this important and exciting research area as well as for his patient guidance and continuous encouragement in both the research process and the writing of this dissertation. Thanks also go to my supervisory committee members Professor Elaine Cohen and Professor Thomas Henderson for their continuous supporting and many helpful ideas.

CHAPTER 1

INTRODUCTION

Today's part geometries are becoming ever more complex and more accurate in manufacturing industry nowadays. Efficiency of the machining process is also a major consideration for designers and manufacturing engineers. Although the current advanced Computer-Aided Design and Computer-Aided Manufacture (CAD/CAM) systems have greatly improved the efficiency and accuracy of machining with the introduction of Numerically Controlled (CNC) machining, excessive materials may still be left on the finished part due to machining constraints, including the inaccessibility of the designed part geometry against the cutter, machine motion constraints like ramp angle, specific cutting patterns, etc. Polishing operations such as grinding and hand finishing are quite time consuming, expensive and may damage the surface of the part or cause the loss of accuracy because of human errors. Although most of the existing machining approaches attempt to reduce those excessive restmaterials by modifying NC tool paths, none of them is satisfactory today. They can be time consuming, error prone, computationally intensive, too complicated to implement, and limited to certain problem domains. This research develops a new approach–"Automatic Compensating Cleanup Tool Path" to efficiently remove restmaterials by CNC machine tool and subsequently to reduce the burden for hand polishing or grinding.

In the following sections of this chapter, we first describe the problem of restmaterial removal. Other approaches for restmaterial removal will be reviewed, which will be followed by the introduction of a new approach, namely, automatic compensating cleanup tool path. Finally, an overview of the whole thesis and a summary of the contributions of this research are outlined at the end of this chapter.

1.1 The Problem—Restmaterial Removal

A tool path is a sequence of machine tool motion commands that controls the machine tool to remove a certain amount of material from the stock to produce the desired intermediate part shape. This intermediate part will be further machined into another intermediate part that is closer to the shape of designed part. A sequence of tool paths are programmed by NC programmers to progressively remove material from stock to produce the final designed part. These tool paths are usually divided into two groups: roughing tool paths and finishing tool paths:

- ***Roughing tool paths*** remove as much material from the stock as possible to produce an intermediate part. A larger tool is usually used in this step to increase the efficiency of the machining;
- ***Finishing tool paths*** remove the residual materials left on the part surfaces after completion of roughing tool paths to produce satisfactory finishing part surfaces. A smaller tool is usually used in this step to achieve a higher accuracy finish.

To machine a part from a given stock, roughing tool paths are usually executed first that remove most of material from the stock to produce an intermediate part which is roughly close to the final designed part. Efficiency of material removal is usually the main concern for the roughing phases; therefore larger tool is usually used in these roughing operations. After finished roughing, finishing tool paths contour the part surface to produce a finish part which is within the machining tolerance specified; therefore, smaller tool is usually used in finishing operations to produce an accurate part. The following is an example of a part model machined from a given block.

In Figure 1, two layers of material are removed from a block stock by roughing tool paths in the first phase of producing a part. A large tool with large steps is preferred in the execution of roughing tool paths in order to remove material more efficiently. Thereafter, a thin layer of material is left in the bottom and large scallops are left on the side walls. These materials are usually removed by a finishing too. The finishing tool path contours the part surface with a smaller tool and smaller stepovers (a stepover is the distance between two adjacent tool passes as shown in Figure 2), which removes all scallops (scallops are the excessive material left between two adjacent tool passes, as shown in Figure 1) from the side walls and the thin layer material from the bottom as shown in Figure 2.

Although scallops may still be left as shown in Figure 3 due to the cutter shape, the height of these scallops should be maintained within the machining tolerance. These scallops are beyond the consideration of this thesis research. Although, by reducing the stepovers, we can reduce the scallop height to improve the machine accuracy, limit may be reached when there are area on the surface where the tool cannot reach such as the areas around concave edges on the part surface. For instance, if a ball-end tool is used in the finishing tool path as shown in Figure 4, the scallop height left in the corner area is always bigger than $r/\sin(a/2) - r$, whereas the scallop height in the flat area as shown in Figure 5. is $r - \sqrt{r^2 - \delta^2/4}$. When $\delta \rightarrow 0 \Rightarrow r - \sqrt{r^2 - \delta^2/4} \rightarrow 0$, therefore scallop height in flat area can be close to zero.

It is assumed in this thesis, that the process will always generate tool paths that produce parts within specified machine tolerance under normal conditions. Exceptions only occur when restmaterial are left due certain cutting limitations such as tool shape, tool motion limitation etc. Excessive material left under these exceptional conditions is the problem that this research work tries to solve.

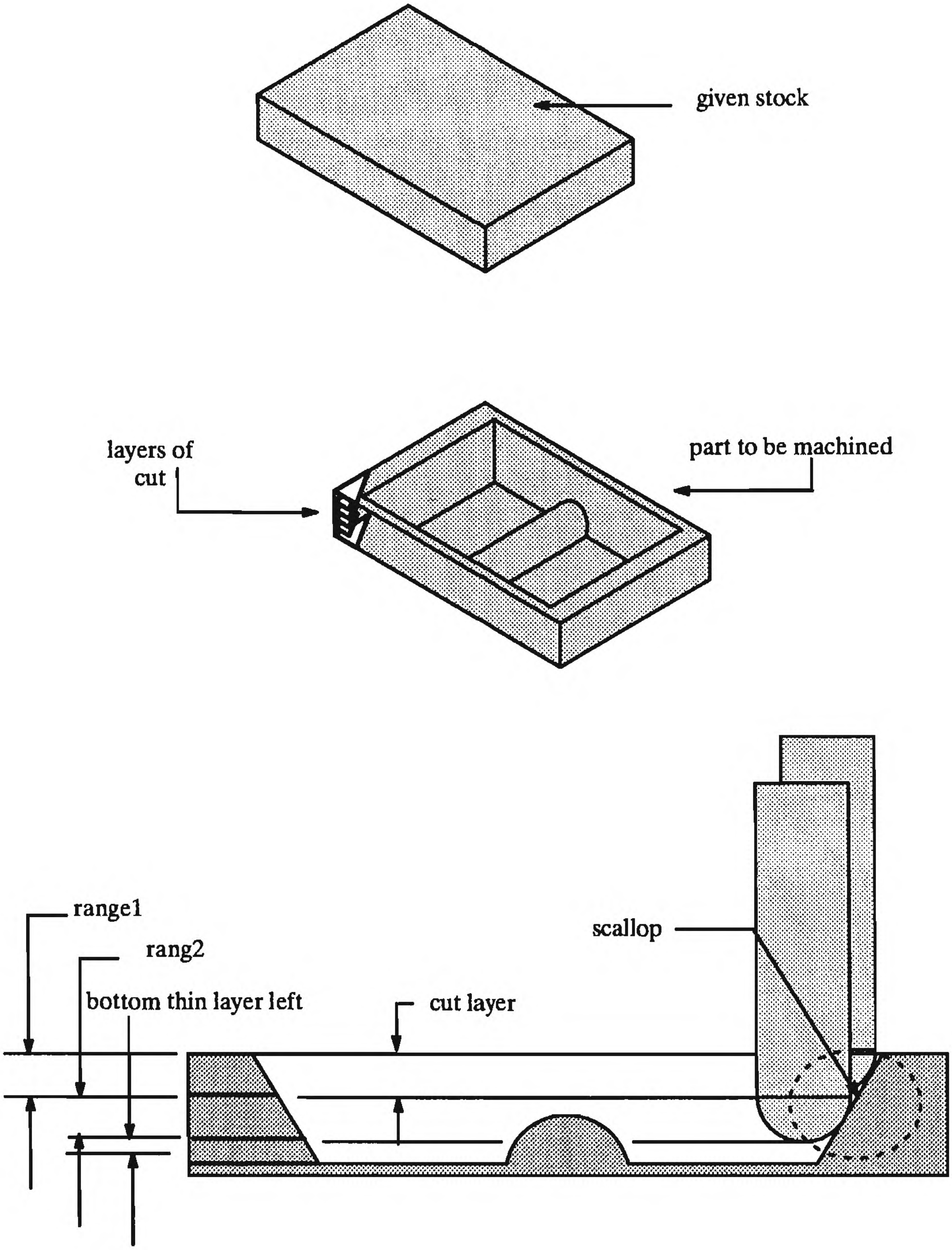

Figure 1 Remove materials with roughing tool path

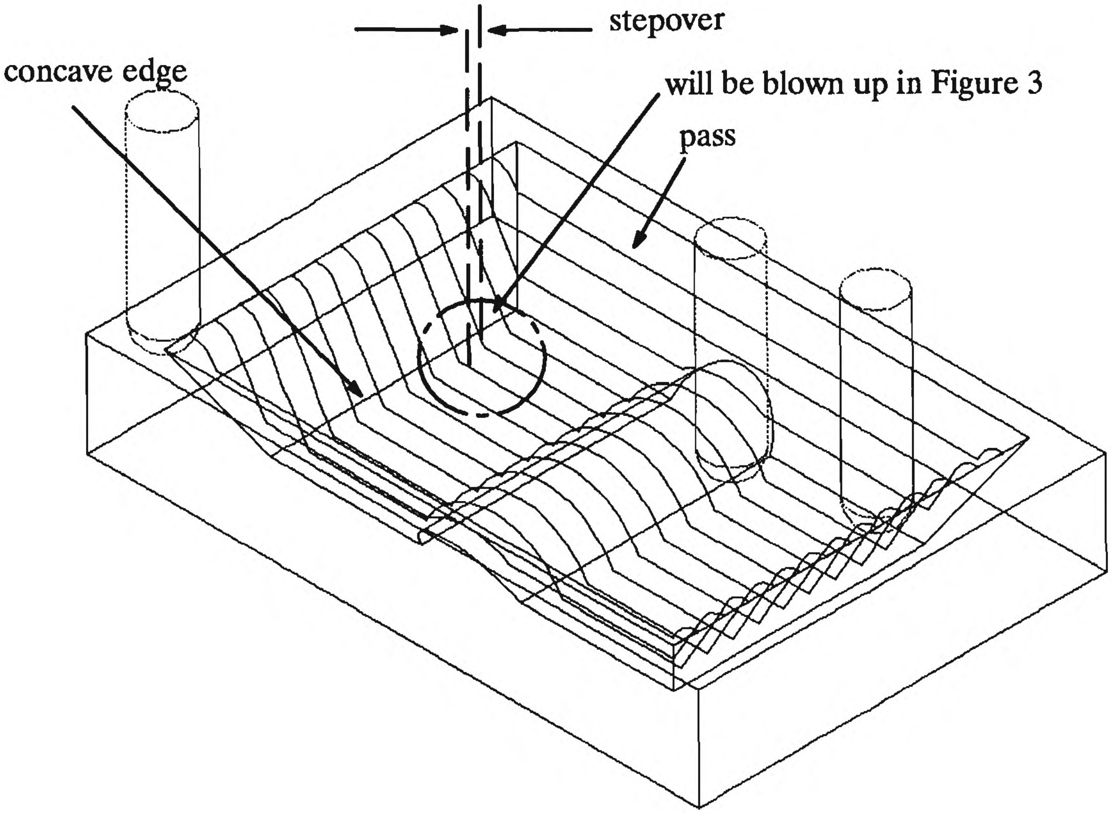

Figure 2 Contour surfaces with finish tool path

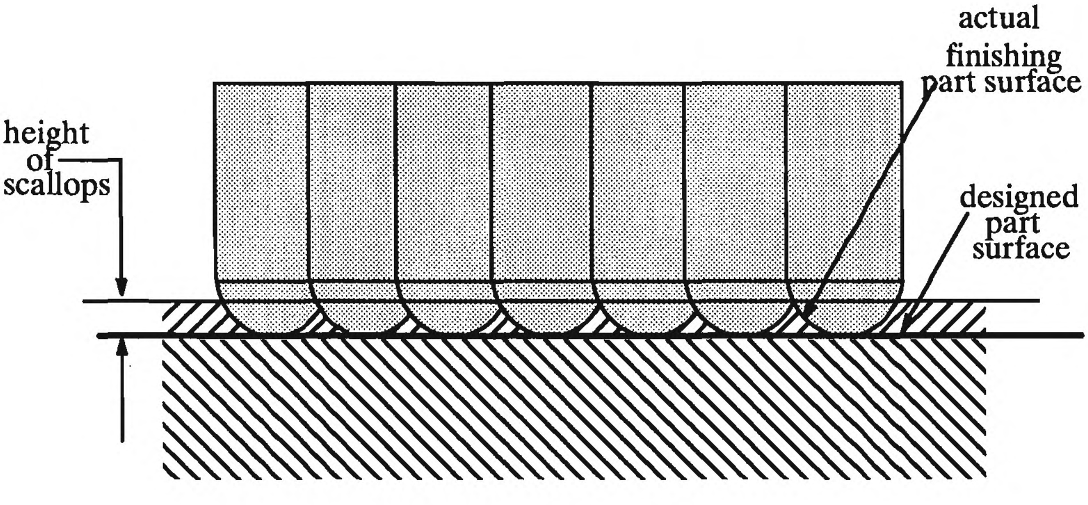

Figure 3 Actual part surface after finishing operation

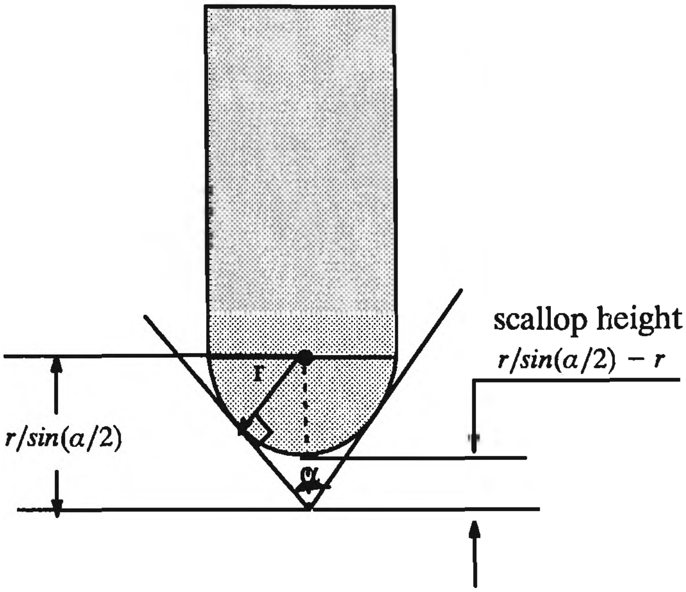

Figure 4 Scallop height at concave corner

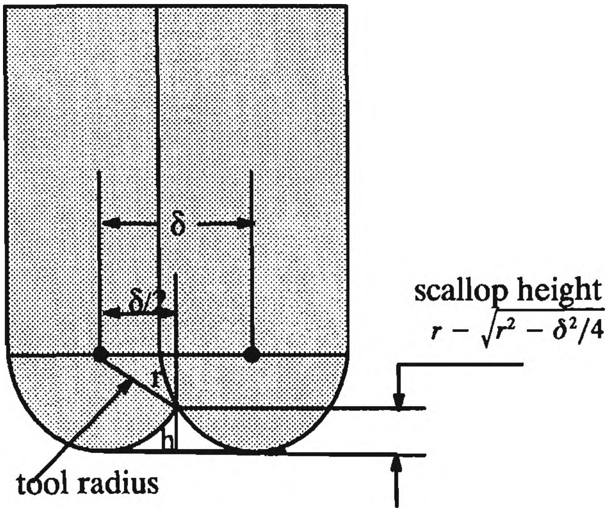

Figure 5 Scallop height at the flat area

1.1.1 Restmaterials

Restmaterials are defined as those excessive materials exceeding machining tolerance, which is intended to be removed by the tool path but left unmachined due to exceptional geometric constrains after the execution of a finishing tool path. Areas on part surfaces that contain restmaterials are called ***uncut areas*** in this thesis.

Theoretically, tool path generation is always gouge free and is accomplished with tolerance control in current CAM systems. After execution of these tool paths, a part within tolerance should be produced. Unfortunately, excessive materials may still be left on the finished part because of the inaccessibility of a tool, Ramp Up and Ramp Down angle constraints (Ramp Up Angle and Ramp Down Angle allow one to specify the up and down angular motion limits of the tool. The angles are measured from the plane perpendicular to the Tool Axis), or specifically designed cut patterns (certain cut patterns may intentionally skip some area of the part surface to improve cutting speed). With the following example, the concept of restmaterial can be more easily understood.

1.1.2 Restmaterial Removal

Zero finishing surface is a term used in manufacturing practice, which means that the surface of a finished part is smooth, the scallop height is evenly distributed all over the part surfaces and there is no restmaterial left on it after the finishing operation. As we have mentioned above, restmaterial is a major problem that considerably affects the accuracy in machining. Zero finishing a surface is a goal that cannot be achieved in true manufacturing. Therefore, to approach an efficient solution for restmaterial removal to increase the accuracy of part machining is desired.

1.2 Approaches for Restmaterial Removal

Restmaterials have been traditionally removed by polishing and hand finishing, which is time consuming, expensive, and inefficiency. NC machine tools can be precisely controlled and have given a more accurate solution for restmaterial removal than polishing and hand finishing. The introduction of high speed machines has also improved the efficiency of surface finishing. However, the present NC tool machining still involves a lot of manual procedures which inevitably introduce human errors. Different approaches that have already been proposed in the literatures for restmaterial removal will be reviewed in this section.

1.2.1 Manufacturing Shop Floor Practice

Using a machine tool to remove restmaterials was first introduced and is now widely used in the manufacturing shop floor. After loading and executing the preprogrammed NC tool path, experienced machine operators visually inspect the finished part and find problematic areas where restmaterials may be left. Then, they may either manually reconfigure the machine tool (such as change the tool or tool axis, etc.) or use simple CAD/CAM software to draw some geometry to generate refining NC tool paths to remove the restmaterials. This process relies on human eyes and experience that may introduce errors. Some areas with restmaterials may be missed, and some other areas without restmaterials may be remachined. In some extreme cases, the whole part may be remachined in order to avoid missing any restmaterial. This is very time consuming. Therefore, manually controlling the machine tool to remove restmaterials, as used in manufacturing shop floor practice, is not only unreliable but also inefficient.

1.2.2 Tool Path Refining

In order to generate a tool path to remove all of the restmaterials and produce a zero finishing surface, small tools are used to reduce the possibility of inaccessibilities of part geometry. This idea has been discussed in [5, 6, 7, 13, 49, 51, 56]. In these papers, the inaccessibility of the cutter against part geometry was globally analyzed to find out all of the potentially unreachable areas on the part, and then, an optimal tool was (or could be) chosen to generate the NC tool path. [8, 21, 34] studied the local interference between cutter and part geometry, offered some methods to avoid the interference and also revealed the restmaterial problem caused by this interference. Research on scallop height analysis has attempted to generate a smooth surface finishing tool path to reduce restmaterials[38, 39, 43].

Although that research mainly concentrated on analyzing the restmaterial problem rather than developing restmaterial free tool paths, some optimization of finish tool paths has been achieved to avoid restmaterials. However, the definition of an optimal tool path has not been generally accepted because of the complexity of manufacturing practice. Moreover, the criteria for optimization are quite different in different manufacturing shops and also depend on the machines being used. Therefore, it is difficult to define an optimal tool path for a part manufacturing process. The solutions provided by these methods are not practical. For example, for those small but delicate and important areas in a big part, these methods will suggest using a small tool to cut all through the whole part to generate a tool path that produces part surfaces close to zero finish. This will greatly increase the cost since it takes a long time to machine the whole part and the wear-off of small tools is also faster than that of larger tools.

1.2.3 NC Tool Path Verification

These approaches were developed from shop floor practices. NC tool path verifications simulate the cutting motion of the machining operations performed on a part and calculate those factors of concern such as gouging, feedrate, material removal rate, or whether a part is actually produced as designed and the like [19, 24]. With the visual display of the area where excessive material is left, users can easily identify them and draw some simple geometry to generate auxiliary tool paths to remove the restmaterial. The human error in shop floor practice can be greatly reduced by this, but it is still not an automatic method. Different approaches have been used in NC tool path verification methods to calculate the difference between the machined part and its designed geometry.

These approaches can be found in the section of literature review for NC simulation and verifications.

1.2.4 Graphical NC Tool Path Editing

Graphical NC tool path editing is an extension of NC tool path simulation and verification methods. Although NC tool path simulation can replace the manufacturing processes of prototype parts, the NC tool path has to be reprogrammed if any error is found during the simulation. This is time consuming. [52, 53, 54, 55] presented a method for editing the NC programming code in the verification stage without going back to the part programming stage. A data structure and an algorithm to interactively verify and edit the NC programming code with the aid of graphics are introduced in these papers. It can also be found in the Unigraphics CAM system. This method still relies on human interactions, and is a manual approach. With the introduction of high speed machining and empowering computing power to NC machine tools, tool path sometimes is considerably huge. Therefore, manual editing is tedious and due to human interaction, error prone. Because of these factors it can become prohibitive.

1.2.5 Comparisons

In the previous sections, we discussed those existing methods that have been used in restmaterial removal. All of those approaches attempted to use machine tools to remove restmaterials in different ways. They can be divided into two different categories:

- refining a ***finishing tool path*** to ensure the removal of restmaterials (this includes the approaches discussed in section 1.2.2),
- designing a ***compensating tool path*** to remove restmaterials left by finishing tool path (this includes approaches discussed in section 1.2.1, 1.2.3, and 1.2.4).

Because of the complexity of part geometry and constraints of machine tools, there may not exist such a finishing tool path that can ensure the removal of all the restmaterials. It should also be noticed that trying to develop a zero finish tool path will introduce significant computational complexity because of the complexity of part geometry and machine tool constraints. The algorithms associated with these approaches has typically exhibit the following characteristics:

- complicated,
- restricted to certain problem domain, and
- computationally intensive.

On the other hand, using a compensating tool path to remove restmaterials is more flexible and feasible. Intrinsically, it follows the general strategy of divide and conquer. Instead of trying to remove all the material in one step, it removes most of the excessive material in the finishing tool path and uses a separate compensating tool path to remove restmaterials. This method is simple and straightforward. The problem is the lack of a computer tool to automate the compensating tool path generation.

The manual human interactions are tedious and error prone. The shop floor practice method may cause over cutting of nonrestmaterial areas and missing restmaterials in other locations. NC verification methods can reduce the error in inspecting restmaterial areas as occur in the shop floor practice method, but manually generating a compensating tool path may still cause variational errors within restmaterials; the graphical tool path editing approach is very time consuming and cannot be so accurate as automatic tool path generation. The disadvantages of this group of methods include the following characteristics:

- Tedious
- Time consuming
- Error prone
- Only suitable for small tool path with small restmaterials

1.3 Automatic Compensating Cleanup Tool Path

As discussed above, in order to improve efficiency and accuracy, NC machining should be used in restmaterial removal. The automatic compensating tool path approach we established here belongs to the compensating tool path category, but it develops algorithms to automatically generate cleanup tool paths. This automation is achieved by two steps: automation of restmaterial detection and automation of NC tool path generation for restmaterial removal. This method explores the NC verification technology, automatically displays and computes part geometry to compare the CAD data with CAM data, and hence determines restmaterials. The cleanup tool path discussed here is just a compensating tool path of the finishing tool path; therefore, it shares the same fixture of the finishing tool path. Because of this, fixture will not be discussed in this research. A relationship between the cleanup tool path and finishing tool path will introduce the automatic setup for cleanup tool path generation. Therefore, a compensating cleanup tool path can be generated automatically based on the computed CAD geometry data along with the preset CAM data.

This approach will greatly reduce the errors introduced by intensive human interactions required by the manual generation of compensating cleanup tool paths; it also overcomes the limitation that refining tool path approaches have. Therefore, this cleanup tool path is easy to use and much more robust and efficient.

1.4 Overview

Three major aspects are involved in this study: problem identification, establishment of compensating cleanup tool path methodology, and development of algorithms and data structures. The organization of the whole dissertation can be described as following:

In Chapter 2, a survey of recent NC verification research will first be given, because the approach proposed here is based on some theories of NC verification. An analysis of the pros and cons of existing methods will also be discussed. A comparison of those existing approaches versus the automatic compensating tool path approach we are going to establish will be carried out. Finally, current developments in CAM commercial software industry will be briefly introduced.

Chapter 3 subdivides the problem domains into smaller problems. This division is based on the geometric properties of the excessive materials and the desirable cleanup tool path cutting patterns. This division is preferable to the professional NC programmers, because it simplifies the solutions and, therefore, improves the computation efficiency.

Chapter 4 introduces a new approach based on the IOPM model. The IOPM model provides the mathematical foundation for tracking the part models in manufacturing process. With the IOPM model, the course that a part is manufactured from a given stock is formally defined, and the dependencies between two consecutive tool paths can be easily represented and studied. These provide a basis for discussing the automation of feature machining and process planning in general.

Chapter 5 discusses the solutions for valley area detection and cleanup. In this chapter, an annealing algorithm along with the spatial directory algorithm is introduced as the fundamental approach for computation.

Chapter 6 discusses the solutions for steep area detection and cleanup tool path generation. An adaptive multidimensional G-buffer algorithm will be used to compute steep areas. Cleanup tool path generation is also discussed.

Finally, we conclude the thesis with a discussion of the contributions of this research and future desired work.

CHAPTER 2

BACKGROUND STUDY

In this chapter, an overview of CAD/CAM/NC simulation and verification is first given to provide a background for this study. As discussed in Chapter 1, automation of cleanup tool path generation includes automation of restmaterial detection and cleanup tool path generation. The computation of restmaterial geometry is closely related to NC verification; therefore, a review of NC verification is conducted. This is followed by a brief introduction to NC milling tool path generation. Recent developments that have been done in CAD/CAM commercial software industrials are reviewed. As this research is implemented in the UNIGRAPHICS CAM system, UNIGRAPHICS is also briefly introduced.

2.1 Overview of CAD/CAM/NC Simulation and Verification

A quarter of a century ago, the desire for manufacturing complex parts with accuracy, efficiency, reproducibility, and minimization of errors brought Computer-Aided Design (CAD) into life. CAD techniques have been widely used in manufacturing processes to produce different part geometries. Since the advent of Numerically-Controlled machines, many algorithms for tool path generation have been developed. Due to the simplicity of part models and machine control features dealt with before, the tool path generation has not attracted adequate attention of researchers. Currently, with the improvement of CAD technologies, parts designed by advanced CAD technologies are becoming more complex in order to achieve the varieties of functionalities from customers. Therefore, the demands for manufacturing such complex part models are becoming an important stage of putting the design into reality. To ensure the geometric validity and accuracy of NC tool paths, the test cutting is usually performed before the final machining. However, the test cutting results in a huge waste of manpower and reduction of machine utilization, computer-aided NC tool path simulations and verifications have been developed to perform a visual simulation and analysis to replace the test cutting process.

In the following sections of this chapter, we analyze the process of how a part is traditionally produced and explain the roles and functionalities of CAD, CAM, and NC verification in modern manufacturing practices, followed by the introduction of automatic cleanup tool path into the system.

2.1.1 Process of Producing a Part

The process of producing a part can be simplified, as shown in Figure 6. A designer first lays out the dimensions of the part geometry based on requirements and sends these messages to machinists. The machinist takes the stock, and the design layout produces a final part.

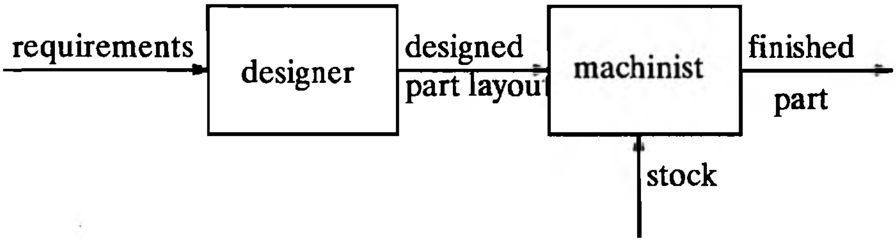

Figure 6 Produce a part without CAD/CAM

Let us look more closely into the Machinist box. As shown in Figure 7, a machinist makes the plan of machining operations based on experience and the design layout and then performs these operations in sequence on the stock to produce a final part.

Before the actually machining, a soft stock is usually chosen to produce a **prototype part** to verify that the consequence of the whole manufacturing process controlled by the designed part layout and tool paths is exactly the part wanted by users. The ***prototype part*** is a test part made of material that is cheaper and easier to cut for validation of machining processes. If any error occurs, the whole process will be modified, and another prototype part will be reproduced. This loop will be repeated until an acceptable part is produced. This process is illustrated in Figure 8.

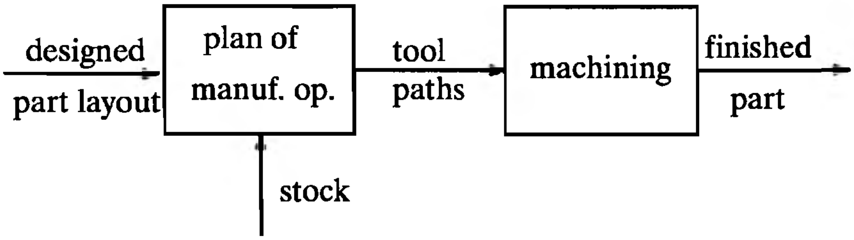

Figure 7 Detail in the “machinist” box in Figure 3

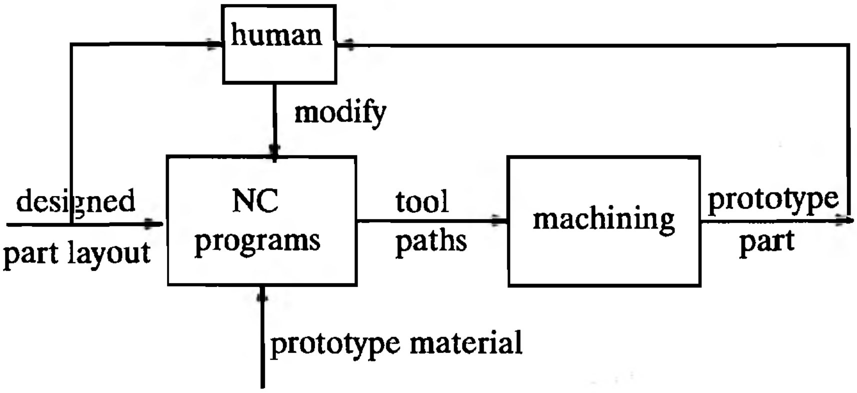

Figure 8 Prototyping loop without effecting design

Furthermore, the prototype part can also be used as a feedback to correct the design. The prototype will be inspected to introduce design changes according to the design requirements. Therefore, the prototype loop can be extended into Figure 9. This loop will be ended when an acceptable prototype part is produced.

Instead of modifying the NC programs after the prototype part is finished, changes in NC programs can be also made earlier based on the intermediate part in machining processes as shown in Figure 10.

2.1.2 CAD, CAM and NC Verification

With the fast growing and wide application of CAD/CAM systems in designing and manufacturing, it is desired that computer graphics can be used to perform NC verification in an extent to replace the physical prototyping process. This could greatly reduce the manufacturing cost, and also speed up the production. Figure 11 illustrates the roles of CAD, CAM, and computer-aided NC verification in the part production processes. CAD systems assist in the design stage, CAM systems performs NC program functionalities, and NC verification replaces the process of producing the prototyped part.

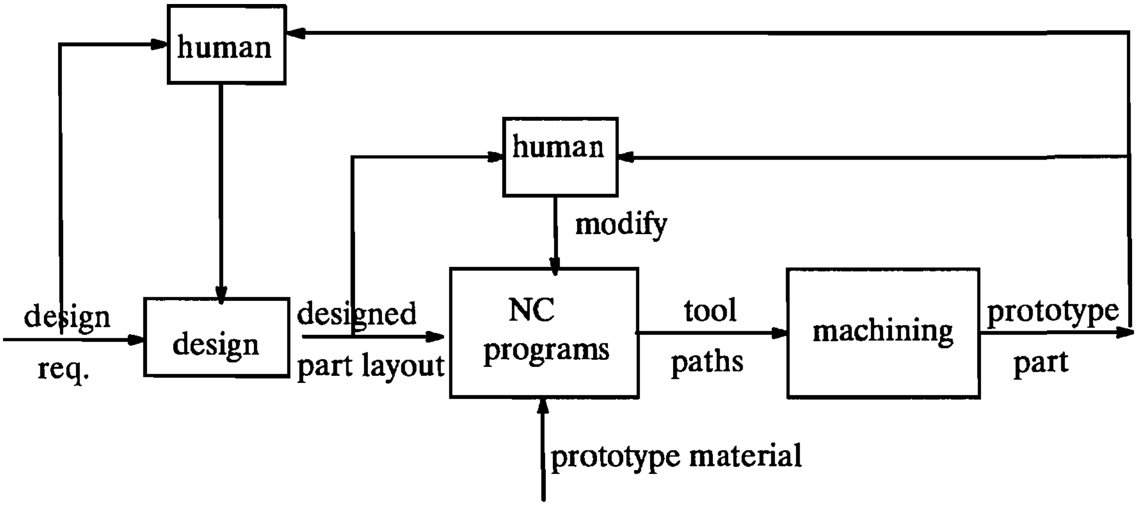

Figure 9 Prototyping loop with feedback to CAD design

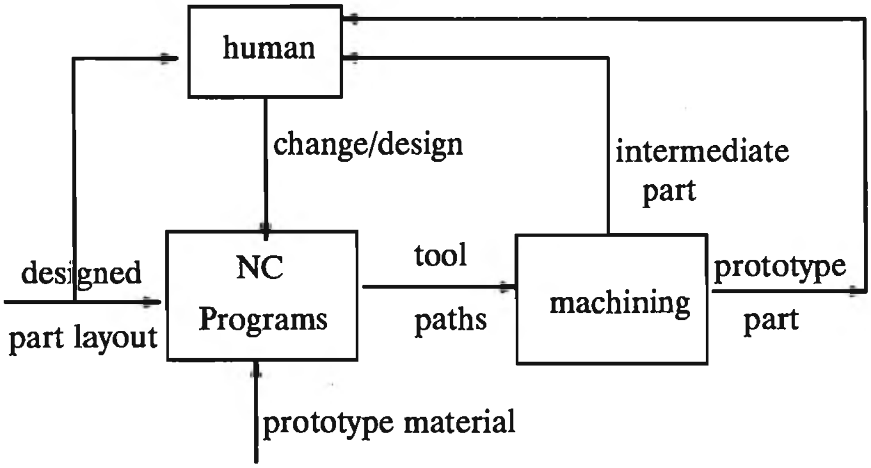

Figure 10 Incremental NC program changes

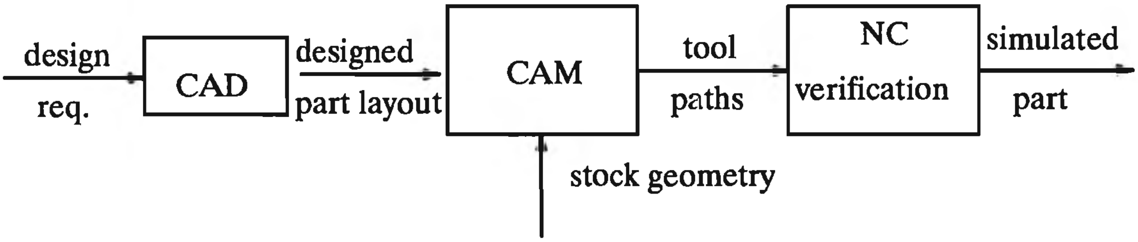

Figure 11 Current roles of CAD, CAM and NC verification

As we have analyzed in Figure 4 in the previous section, CAD, CAM, and NC verifications should be closely related to each other in analogy to Figure 9, as shown in Figure 12.

Traditionally, design and manufacturing are considered as autonomous stages as in [14]. The automatic generation of cutter paths is regarded as one of the key processes to the integration of CAD and CAM, and it is vital for the survival of the manufacturing industry [1, 14]. The integration of NC verification into the use of automation of tool path generation in this thesis will further integrate the NC verification into the CAD/CAM systems. The use of NC verification here in this dissertation will explore its greater application and a new direction for NC verifications.

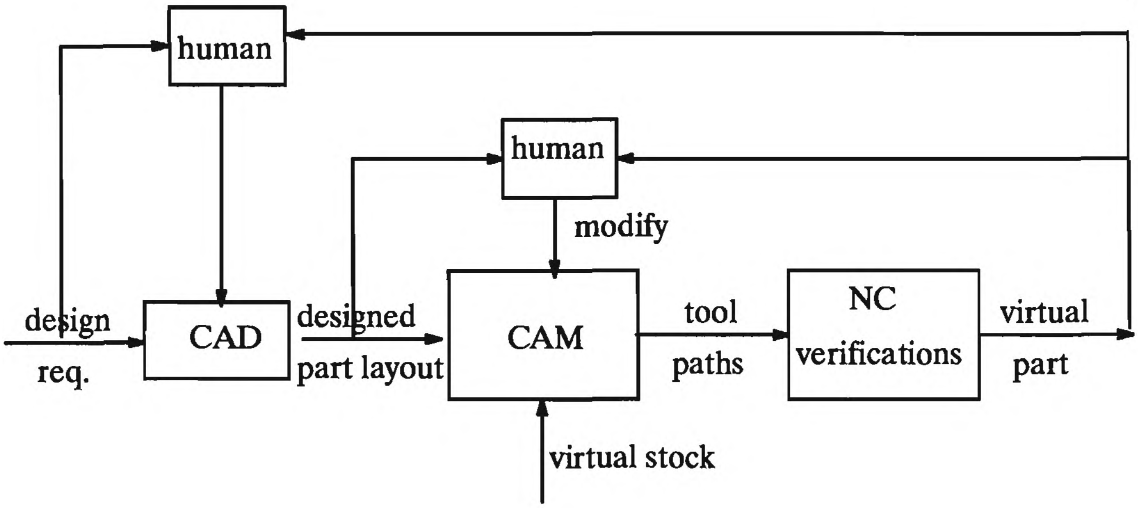

Figure 12 CAD, CAM and NC verification relationships

2.1.3 Introduction of Automatic Cleanup Tool Path

The automatic cleanup tool path is a tool path for cleanup which is generated automatically, without any human interaction. This means that a functionality that provides the automatic generation of the cleanup tool path based on the NC verification information should be implemented in the CAM module, which is the main concern of this dissertation. This is illustrated in Figure 13.

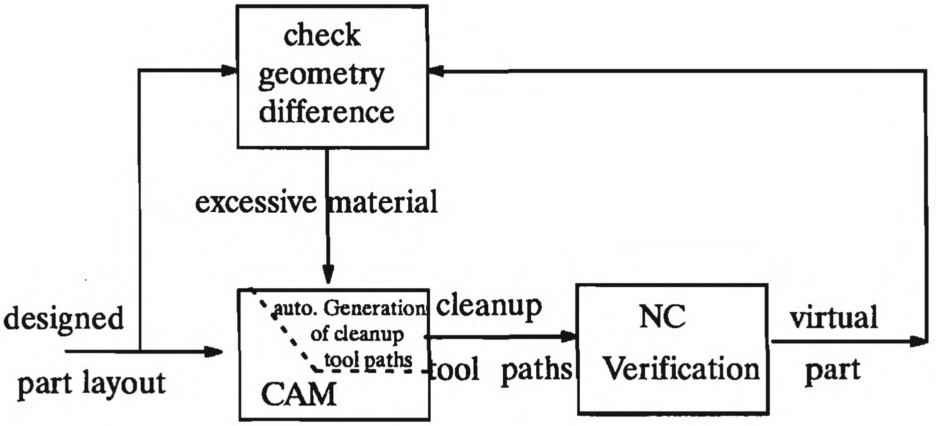

Figure 13 Automatic cleanup tool paths

As shown in Figure 13, this research provides the tool to compute the geometric difference between the virtual final part and the designed part and then to automatically generate cleanup tool paths which refine the virtual part into an acceptable part compared to the designed part.

2.2 Literature Review of NC Verification and Simulation

Simulation systems for NC machining which also provide an analysis and verification component are a powerful tool to detect errors in NC codes off-line. Although there are many CAD/CAM systems commercially available today, none of them can produce a perfect tool path. Even if the part program is carefully coded, some errors may still exist in NC machining. The NC machining errors may come from the mistakes of programmers or machine operators. To ensure the geometric validity and accuracy of NC programs, a prototype part is cut before final machining. However, the prototype part result in a waste of manpower and machining utilization. More important is that it is difficult to detect the machining errors and to feed them back to CAD data or NC programming. For this reason, the prototyping process may run several times until it is acceptable.

NC simulations were originally developed to replace the prototyping processes (as we have discussed in Figure 9 in Chapter 1). Previous studies were mainly focused on only the simulation. One of the most popular approaches for NC simulation is to employ the swept-volume of tool motion, display and analyze the result by Boolean operations. These approaches can be divided into different categories as followings based on mathematical models that are used. These categories are direct solid modeling, discrete vectors, dexine representation, quadtree representation and voxel representation.

2.2.1 Direct Solid Modeling

Direct solid modeling uses regularized Boolean operations to substract the successive tool sweep volumes from the workpiece to get an explicit workpiece solid model. This method can be traced back to the late 70s.

The use of solid modeling was first introduced by Volcker and Hunt who did an exploratory study on the feasibility of using PADL constructive Solid Geometry (CSG) modeling system for the simulation of NC programs [20]. As in Figure 14, the tool moves from position 1 to position 2 and then to position 3, the shaded area is the sweep volume of the tool motion. Subtracting the shaded tool sweep volume from the block is the direct solid modeling of the operation, which produces a direct solid model of the in-process part.

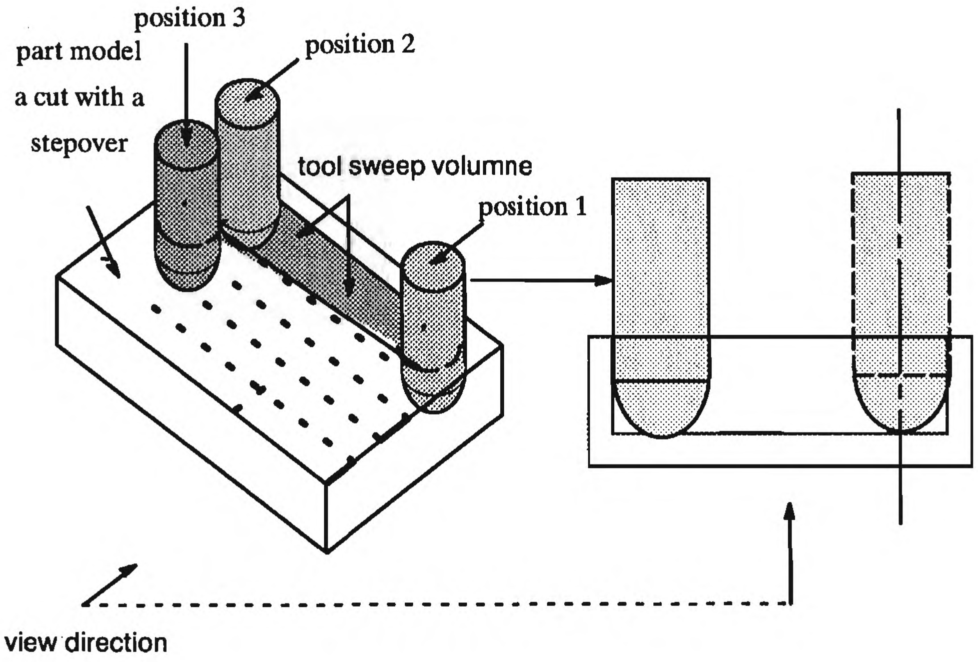

Figure 14 Direct solid modeling

This method is straightforward, it computes an exact solid model, but the problem is that regularized Boolean operations are computationally expensive. It can be used to perform off-line NC tool path analysis, such as tolerance analysis, etc., but is not applicable for the real time NC tool path simulation. For example, the cost of the simulation using the CSG approach in [20] is $O(N^4)$, where N is the number of tool movements. A complex NC program may contain ten thousands (or billion for high speed machining) movements, thus making the computation intractable. Therefore, the numerical approximation methods have been developed and widely used in real time simulation. In order to increase efficiency, a number of approximation simulation methods have also been developed. These approximation methods are O(N); i.e., the simulation time grows only linearly with the number of tool movements. These approximation methods are described in the following sections. A SEDE approximation method has been developed in [52], to improve the speed, while accuracy is sacrificed. This method is good for simulation and visualization of NC tool path, but is not accurate enough for NC verifications.

2.2.2 Discrete Vectors

Initially proposed as a display technique for constructive solid geometry (CSG) models, this type of approach involves intersections of sightlines with surfaces of an image space representation of a solid. Boolean operations are then resolved in one dimension along each sightline. These so-called visual solid modelers are very efficient computationally since the solid represented by constructive Boolean operations is not explicitly evaluated. Similar techniques have been applied to simulate material removal via NC milling in [18, 55] with an impressive computational speed. However, these fast simulation techniques are not well suited for NC verification because the representations of tolerances and general offset surfaces have not been adequately addressed by current solid modelling technology [44, 45]. In addition, discrepancies found via image space Boolean operations are not accurate since they are measured relative to an arbitrary selected sightline instead of a designed part model datum.

An improved method—surface-based technique for three–axis NC verification can be found in [11, 12, 22, 23]. With this approach, a first order curvature estimation is used to evaluate a grid of points on the surface that is dense enough to account for surface irregularities relative to the size of the milling tool.

A slightly different approach can be found in [4, 39, 40]. This technique is based on the intersection of part surface normal vectors with cylindrical models that represent the machine tool at discrete sequential positions. This idea is more efficient compared to direct solid modeling. Meanwhile, it allows straightforward incorporation of toleranced dimensional verification as milling discrepancies can be measured relative to surface normal vectors of a designed part model, but it is computationally more expensive than the approach found in [11, 12, 22, 23]. The following dexine representation is a variation of this discrete vector method.

2.2.3 Dexine Representation

Dexine representation has been given several different names, such as Z-mapping[25, 26], dexel model[50], and G-buffer[47,48]. It is similar to the method of discrete vectors, the only difference is that part models are approximated by dexel columns (three-dimentional (3D) volume) instead of vectors. Therefore, it allows efficient dynamic update of the workpiece and provides many facilities for analysis operations. For example, [47,48] used this model to integrate the generation and verification of tool path. [25, 26] analyzed the error, and allows an immediate edition of tool path. The disadvantage of this approach is the sampling error. Because the height is calculated only at the center of each dexel, sampling errors at less than the dexel interval in horizontal directions are not predictable.

2.2.4 Quadtree Representation

[2, 35] used quadtrees to represent part surface at a fixed layer and then applied two-dimentional (2D) Booleans against tool cross sections to compute the image. They succeeded in reducing memory capacity by employing the quad-tree data structure for path generation. However, the quad-tree data structure makes the image processing operations for the NC function much more complicated than that in those two approximation methods mentioned above. Considering the persistent trend of rapid improvement in computer hardware, with the future trend in mind, the reduction of memory size here increasingly less important.

2.2.5 Voxel Representation

This is a widely used approximation method in volume-rendering. It is done by partitioning the workpiece into cubes that are small enough to satisfy the tolerance restriction of the computation. It is slower than the dexel method and has the same disadvantages.

2.3 NC Tool Path Generation

In the past few decades, computers have been used to aid an ever increasing number of manufacturing-related tasks. CAD/CAM programs are used at all stages of the manufacturing process. One of the most complex tasks (mathematically and otherwise) attempted by CAD/CAM has been the generation of NC tool path.

Milling machines are widely used types of manufacturing equipment, called machining center. Compared with 5-axis, 3-axis milling is more popular, less expensive, and much more efficient in terms of machining and programming time. It has been widely used in the auto and aerospace industries. The new developed "high speed machining" technology makes 3-axis machining more valuable. This research focuses on 3-axis machining and also considers the requirements from "high speed machining." Most of the research done in this dissertation can be applied to 5-axis machining. Due to its flexibility in orientation, 5-axis machining has less restmaterial problem to deal with than 3-axis machining. There are two types of milling operations: roughing operation and finishing operation. An operation in this dissertation is a set that contains all the information needed by the CAM system to generate a corresponding NC tool path. Roughing and finishing operations can be defined as follow:

- ***Roughing operation:*** A set that contains all information needed for a CAM system to generate NC ***roughing tool path*** (Definition of roughing tool path can be found in section 1.1).
- ***Finishing operation:*** A set that contains all information needed for a CAM system to generate NC ***finishing tool path*** (Definition of finishing tool path can be found in section 1.1).

Semifinishing operations may be introduced to remove some material left after execution of roughing tool path. It can be considered as a subtype of finishing operations. In the following, we will discuss NC tool path generation algorithms for roughing and finishing operations, respectively.

2.3.1 Roughing Tool Path Generation

Roughing is usually addressed in the literature as pocketing or cavity milling. Roughing is used when a large portion of raw material needs to be removed. The earliest research can be found in 1978 when Persson [41] developed algorithms for planar pocketing.

The most common technique used in generating roughing tool path is to intersect a part with a sequence of horizontal planes and then use the curves resulting from the intersection of pocket boundaries with these planes as input to a planar pocketing algorithm. For a ball-end tool, a better cut may be obtained by offsetting the surfaces before performing the horizontal slicing [9]. These algorithms are suitable for ball-end tools but are difficult to implement for parts bounded by multiple surfaces. Simpler algorithms for general tool shapes can be found in [15, 17, 33]. Similar algorithms may also be found in [32].

Recently, advanced algorithms for optimizing NC tool path generation for cavity milling/pocketing can be found in the literature. [31] used Octrees to optimize cutter tool selections. A discretize grid schema similar to the dexine schema discussed in the previous section has been used in [57] to generate roughing cuts.

2.3.2 Finishing Tool Path Generation

Different from the roughing tool path, finishing tool path contours part surfaces to remove the extra material left after the execution of a roughing tool path to produce an exact part. A variety of algorithms has been developed for this tool path generation. They can be summarized into three categories: parametric, intersection, and projection methods.

The parametric method exploits the parametric representation and generates isocurves that are uniformly distributed across the parametric domain [16, 36]. This method is not optimal if the surface mapping into Euclidean space is not isometric. Another disadvantage of this method is the lack of interference checking. Positioning a tool perfectly tangent to one part of a surface may cause it to gouge either some other part of the same, or some other surface. Several algorithms have been developed to rectify this situation. Parametric approaches can also be found in [29, 30], and [9].

The intersection method offsets part surfaces with tool radius and then intersects the offset surfaces with planes equally spaced in Euclidean space, resulting in a piecewise linear tool path approximation. The advantage of this method is that it is gouge free, but this method is suitable for 3-axis not for 5-axis. Similar to this category, a G-buffer was used in [48]. They used a dexel model to approximate the part, and then computed offset surfaces, sliced the offset surfaces and got a final NC tool path.

Another major tool path generation method–the projection method is discussed in [15]. With this method, a sequence of drive segments is first generated above the surface, and then the system positions the tool on points along each drive segment and projects it onto the part surface along a given projection vector. The result is a tool path by which the tool moves in contact with the part surface, and the entire path is constrained to the surface defined by the drive segments and the projection vector.

Other totally different approaches can also be found, such as in [46], copy milling techniques were used for 3-axis copy milling machines. This can be called rapid prototyping or reverse engineering. It is also widely used in industry to produce parts when their engineering drawing are not available.

Recently, two survey papers are published in the area of NC tool path generation. One is [37] and the other is [10]

2.4 Recent Development in CAD/CAM System

The concept of a "Cleanup Operation" was first introduced by CAM users (manufacturing engineers). Cleanup operations greatly reduce the burden of final grinding or hand finishing. Currently, in major CAM software such as Unigraphics, GM cut, CAMAX, WORKNC, TEBIS, etc., different modules have been developed to generate cleanup operations. Some simulation packages like dCADE also developed tools to detect some kinds of restmaterials.

Most of these newly developed modules deal with limited cleanup problems. For example, GM cut provides flow cut operation–which generates tool path along concave edges and then changes to a smaller tool to cleanup materials left in these concave edges. Besides the incompleteness of the problem domain it addressed, manually specifying the concave edge is also tedious, hard, and not robust. dCADE, a simulation package, also detects the uncut areas where surface areas are too steep to be machined accurately.

Unigraphics, WORKNC, and CAMAX developed more general approaches to target the uncut area caused either by inaccessibility of machine tools or by particular cut patterns, but they all have limitations, and are also too complex to use. For example, in UNIGRAPHICS, the uncut areas are first displayed while the previous finished tool path is generated, and then different mechanisms are provided to let the user select uncut boundaries to prepare the geometry configuration for the cleanup operation. Finally, other parameters, such as cleanup tool diameter, cut pattern, and the like, are specified to generate cleanup operations.

To remove restmaterials left on finished parts, we need first to identify these uncut areas, and then to create efficient tool paths to clean them up. Manually identifying these areas and generating operations to cleanup these areas are a difficult processes and extremely time consuming. In some cases, the number of the uncut areas is large, so manually determining the uncut areas and creating corresponding cleanup operations are formidable tasks. Therefore, automatic generation of the re-machining operation is desired.

2.5 Unigraphics System

Unigraphics is an interactive CAD/CAM system designed as a flexible and cost-effective method of automating designing, drafting, and manufacturing processes. Its CAM system is widely used in manufacturing industry, and it provides generalized solutions to a variety of practical problems. It is accessible in full detail to this research. Therefore, this research is developed under the Unigraphics environment. In this section, we will briefly introduce the research environment—Unigraphics CAM system.

Unigraphics CAM is a leading software in CAM commercial systems; it is widely used in auto and aerospace industry. It can robustly generate a variety of tool paths based on a wide range of applications.

A tool path can be automatically generated once the part geometry, tool specification, and other parameters that confine the tool motion are specified. The entity that contains information needed for tool path generation is called an operation. Part geometry is designed from a CAD module or imported from other modeling systems via IGES files. Based on different types of machines, operations are classified as lathe, wire EDM, or milling operations. With different goals in mind, milling operations are further divided into roughing and finishing operations. The difference between roughing and finishing operations is that roughing is designed to remove volume of materials from the stock whereas finishing is designed to remove materials from the part surface to produce a satisfactory finishing surface.

An NC program can be generated through the following sequence: loading or creating a part (in terms of geometry), creating certain operations and selecting corresponding geometry from the part, setting up detailed parameters such as ramp angles, and the like. After this information is set, corresponding tool paths can be generated automatically.

CHAPTER 3

PROBLEM CLASSIFICATION

Milling is very important and mathematically the most complex process in manufacturing. Most parts are produced by milling. As we have mentioned in the introduction, this study is focused on 3-axis milling operations; 3-Axis milling operations usually can be divided into two groups: finishing operations and roughing operations.

As we have defined in the introduction, restmaterials are the excessive materials left uncut after a finishing operation. Although pocketing or cavity milling is a roughing operation, in some extreme cases, the user may also apply it as a finishing operation. Therefore, in the following sections, a brief description of restmaterials in pocketing or cavity milling is also included besides the discussion of different types of restmaterials left on part surfaces after a finishing operation.

3.1 Restmaterials in Finishing Operation

A finishing operation contours the part surface, with the limitations of the tool shape, ramp angle or constant stepovers; it may leave restmaterials on part surfaces. There are mainly three types of restmaterials, namely,

- valley restmaterial
- ramp restmaterial
- steep restmaterial

3.1.1 Valley Restmaterial

In finishing operations, when the tool makes multiple contacts with part surfaces, restmaterials will be left. Figure 15 shows a side view of a tool making double contacts with two surfaces of a concave edge. Materials between the tool and the concave edge shown in Figure 15 are the uncut areas, which are called valleys in the following context.

3.1.2 Steep Restmaterial

Tool paths are a set of curves embedded in free-form surfaces, which contours part surfaces. These curves are usually generated by intersecting parallel planes with the part surfaces, and the resulting intersections curves are considered as traces of cutter contact points of the part surfaces. Therefore, with the surface normal the cutter location points can be computed. This method has been widely used in 3-Axis tool path generation.

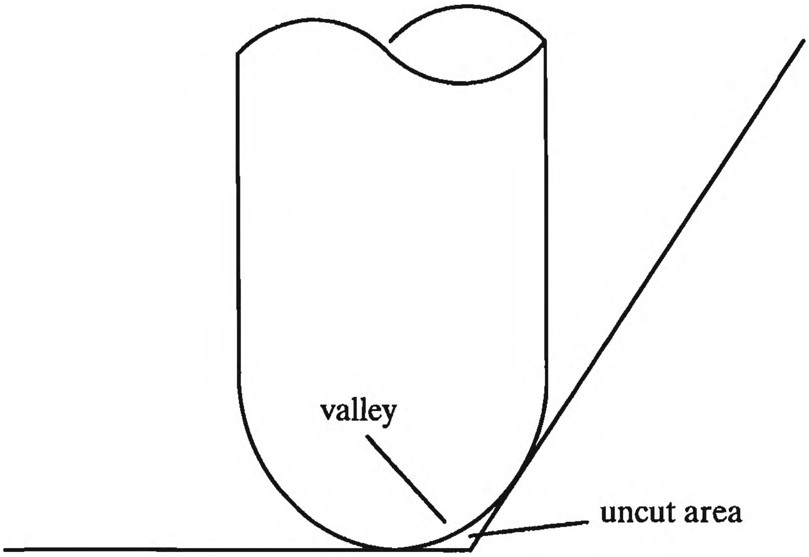

Figure 15 Uncut area left in a concave valley

Assume the coordinate system as follows: the Z axis is parallel to tool axis, and the X axis is perpendicular to the set of parallel space planes. Therefore, the set of parallel planes can be represented as $\{y = y_0 | y_0 \in [y_{min}, y_{max}]\}$, y_{min} and y_{max} are the minimum and maximum y coordinate of the part surfaces that are to be intersected. In tool path generation, normally the parallel planes are evenly spaced, which means, the set of planes can be represented as $\{y = y_0 + d * i | 0 \leqq i \leqq (y_{max} - y_0)/d - 1, i \in \mathcal{N} \text{ and } y_0 \geqq y_{min}\}$. An example of the intersection lines can be found in Figure 16.

In some extreme cases, no matter how small d is, the tool position changes abruptly as shown in Figure 17. The angle α between the tool axis and a part surface normal is called the steepness angle in this paper. When $\alpha \cong 90°$, $1/\cos\alpha \rightarrow \infty$, therefore, $d/\cos\alpha \rightarrow \infty$. This means the tool position is indefinite. Therefore, the generated tool path is not robust and may cause the tool to vibrate or plunge.

As analyzed above, when the steepness angle of a part surface exceeds a certain value (usually input from mechanist), this portion of the part surface can be avoided in the preset cutting operation, a separate operation to clean it up is desired to achieve its efficiency.

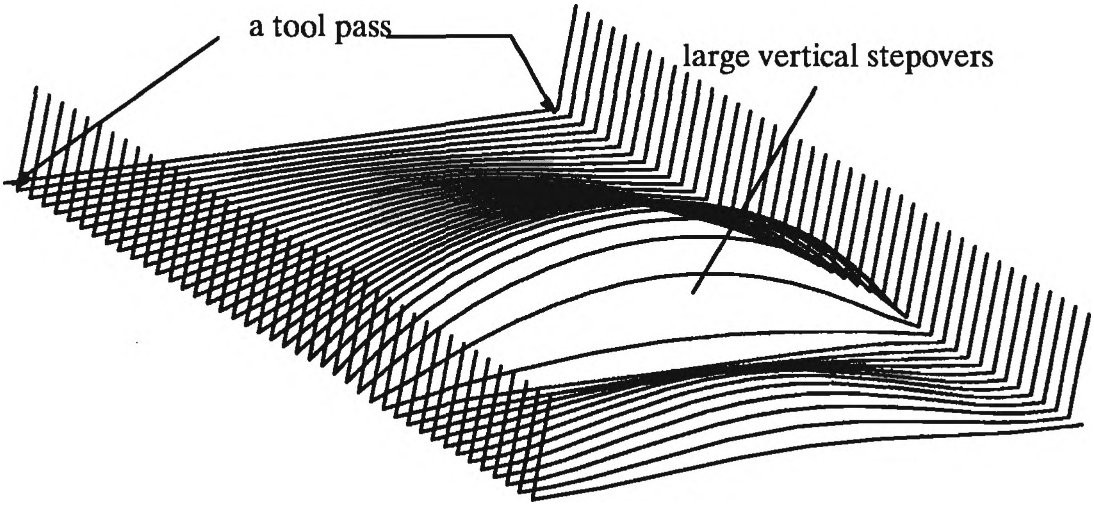

Figure 16 Unsatisfactory surface finish due to constant stepover

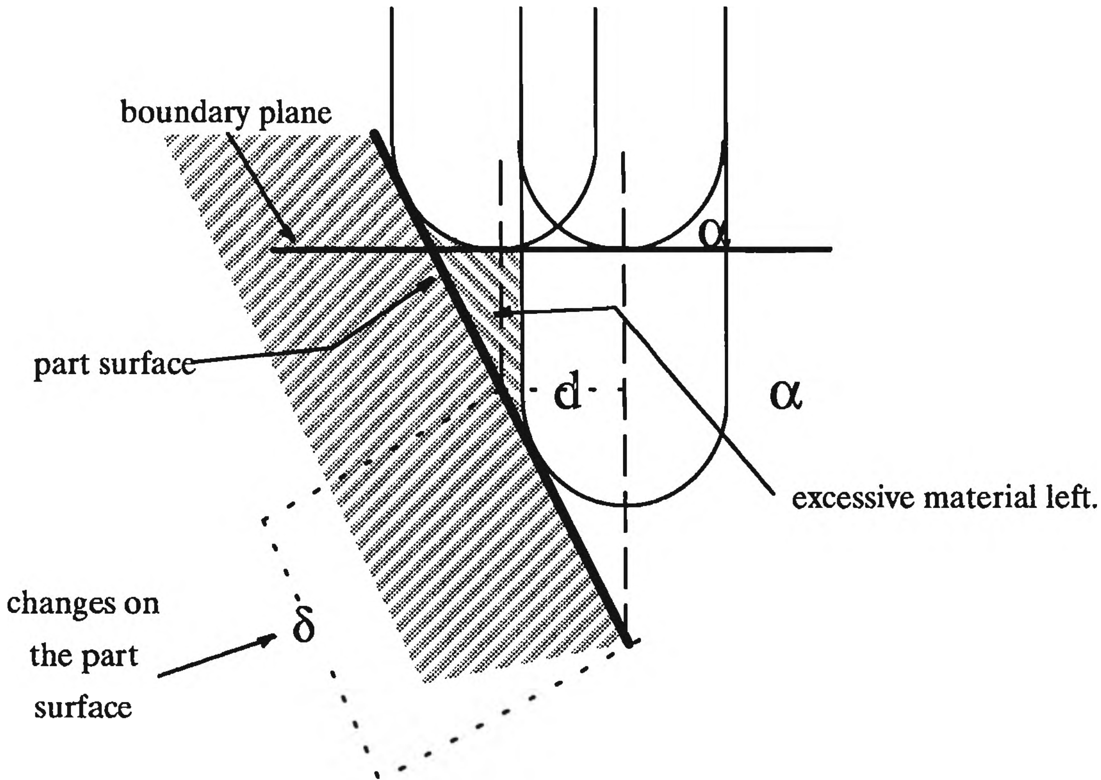

Figure 17 Problem with projection when area is too steep

3.1.3 Ramp Restmaterial

Material can also be left uncut because of the limitation imposed on the ramp angle. As shown in Figure 18, although the tool diameter allows it to cut into the shaded area, the ramp angle limitation would prevent this and the tool can only move from the start contact point to the end contact point. Contact points of the start of a ramp and the end ofof a ramp are called r***amp points***. Areas enclosed by ramp points are the ramp areas.

3.2 Restmaterial in Cavity Milling/Pocketing

Cavity milling is usually used to remove volume of materials enclosed by part surfaces. These materials are removed layer by layer; each cut removes the material from each range as shown in Figure 1. Uncut materials may be left around the concave edges on the walls, the bottom, or flat bottom floors in cavity machining. Three different types of uncut areas in cavity milling are explained in the following sections.

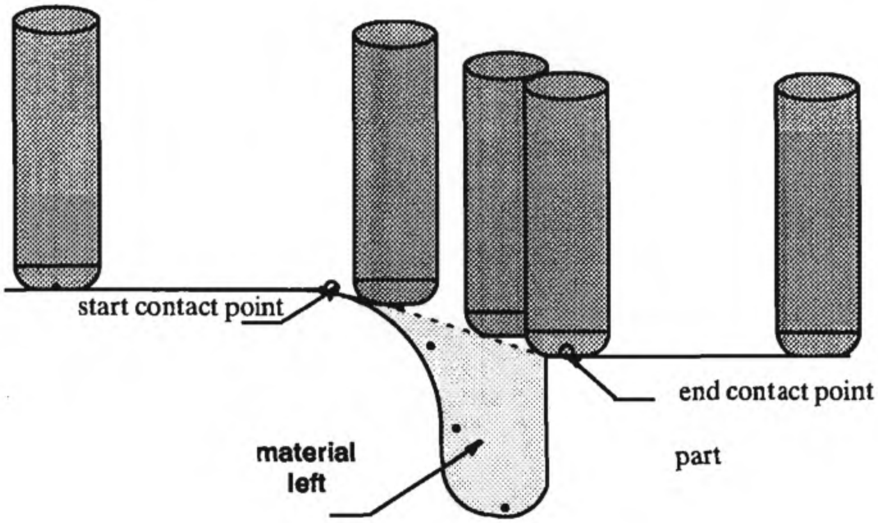

Figure 18 Uncut material due to the maximum ramp angle

3.2.1 Uncut Areas on Walls

It is common that there are concave edges on the walls of a cavity. In Figure 19, there are four edges on the cavity walls. When a tool cuts through the concave edges, it cannot reach the corner, as shown in a top view of a single layer cut in Figure 19. Uncut areas in Figure 19 can accumulate when the tool cuts through layer by layer as illustrated in Figure 20.

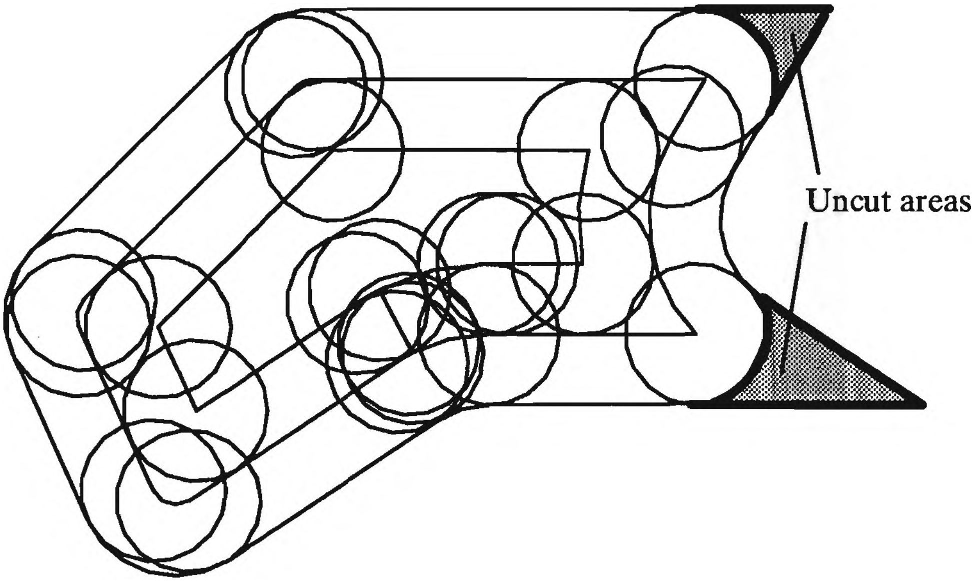

Figure 19 Top view of uncut areas due to a narrow corner

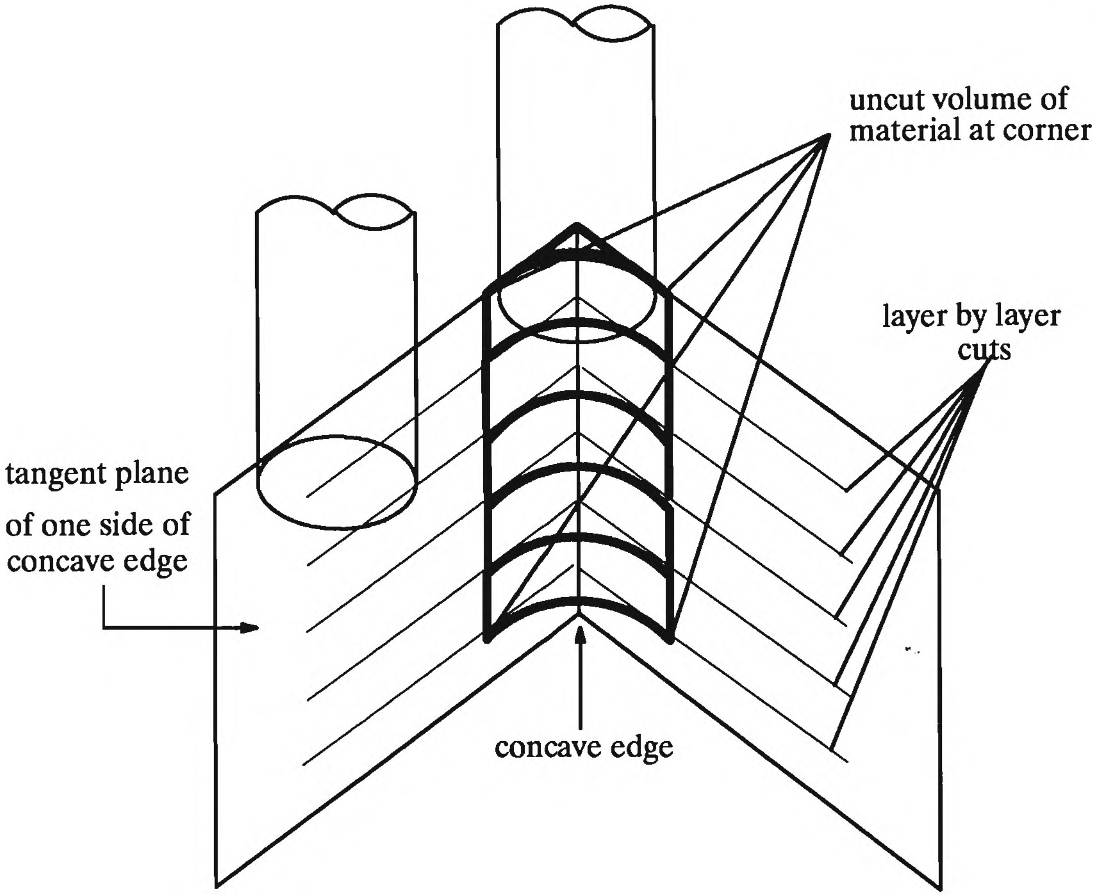

Figure 20 Isometric view of uncut material in cavity machining at concave edges

3.2.2 Restmaterial on Floors in Cavity Milling

In addition to the problems mentioned above, restmaterials may also be left on the floor of a part machined by cavity machining when stepping in the Z direction. Symmetrically to steep area in surface machining, when the angle between a part surface normal and a tool axis is small or close to 0°, excessive material may be left. This usually occurs in areas close to the floor of cavity machining as shown in Figure 21. In Figure 22, α is the angle between tool axis and part surface normal which is called the flatness angle in this paper. When $\alpha \cong 0°$, a small step in the Z direction, d, will be amplified by the factor1/(tanα). If $\alpha \cong 0°$, 1/(sinα) ->∞, then, d/ (tanα) -> ∞. This means the tool position moves dramatically when $\alpha \cong 0°$. Therefore, excessive material will be left on the floor of the cavity.

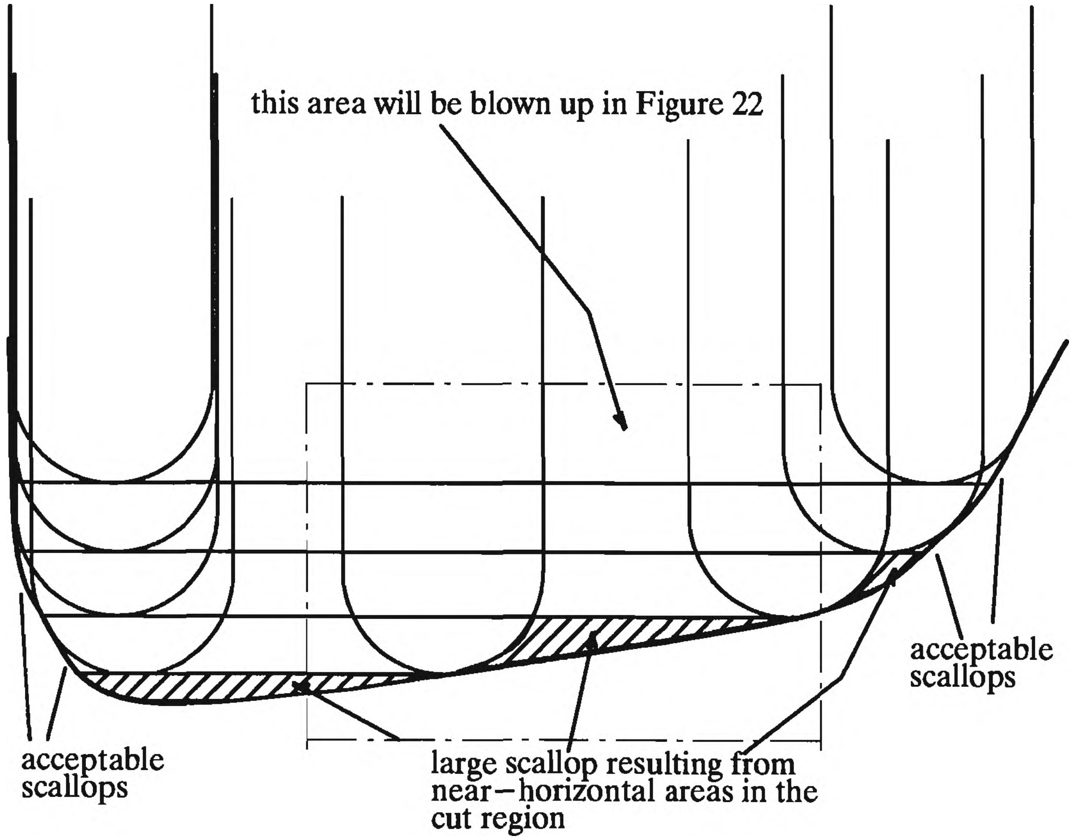

Figure 21 Large horizontal stepovers and scallops resulting from planar cuts

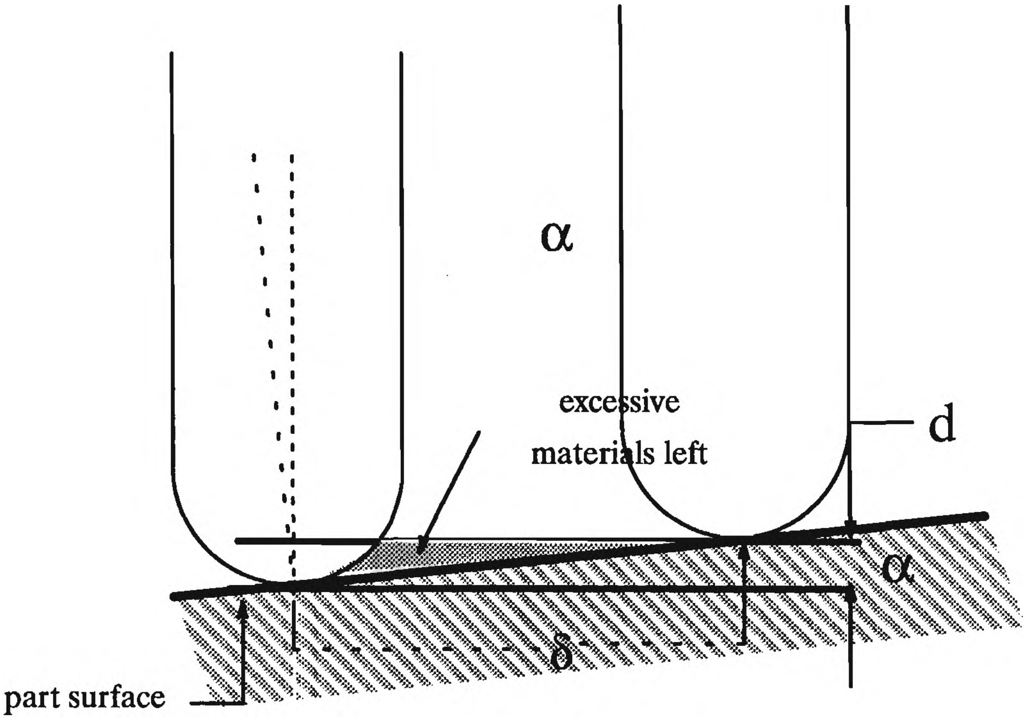

Figure 22 "Blow up view" of Figure 21 dashed region

3.2.3 Restmaterial at Concave Edges on the Floor

Around the bottom floor, concave edges also force the tool to leave excessive materials; this is the same geometric configuration as valley uncut areas in surface machining.

CHAPTER 4

IOPM MODEL

Before discussing the detailed algorithms for removing those restmaterials described in the previous chapter, we will study a mathematical model–In-Operation Part Model (IOPM). This IOPM was developed to model the execution of NC programming stages. With this model, generation of cleanup tool paths and its relationship with finishing tool path can be more easily understood. This model provides a more generalized view of cleanup tool path. In this chapter, we will first introduce the definition of IOPM model followed by an example of using IOPM model to represent generating NC programs for machining a part. Finally, dependencies of cleanup tool path to the finishing operation based on the IOPM model are studied.

4.1 IOPM Model

The development of solid technology in CAD system maintains an "informationally complete" representation of solids, and it enables much more powerful NC programming systems. With the development of this IOPM model, we wish to explore more automation capabilities in NC programming.

The NC tool path is determined by part geometries and machine constraints. An operation contains all information but the part geometry needed for NC tool path generation. The information mainly includes machine constraints, which are usually customized based on different machining practices. Therefore, the part geometry is the major variational factor in the process of producing a part. The IOPM model represents the changes of part geometry along with its associated operation information.

In-Operation Part Model is a series of intermediate solid models, which represents the geometric information of an intermediate part after each machining operation performed. The manufacturing process can be represented as a sequence of IOPMs. Once a process plan is selected, there will be a unique sequence of IOPMs representing the intermediate part geometry during the manufacturing process. Different process plans will produce different intermediate part geometry with different tolerance sequences. A mathematical model of a machining process can therefore be represented formally as a set: {B, IOPM1, IOPM2.,...,IOPMn, D}. B represents geometric solid of the given stock, D represents the geometric solid of designed finished part, and IOPMi(i =1,2,...,n) is the geometric solid of the intermediate part after the machining operation Pi (Pi is an operation which transforms IOPM(i–1) into IOPMi).

The IOPM models are aimed at efficiently tracking the part geometries during machining processes. To machine a part, a stock, a solid model of a finished part, and a sequence of machining operations that will be performed on the work-piece will be given. From the work-piece and the specified operations, IOPM models will be created to represent the intermediate geometry of work-piece in progress. This can be illustrated in the following example.

4.2 An Example of IOPM

Figure 23 shows an example of IOPM used in representing a NC machining processes. Given a stock represented by a geometric solid B, and a solid model of finished part D, and a sequence of operations {P1, P2, P3,P4}, the IOPM sequence for this part manufacturing process will be {B, IOPM1, IOPM2, IOPM3, IOPM4,D}, where IOPMi (i=1,2,3) is the intermediate solid model generated after operation Pi is performed on the stock piece. With these intermediat part models representing stages in machining process, we

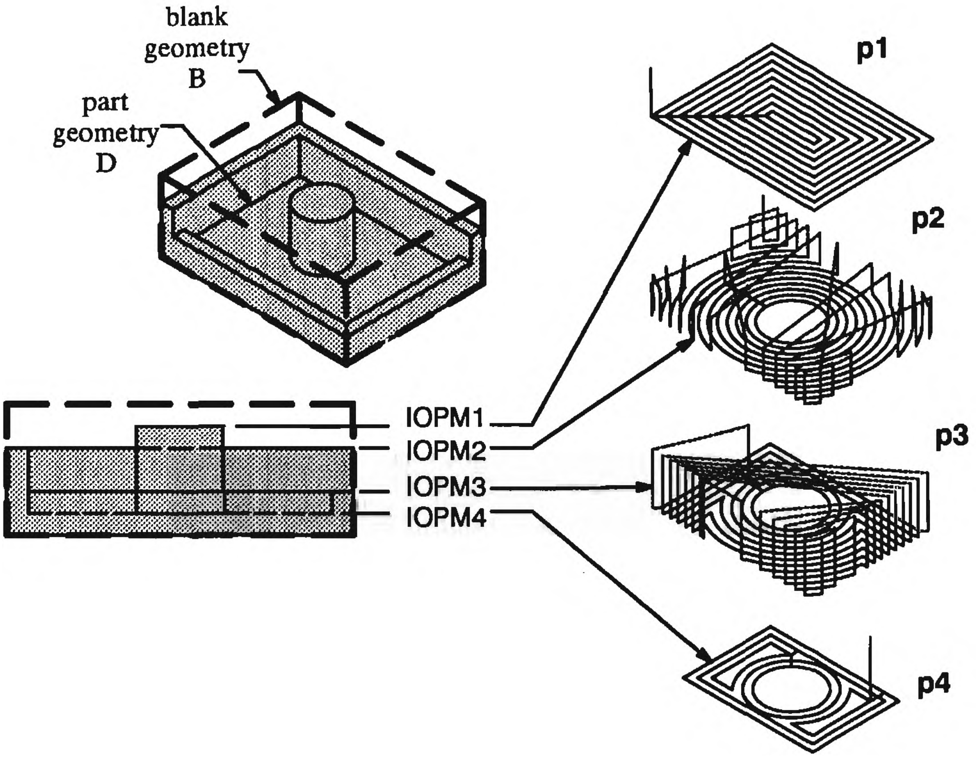

Figure 23 In-Operation Part Model chart

can optimize the machining process based on the intermediat IOPM model. The compensating cleanup tool path is an optimization based on the final stage of the IOPM model, as for this example the IOPM4.

4.3 Modeling of Cleanup Tool Path

Study of the geometric differences between two consecutive IOPMs may lead to the automatic generation of successive operations. The restmaterial discussed in this research is the geometric differences between the one next to the last IOPM and the last IOPM—D. By study of this type of restmaterial, cleanup operations will be generated automatically.

An IOPM is the result of the last operation performed on its previous IOPM and the given geometry of next operation. It ties two consecutive operations. With the IOPM model, associativity between operations can be discussed. The study of associativity between operations can provide optimization and automation of operation generations.

With the refined uncut area boundaries and the given part surface, cleanup operation can be created. We can use the same tool but a different cut pattern to cleanup steep and horizontal areas, or use a tool with smaller radius to cleanup valley or concave edges. Since we have classified uncut area boundaries into different categories (valley, steep, horizontal, etc.) and designed different cut patterns that can efficiently remove corresponding areas, cleanup operations can be created automatically in the system by inheriting other parameters from their previous operation. This can be clearly illustrated by In-Operation Part Models (IOPM).

4.4 Automatic Generation of Cleanup Tool Path

Three types of parameters can be specified to provide users a full control over the cleanup operation while the generation of cleanup operation is hidden from user. These three types of parameters are as follows:

- Cleanup machine tool: The user can specify a cleanup tool when the parent operation is created. When this parameter is missing, a default cleanup tool will be provided by the system.
- Cleanup geometry optimization factors: These factors include merge distance, overlap distance, minimum area, and smoothness/accuracy scale. If any of these parameters is missing, default values will be provided by the system.
- Cut Pattern.

Cleanup operation is dependent on part geometry, operation parameters, and tool path of the finish operation that we call its parent operation: the uncut valley area is dependent on the reference tool used in the parent operation and the part geometry. An uncut ramp area is caused by part geometry and ramp angle limitation imposed in parent operation; an uncut steep area results from parent cut direction dependencies and the surface normal of part geometry. Therefore, the cleanup operations are dependent on their parent operations:

- Parameters will be specified in the ancestor operation to allow the creation of the linked cleanup operations.
- If the parent operation is deleted, the linked operation should also be deleted.
- If the parent operation has been updated, the linked operation should also be updated.
- A limited set of cut patterns will be supported by the system. The user can either choose his or her own cut pattern or use the system default cut pattern.

CHAPTER 5

VALLEY AND RAMP RESTMATERIAL REMOVAL

In this chapter, we discuss detailed methodologies used for valley and ramp restmaterial removal. The definitions of valley and ramp restmaterials can be found in Chapter 3. We combine ramp and valley into one chapter because they share the same algorithms for detecting restmaterials; their cleanup tool path generation is also similar. The key to automatic cleanup tool path generation is the reconstruction of geometric information of restmaterials. The computed geometric information should be accurate, smooth, and useful for NC tool path generation algorithms.

5.1 Restmaterial Areas Detection

In discrete vector approximation method (as in section 3.1.1), valley-shaped restmaterial areas have been well studied. Different approaches for analyzing these restmaterial areas can be found in the review section of NC tool path simulation and verification in Chapter 2. Although no explicit computation of restmaterial areas has been carried out in those studies, technologies to detect restmaterials introduced in those studies can be used here. In this section, an algorithm that explicitly constructs restmaterial geometry information will be discussed. Different from other NC tool path simulation and verification methods, this approach uses information from previous finish operation generation as input and uses numerical methods to approximate restmaterial geometry as sufficient as needed by cleanup tool path generation.

5.1.1 Double Contact Points

As we mentioned above, using extra simulation of NC tool paths could also detect restmaterial areas, but all of these methods like direct vector method, dexel method, and voxel method discussed in the background chapter require expensive computations. Here, instead of doing an extra NC tool path simulation and verification, double contact points detected in finish tool path generation are saved and used in computing the boundaries of restmaterial areas. Similar information of ramp restmaterial areas will be saved and used for computing ramp restmaterials.

An example algorithm for how these double contact points are computed can be found in [Hansen89] as described in Chapter 4. Some other gouge free tool path generation algorithms have also discussed these double contact points. Rather than reiterating Hansen's work, we refer the interested reader to the original work in [Hansen89].

5.2 Boundaries of Restmaterial Areas

In this section, boundaries of restmaterial areas are computed based on the saved double contact points and/or ramp points. Two major algorithms—**2D Spatial Directory** and **Annealing** are developed/used to compute the boundaries. The followings are details of how these algorithms were developed and applied in restmaterial construction.

Before discussing the details of these algorithms, an intuitive example of a valley restmaterial boundary is given in Figure 24. The working tool is too big to remove materials in the dark-shaded area between the tool and the part surface. With the three adjacent tool positions along the valley, uncut areas boundaries can be approximated by polygon $AA'A''B''B'B$, as shown in Figure 24 b. Material between the light-shaded area (as shown in Figure 24 a) and the part surfaces is considered as an approximation of the restmaterial; therefore restmaterial areas such as valley areas, concave edges on walls, and concave edges in the bottom floor can all be approximated by connecting their contact points together into boundaries.

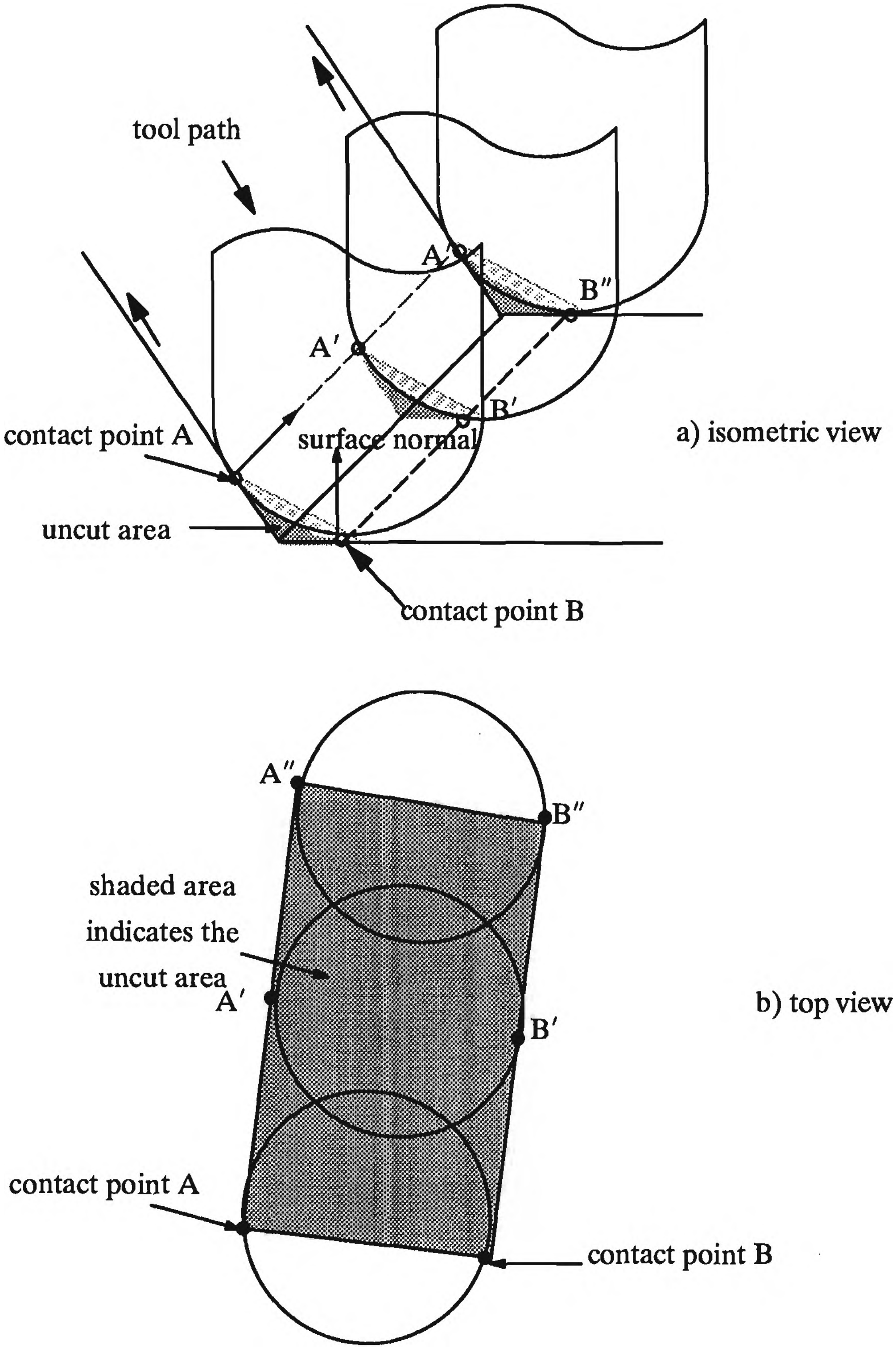

Figure 24 Valley restmaterial

As we can see from this example, the excessive material will rest inside the area enclosed by those pairs of double contact points, restmaterial boundaries can be constructed by connecting these adjacent double points together. Two ways can be used in connecting these double contact points:

- When a zigzag cut pattern is used in the finish operation, adjacencies of double contact points can be easily retrieved by recording the pass number. The definition of a pass can be found in Figure 2;
- When irregular cut patterns are chosen, adjacency information of double contact points are too complex to record or retrieve. An approximation method is used in this circumstance.

Because the goal of computing restmaterial geometry is to provide enough information for cleanup tool path generation, the approximation of restmaterial boundaries is allowed when all restmaterials are included and the deviations of the boundaries should be as small as possible to avoid unnecessary cleanup tool path. To ensure the generality of this approach, irregular cut patterns are assumed, and algorithms that anneal the double contact points together are developed. Before double contact points are annealed into boundaries, they are first grouped into different regions based on the 2D Spatial Directory Algorithm.

5.2.1 2D Spatial Directory Algorithm

The center of each pair of contact points is first computed and then projected down to a plane perpendicular to the tool axis. A rough grid on the projection plane is created so that these centers can be stored to their closest grid vertices; each vertex is like a directory for a cluster of centers. Usually, the size of a grid is half of the tool radius. Nonempty vertices (vertices with at least one center attached) are colored as black; other vertices are left as default white. An example for this is given in Figure 25.

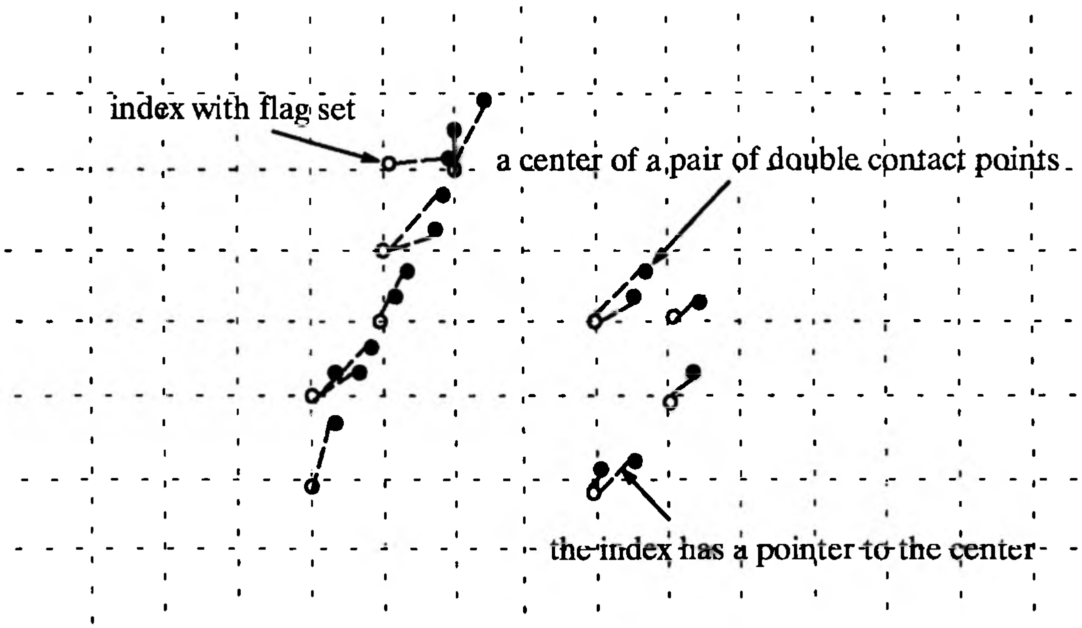

Figure 25 2D Grid for center of double contact points

The dimensions of the grid are n x m:

n = (x_max – x_min) / tol + 1;
m = (y_max – y_min) / tol + 1;

Where x_max, x_min, y_max, and y_min are maximum, minimum x, y coordinate values, respectively, and tol is set as half of the tool radius used in finish operations. The algorithm for grouping the center points is detailed as follows.

After the creation of the grid, all grid vertices are initialized as white-colored pixels; then center points of double contact pairs will be projected and indexed into grid vertices; the vertex that contains more than one center point is flagged as black.

A search algorithm starts from the first row of the grid to the last row to connect adjacent center points into polylines (with the grid vertices as vertices of the polyline). One linked list is used to keep tracking active polylines and another list is used to keep tracking completed polylines.

An active polyline has one or more new adjacent neighbor grids to connect to. When a polyline does not have new adjacent grid index to continue with in the current row of searching, this polyline is considered completed and is moved from the active polyline list into the complete polyline list. Adjacent vertices to a vertex (i, j) are {(i–1, j–1), (i, j–1), (i+1, j–1), (i, j+1), (i+1, j–1), (i+1, j), (i+1, j+1)}, as shown in Figure 26.

(Centers of points are first projected onto the Spatial Directory (SPD) map. Indexes of SPD map are initialized as white color indexes. A pointer will be created pointing to a center point for the closest SPD index to the projected center point, and this index is also marked black.)

```
Active_polyline=NULL, Completed_polyline=NULL;
For each row do
      For each index in the row
         If (index is black)
            If(neighbor of any polyline)
               add to the polyline;
```

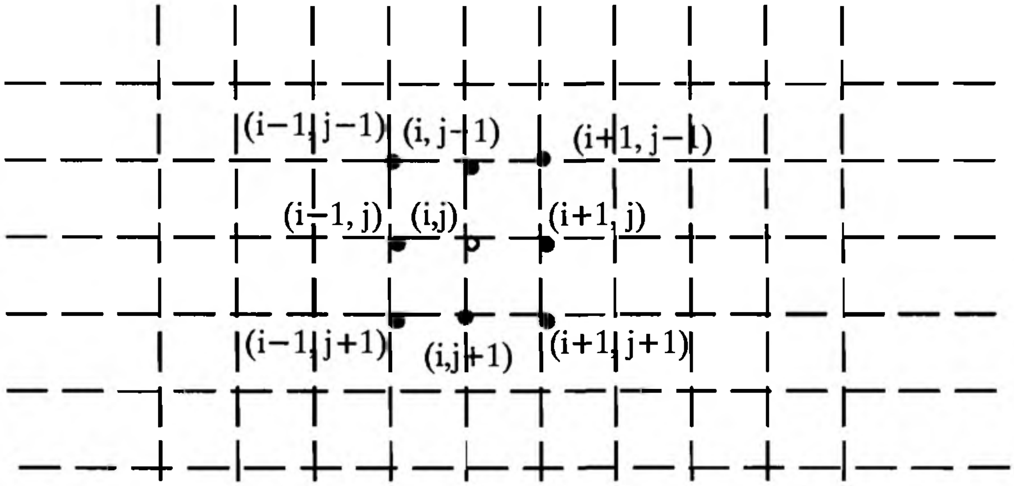

Figure 26 Adjacent vertexes of a grid point

```
            Else
              create a new polyline with current index as the starting point
            Endif
          Endif
    /*check Active_polyline list*/
    If (a polyline does not have a new adjacent grid vertex added to it)
        move it from the Active_polyline list to the Completed_polyline list
    Endif
    /*check points left*/
    If ( a grid point has not been added to a polyline)
        create a new polyline and add to the Active_polyline list with this point as a start
    point.
    Endif
Endfor
```

It should be emphasized that a vertex could be a branching point. A branching point has more than one adjacent index in the new grid row, as Point P shown in Figure 27, where a, b, and c are all adjacent points to Point P. In such case, the polyline will prefer a straight line connection rather than a diagonal connection. As shown in the figure, b is chosen to be the next connecting point over points a and c. The unconnected index will become the starting point of a new polyline.

If the branching point (as shown in Figure 27, the point M) has multiple adjacent points (Points l and k) whose index differences are the same, then an arbitrary one (Point k) can be connected, and the remaining point(Point l) becomes the starting point of a new polyline.

Based on the connectivity of the center points, contact points therefore can be connected. The idea that roughly puts clusters of 3D points into 2D grid vertices and refines information by doing computations within each vertex is called the 2D Spatial Directory.

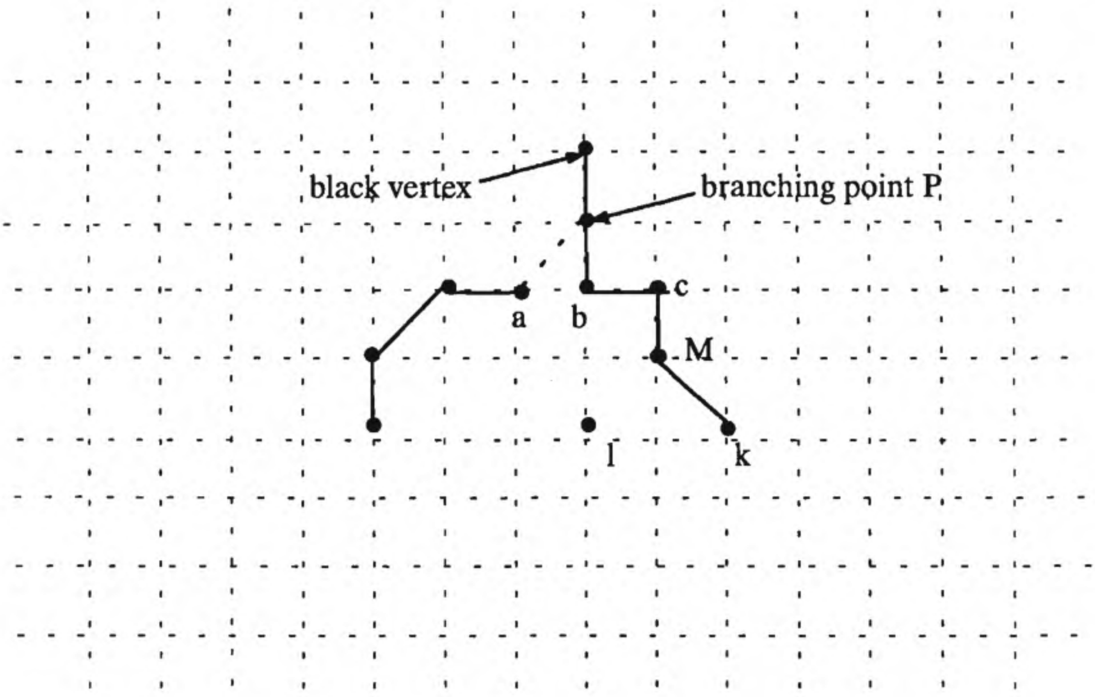

Figure 27 Branching point

It should be clarified here that connecting into polylines does not ensure that all points in one polyline belong to the same restmaterial boundary; they may be disconnected later in a filtering process. On the other hand, the disconnected points may be connected to each other later as two or more polylines may be joined at branching points. Exceptions exist in the above computation where there may be a thin wall between two adjacent center points. Therefore, normals of the contact points are used to detect thin walls between neighbor points or discontinuity of uncut areas under certain geometric configurations. Since the cleanup operations are created to remove restmaterials, as long as the boundaries enclose all uncut areas, the result is acceptable only if there is no thin and high walls inside the boundaries of uncut areas. This is because it may cause a tool to move abruptly or waste too much machining time when the tool travels on the thin and high walls.

5.2.2 Annealing Points into Boundaries

Simulated annealing [3, 27, 28] is a technique which was developed to help solve large combinatorial optimization problems. It can be applied to problems where one can reach all feasible points by performing a sequence of steps and where for each step the amount of change in the measure of quality to be optimized can be quantified. It is based on probabilistic methods and is designed to avoid getting stuck at local minima. At each stage of the process, both good steps (steps that improve some measure of quality) and bad steps (steps that do not improve measure of quality) are made. The bad steps are chosen randomly, and as one proceeds from stage to stage, the probability that they achieve is slowly reduced according to a prescribed annealing schedule. The detailed simulated annealing algorithm and analysis of their complexity for closest point connection (traveling salesman problem) can be found in [42].

5.2.3 An Example

A real part is used for the demonstration of how the 2D Spatial Directory worked. Figure 28 shows a corner of a real part, the dashed lines are the 2D grid. The “*” at each grid represents the node that have center points attached to it. The grids marked with “*” are connected by solid lines if they are adjacent (as defined in Figure 26.) As you can see from the Figure 28, these connected solid lines approximates the shape of valley area of the part surfaces.

Center points are therefore grouped into different connected groups. By using the annealing algorithm, these center points are connected into smoother polylines as shown in Figure 29 by the white-colored and thicker polylines. The double contact points attached to these polylines can be connected to represent the boundary of the valley area. These boundaries are shown in the figure by the black-colored thinner polylines.

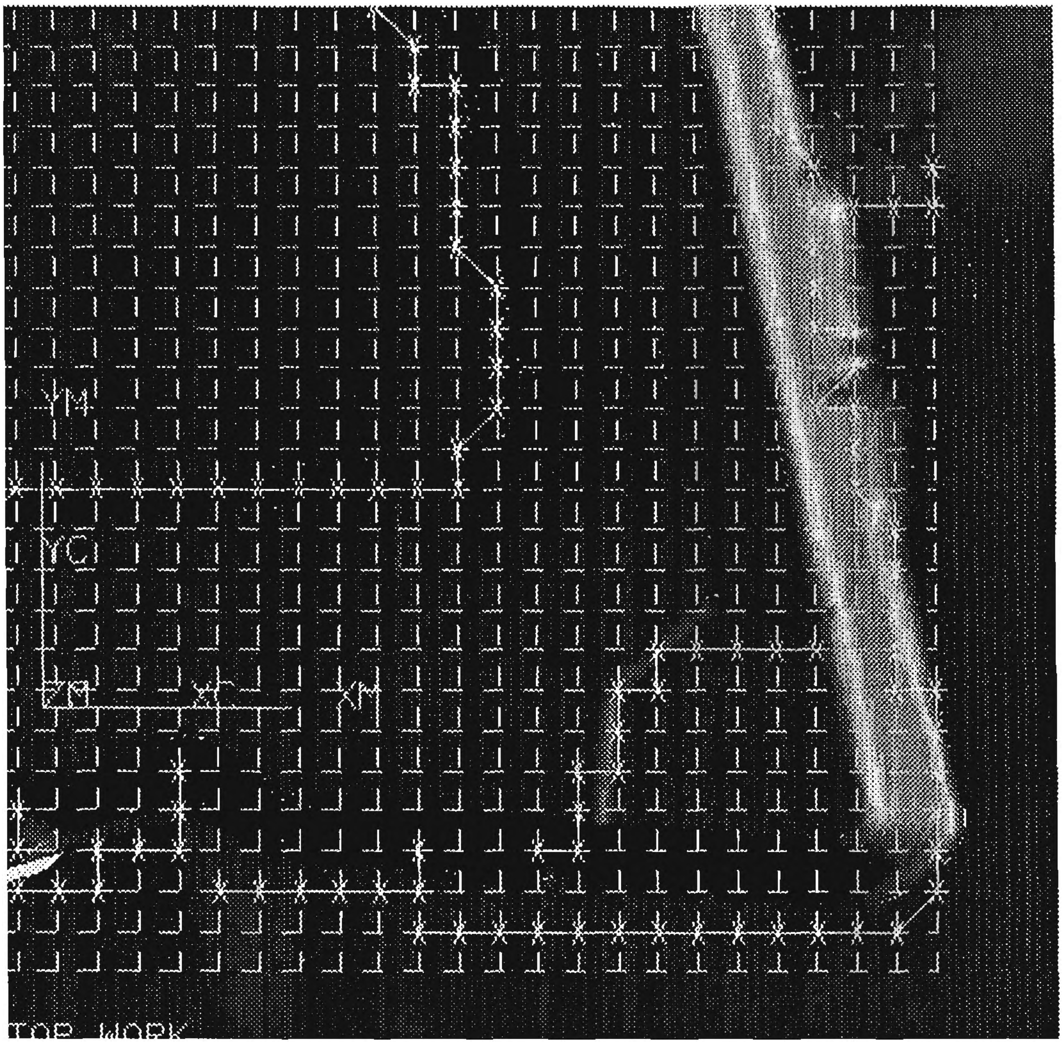

Figure 28 2D spatial directory holding double contact point information

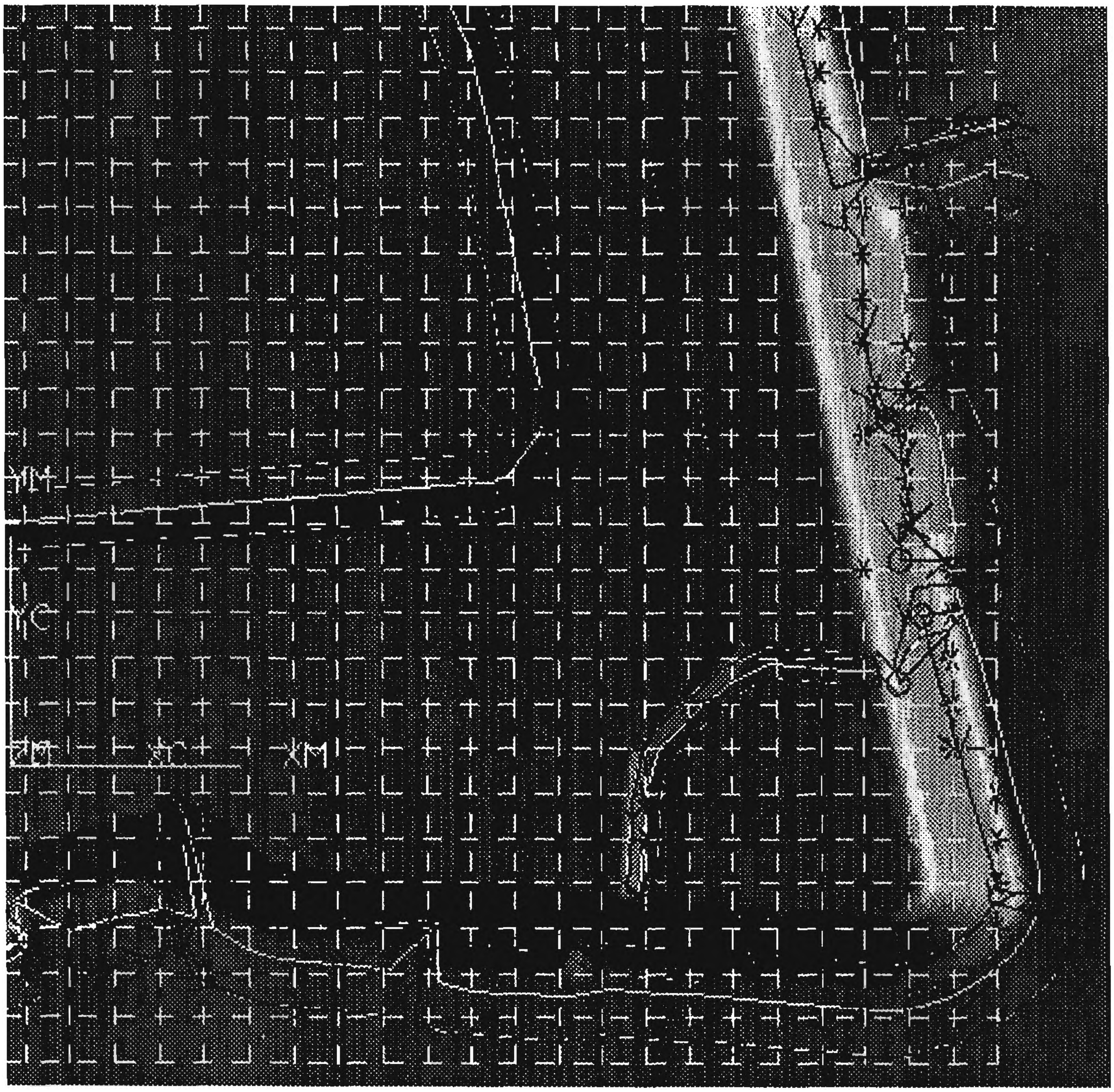

Figure 29 Connecting double contact points based on connectivity of their center points

5.3 Restmaterial Boundaries Modification

In practical manufacturing industry, there are other factors involved besides accuracy. Smoothness of the surface is also an important factor in machining a part, because this makes the part aesthetically more pleasing to customers. It should be noted here that the goal of constructing restmaterial boundaries is to provide sufficient geometric information for the efficient generation of a cleanup tool path. Therefore, after the computation of restmaterial boundaries, certain modifications should be made to transform the restmaterial geometry into the form by which efficient and smooth cleanup tool paths can be generated. In this section, we will discuss some modifications of the restmaterial boundaries, which will improve the efficiency and smoothness of the cleanup tool path. They are merging, deleting, extending, and smoothing boundaries of the cleanup regions.

5.3.1 Smoothing Cleanup Boundaries

Because of involvement of numerical approximation in the restmaterial region computing stage, aliases may cause the actually area of restmaterial to be larger than the computed cleanup region boundaries. Unfortunately, in real industrial manufacturing, smoothness is sometimes preferred over accuracy regarding the cleanup region boundaries. Using a smoother cleanup region boundary along with the designed part geometry will generate a cleanup tool path that has smooth interfaces with the rest of the part surfaces. This will produce a much more aesthetically pleasing surface finish. There are several smoothing algorithms including line fit, spline fit, and low pass filter that can be used to construct smooth cleanup boundaries within the desired tolerance. Normally, second order continuity is good enough to introduce a shine and smooth surface finish; therefore, a least square spline fit algorithm is used. Based on the algorithm, the connected double contact points form a polygonized regions. By using spline fit, the regions are enclosed by continues spline curves, leave much smoother boundaries, which can be seen in Figure 30.

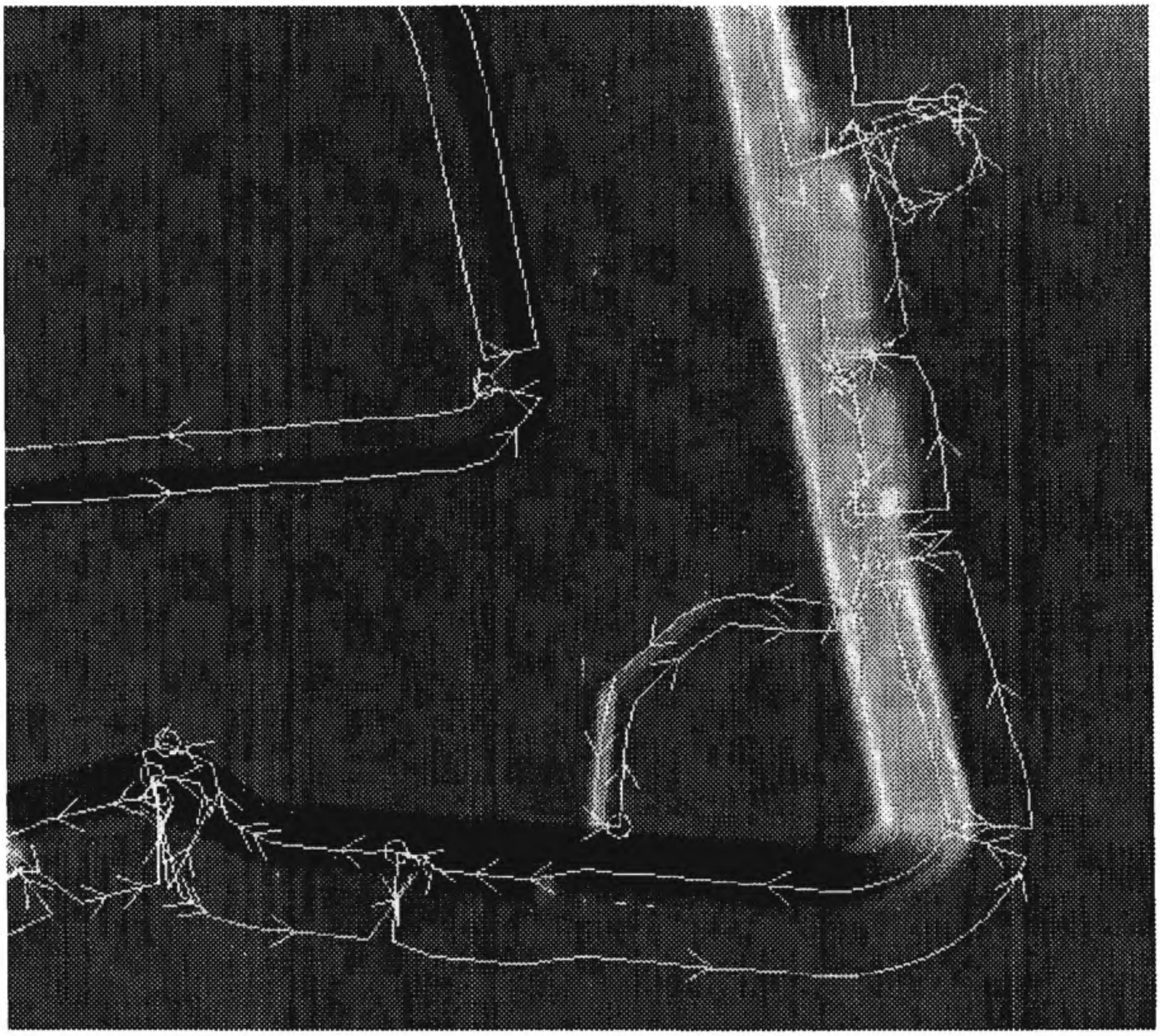

Figure 30 Boundaries without smoothing

5.3.2 Merging Cleanup Region

The "Engage/Retract" is a time and tool consuming machine motion compared to normal cutting tool motion. For each disconnected cleanup region, at least one "Engage/Retract" is required. Therefore, the number of disjoint cleanup regions is proportional to machining "Engage/Retract" numbers, which directly affects the machining time. Therefore, when two disjoint uncut areas are close within a specified threshold distance, they should be considered for merging into one bigger uncut area to reduce the number of "Engage/Retract" motions. This is illustrated in Figure 31. The left half of the figure shows three disconnected but close by uncut areas; they can be merged into one bigger uncut area as shown in the right half of the figure.

Figure 32 shows an example of Figure 30 after the merging operation. The merge distance used here is 2 mm in this part. The white curves with arrows are the boundaries of the valley uncut area. Small nearby valley uncut boundaries shown in Figure 30 are merged into bigger regions as shown in Figure 30.

5.3.3 Extending Cleanup Boundaries

Similar to that mentioned in the previous section, surface finish appearance is of great importance in the real manufacturing world. Certain types of overlaps have been desired by machinists to achieve different surface finish appearances. For instance, overlaps may be desired in the fillets in car body die manufacturing. Figure 33 shows the geometric definition of overlaps.

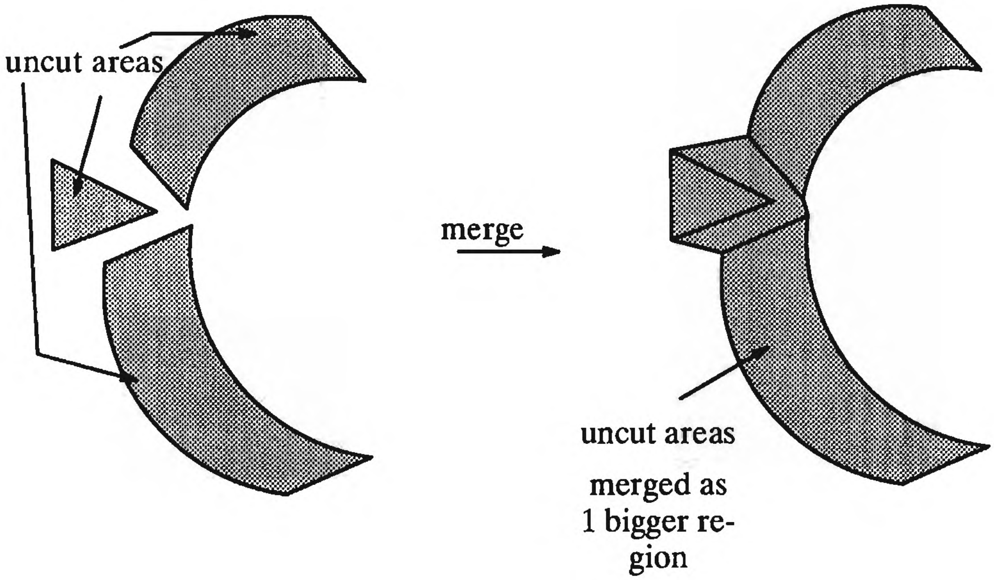

Figure 31 Merge two nearby areas into one big region

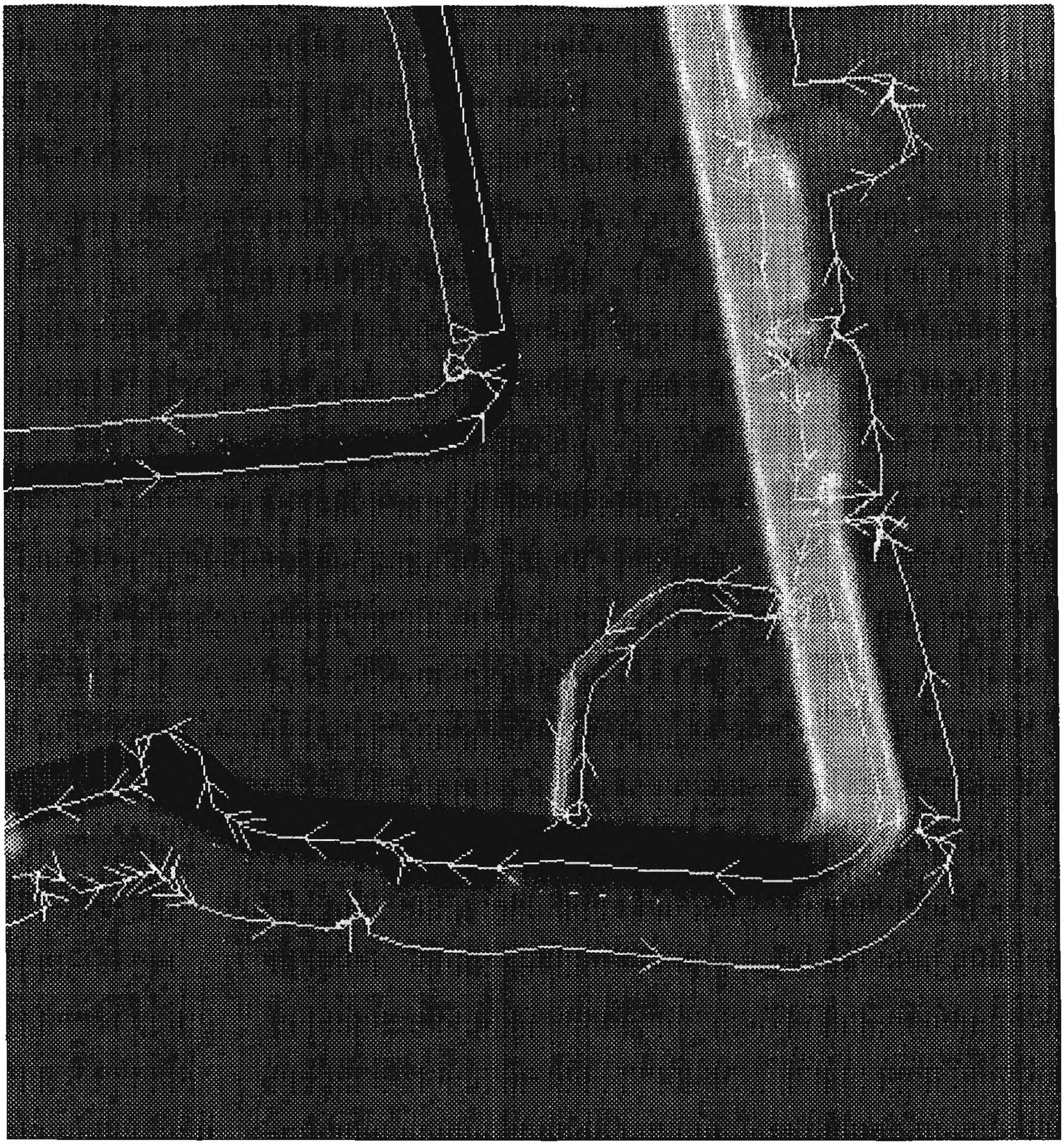

Figure 32 Boundaries after merging operation

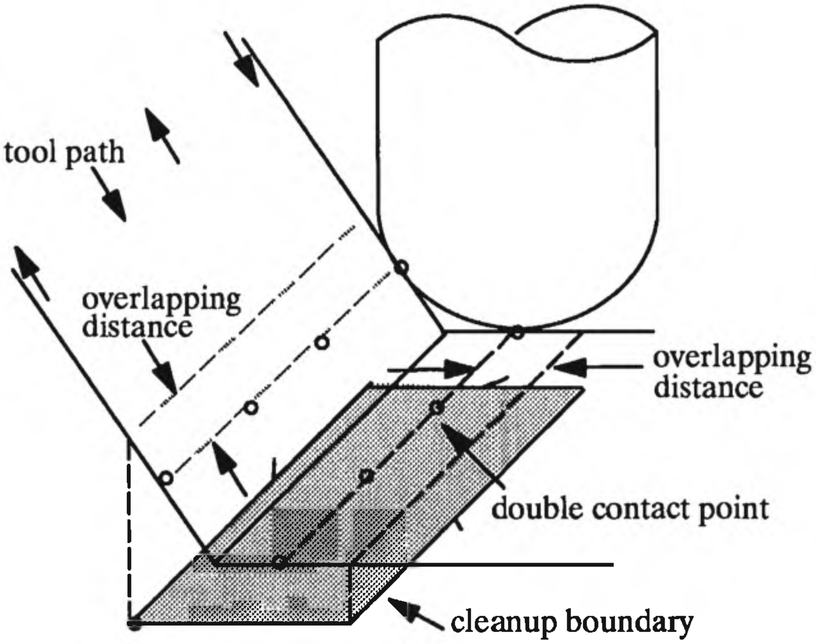

Figure 33 Cleanup boundary with overlapping distance

In Figure 34, a 2mm overlap has been applied to the uncut valley boundaries shown in Figure 32. These boundaries in Figure 34 are even more smoother and cover a larger area compared with boundaries in Figure 32. This overlap is designed to insure the coverage of restmaterial area and conducting a smoother transition between the finishing tool path and cleanup tool path.

5.3.4 Deleting Small Cleanup Boundaries

It is often the case that there are pieces of small uncut areas besides several big uncut areas. These small areas can be merged into bigger uncut areas if they are close to each other, but when small areas are too numerous and too far apart from each other, cleaning up these areas may require a large number of “engage/retract” motions, and it may be desired that these small uncut areas be ignored to speed up the cleanup operation in some cases. Since there may be very small amounts of restmaterials left in these small uncut areas, grinding may not be that costly compared to the situation when a large number of “engage/retracts” motions are required.

Figure 34 Boundary with overlaps after merging

5.4 Cleanup Tool Path Generation

Once the uncut boundaries are computed, their enclosed restmaterials can be removed more efficiently by the following specially designed tool path. The following cut patterns have been developed to assist the user in generating the cleanup operations. These cut patterns are designed specifically for different types of restmaterials; they are flow cut and follow pocket cut patterns:

- The flow cut pattern may be used for both cleanup and precut of concave edges. This pattern generates one or more parallel cuts along curves of bitangency for the cutting tool. It is shown in Figure 35.
- The follow pocket cleanup cut pattern is performed by 3-Axis contouring. It cuts along the uncut area boundaries, and gradually works toward the center of each unmachined area with a constant stepover along the surface. Figure 36 illustrates how the cut is generated.

5.5 An Example

In this section, an example of valley restmaterial cleanup for a practical part is demonstrated. The system used for the cleanup tool path generation is based on algorithms discussed in this chapter.

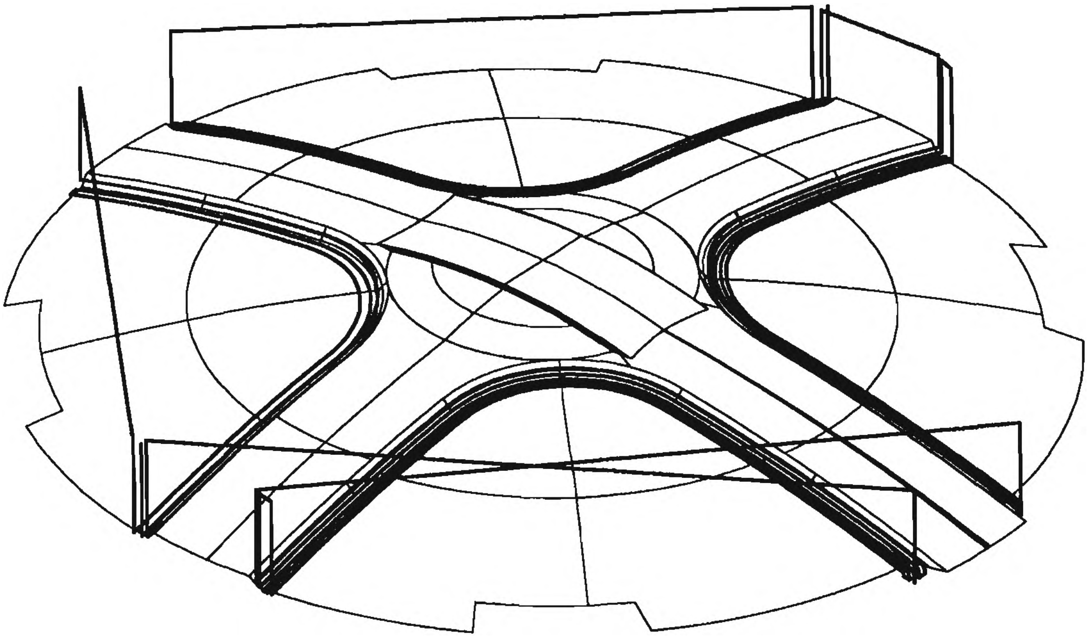

Figure 35 Flow cut pattern

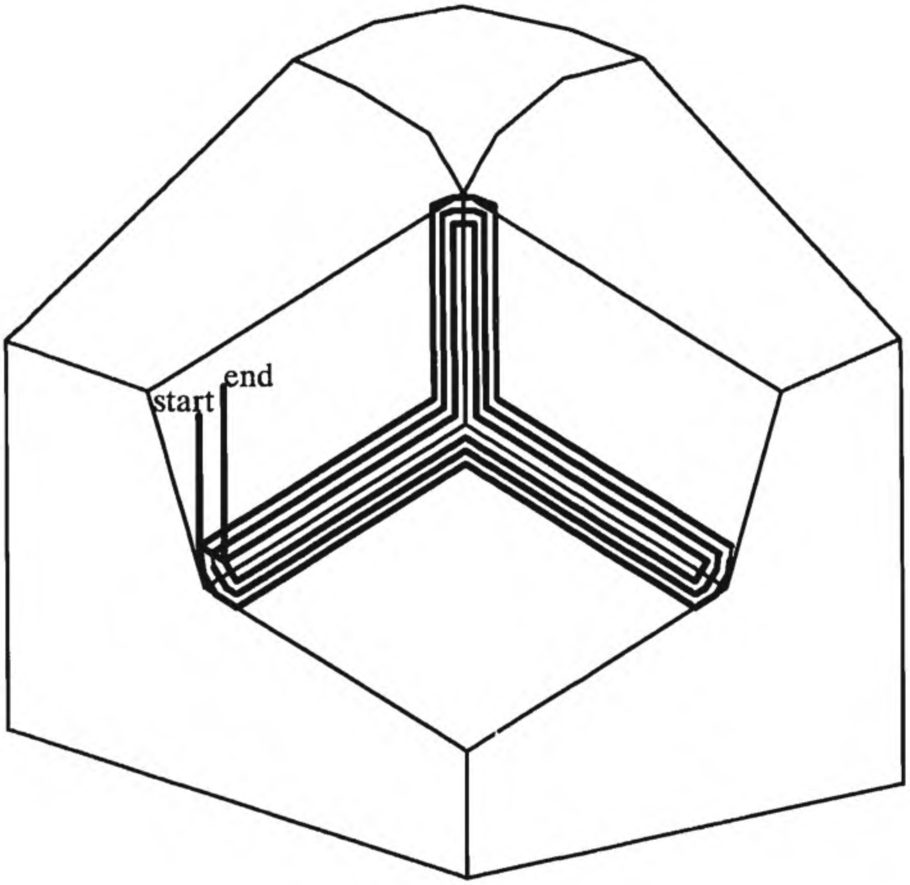

Figure 36 Follow pocket cut pattern

Figure 37 shows a shaded and wire-frame 3D display of a part designed for a car die. It is used to absorb vibrations. The design is done in the Unigraphics CAD system. There are more than 2000 freeform trimmed surfaces. Gaps and overlaps may exist among surface patches. Figure 37 is generated by a simple shading algorithm and is a rough image, just to give the reader an impression of the overall 3D structure. Figure 38 details surface patches and complexities.

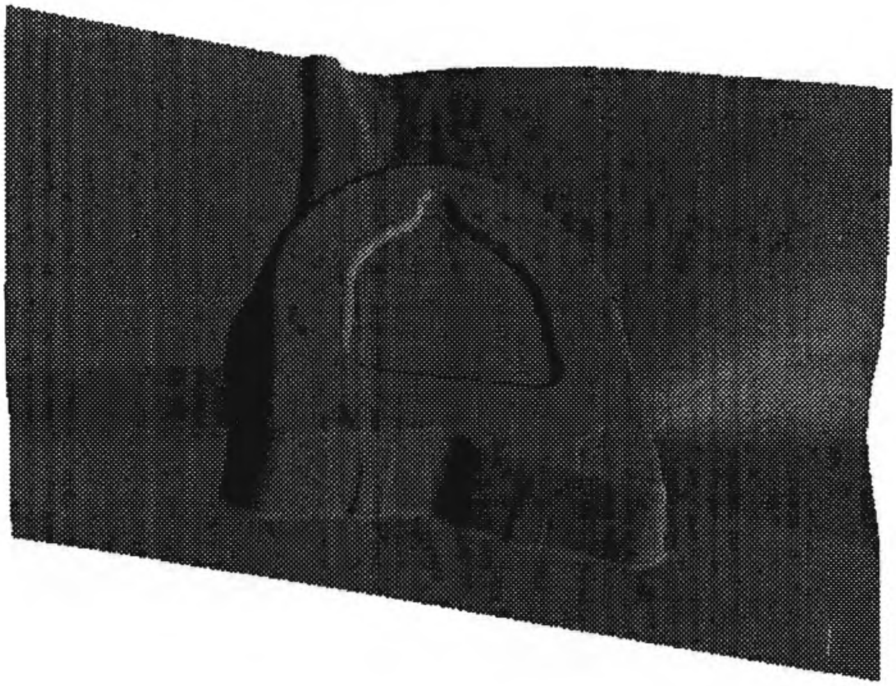

Figure 37 The car die model (shaded)

Figure 39 shows a regular finishing lathe cutting tool path for machining the car die part. In Figure 40, around the center core area, there are gray colored "+" s displayed, which indicates the double contact points are found. The area between these double contact points contains extra material that has not been removed by the finishing tool path. These restmaterial will be first configured by boundaries as shown in Figure 41. With these computed valley uncut boundaries a cleanup tool path with selection of smaller tool radius is generated as shown in Figure 42.

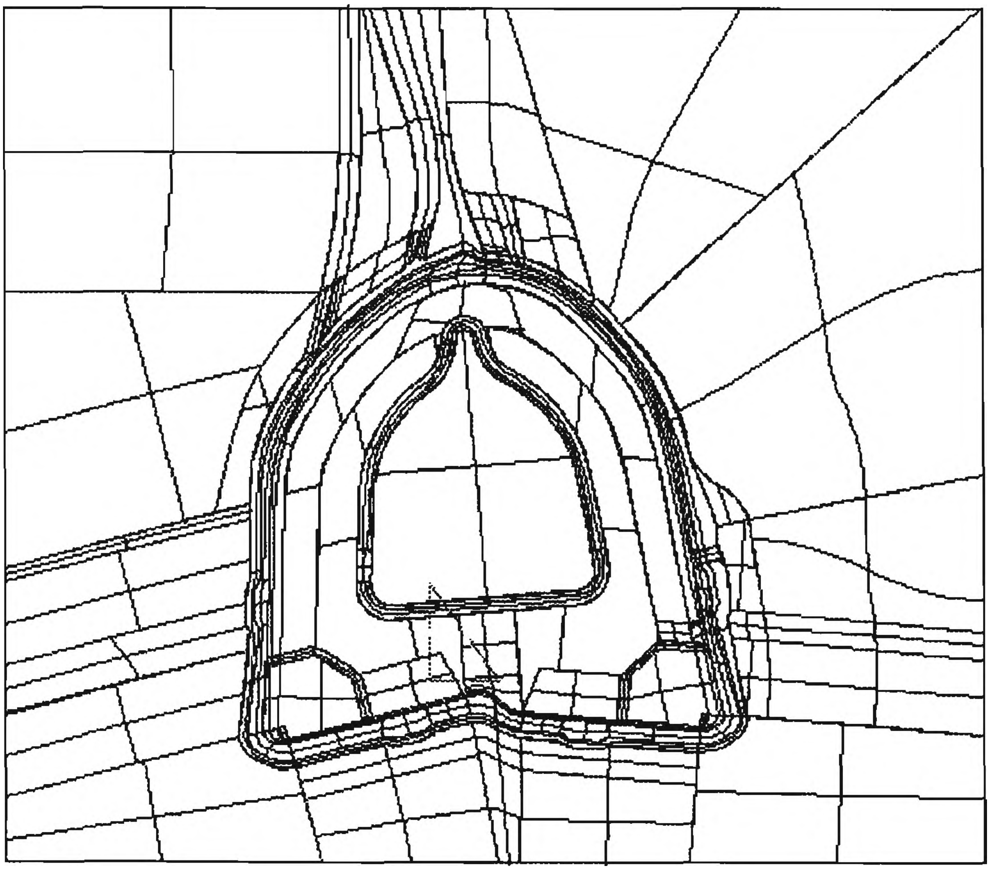

Figure 38 The car die model (wire-frame)

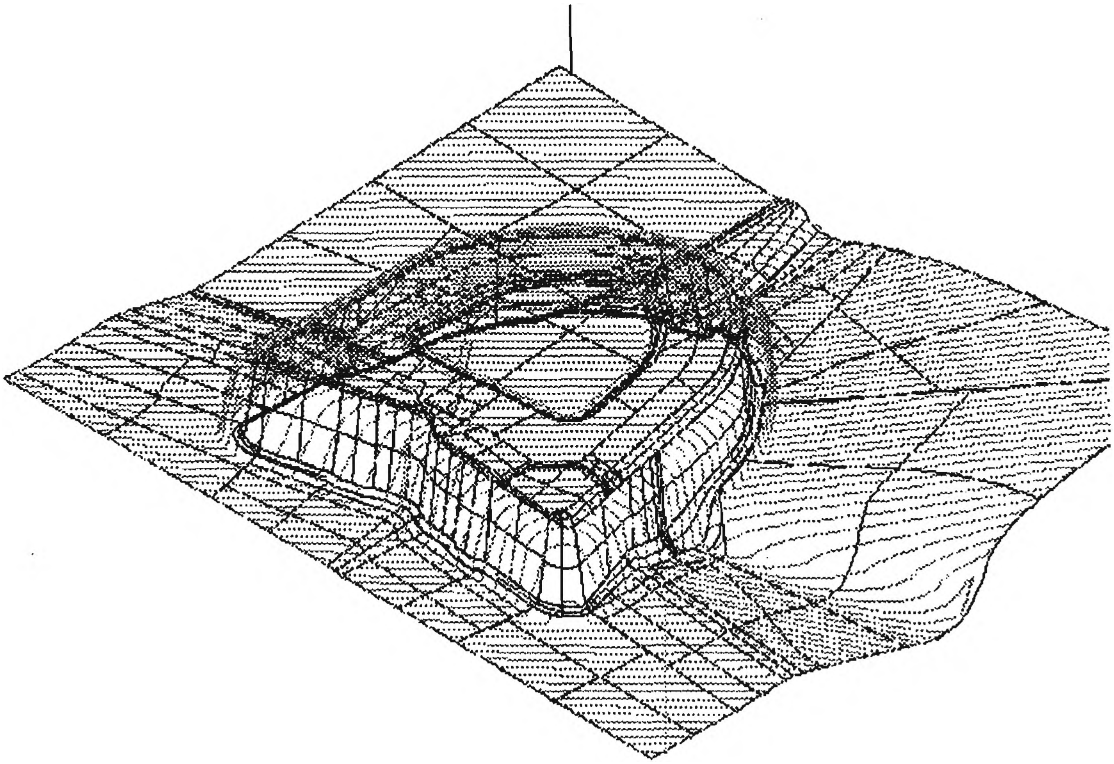

Figure 39 A finishing tool path generated for the part (top view is used)

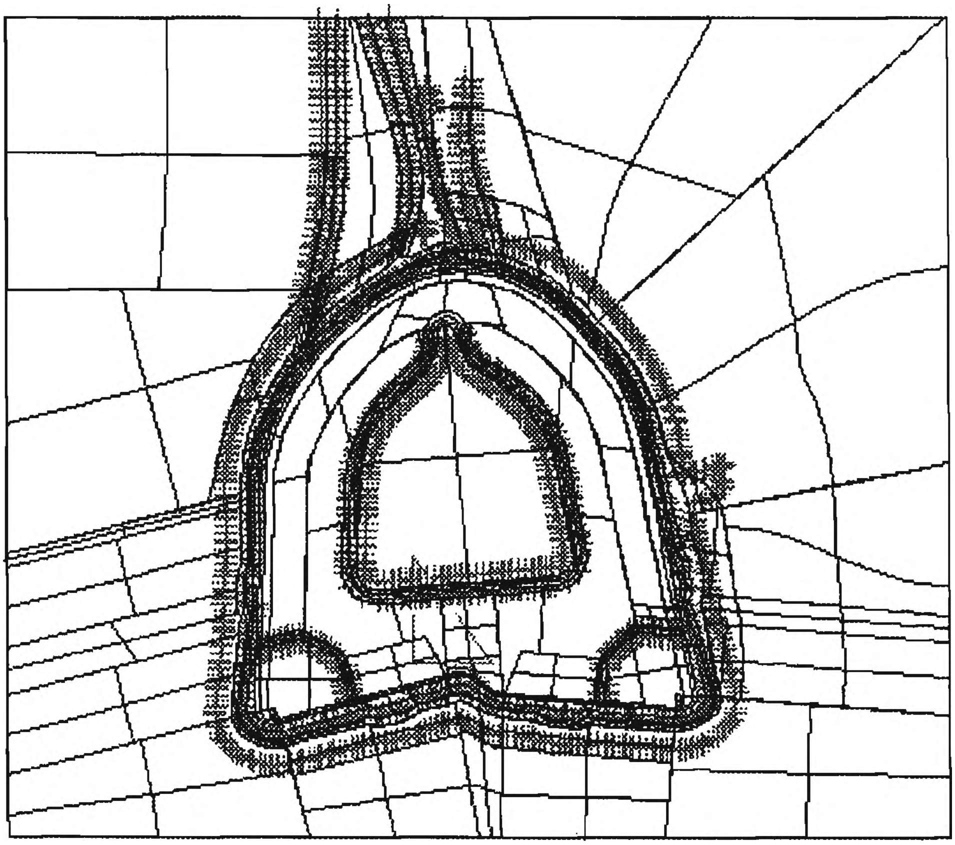

Figure 40 Double contact points detected for restmaterials (top view is used)

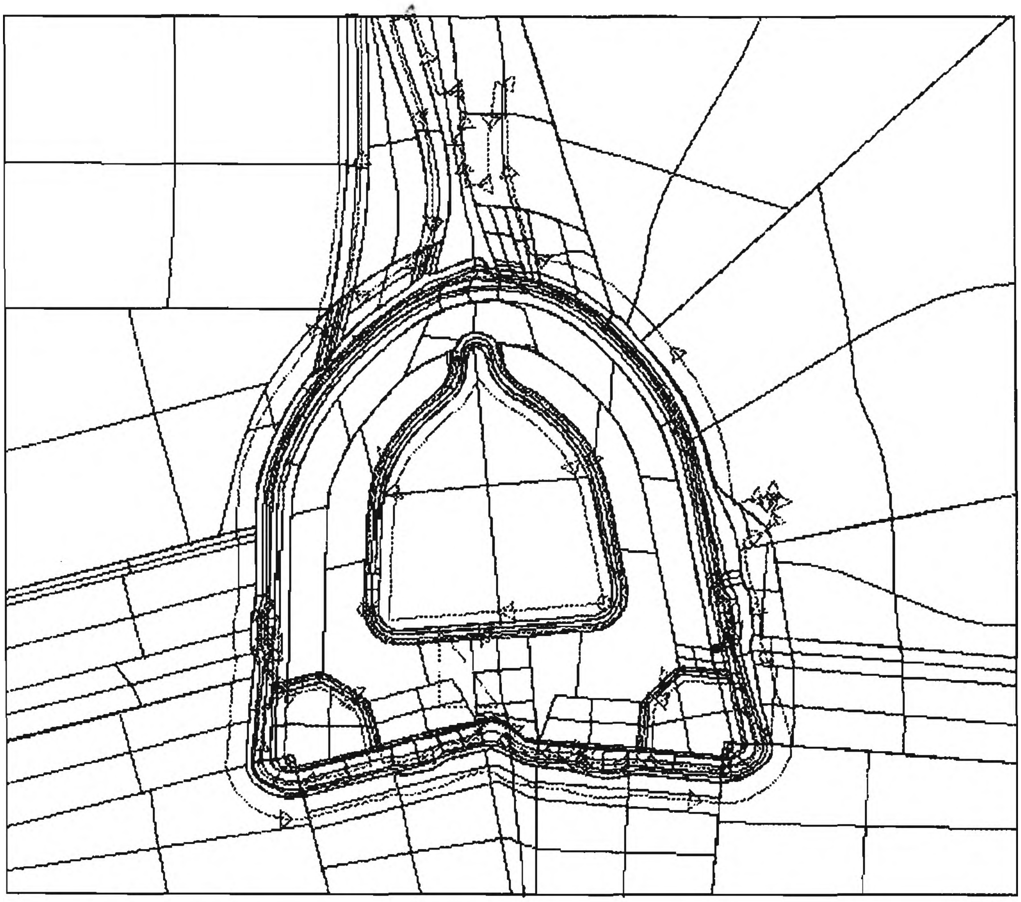

Figure 41 Boundaries for restmaterial (top view is used)

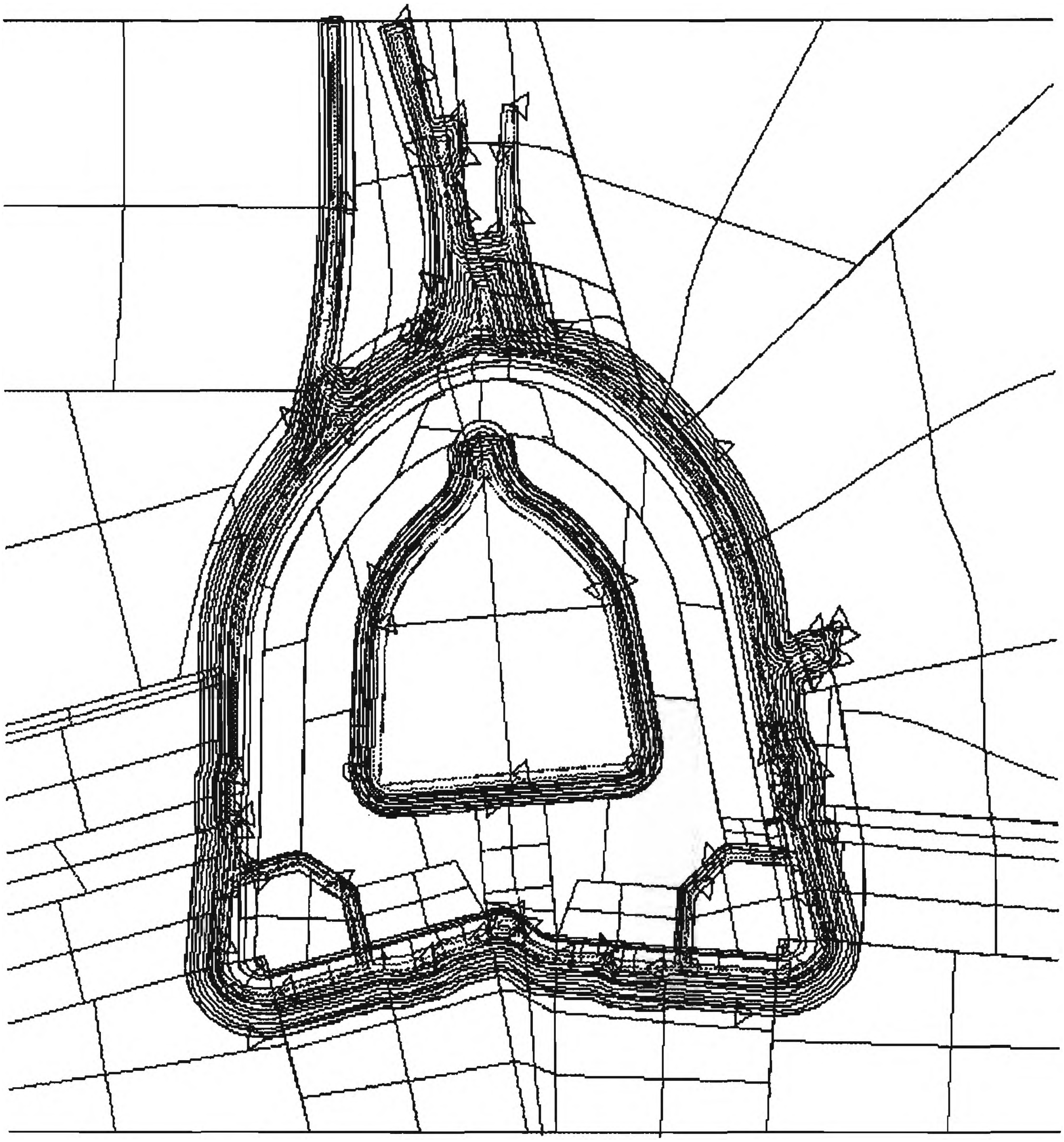

Figure 42 Compensating cleanup tool path (follow cut pattern)

CHAPTER 6

STEEP RESTMATERIAL REMOVAL

Currently, two major approaches have been used in generating NC tool paths: the parametric method and the intersection method. The parametric method is based on the parametric representation of part surfaces, which generates isocurves that are uniformly distributed across the parametric domain. The intersection approach uses evenly placed planes to intersect with part surfaces. Both of these approaches have relative advantages and disadvantages. The parametric method is good for surfaces with parametric representations, but machining of multiple surfaces may require aligning and sewing the part surfaces into a uniformed parametric space. This is difficult in the parametric method and will also introduce computation error and reduce the accuracy of part surfaces. Although the intersection method can be used for all types of surface representations, the tool path is not satisfactory. An example of the problem is given in section 3.1.2 in Figures 15 and 16. In this chapter, we use the cleanup tool path as a remedy for the intersection approach to generate more evenly spaced tool path for those arbitrary part surfaces.

The automatic compensating cleanup tool path approach has been developed from the intersection method; it will first analyzes part surface geometry, subdivides surfaces into three regions: steep regions, horizontal regions, and the rest (the rest are the regions that are neither steep nor horizontal regions). Next a major portion of the surfaces is sliced by vertical planes, and the resulting curves can be connected into a tool path that removes material from the horizontal regions and, sometimes, the rest regions. Thereafter, the steep regions are sliced by horizontal planes. The resulting intersection curves can be connected into a complete tool path.

6.1 Part Geometry Classification

To be general, part surfaces are assumed to be represented as triangle meshes. Other types of surfaces can be easily converted into such triangle meshes. In this section, two algorithms will be developed for analyzing and subdividing part surfaces into three types based on their geometry. The Adaptive Multiple Dimension Buffer (AMDB) is used for analyzing and storing part geometry information, and is developed based on the G-buffer method (or dexine model). The Goboard algorithm is used for construction of the subregions.

6.1.1 AMDB

Today, the fast increase of computer memory size makes the use of discrete computation models more feasible. With the future trends in mind, discrete models are preferred in this thesis. Because the problem of concern here is to remove restmaterials left on the part surface, a 2D discretization model will be sufficient. The dexine method discussed in section 2.2.3 is suitable in solving this problem. An improved G-buffer algorithm called Adaptive Multiple Dimension Buffer (AMDB) is introduced in this section.

A G-buffer is a 2D array, like an image. Each G-buffer contains one geometric property for all pixels such as Z values, or surface normal, etc. A G-buffer is usually generated from a parallel projection. Similar to the G-buffer algorithm, a 2D plane grid buffer is used to store multidimension geometric information from part surfaces instead of one geometric property. The following study is a brief discussion of how the AMDB is created and used in computing steep areas and horizontal areas.

6.1.1.1 Creation of AMDB

A triangular mesh is first generated to approximate the part surface within a specified tolerance. After that, the tool offset triangle is computed for each triangular patch. A tool offset triangle is computed by offsetting each triangle vertex along its normal direction with the distance of a tool radius. A 2D grid is created in an arbitrary projection plane (in this dissertation, the plane perpendicular to the tool axis is preferred and each grid size is $r \times r$, where r is the tool radius). Then the offset triangles are projected onto the 2D grid. For each grid point which is covered by the projected tool offset triangle, a pointer to this triangle is attached to the grid point. In the meantime, a test of the steepness for this triangle is carried out and grids will be marked as steep, horizontal, and the rest, respectively.

If the tool offset surfaces are computed with self-intersections removed, then a gouge free tool path can be easily generated by using evenly space parallel planes in Euclidian space to intersect these offset surfaces. The resulting intersection curves are the tool passes that can be easily connected into a tool path. But the explicit computation offset surfaces of free form surfaces that are free of self-intersection is extremely difficult. Therefore, in this dissertation, although the tool path generation method is based on the idea of offsetting surfaces, the complexity of removing self-intersections is avoided by implicit application of such offset surfaces. Details of this method can be found in the following discussion of cleanup tool path generations. In this section, we focus on the classification of surface regions based on AMDB and the Go-board algorithm.

6.1.1.2 Advantages of AMDB Over G-buffer

In this section, a simplified example is presented to detail the procedures involved in AMDB creation. This example is also a typical example, which demonstrates the advantages of AMDB over G-buffer. The step of offsetting surface triangles is omitted here.

Gaps and overlaps among offsetting triangles are ignored here to maintain the simplicity while still preserve its generality.

First, a 2D grid is created in an arbitrary projection plane (in this example, a projection plane which is perpendicular to the tool axis is chosen). A meshing scheme is used to generate triangles within tolerance to approximate the part surfaces. As shown in Figure 43, these triangles are labeled from 1 to 10. The triangles are projected down to the projection plane with the tool axis as projection vectors. They are buffered into one of the closest grid vertex: Triangle 1 and 2 are buffered into Vertex A; Triangle 3,4,5,6,7, and 8 are buffered into Vertex B; and Triangles 9 and 10 are buffered into Vertex C. After all the triangles have been buffered into the 2D grid vertices, an AMDB of the part surface geometry is said to be created.

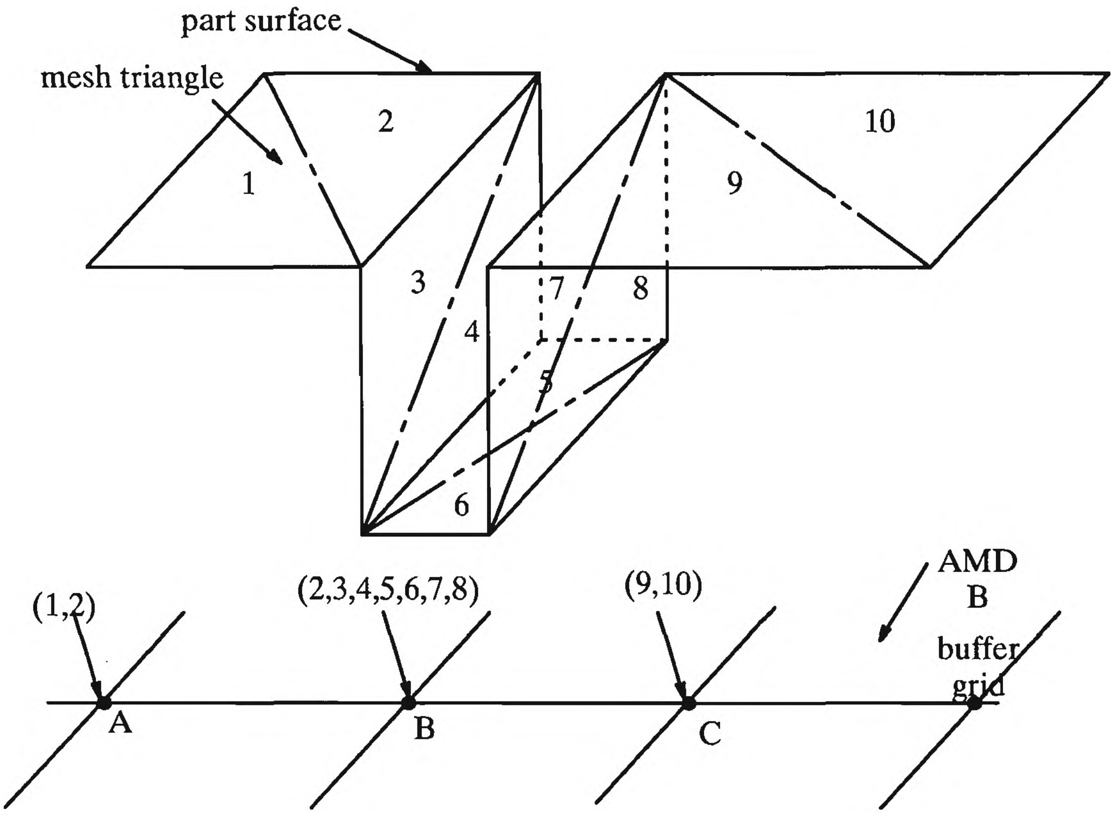

Figure 43 AMDB for part surface

The reason that a G-buffer cannot be used here is that a G-buffer only buffers Z values at sampling points; this overlooks some important manufacturing features. Considering the above projection plane, the geometric information of a very steep or vertical surface may be lost. One such cases is shown in Figure 44.

In Figure 44, Z values are sampled at the discrete sampling points in a G-buffer, and they are used as the heights of the dexels. Therefore, the indicated very steep walls will be missed, and the buffered model will imply a flat part surface. One can argue that this can be considered as an acceptable error with missing detailed geometry, but when the steep wall is large, it can no longer be considered as a detail. It may be an important feature of the part. Therefore, AMDB is created and used here to resolve such a problem. For the horizontal areas, if the projection plane is parallel to the tool axis, the same problem in G-buffer as explained above will arise.

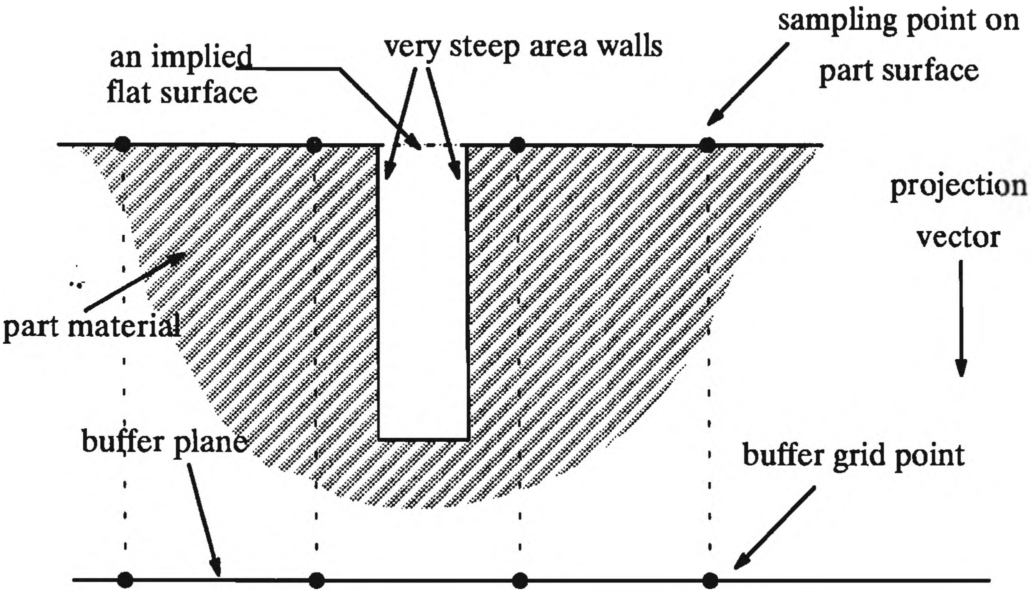

Figure 44 Side view of G-buffer missing steep areas

6.1.2 Collecting Steep Regions—Go Board Algorithm

We can consider the AMDB buffer grid as a Go board with variable lengths (a Go–board is a 19×19 grid map). Its lengths can be determined by the maximum and minimum values of part surface projections. Indexes covered by steep area projections can be considered to be occupied by black Gos while indexes covered by horizontal areas projections can be considered to be occupied by white Gos and the rest to be unoccupied. Assume that the threshold angle for steepness and flatness are α and β, respectively, and that $\cos\alpha$ is always smaller than $\cos\beta$, this ensures that steep areas will never intersect horizontal areas. The following sequence shows how to create a Go board map:

```
(All index status are defaulted to 0)
For each grid index do
    For each triangle buffered by the index do
        measure = dot product of the triangle normal  and the tool axis
        if  measure < $\cos \alpha$ then
            index_status = 1   /* steep*/
        if  measure > $\cos \beta$ then
            index_status = 2   /* horizontal */
```

After the Go board map is created as shown in Figure 45, a tracing scheme that computes the boundaries of black and white areas will be used. The indexes whose status are 0 are the unoccupied areas; the indexes whose status are 1 are the areas occupied by black Gos; and the indexes whose status are 2 are the areas occupied by white Gos (as shown in Figure 45). With the rough boundaries of the black and white areas, the triangles that are attached to these boundary indexes are used to compute the final steep and horizontal areas.

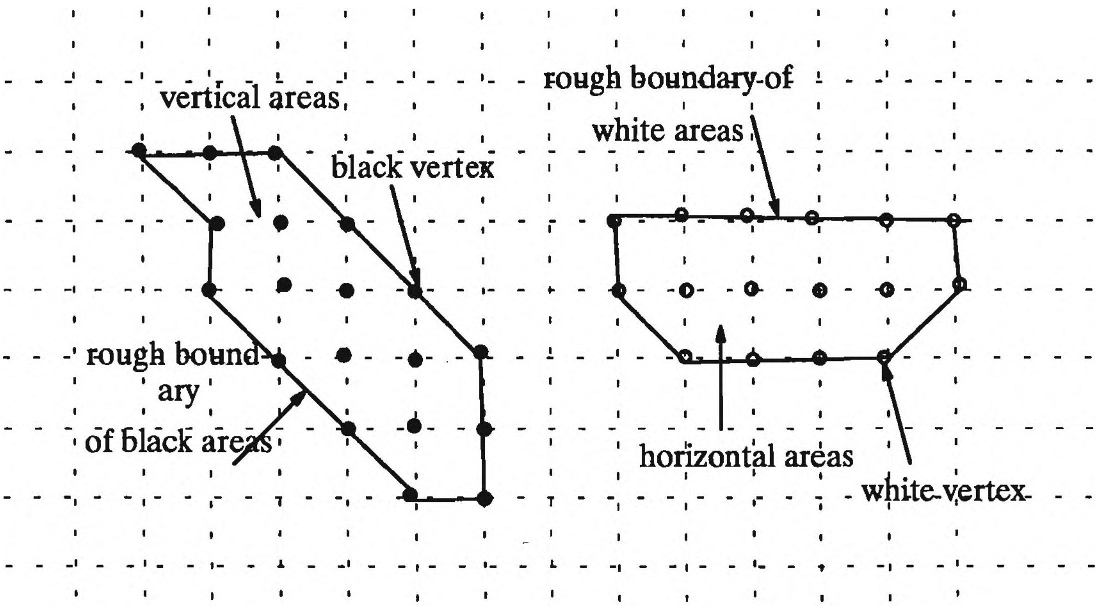

Figure 45 Go board map for computing steep and horizontal areas

Based on the above classification of grid indexes, steep or horizontal triangles can be easily found through previously buffered triangle pointers. The steep triangles can be found by collecting their pointers from the black marked indexes and the horizontal triangles can be found by collecting their pointers from white marked indexes. Therefore, different cutting methods can be applied to these three different types of surface regions.

6.2 Cleanup Tool Path Generation

As we said earlier in this chapter, parallel planes that are parallel to the tool axis can be used to intersect tool offset surfaces to generate a finish tool path that machines major portions of the part surfaces. Parallel planes that are perpendicular to the Z axis can be used to intersect with these steep regions to generate compensating cleanup tool paths that machine steep areas. Figure 46 shows slices in the horizontal areas and Figure 47 shows an example of Z slicing in steep regions.

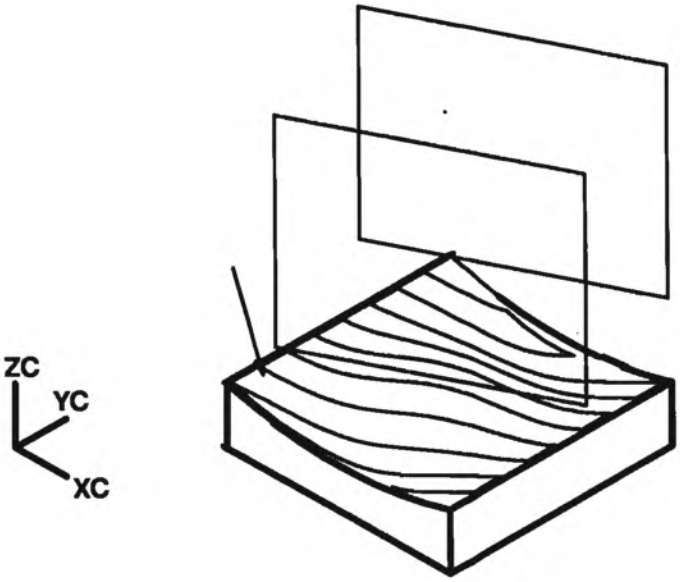

Figure 46 Zigzag contouring cut pattern for horizontal areas

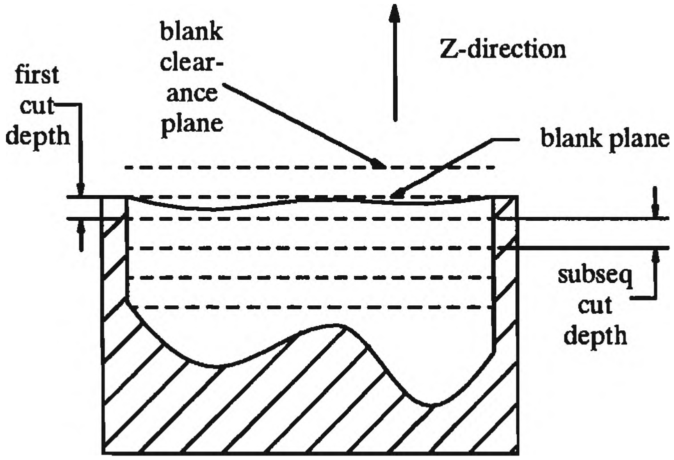

Figure 47 Steep area cleanup cut pattern

6.2.1 Machining Major Portions of Part Surfaces

In this section, algorithms for computing the tool path that machines the major portion of the part surfaces (including the horizontal and rest areas) are discussed. All the computations are carried out on an AMDB.

With the AMDB model of the part created as described in previous sections, parallel planes that are parallel to the Z axis are used to intersect with AMDB part surfaces. The constant interval between two parallel planes is determined first. Next, intersections of these planes against AMDB part surfaces are computed. Sampling points are taken at each grid interval to get the maximum Z value to finally locate tool positions. In the following, the spacing of these parallel planes is first computed based on the given manufacturing tolerance requirement, and evenly spaced planes are used to intersect with AMDB part surface triangles and to locate tool end traces for tool path. Finally, these traces are connected into a finish tool path. In this dissertation, the horizontal and the rest regions are first machined by the finish tool path, and the steep regions are machined by a later tool path—the automatic compensating tool path.

6.2.1.1 Interval Between Two Adjacent Parallel Planes

The spacing of parallel planes can be determined based on the machining tolerance requirement. The following figures illustrate the relationship between the constant interval and machining tolerance.

Figure 48 shows the motion of a flat-end tool. The distance d is the interval between two adjacent parallel planes. The maximum scallop height is h. The slope of the part surface is assumed to be the maximum steepness of the allowable nonsteep areas. The steepness angle at that point is ß. It is easy to get the relationship $h = d*\sin ß$. h is proportional to sin ß. As defined above, ß's of all the nonsteepness areas are less than the steepness angle; therefore, the interval should be:

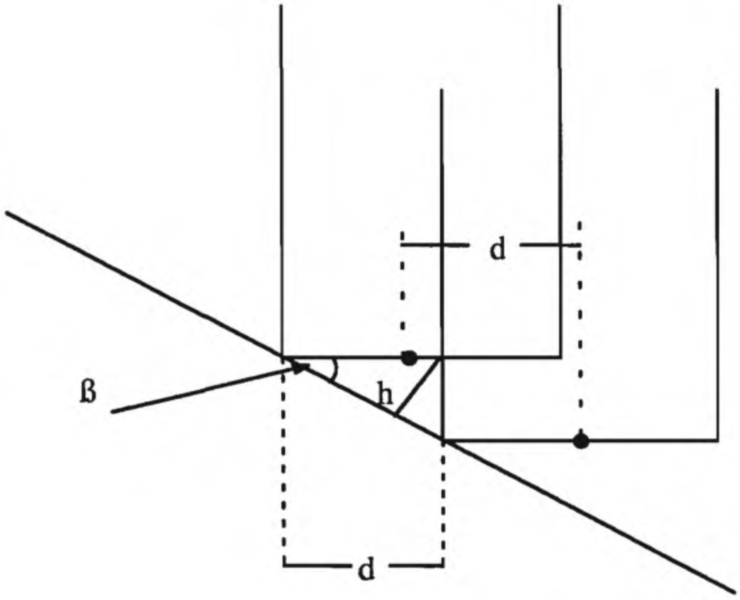

Figure 48 Flat-end Tool

Similar to Figure 48, Figure 49 shows the motion of a ball-end tool. d' is the distance of the point where maximum scallop height of that tool motion is reached. As we can see from the Figure 49, d' is always smaller than the assumed interval. Therefore, the interval calculation results are always bigger than for the flat-end case. In conclusion, the safe interval which satisfies tolerance requirement should be tolerance / sin (steepness_angle).

6.2.1.2 Intersection AMDB

After interval of the parallel plane of spacing is determined, we use these evenly spaced parallel planes to intersect with triangles that are stored in the AMDB. Because the interval of parallel planes is smaller than the grid interval of AMDB created for the part surfaces, the intersections need only be carried out in the two adjacent rows (columns) near the projection line of the plane. This is shown in Figure 50. Triangles stored in columns i and i+1 will be intersected with the Plane γ. Sampling points are then taken along the projection line; the point that has the highest Z value is used as a tool path location.

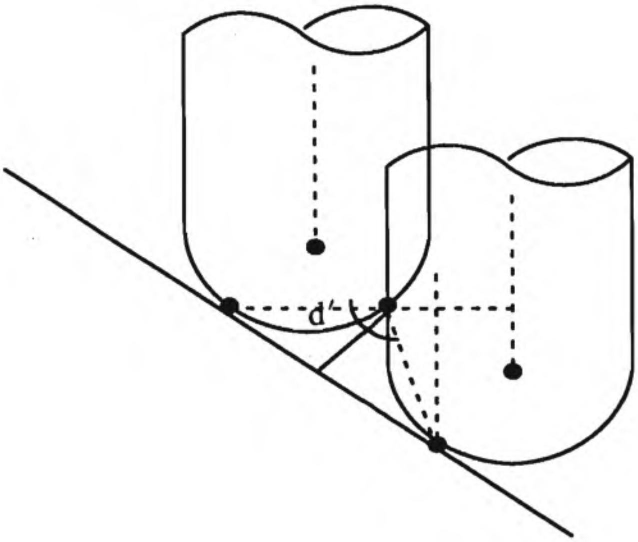

Figure 49 Ball-end tool

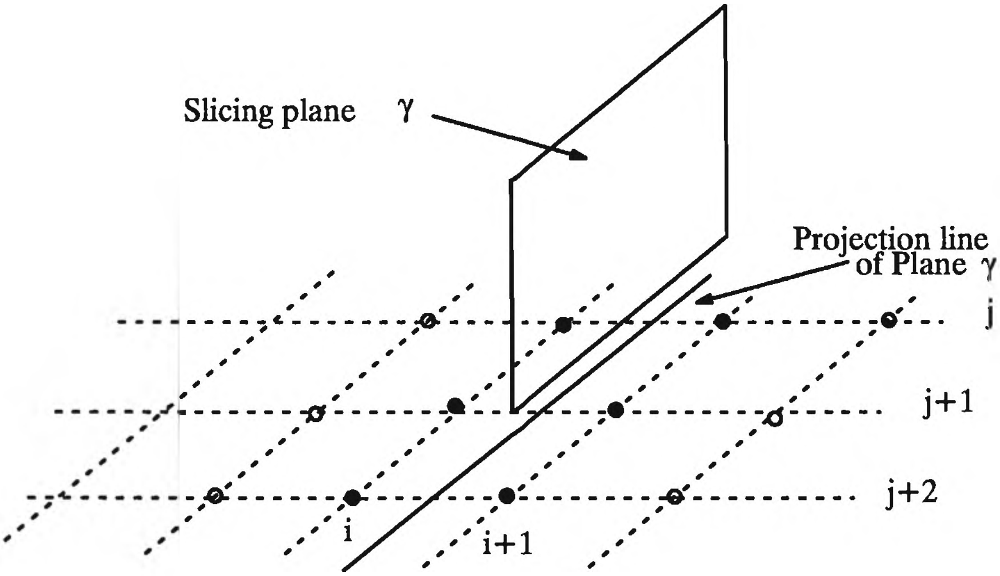

Figure 50 Neighbor rows of a slicing plane

To use the point with the highest Z value is the process of implicitly removing gouge positions from the tool path. In the meantime, AMDB reduces the number of intersections between triangles and parallel planes. As shown in Figure 50, slicing plane will only intersect triangles that are indexed into the two neighbor rows of it.

Figure 51 details the intersection of parallel planes with the triangles in a chosen AMDB index. As multiple triangles can cover the same grid index, multiple triangles should also be intersected with the slicing plane. These triangles may not be connected, and they may also have overlaps. Because these triangles are offset triangles from the original part surface triangles, self intersections, overlaps, and gaps can occur. With the intersection of the slicing plane, however, z values at each sampling point along the projection line can be compared, and the point that has the maximum Z value is used as the final tool location point. Therefore, overlaps and self intersections are implicitly removed; gouge points are all removed. Gaps of the

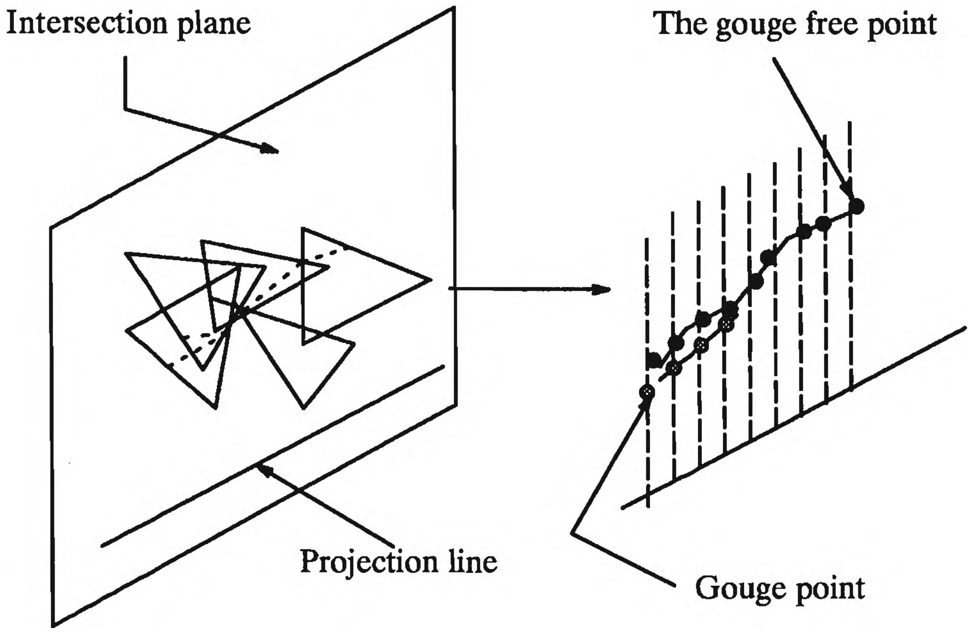

Figure 51 Intersecting triangles

triangles can all be filled.

We can use a trick here to simplify the computation. Instead of intersecting planes with triangles as in Figure 45, sampling points based on the tolerance analysis can be used to compute the intersection lines. Spacing of the sampling points is the same as the interval used in parallel planes. One thing should be emphasized here; namely, the points should be connected based on their adjacency. When a point is found belonging to a steep triangle, a break of the line should be made.

6.2.2 Cleanup Tool Path for Steep Areas

The tool path discussed in the previous sections are the finish tool paths. In this section, we discuss the compensating cleanup tool path that is used to remove restmaterials from steep areas. The generation of this cleanup tool path can be carried out in two steps: 1) slicing steep offset triangles to get tool path segments and 2) removing gouge segments.

6.2.2.1 Slicing Steep Offset Triangles

Parallel planes that are perpendicular to the tool axis are used to slice steep offset triangles. These parallel planes are evenly spaced in Euclidian space. Their intervals can be determined based on the machine tolerance. With the same interval, ball-end tools or other convex shaped tools will leave more scallop height than flat-end tools; therefore, the interval is calculated based on the flat-end tools. This is illustrated in Figure 52.

In Figure 52, d is the interval between two parallel planes. ß is the surface steepness angle at that point. h is the scallop height between two cuts. From the above figure, we can see that $h = d * \cos ß$; this means that the bigger ß (ß is smaller than 90°) is, the smaller h will be, where d is fixed. As defined, all steepness angles of steep area are bigger than the given threshold steepness angle. Therefore, the maximum interval that satisfies machine tolerance is : Tolerance / cos (threshold steepness angle).

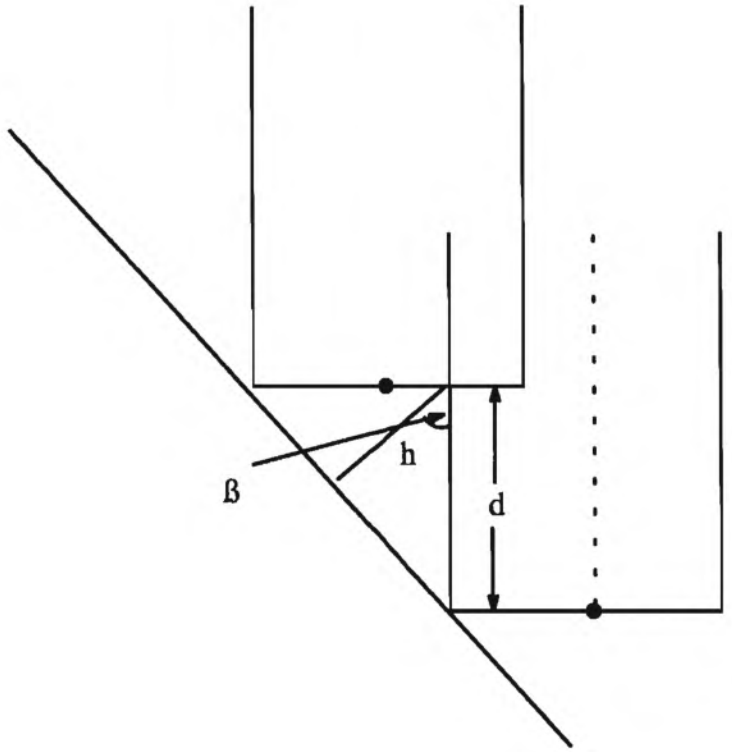

Figure 52 Determination of the interval between two slicing planes

Ideally, slicing of steep offset triangles should provide a steep cleanup tool path, but in reality, gouges may be overlooked. In real manufacturing, gouge free tool path is required. Therefore, neighbor triangles of the steep triangles should also be considered when gouge checking is performed. In the next section, an implicit gouge checking is performed on these cleanup tool passes.

6.2.2.2 Trimming Tool Path Segments

As mentioned in the AMDB creation, when steep triangles are projected onto the grid indexes, corresponding grid indexes will be flagged as steep indexes. Slicing planes with a constant interval (calculated in the previous section) are used to intersect with triangles from the steep indexes. The resulting intersection line segments are the tool passes that need to be trimmed in order to remove gouge tool positions. The trimming process is described below.

For each slicing plane, all or portions of the steep and neighbors of steep triangles that are above the slicing plane are projected on to it. Segments on the slicing plane that are covered by shadows of these projections will be removed. After this trimming is done, the rest of the line segments will be connected into the compensating cleanup tool path. With this trimming, it is guaranteed that this cleanup tool path is gouge free. An example of this trimming process is given in Figure 53.

In Figure 53, shaded triangles are the portions of offset triangles that are above the slicing plane. The tool path segments covered by these shaded areas are removed from the final tool path. We can see from the figure, after trimming, the final tool path includes line segments AB and CD, and a polyline EFGHI. These segments are further smoothed by spline fitting algorithms to generate a smooth tool path.

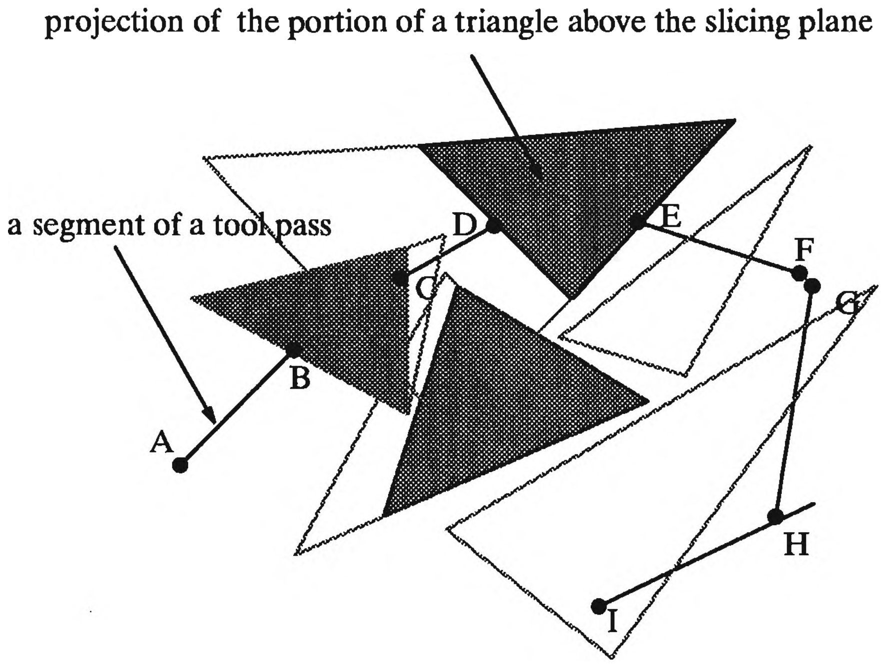

Figure 53 Trimming tool paths by projected triangles

6.3 Test Cases

As indicated previously, the implementation is done in Unigraphics, and the tool provided by this implementation has been widely used in auto and aerospace industries such as GM, GE, MDC, Boeing, etc. A practical part from a customer is used here to demonstrate the result of this research. Figure 54 shows a traditional toolpath that contours the part and leaves a finishing surface yet within the specified machining tolerance but leave nonsmooth traces on the steep walls. It is desired to generate evenly spaced traces in this part to give a good lighting effect. Therefore, a cleanup tool path is generated and gives an evenly spaced traces on the steep walls as shown in Figure 55. Because the steep areas are only one hundredth of the total part surface and there is no additional setup required, the cost therefore is only 1% of machining the whole part surfaces. With the assistance of the cleanup toolkit, the user can increase the stepovers as much as possible while still maintaining the machining tolerance on the flat areas of the part and using the cleanup tool path to remove excessive scallops from the steep walls. As we know, any small increase of stepovers will reduce machining time significantly. For instance, if the stepover is increased by 10%, the time to contour the whole will be reduced by 10%. In high speed machining, the steep walls are normally avoided at the first tool path execution to avoid bearing and damage the tool. The dark curves in Figure 56 show the boundaries of steep areas in the part. These areas will be further cleaned up by cleanup tool paths shown in Figure 55.

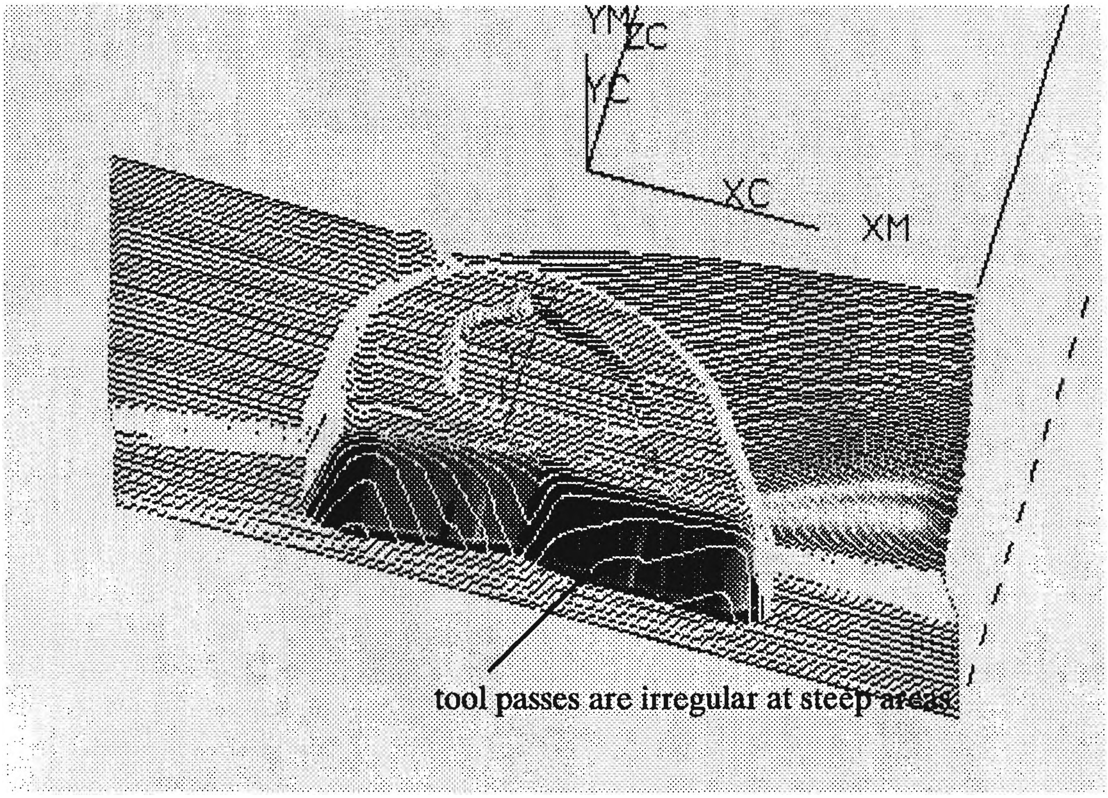

Figure 54 Traditional tool path has irregular passes at steep areas

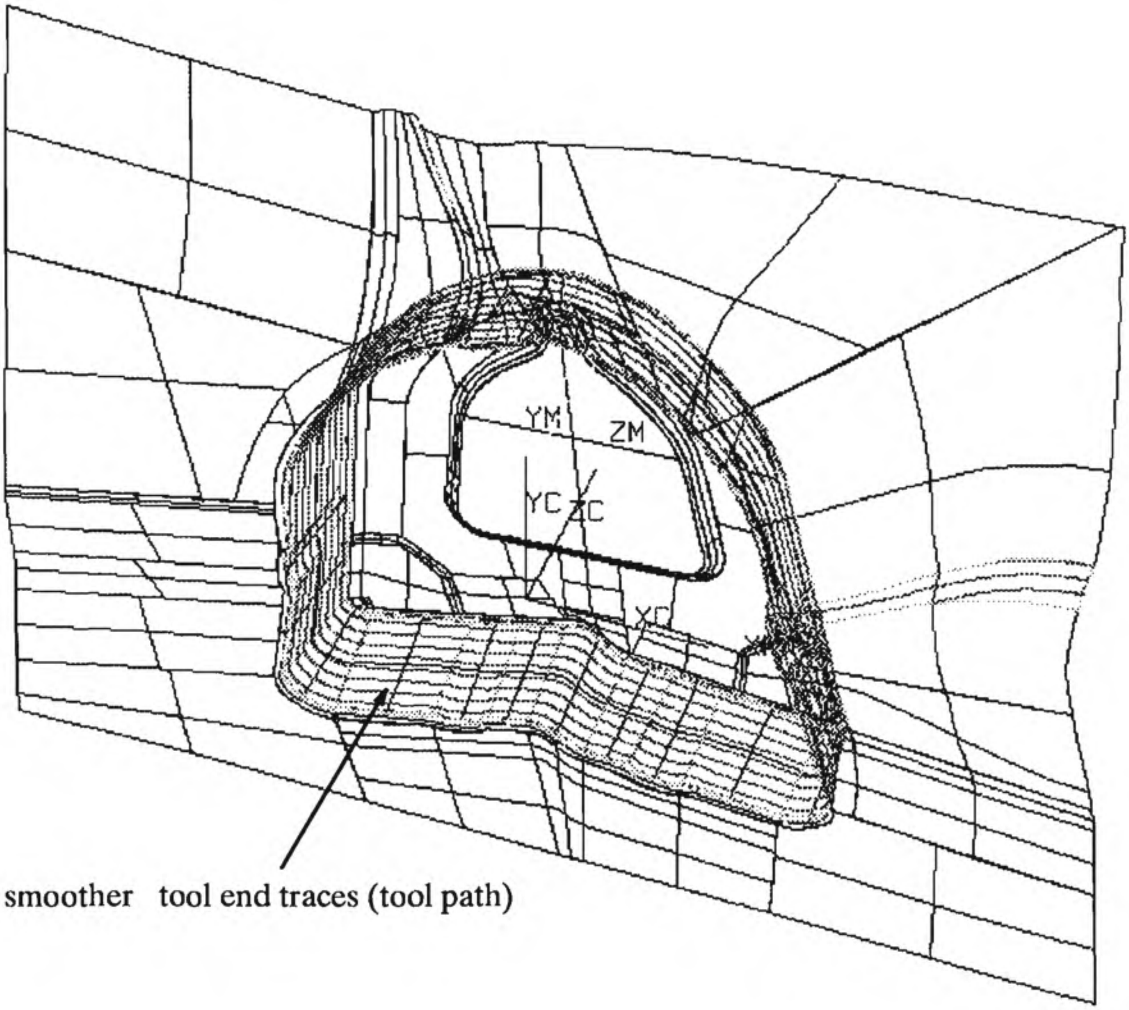

Figure 55   Cleanup tool path for steep areas

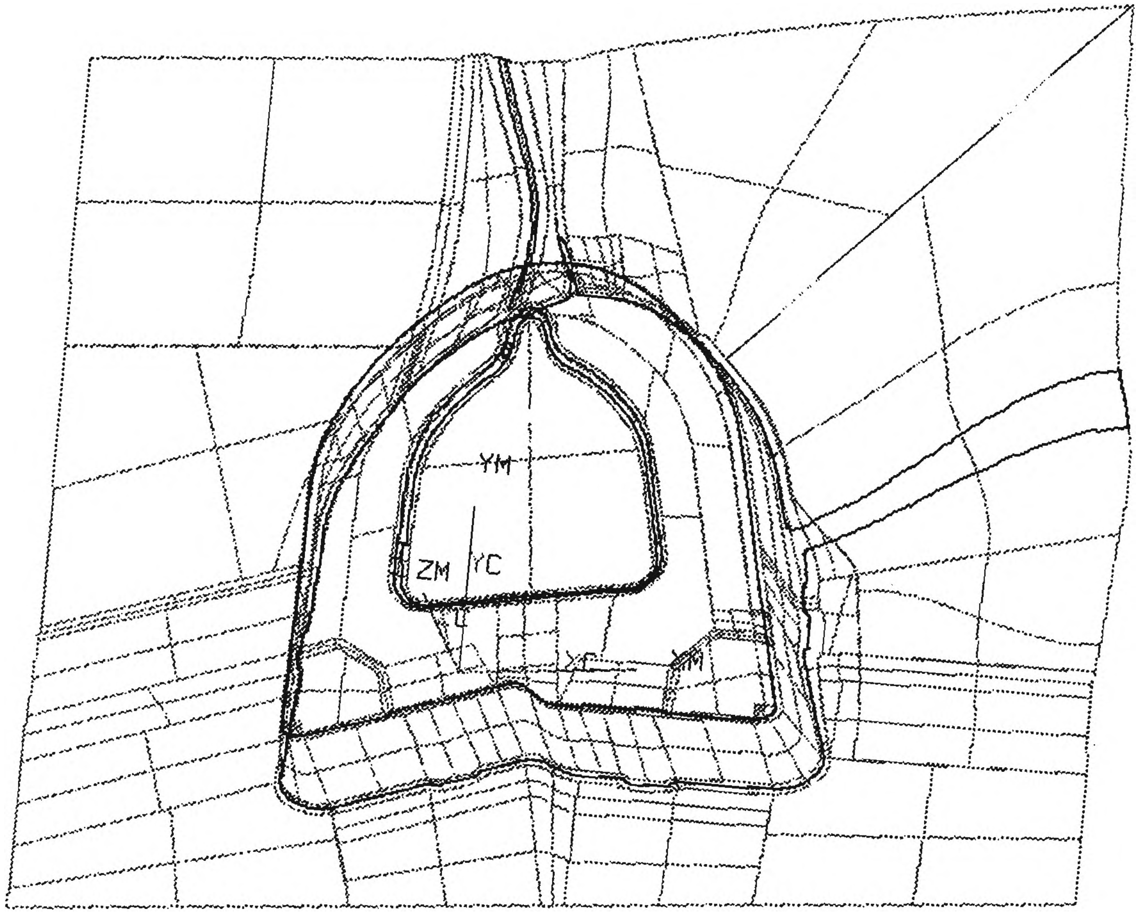

Figure 56 Boundaries for steep areas

CHAPTER 7

CONTRIBUTIONS AND FUTURE WORK

The major contribution of this research is the introduction of the compensating tool path into a CAM system. Theoretically, this compensating tool path concept provides a divide and conquer paradigm in NC tool path generation. The exploration of dependencies between two consecutive tool paths provides a new direction for automation of NC tool path generations. The application of integrating NC verification methods into generation of successive tool paths not only contributes to the integration of CAD, CAM, and NC verifications but also indicates a new direction for NC verification. On a practical level, this research will significantly reduce the cost and time of machining through reducing the burden in hand polishing and grinding. Three major aspects have been involved in this study, i.e., problem identification, establishment of compensating cleanup tool path methodology, and the development of algorithms and data structures. Contributions are summarized in these three aspects accordingly in the following section. Future work is also discussed.

7.1 Contributions

A new and nontraditional type of tool path, a compensating cleanup tool path was developed in this study. The automatic generation of the tool path makes the use of the machine tool to remove restmaterial more efficient, more robust, and much easier, therefore, greatly reducing the cost of machining a part by off–loading the burden for grinding and hand polishing processes. An itemized contribution list is as follows:

- This research studied and subdivided the restmaterial problem into smaller problems, hence provided the basis for developing solutions to restmaterial removal.
- This research also developed the compensating cleanup tool path, a new concept with regard to traditional NC programming. It uses the divide and conquer paradigm–instead of generating one finish tool path to remove all excessive materials, it uses two tool paths, one finish tool path for the majority of materials need to be removed and an automatic compensating cleanup tool path for the rest.
- With the compensating cleanup tool path, the previous finish tool path can be designed to remove material more efficiently. In addition, with the cleanup operation, less excessive materials will be left for grinding or hand finishing; therefore, both the machining time and grinding tools (usually grinding tools cost more than milling tools) are greatly saved.
- This research also developed a mathematical model, the IOPM model, that provides a foundation for tracking part models in manufacturing process. With this IOPM model, the course that a part is manufactured from a given stock is formally defined; dependencies between operations can be easily represented and studied. The development of this model will also provide the basis for integrating NC simulation and verifications into CAD/CAM system. The data communications among CAD/CAM/NC verification can be formally discussed based on IOPM model.
- This research also developed an Adaptive Multiple Dimension Buffering (AMDB) model. This model can be used to discretize 3D surfaces into a 2D buffering system while still preserving 3D geometric information which is superior to the recently developed G–buffer algorithm. This algorithm not only preserves the simplicity and efficiency of the G–buffer algorithm but

also avoids the sampling error in Z direction that the G–buffer cannot overcome.

- The Goboard model along with its tracing schema developed in this research is a useful algorithm in computing boundary traces. Because the nature of the problem it solved is to generate boundary traces out of points of 3D surfaces, the Go–board model can be used in other applications with the same abstract models.
- The automation of cleanup operation will simplify the operation generation, and make it easy to use. It also improves the robustness and consistency of the cleanup operation over the manual generation. With less user interaction, the overall processing time is greatly improved. In addition, as discussed in the introduction, the NC tool path is the foundation of feature–based machining and processes planning. The automatic generation of cleanup operations provides a direction to automate machining process for surface features. Generally, features can be divided into surface features and volume features.
- Ideas and algorithms are implemented and tested in an industrial, commercially available CAD/CAM system. This system can be used in leading manufacturing industrials, such as GM, OPEL, BOEING Aero space Company etc.

7.2 Future Work

As discussed in Chapter 3, materials left after roughing tool paths can be detected and their geometric information can be stored and constructed into CAD/CAM usable data. In this way, a cleanup tool path compensating a roughing machining can be automatically deduced similar to the compensating cleanup tool path for finishing machining. Such compensating tool path will increase the roughing efficiency by allowing the use of a bigger tool while still leaving reasonable finish surfaces by the compensating tool path. The further development of this approach will provide a background for automation of feature-based machining (volume features only are address here).

Further improvement of the IOPM model needs to be done in order to model the data communications among CAD/CAM/NC simulation and verification modules. Extending the compensating cleanup tool path to the cleanup tool path that can have independent fixtures will further automate the NC tool path generations.

The application of the AMDB data structure should be further explored in other applications where discretization of 3D surfaces is involved . A more formal discussion of error analysis for the AMDB model may be more helpful.

REFERENCES

[1] Bedworth, D.D., Henderson, M.R. and Wolfe, P.M., Computer-integrated design and manufacturing. McGraw Hill Series in Industrial Engineering and Management Science Se- New York, Feburary 1991.

[2] Brunet, P. and Navazo, I., Solid representation and operation using extended octrees. *ACM Transactions on Graphics,* 1990, **9**(2), 170–179.

[3] Cerny, V., Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization theory and applications,* 1985, **45**(1), 41–51.

[4] Chapel, I.T., The use of vectors to simulate material removal by numerically controlled milling. *Computer–Aided Design,* 1983, **15**(3), 156–158.

[5] Chen, L.L., Chou S. and Woo T., Separating and intersecting spherical polygons: computing machinability on three-, four-, and five-axis numerically controlled machines. Technical Report 91–9, Department of Industrial and Operations Engineering. University of Michigan, March 1991.

[6] Chen, L.L. and Woo, T.C., Computational geometry on the sphere with application to automated machining. *Transaction of the ASME.* June 1992, **114**, 288–295.

[7] Chen, L.L. and Woo T.C., Computational geometry on the sphere with application to automated machining. Technical Report 89–30, Department of Industrial and Operations Engineering, University of Michigan, August 1989.

[8] Chio, B.K. and Jun, C.S., Ball-end cutter interference avoidance in NC machining of sculptured surface. *Computer–Aided Design,* 1989, **21**(6), 371–378.

[9] Chou, J. J., Numerical control milling machine toolpath generation for region bounded by free from curves and surfaces. Ph.D. dissertation, University of Utah, June 1989.

[10] Dragomatz, D. and Mann, S., A classified bibliography of literature on NC milling tool path generation. *Computer–Aided Design,* 1997, **29**(3), 239–247.

[11] Drysdale, R.L. and Jerard, R.B., Discrete simulation of NC machining. In *Third Annual Symposium on Computational Geometry,* Waterloo, Canada, June 1987, pp. 126–135.

[12] Drysdale, R.L., and Jerard, R.B., et al., Discrete simulation of NC machining. *Algorithmica,* 1989, **4**(1), 33–60.

[13] Elber, G., Accessibility in 5-axis milling environment. *Computer–Aided Design,* 1994, **26**(11), 796–802.

[14] Greenwood, F., Introduction to computer-integrated manufacturing, Harcourt Brace Jovanovich, October 1989.

[15] Hansen, A. B., Tool Positioning and path generation algorithms for computer-aided manufacturing. Ph.D. dissertation, University of Southern California, Computer Science Department, December 1989.

[16] Hansen, A. B. and Arbab, F., Contemporary tool positioning methods. Technical Report No.87–58, Computer Science Department, University of Southern California, 1987.

[17] Hansen, A. B. and Arbab, F., Fixed-axis tool positioning with built-in global interference checking for NC path generation. *IEEE Journal of Robotics and Automation,* December 1988, **4**(6).

[18] Hook, T. V., Real time shaded NC milling display. *SIGGRAPH 86*, ACM, 1986, 15–20.

[19] Huang, Y. and Oliver, J. H., NC milling error assessment and tool path correction. *SIGGRAPH 94,* 1994, 287–294.

[20] Hunt W.A., and Voelcker, H.B., An exploratory study of automatic verification of programs for numerically controlled machine tools. TM–34, 1982, University of Rochester, Production automation project.

[21] Hwang, J.S., Interference-free tool-path generation in the NC machining of parametric compound surfaces. *Computer–Aided Design,* 1992, **24**(12), 667–676.

[22] Jerard, R.B., Angleton, J.M., et al., The use of surface points sets for generation, simulation, verification, and automatic correction of NC machining programs. In *Tempe, Az, Society of Manufacturing Engineers,* Jan. 8 – 12, 1990, pp. 143–148.

[23] Jerard, R.B., Hussaini S.Z., et al., Approximate methods for simulation and verification of numerically controlled machining programs. *The Visual Computer,* 1989, **5**, 329–348.

[24] Kaul, A., Minkowski sums: a simulation tool for Computer–Aided Design/CAM. *Computers in Engineering,* 1992, **1**, 447–456.

[25] Kim, C.B., Park, S. and Yang, M.Y., Verification of tool path and manual and automatic editing of NC code. *International Journal of Production Research,* 1995, **33**(3), 659–673.

[26] Kim, C.B., Yang, M.Y. and Park, S., Tool path verification and NC program editing under a multi-window environment. In *Proceedings of the 1992 Japan–USA Symposium on Flexible Automation,* Vol. 1. ASME, New York, 1992, pp. 409–414.

[27] Kirkpatric, S. and Gelatt Jr., C. D., and Vechhi, M. P., Optimization by simulated annealing. *Science,* 1983, **220**, 671–680.

[28] Kirkpatric, S. and Gelatt Jr., C. D., and Vechhi, M. P., Optimization by simulated annealing: Quantitative Study. *Journal of Statistical Physics,* 1984, **34**, 975–986.

[29] Kuragano, T., FRESDAM system for design of aesthetically pleasing free-form objects and generation of collision-free tool paths. *Computer–AidedDesign,* 1992, **24**(11), 573–581

[30] Kuragano, T., Sasaki, N. and Suzuki, A., The FRESDAM system for generating tool paths for manufacturing aesthetically pleasing injection molded products. In *Computer Applications in Production and Engineering CAPE'89,* North–Holland, Amsterdam, 1989, pp. 411–418.

[31] Lee, K., Kim, T. J. and Hong, S. E., Generation of tool path with selection of proper tools for rough cutting process. *Computer–Aided Design,* 1994, **26**(11), 822–831.

[32] Lee, Y. and Chang, T., CASCAM—An automated system for sculptured surface cavity machining. *Computers in Industry,* 1991, **16**, 321–342.

[33] Lee Y.S., Choi, B.K. and Chang, T.C., Cut distribution and cutter selection for sculpture surface cavity machining. *International Journal of Production and Research,* 1992, **30**(6), 1447–1470.

[34] Li, S.X. and Jarard, R.B., 5-axis machining of scuptured surfaces with a flat-end cutter. *Computer–Aided Design,* 1994, **26**(3), 165–178.

[35] Liu, C, Esterling, D. M., Fontdecaba, J. and Mosel, E., Dimensional verification of NC machining profiles using extended quatrees. *Computer–Aided Design,* 1996, **28**(11), 845–852.

[36] Loney, G.C. and Ozsoy, T.M., NC machining of free form surfaces. *Computer–Aided Design,* 1987, **19**(2), 85–90.

[37] Marshall, S. and Griffiths, J.G., A survey of cutter path construction techniques for milling machines. *International Journal of Production Research,* 1994, **32**(12), 2861–2877.

[38] Mullins, S.H and Jensen, C.G. and Anderson, D.C. , Scallop ellimination based on precise 5-axis tool placement, orientation, and step-over calculations. *Advances in Design Automation,* 1993, **2**, 535–543.

[39] Oliver, J.H., Efficient intersection of surface normals with milling tool swept volume for discrete three-axis NC verification. *Journal of Mechanical Design,* 1992, **114**, 283

[40] Oliver, J.H. and Goodman, E.D., Direct dimensional NC verification. *Computer–Aided Design,* 1990, **22**(1), 179–188.

[41] Persson, H., NC machining of arbitrary shaped pockets. *Computer–Aided Design,* 1978, **10**(1), 169–174.

[42] Press, W. H., Flannery, B.P., Teukolsky, S. A. and Vetterling, W. T., Numerical Recipes in C. Cambridge, 1986.

[43] Ralph, W.L. and Loftus, M., Cusp geometry analysis in surface machining. *International Journal of Production Research,* 1992, **30**(11), 2697–2711.

[44] Requicha, A.A.G and Chan, S.C., Representation of geometric features, tolerances, and attributed in solid modelers based on constructive geometry. *IEEE Journal of Robotics Automation,* 1986, RA–2(3), 156–166.

[45] Requicha, A.A.G. and Rosssignac, J.R. Solid modeling and beyond. *IEEE Computer Graphics & Applications,* 1992, Vol12, No.5, 31–44.

[46] Saglio, E. and Torinese, S. M., Super-finishing tool path generation for digitized models supported by super-finishing form feature recognition. *Computers in Engineering, ASME,* 1994, **1**, 171–181.

[47] Saito, T. and Takahashi, T., Comprehensible rendering of 3-D shapes. *Computer Graphics (Proc. SIGGRAPH '90),* 1990, **24**(4), 197–206.

[48] Saito, T. and Takahashi, T., NC machining with G-buffer method. *Computer Graphics (Proc. SIGGRAPH '90,)*1991, **25**(4), 207–216.

[49] Spyridi, A. J. and Requicha, A. A. G., Accessibility analysis for polyhedral objects. In *Engineering Systems with Intelligence: Concepts, Tools and Applications,* ed. Dordrecht, Holland: Kluwer Academic Publishers, Inc., 1991, pp. 317–324. (Published also as IRIS Rept. No. 280, Institute for Robotics and Intelligent Systems, University of Southern California, June 1991.)

[50] Stifter, S., Simulation of NC machining based on the dexel model: a critical analysis. *International Journal of Advanced Manufacturing Technology,* 1995, **10**(3), 149–157.

[51] Tang, K., Woo, T. and Gan, J., Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining. *Advances in Design Automation,* 1990, **1**, 183–192.

[52] Wang, L.P., Leu, M.C. and Blackmore, D., Generating swept solids for NC verification using the SEDE Method. *Forth Symposium on Solid Modeling and Application,* May14–16, 1997,364–375.

[53] Wang, W.P and Wang, K.K., Real-time verification of multiaxis NC programs with raster graphics. *Proceedings of the IEEE International Conference On Robotics and Automation,* San Francisco, California, April 1986, pp. 166–171.

[54] Wang, W.P and Wang, K.K., Geometric modeling for swept volume of moving solids. *IEEE Computer Graphics ad Applications,* 1986, **6**(12), 8–17.

[55] Wang, W.P, Solid geometric modeling for mold design and manufacturing. Ph.D. Dissertation, Cornell University, 1984.

[56] Woo, T. C., Visibility maps and spherical algorithms. *Computer–Aided Design,* 1994, **26**(1), 6–16.

[57] You, C. and Chu, C., An automatic path generation method of NC rough cut machining from solid models. *Computers in Industry*, 1995, **26**, 161–173.