WIRE ELECTRICAL DISCHARGE MACHINING (WEDM) AS A SUBTRACTIVE RAPID MANUFACTURING TOOL

A Dissertation in
Industrial Engineering and Operations Research
by
Zhi Yang

© 2010 Zhi Yang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of
Doctor of Philosophy

December 2010
The dissertation of Zhi Yang was reviewed and approved* by the following:

Richard A. Wysk
Professor of Industrial Engineering and
Leonhard Chair in Engineering
Dissertation Co-Advisor
Co-Chair of Committee

Sanjay Joshi
Professor of Industrial Engineering
Dissertation Co-Advisor
Co-Chair of Committee

Timothy W. Simpson
Professor of Industrial and Mechanical Engineering

Henry J. Sommer III
Professor of Mechanical Engineering

Paul M. Griffin
Peter & Angela Dal Pezzo Department Head Chair of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School
Abstract

A modern Wire Electrical Discharge Machining (WEDM) system is capable of producing more complex geometry than 2D, 2½D or ruled surface parts. The improved geometric capability and the other well-known advantages, such as capability of cutting hard materials, ignorable cutting force, make WEDM a potential Rapid Prototyping (RP) process. Unfortunately, limited capability in process planning automation for six-axis WEDM requires significant process planning time and effort.

This research presents the development of a methodology and the associated algorithms for using WEDM as a subtractive rapid prototyping process. Specifically, an automatic process planning system is built for six-axis WEDM process. A boundary representation is used to model part geometry and the manufacturability of a part on a six-axis WEDM machine is addressed in this research as well. In particular, a stereolithography (STL) file is used as the input for the algorithms. Global tangent visibility, fabrication orientation, and the setup requirements and numerical control code are the output.

Similar to the importance of visibility analysis in the automation of the milling process planning, global tangent visibility analysis is the foundation in the automation of WEDM process planning. In this research, an algorithm to calculate the global tangent visibility for polyhedral computer-aided design (CAD) models was developed. A greedy algorithm is also developed to classify the global visibility results and to determine the
setup plans for producible geometries on six-axis WEDM. Number of setup orientations is determined automatically for six-axis WEDM machines. A method to generate machine instructions automatically based on tangent visibility results and setup plans was also developed. In order to verify the methodology, a verification system is built; and several sample parts are fabricated.

The methodology and algorithms developed in this research result in a systematic automatic process planning method for applying WEDM as a subtractive RM tool. Those results are not limited to WEDM process; it can be applied to other in-line cutting systems such as laser, water jet and hot wire foam cutters.
Table of Contents

List of Figures ................................................................................................................... ix

List of Tables ..................................................................................................................... xv

Acknowledgements .......................................................................................................... xvi

Chapter 1. Introduction ...................................................................................................... 1

  1.1 Background .................................................................................................................. 1

  1.2 Motivation .................................................................................................................... 2

    1.2.1 Material Capability ............................................................................................... 2

    1.2.2 Visibility for In-Line Cut Processes .................................................................... 3

    1.2.3 Manufacturing Processes for Rapid Manufacturing ........................................... 5

  1.3 Research Objectives ................................................................................................... 6

  1.4 Outline of the Dissertation ......................................................................................... 7

Chapter 2. Literature Review ............................................................................................. 9

  2.1 Rapid Manufacturing ................................................................................................ 9

    2.1.1 Additive Rapid Manufacturing ........................................................................... 9

    2.1.2 Importance of Conventional Manufacturing Processes for RM ........... 12

  2.2 Wire Electrical Discharge Machining ........................................................................ 15

  2.3 Visibility for Manufacturing Processes ................................................................. 20
2.3.1  Definition of Visibility .................................................................................. 20
2.3.2  Applications of the Visibility Problem ......................................................... 23
2.3.3  Lack of Research in Global Visibility Analysis for In-line Cut Systems ........ 25
2.4   Part-building Orientation Problems for Additive RP Processes ...................... 26
2.5   Summary ........................................................................................................... 27

Chapter 3.  Global Tangent Visibility Analysis for Polyhedral CAD Models .... 29
3.1   Importance of Visibility Analysis ...................................................................... 29
3.2   Differences between Common Visibility and Tangent Visibility ................. 31
3.3   Definitions of Tangent Visibility ....................................................................... 33
3.4   Calculating Global Tangent Visibility ............................................................... 42
  3.4.1  Polygon Plane Intersection ........................................................................... 42
  3.4.2  Computing the Tangent Visible Percentage .................................................. 47
  3.4.3  Complexity Analysis ..................................................................................... 51
3.5   Example of Tangent Visibility .......................................................................... 52
3.6   Conclusion ......................................................................................................... 57

Chapter 4.  Polyhedral Model for WEDM-RP .......................................................... 59
4.1   Introduction ....................................................................................................... 59
4.2   Attributed Adjacency Graph .............................................................................. 60
4.3 Algorithm to Simplify the STL Polyhedral Model for WEDM-RP ........ 63

4.4 Accuracy Problems in the STL file ......................................................... 68

4.4.1 Common Errors in the STL File .......................................................... 68

4.4.2 STL Errors and Solutions in WEDM-RP ............................................. 71

4.5 Summary .................................................................................................. 80

Chapter 5. Manufacturing Coordinates Determination and Setup Optimization
.................................................................................................................................................. 82

5.1 Interpretation of Tangent Visibility Results ................................................. 82

5.2 Manufacturing Coordinates in WEDM-RP ............................................... 88

5.3 Classification of Tangent Visibility Results ............................................... 93

5.3.1 Classification of Edges ........................................................................ 93

5.3.2 Classification of Polygons without Concave Edge ............................... 95

5.4 Determination of Intermediate Coordinates for WEDM-RP .................... 97

5.5 Summary .................................................................................................. 105

Chapter 6. Wire Path Planning Methodology .................................................. 106

6.1 Wire Path Generation and Transformation .............................................. 106

6.1.1 Wire Path Generation .................................................................... 106

6.1.2 Wire Path Transformation ................................................................. 110

6.2 Sequence of the Wire Cutting Operation ................................................. 112
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Cut off Plane</td>
<td>120</td>
</tr>
<tr>
<td>6.4</td>
<td>Summary</td>
<td>122</td>
</tr>
<tr>
<td>7.1</td>
<td>Software Implementation</td>
<td>123</td>
</tr>
<tr>
<td>7.2</td>
<td>Examined Prototypes</td>
<td>125</td>
</tr>
<tr>
<td>7.3</td>
<td>Model Verification System</td>
<td>134</td>
</tr>
<tr>
<td>7.4</td>
<td>Summary</td>
<td>136</td>
</tr>
<tr>
<td>8.1</td>
<td>Research Contributions</td>
<td>137</td>
</tr>
<tr>
<td>8.2</td>
<td>Restrictions and Limitations</td>
<td>138</td>
</tr>
<tr>
<td>8.3</td>
<td>Future Research Directions</td>
<td>142</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Efficient Wire Path</td>
<td>142</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Edge Convexity Problem in the STL file</td>
<td>143</td>
</tr>
<tr>
<td>8.3.3</td>
<td>Integration of Tangent Visibility and Common Visibility</td>
<td>143</td>
</tr>
<tr>
<td>8.4</td>
<td>Conclusions</td>
<td>144</td>
</tr>
</tbody>
</table>

References 145
List of Figures

Figure 1-1 Illustration of different contact models ................................................. 4

Figure 1-2 Six-axis Wire Electrical Discharge Machining Design .................................. 5

Figure 2-1 Qualitative assessment of methods available for direct metal component production (Levy, Schindel et al. 2003) .......................................................... 13

Figure 2-2 Example models with mass/enclosed volume that is (a) low and (b) high .......................................................................................................................... 14

Figure 2-3 A machined Trabecular Metal™ human tibia fracture replica (image from (Frank 2009)) ........................................................................................................... 15

Figure 2-4 Wire EDM process description ...................................................................... 16

Figure 2-5 Wire cut slot produced in a WEDM cut .......................................................... 17

Figure 2-6 Micro-scale WEDM parts [photograph from (Schoth, Forster et al. 2005)] ............................................................................................................................ 19

Figure 2-7 Demonstration of visibility ............................................................................ 21

Figure 2-8 Visibility map example .................................................................................. 22

Figure 3-1 Difference between common visibility and tangent visibility .................. 32

Figure 3-2 Comparison of common visibility and tangent visibility ............................. 33

Figure 3-3 A point with its tangent surface ..................................................................... 34

Figure 3-4 Convex edge and concave edge examples ..................................................... 35

Figure 3-5 Point edge examples ..................................................................................... 36

Figure 3-6 Lemma proof ............................................................................................... 39
Figure 3-7 Illustration of an intersection graph............................................. 45
Figure 3-8 Algorithm for computing the intersection graph.............................. 46
Figure 3-9 Flowchart of the computing tangent visibility for a polygon ............ 48
Figure 3-10 Algorithm for computing the tangent visibility................................ 49
Figure 3-11 Algorithm for the sub-function Edge-Intersect_Test...................... 49
Figure 3-12 Algorithm for sub-function FormRegion....................................... 50
Figure 3-13 Algorithm for the sub-function Form_Rects.................................. 50
Figure 3-14 Algorithm for the sub-function Form_Triangle............................. 51
Figure 3-15 Algorithm for the sub-function Form_Cone................................. 51
Figure 3-16 Example part I ............................................................................ 53
Figure 3-17 Example part II ........................................................................... 55
Figure 3-18 Example part III ......................................................................... 56
Figure 3-19 Example part IV ......................................................................... 56
Figure 3-20 Tangent invisible triangles on human femur model ...................... 57
Figure 4-1 Relationships of 2 planes ................................................................. 61
Figure 4-2 Cross-slot facet index illustration.................................................... 62
Figure 4-3 Complete AAG for cross slot STL file.......................................... 63
Figure 4-4 Data structures for NodeList, NList, and HeadList......................... 64
Figure 4-5 Relationships between NodeList, NList, and HeadList................... 64
Figure 4-6 Algorithm to simplify AAG............................................................. 65
Figure 4-7 Algorithm to find co-planar set..................................................... 66
Figure 4-8 Surface index after simplification................................................. 67
Figure 4-9 Simplified AAG for cross slot example .................................................. 68
Figure 4-10 Gaps due to missing facets (Leong, Chua et al. 1996) ...................... 69
Figure 4-11 Example of degenerate facet .............................................................. 69
Figure 4-12 Overlapping facets example .............................................................. 70
Figure 4-13 Non-manifold edge example (Leong, Chua et al. 1996) ................. 70
Figure 4-14 A frustum of a cone and its dimensional information ...................... 73
Figure 4-15 Method to calculate adjacent facets angle ....................................... 75
Figure 4-16 Procedure to determine the facet angle ........................................... 76
Figure 4-17 A pagoda model .................................................................................. 76
Figure 4-18 Cumulative error in co-plane determination .................................... 78
Figure 4-19 Cumulative error detail example ..................................................... 79
Figure 4-20 Detail of Co-planeCheck .................................................................... 80
Figure 5-1 Demonstration of intersections between polygon and “large enough”
triangle ............................................................................................................... 84
Figure 5-2 Intersections between a polygon and a “big enough” rectangle ........ 85
Figure 5-3 Example of polygon without concave edges ..................................... 86
Figure 5-4 A triangle cover example ..................................................................... 87
Figure 5-5 A rectangle cover example .................................................................. 88
Figure 5-6 A coordinate system for WEDM-RP .................................................... 91
Figure 5-7 Definitions of \textbf{RN} and \textbf{RO} .............................................................. 91
Figure 5-8 Demonstration of intermediate coordinate and manufacturing
coordinate ............................................................................................................. 92
Figure 6-8 Flowchart of GOTOZeor function ................................................................. 117
Figure 6-9 Cutting path example 1 .................................................................................. 118
Figure 6-10 Cutting path example 2 ................................................................................. 119
Figure 6-11 Cutting path example 3 ................................................................................ 120
Figure 6-12 An example of cut off plane ........................................................................... 121
Figure 6-13 Cut off plane for the model pagoda .............................................................. 121
Figure 7-1 Software structure of the WEDM-RP system ............................................... 125
Figure 7-2 Test using the CAD models with various islands ............................................. 126
Figure 7-3 Slot part with its design coordinate system ...................................................... 127
Figure 7-4 Setup orientation and rotational orientations for the slot part ................. 128
Figure 7-5 Wire trajectory under R\textit{N1RO}(1) coordinates .............................................. 128
Figure 7-6 Inner feature example with its design coordinate system ......................... 129
Figure 7-7 Setup orientation and rotational orientation for inner feature example ................................................................. 130
Figure 7-8 Model pagoda example with its design coordinate system ................. 131
Figure 7-9 Setup orientation and rotational orientation for model pagoda example ......................................................................................... 131
Figure 7-10 A model hourglass ....................................................................................... 132
Figure 7-11 A complex model pagoda with unreachable triangles .............................. 133
Figure 7-12 Four-axis foam cutter with a rotary device ................................................. 134
Figure 7-13 Finished model pagoda ................................................................................. 135
Figure 8-1 A prismatic part with a blind hole ................................................................... 139
Figure 8-2 A frustum of a cone ................................................................. 140

Figure 8-3 Round-up errors cause an edge coverage problem ...................... 141

Figure 8-4 The STL file for a model pagoda ............................................. 142
List of Tables

Table 3-1 Computation results of three CAD models ........................................... 57

Table 4-1 Cross slot example result ........................................................................ 67

Table 4-2 Facet relationship errors summary ......................................................... 74

Table 4-3 Auto-correction result for the pagoda model .......................................... 77

Table 7-1 Computation results of five CAD models ............................................. 134
Acknowledgements

I am deeply indebted to Dr. Richard Wysk and Dr. Sanjay Joshi for their advice, direction, and support. I certainly could not have completed this dissertation with their constant support and encouragement. They are the best co-advisors that you can ever image to have. In every way, they have helped me developed the knowledge and skills, supported and encouraged me to finish my exciting and challenging journey at Penn State. I am also thankful to Dr. Timothy Simpson and Dr. H J Sommer III for their generosity in sharing their knowledge, experience, and time while serving on my committee.

I wish to express my sincere thanks to all my wonderful friends at Penn State for their support. Special thanks go to my best friends at Penn State: Chumpol Yuangyai, Rachel Abrahams, Rohan Shirwaiker and Ming-Chuan Chiu. I also want to thank Mr. Daniel Supko and Mr. Randy Wells, technicians in the PSU FAME lab, for all their help, guidance and suggestions.

I have to thank my parents, who have always being there to give me everything I needed to succeed in life. I am grateful to have a wonderful brother and a great sister-in-law; they take good care of our parents and make me feel less worried about our parents thousands of miles away. Finally, and most importantly, I want to thank my Fiancée, Ying Chen, for her love and support over all these years. She makes my Ph.D study more fun and meaningful at Penn State. She becomes my most important source of love, joy, support and motivation.
Chapter 1.

Introduction

1.1 Background

Traditional rapid prototyping (RP) is also identified as solid free form fabrication or layered manufacturing. This technology is broadly used for producing the physical model of a new product design directly from computer aided design (CAD) without the use of any special tooling. This rapid manufacturing procedure reduces the lead time to produce a prototype of a product while improving the visualization ability due to its physical existence (Noorani 2006). The advantages of RP have made a substantial contribution to the soaring global markets for quick-to-market, highly engineered products. The market for RP, consisting of all products and services globally, grew 16% to an estimated $1.141 billion in 2007, according to the Wohler’s Report (Wohlers 2008). Potential application RP areas include: quoting, ergonomic studies, rapid tooling, visual aid for engineering, and function models (Noorani 2006).

Even though RP processes have developed dramatically during the last two decade, traditional RP processes are still limited by the accuracy, material variety and development speed, especially for producing end-use parts. Technologies like Rapid Manufacturing (RM) and Rapid Tooling (RT) attempt to overcome many of the current limitations in the production of final end-use parts using RP technologies (Hopkinson, Hague et al. 2006). Current research for RM/RT/RP mainly focuses on additive processes,
which leaves a research gap in subtractive processes as an RM tool. Well developed traditional manufacturing processes can be used as subtractive RM processes to address the problems with accuracy and materials requirement. On the other hand, this approach has its own problems that are associated with tool path generation, tooling, and fixturing (Hassold 1995). The engineering time spent to overcome these limitations is significant and will diminish the advantages of subtractive rapid prototyping processes for small batch products. The research gap in subtractive RM is getting recognized during the past several years. However, most investigations on subtractive RM processes focus on the milling process (Hassold 1995; Frank, Wysk et al. 2004; McBrearty, Wysk et al. 2006; Zhu and Lee 2007). Despite the capability of handling variety of materials, milling processes still have noticeable difficulty to process some popular medical RP application materials, such as titanium, stainless steel and cobalt alloy (Bártolo and Bidanda 2007). Thus, other subtractive processes, which can fabricate hard-to-machine materials, could be explored as an alternative subtractive RM process. Investigations are also needed to overcome limitations associated with tool path generation, fixture design and tooling for those newly-considered subtractive processes in order to achieve maximum utilization of the subtractive RM process.

1.2 Motivation

1.2.1 Material Capability

As a non-traditional linear cutting process, Electrical Discharge Machining (EDM) is used to machine electrically conductive materials. It cuts by using precisely controlled
sparks that occur between an electrode and a workpiece in the presence of dielectric fluid (Jameson 2001). EDM is a thermal non-contact process. The sparks generate heat between electrode and workpiece, which will vaporize the workpiece material and result in material removal. Because EDM is a thermal process, it can be used to fabricate all conductive material regardless the material hardness. Wire EDM (WEDM) uses a continuous wire as the electrode, and sparking takes place from the electrode wire-side surface to the workpiece.

Besides being appropriate for a wide range of materials, the WEDM process is capable of producing a fine, precise, corrosion-resistance and wear-resistance surface (Liao, Huang et al. 2004). Research shows that a protective layer will be formed on the surface of the metallic biomaterial after the material is electric discharge machined (Yang, Wysk et al. 2009). This protective layer generally has uniform surface properties and prevents further oxidation (Zinelis 2007). Those properties make WEDM a great process candidate for producing metallic biomaterial parts.

1.2.2 Visibility for In-Line Cut Processes

Similar to other in-line cut manufacturing systems, such as abrasive water jet and laser material processing, WEDM has a noticeable line contact characteristic with the part being produced. Unlike some manufacturing processes which involve point contact between the tool and the workpiece, the contact mode in in-line cut manufacturing processes is a linear surface. Figure 1-1 illustrates the contact models for traditional machining processes and in-line cut manufacturing systems.
Because of the significant difference in the contact model between in-line cut process and other subtractive process, the visibility problem for in-line cut process is not broadly discussed in literature. This lack of visibility discussion for in-line cut processes is caused by the general application of those processes. Because the common geometries produced by those processes are generally in 2D or 2½D, the visibility problem becomes trivial to solve. However, with the development of modern technology, more complex machines are available in the market. For example, a structure sketch of a modern six-axis wire EDM is illustrated in Figure 1-2. The combinations of axis movement make six-axis WEDM capable to produce complex geometry product. Consequently, the non-trivial visibility problem for in-line cut processes with complex 3D parts needs to be investigated to achieve maximum utilization of the WEDM as a subtractive RM process.
1.2.3 Manufacturing Processes for Rapid Manufacturing

From the literature, the common limitations for current RM processes are material variety and properties, processing speed, dimensional accuracy, surface finish, repeatability, geometry capability and cost effectiveness (Grimm and Wohlers 2001; Ruffo and Hague 2007). From the manufacturing point of view, a large variety of processes are capable of producing RM products with one or several limitations mentioned above.

There is no dominate RP technique in rapid prototyping manufacturing (Paulo Bártolo 2007). Most additive processes present few geometry limitations. The ability to produce almost endless geometric shapes and features using additive processes makes them a desirable candidate for nearly any geometry. However, some of other limitations,
such as: material limits, processing speed, dimensional accuracy, surface finish and cost effectiveness impose severe limits for the use of additive processes. Generally speaking, parts to be made using additive processes are typically constrained by material and dimensional accuracy (Hopkinson et al., 2006). On the other hand, subtractive processes are commonly limited by the geometry they can produce, but are capable of manufacturing products using a variety of materials and can fabricate products of high dimensional accuracy.

In summary, one major advantage of WEDM is zero cutting force regardless of material hardness. This advantage makes WEDM a popular process for medical applications such as medical devices and medical substitution parts. Furthermore, the good material handling capability and good manufacturing accuracy make WEDM suitable for RM applications. However, WEDM is also limited by the part geometry and high engineering cost for complex geometry. In this research, we developed a systematic method to use WEDM process as a RM tool by significantly reducing the engineering time and built a push-button manner system for WEDM to producing complex geometry products.

1.3 Research Objectives

The purpose in this research is to investigate the use of WEDM as a subtractive RM process by reducing the amount of human intervention involved in WEDM process planning. This research focuses on developing a method called WEDM–rapid prototyping (WEDM-RP) in order to improve the efficiency of building complex parts.
The RP concept in this research is different from traditional RP solutions. Traditional RP techniques are additive processes based on layer manufacturing. In WEDM-RP, the RP concept emphasizes the rapid manufacturing of complex parts through material removal using WEDM. This RP concept is also focused on eliminating much of the manual effort required for planning and programming.

The objectives of this research are to develop a method, named WEDM-RP to solve the following problems:

1. **Manufacturability analysis for in-line cut processes.** Manufacturability analysis, as referred to here, is solely based on visibility analysis for in-line cut processes. Global tangent visibility is investigated in this research to support the manufacturability analysis for in-line cut system.

2. **Fabrication orientation.** Workpiece orientation is a critical considerations for manufacturing processes (Radzevich and Goodman 2002). The purpose of this objective is to find an orientation to minimize the total number of setups so that the efficiency of the WEDM process can be increased.

3. **Wirepath generation.** Toolpath generation is a common limitation for employing subtractive processes as a RM tool. To overcome this limitation, an automatic wirepath generation algorithm is developed in this research.

### 1.4 Outline of the Dissertation

This dissertation is organized as follows. Chapter 2 presents a literature review related to the specific problems addressed in this research. In Chapter 3, global tangent
visibility analysis for polyhedral CAD models is discussed. Chapter 4 presents the method of converting general stereolithography (STL) format into polyhedral format. Orientation determination and optimization is presented in Chapter 5, following by wirepath generation in Chapter 6. Chapter 7 presents some examples and results from this research. Finally, Chapter 8 discusses the contributions, limitations and research extensions related to this research.
Chapter 2.

Literature Review

In this chapter, an introduction to rapid manufacturing (see Section 2.1) and discussion on the importance of the subtractive process as a rapid manufacturing tool are provided. Section 2.2 introduces the WEDM process along with the current research that focuses on this subject. Section 2.3 discusses the common visibility problem and its applications. The lack of research on the visibility problem for in-line cut system is also discussed. In Section 2.4, a review the set-up orientation problems in various processes and related solving methods are provided.

2.1 Rapid Manufacturing

2.1.1 Additive Rapid Manufacturing

Rapid manufacturing, also known as direct manufacturing, is commonly defined as “the use of a CAD based automated additive manufacturing process to construct parts that are used directly as finished products or components” (Hopkinson, Hague et al. 2006). This definition reflects the fact that most of the current RM research focuses on additive manufacturing processes.

The literature has established that current RM processes are commonly subject to limitations inhering in material variety and properties, processing speed, dimensional accuracy, surface finish, repeatability, geometry capability, and cost-effectiveness
(Grimm and Wohlers 2001; Ruffo and Hague 2007). Majority of the RM research intends to focus on overcoming one or several limitations mentioned above.

All additive processes use a similar layer manufacturing concept; that is, products are manufactured layer by layer. These additive processes are distinguished from each other by the processable materials used in each. Researchers have tried to introduce new materials and new technologies to overcome the material-related problems, such as material variety and properties, surface finish, etc. One of the major problems discussed in the literature is that of product-building orientation determination (Xu, Wong et al. 1997; Hu, Lee et al. 2002; Masood and Rattanawong 2002; Masood, Rattanawong et al. 2003; Byun and Lee 2006; Byun and Lee 2006). In those studies, the researchers developed different methods to find an optimal product-building orientation to overcome the dimensional accuracy problem. Other research has focused on new material, such as polyamides, acrylonitrate butadiene styrene (ABS), polycarbonate, polyphenylsulfone (PPSP), and even metallic material, such as titanium and stainless steel (Hopkinson, Hague et al. 2006).

Economic considerations and process cost comparisons have also been broadly discussed in the literature. Hopkinson and Dickens studied the cost of RM (Hopkinson and Dickens 2003), and Hopkinson updated some of their original results in his book Rapid Manufacturing (Hopkinson, Hague et al. 2006). The researchers assumed that the machine produced only copies of the same part using a constant production time. The total cost was broken down into machine costs, labor costs, and material costs. As well
as presenting this cost classification, the researchers also discussed approximation cost analysis for injection molding, stereolithography, fused deposition modeling, and selective laser sintering. The cost analysis for selective laser sintering production was taken as the focus of Ruffo et al. (Ruffo, Tuck et al. 2006; Ruffo and Hague 2007), who built a cost-estimation model for laser-sintering production for low to medium volumes (Ruffo, Tuck et al. 2006). Their laser-sintering production cost model split the overall cost into indirect costs associated with the time of building and direct costs associated with material used during manufacturing. They compared the cost model with Hopkinson’s and provided a detailed and reasonable model for using a selective laser-sintering process with low volumes. Ruffo and Hague conducted further research on cost models for using laser sintering to produce different parts (Ruffo and Hague 2007). Their model considered parallel production on the same machine, which is a more realistic situation than single product assumption for general RP processes.

A major advantage of applying additive processes as an RM tool is the negligible engineering costs related to process planning and other process engineering activity. Despite the high machine costs and material costs, additive processes generally have a cost advantage over conventional manufacturing processes, especially in regard to manufacturing a low-volume product. On the other hand, well developed conventional processes, such as CNC milling and WEDM, have low material and manufacturing costs but very high engineering costs, especially when they are used to produce products with complex geometries. Despite the successful implementation in large-volume product
manufacturing, these high engineering costs diminish the cost-effectiveness of conventional processes for low-volume fabrication. In this research, a WEMD-RP method is developed to decrease and eliminate the high engineering costs for the WEDM process with the aim of making it a compatible RM process candidate.

2.1.2 Importance of Conventional Manufacturing Processes for RM

Although recent research efforts regarding RM have had additive manufacturing processes as their primary focus, this should not be thought of as the only manufacturing category for RM applications. Specifically, some conventional machining processes should be considered RM processes as well.

First of all, conventional machining processes are relevant to RP technologies, which can generally be divided into two kinds of processes: additive and subtractive (Pham and Gault 1998). RM technologies have been introduced to ensure long-term consistent component use for the entire production life cycle, and one of the largest efforts is focused in the direct manufacture of metal parts (Bakkelund, Karlsen et al. 1997; Levy, Schindel et al. 2003). Six different RP principles are contenders for direct metal parts fabrication. Figure 2-1 illustrates the qualitative situations of the direct metal components production relative to the usual options. Cutting methodologies and layer manufacturing, i.e., the subtractive methodologies occupy a large portion of the graph when quantities are relative low.
Secondly, due to the nature of the manufacturing processes, whether subtractive or additive, different processes have their own innate advantages and disadvantages. Figure 2-2 illustrates an example of two parts with different mass volumes. The part in Figure 2-2 (a) would require an excessive amount of material removal to make the part from a block of stock. This would be reasonably efficient for an additive RP process. On the other hand, for the part in Figure 2-2 (b), an additive RP process would spend an excessive amount of time stacking simple layers, whereas the subtractive process would finish the part very quickly from a block of stock.
Thirdly, conventional subtractive processes can be used for a wide variety of materials. Subtractive processes have been developed over many years and are able to produce a wide variety of materials in solid form. On the other hand, material availabilities are still a challenge for most additive RP techniques. Each RP technique requires a specific form of material, such as powder, solid pellets, or filaments; therefore, some of the RP applications are limited by the choice of material. For example, a variety of RP technologies can be used to fabricate scaffolds in tissue engineering (Landers and Mülhaupt 2000; Lam, Mo et al. 2002; Zein, Hutmacher et al. 2002), but most of them have material weakness, such as requiring rigid filaments or material in powder form (Yeong, Chua et al. 2004). Furthermore, the fabrication process is time consuming, and it is difficult to manufacture lattice structures using additive processes. It would be easier to subtract material from a scaffold block made by traditional processes, such as solvent
casting or gas foaming. Figure 2-3 illustrates an example of a human tibia fragment (for reconstructive surgery) machined from Trabecular Metal™ with CNC-RP.

![Figure 2-3 A machined Trabecular Metal™ human tibia fracture replica (image from (Frank 2009) )](image)

Finally, the attention paid to alternative processes for RP by the industrial community has differed from the focus of academia. According to Wohler’s report in 2006, the term “rapid manufacturing” has gained acceptance over the past few years. About 10% of all RP parts are made for rapid manufacturing purposes, and an increase is expected in coming years. Direct metal fabrication technologies are used in a wide variety of industries, from automotive and aerospace to electronics and dentistry. As the range of technologies and materials expands, the rapid manufacture of metal parts will become increasingly popular (Wohlers 2006).

### 2.2 Wire Electrical Discharge Machining

EDM is the process of fabricating electrically conductive materials by using precisely controlled sparks that occur between an electrode and a workpiece in the presence of a dielectric fluid (Jameson 2001). Spark/electrical energy is generated by a
DC-power source. During each spark, the electrons generated from the cathode travel through the ionized column of the dielectric fluid and bombard the workpiece. The electrons release their energy in the form of heat. This generated heat vaporizes the workpiece surface into a cloud, and because the workpiece has positive polarity, the vapor cloud is also positively charged. This positively charged vapor cloud is attracted to the negatively charged electrode. During the time that the vapor cloud is in transit toward the electrode, the spark electricity is turned off, which eliminates the vapor cloud’s electrical attraction to the electrode. The dielectric fluid is deionized and the vapor cloud is cooled, and, thus, an EDM chip is formed. The entire process is summarized in Figure 2-4. A normal wire-cut demonstration is illustrated in Figure 2-5.

Figure 2-4 Wire EDM process description
WEDM is a widely employed manufacturing process. Though categorized as a non-traditional process and used less frequently than other well-developed traditional processes, such as milling and drilling, WEDM is still considered a core fabrication process in industry for tool making, medical applications, and aircraft applications (Ho, Newman et al. 2004; Kemper 2004; Yan 2004; Schoth, Forster et al. 2005; Sheu 2005; Coursey, Kim et al. 2006; Kwon and Yang 2006; Alemzadeh, Hyde et al. 2007; Yan and Lai 2007).

Much of the research on WEDM has focused on WEDM process optimization and process improvement (Ho, Newman et al. 2004). In the process optimization field, researchers have used statistical methods, such as analysis of variance and the Taguchi method, to find optimal operation parameters for new materials and systems (Yan and
Fang 2004; Kuriakose and Shunmugam 2005; Puri and Bhattacharyya 2005; Lautre and Manna 2006; Mahapatra and Patnaik 2006; Manna and Bhattacharyya 2006; Ramakrishnan and Karunamoorthy 2006; Sarkar, Mitra et al. 2006). Applied voltage, wire tension, workpiece dimensions, and pulse on-off parameters are generally considered input variables. Surface roughness, material removal rate, and sparking gap dimension are commonly considered aspects in the research. Besides the statistical methods, Kuriakose and Shunmugam (2005) used multi-objective optimization to find optimal machining parameters for titanium alloy based on the significant input variables obtained via the Taguchi method. Ramakrishnan and Karunamoorthy (2006) used a similar approach for heat-treated tool steel. Artificial intelligence techniques are widely used in EDM research, such as genetic algorithms (Yan and Fang 2004), expert systems (Lautre and Manna 2006), and artificial neural networks (Sarkar, Mitra et al. 2006).

WEDM is also capable of fabricating micro scale components. Micro WEDM process design and control (Liao, Chen et al. 2005; Yan and Chien 2007), experiment and demonstration of the capability of micro WEDM (A.Schoth, Forster et al. 2005; Liao, Chen et al. 2005; Schoth, Forster et al. 2005; Sheu 2005; Di, Chu et al. 2009), and applications of micro-WEDM ( Förster, Grimm et al. 2005; Müller, Müller et al. 2005) become the main research focus on WEDM in recent years. Schoth et al. successfully manufactured micro-scale steel gear wheel and tungsten probe using 20 μm diameter cutting wire (Schoth, Forster et al. 2005). Their research proved the capability of WEDM to produce microparts in a variety of electrically conductive materials with aspect ratios
up to 100. The micro scale capabilities extend WEDM application domain significantly.

Figure 2-6 illustrates some examples of WEDM parts.

![Steel gear wheel and Tungsten Probe](image)

*Figure 2-6 Micro-scale WEDM parts [photograph from (Schoth, Forster et al. 2005)]*

Wire path generation and tool trajectory are also broadly discussed and well developed for four-axis and five-axis WEDM. Ni et al. (1991) built a rule surface model for a four-axis WEDM. Their model was based on the study of a ruled surface on the top and bottom profiles and derived detailed wire motion equations for 4-axis WEDM. Yan et al. (1995) provided an algorithm for concise and unique tool path generation. Wang and Ravani (2003) discussed the 5-axis WEDM motion model. They normalized the ruled surface curve to generate a smooth wire motion code. Zhai et al. (2004) discussed the NC code generation technique for WEDM, compensation techniques, and the wire inclining angle verification method, all of which are discussed and developed in the present study. However, the previous research concentrated on wire path generation and did not consider the manufacturability analysis for the WEDM process.
The idea of using WEDM as an RM/RT tool is not a new one. In fact, several researchers have discussed the possibility of using WEDM as an RM tool. Lee used the layered manufacturing concept to decompose solid models into slices of geometries, and manufactured each slice based on the automatically generated wire path on the WEDM machine—the slices of the parts are assembled in order to create the final product (Lee, Brink et al. 2003; Lee 2005). Chen et al. (2009) developed a computer assistant system to manufacture micro-/nano-milling tools using a feature-based approach to generate the wire path automatically. In this approach, each micro-milling tool is decomposed into different features, and the wire path is generated automatically for each feature.

The literature confirms that the real obstacle to using WEDM for RM applications is that of automating the wire path generation. Further, very little has been published on wire path automation in feature-free geometries. The present study, though, has solved the automation of wire path generation in 2D, 2½D, ruled surface, and feature based geometry. In addition, we have developed a WEDM-RP that solves wire path automation in feature-free geometry.

2.3 Visibility for Manufacturing Processes

2.3.1 Definition of Visibility

Visibility is a natural phenomenon in everyday life. To see an object necessitates that there be no obstacles between the observer and the object. A point on the object is visible if a straight line segment connecting the point and the observation point does not
intersect with any other object. Figure 2-7 illustrates an example of a visible point and a non-visible point. It is clear that the straight line segment connecting the observation point to the non-visible point intersects with the object.

![Figure 2-7 Demonstration of visibility](image)

The straight lines involved in determining visibility or lack of visibility are an important consideration in manufacturing research. Visibility analysis is important in milling processes, coordinate measuring machining, and mold casting. In manufacturing visibility research, a commonly used tool is the visibility map, which was developed from the Gaussian map. Whereas the Gauss map is a map of surface normals on the unit sphere, the visibility map, as an extension of the Gauss map, is also a spherical region, but with each point denoting a direction that is visible to the visible source (Woo 1994). If a given point is treated as the sphere’s center and the visible direction is normalized to
a unit vector starting at the given point, the end points of those visible directions will form the visibility map. Figure 2-8 shows an example of a visibility map. The point \( p \) is not visible from any direction because of the surrounding obstacles. The visibility map of point \( p \) is given in Figure 2-8 (b).

![Figure 2-8 Visibility map example](image)

This visibility map definition constitutes a milestone in visibility research. Researchers use a visibility map and extensions of it to solve real manufacturing problems, such as optimal set-up orientation and manufacturability. As an important visibility map extension, the visibility cone is also a commonly used definition in visibility research, especially in the domain of manufacturing processes. A visibility cone is defined as the following: a vertex \( v \) of a set \( S \) is said to be visible in viewing direction \( d \) if a ray originates from \( v \) in viewing direction \( d \), which does not intersect with \( S \). The aggregation of all visible directions is defined as the visibility cone of \( v \) (Yin, Ding et al. 2000). The visibility cone is usually used to determine manufacturing set-up orientations,
coordinate measuring machine (CMM) accessibility and mold parting selection. In the next section, we review the literature for visibility applications in various manufacturing processes.

2.3.2 Applications of the Visibility Problem

The visibility problem is an original fundamental problem in computer graphics. It was broadly applied in the manufacturing domain after Gauss and visibility maps had been introduced. The major visibility applications include optimal set-up orientation, tool path automation, CMM accessibility, and mold parting selection.

Optimal set-up orientation and tool path automation research principally focus on the milling process. Despite the development of CAD/CAM software, the complex surface geometry still makes it difficult and time-consuming to determine a good set-up orientation such that the total number of set-ups is minimized. This problem can be solved by applying either the Gauss or visibility map to finding the visibility of a given geometry. Woo (1994) introduced the visibility map and spherical algorithm form to general manufacturing systems. His study classified different manufacturing processes based on the degree of freedom of the process; it also discussed the visibility map in terms of its points, lines, and surfaces for different degrees of freedom. It should be noted here that visibility maps are limited inasmuch as they provide only local visible information. Set-up orientation and tool path automation, though, each require global visible information. The global visibility cone is generally used to calculate global visibility for a given geometry, and in recent years, researchers have developed several
efficient algorithms to compute the global visibility cone. Yang et al. (1999) used a convex hull computation algorithm to compute the visibility cone (Yang, Ding et al. 1999). Li and Frank (2007) computed the non-visibility set for a polyhedral model, and the visibility set is the complementary set of their research result. With the global visibility cone information available, tool path automation is now achievable (Balasubramaniam, Sarma et al. 2003; Li and Frank 2006).

The research on CMM feature accessibility primarily focuses on CMM inspection planning in order to ensure a set of points on the inspection surface reachable by the inspection probe without collisions. The feasible inspection probe orientations in CMM are similar to the feasible tool orientation in the milling process. Early research focused on accessibility using either a visibility map or accessibility cone studies. Kweon and Medeiros (1998) used visibility maps to determine the optimal part orientation for CMM inspection. Spitz et al. studied the accessibility of CMM and provided mathematical proofs and fundamentals for automatic CMM inspection planning (Spitz, Spyridi et al. 1999). More recently, the research focus shifted to feature based, CAD-based, and free-form surface CMM planning, but the basic approach is still based on accessibility or visibility analysis (Ainsworth, Ristic et al. 2000; Lin and Murugappan 2000; Zhang, Ajmal et al. 2000; Kramer, Huang et al. 2001; Elkott, Elmaraghy et al. 2002; ElKott and Veldhuis 2007).

Visibility analysis is also broadly used in mold part line determination. The mold part line is selected to minimize the number of undercuts and the amount of stretched
material (Avery 1998). The visibility map (Chen, Chou et al. 1993; Chen, Chou et al. 1995) and the visibility cone (Yin, Ding et al. 2000) are used to solve the mold part line determination problem.

2.3.3 Lack of Research in Global Visibility Analysis for In-line Cut Systems

Although the visibility problem has been broadly discussed for the milling process and other manufacturing processes, very few articles have even mentioned visibility analysis for WEDM. During the past several decades, common WEDM parts were 2D, 2½D, and those defined by rule surface. Wire path generation for those types of parts is straightforward and can easily be accomplished with commercialized software. However, with the development of the WEDM system, faster and more complex machines have become available in the market. These advanced WEDM machines are capable of producing more complex geometry than was possible previously. For these complex geometry products, wire path generation became a tedious and difficult-to-accomplish procedure. Similar to tool path automation for the milling process, wire path automation is difficult to accomplish without corresponding visibility analysis. In fact the literature has established that the visibility problem is generally solved via visibility map or global visibility cone. Nevertheless, the majority of the visibility problems can be defined as being point –to-point in nature. A point is visible, if a straight line segment as in the visibility map, or a ray orienting from the point as in the visibility cone, does not intersect with any other object. Researchers have used the orientations in the visibility map or global visibility cone as the cutting tool
orientation for accessing and fabricating products. On the other hand, the cutting tools of all the in-line cut systems, such as the cutting wire in WEDM, access the part tangentially, which requires that a whole straight line be free of intersections. It is important to note, then, that the current visibility map or global visibility cone methods are not applicable to the visibility problems inhering in the in-line cut system.

The literature offers scant discussion of line visibility. Though Woo (1994) discussed line visibility for in-line cut systems, such as WEDM, his research is based on the visibility map; therefore, the result is local visibility. To achieve wire path automation, global visibility is required. The present study presents an investigation of global tangent visibility and develops an algorithm to solve tangent visibility.

2.4 Part-building Orientation Problems for Additive RP Processes

In Section 2.3, the set-up orientation problem was solved by applying global visibility analysis to subtractive processes, such as milling. In this section, a review of literature on determining the part-building orientation for additive RP processes is provided.

Additive RP processes have a great capacity for building complex geometry parts because of the layered manufacturing concept applied in those processes. However, the building orientation will affect the quality of the prototype, build time, and part cost significantly (Byun and Lee 2006). Cheng et al. developed a multi-objective optimization model to determine the building orientation in additive RP processes (Cheng, Fuh et al. 1995). They considered the part accuracy, building time and stability
of the part as the objectives of their optimization model. Those objectives are broadly used in the later research. Multiple objectives are also generally considered technique in orientation determination research (Xu, Wong et al. 1997; Hu, Lee et al. 2002; Masood and Rattanawong 2002; Byun and Lee 2006). Variable slicing is another important technique for improving part accuracy. For example, Xu et al. (1997) built a model to calculate the optimal orientation with variable slicing for Stereolithography (SLA) systems(Xu, Wong et al. 1997). Byun and Lee (2006) studied the variable layer thickness and calculated the variable layer thickness by specifying the maximum allowable surface roughness.

The research that used multi-objective models generally provided several orientation candidates—instead of calculating the optimal orientation directly, the best orientation was obtained from among several candidates. Masood and Rattanawong (2002) identified this problem and built an algorithm to calculate the optimal orientation by minimize the total building volumetric errors. Thrimurthulu et al. (2004) developed a model to improve surface finish and decrease building time; they built a generic algorithm to calculate the optimal building orientation.

2.5 Summary

Based on the literature review, the WEDM process has the capability to be applied as an RM tool. However, the problems related to wire path generation, fixture design, and orientation determination constitute huge obstacles to achieving this aim. Despite the good accuracy and improved manufacturing speed, the problems, the long
engineering time and high planning costs are high in comparison to additive processes. These drawbacks have so far diminished the WEDM’s potential as an RM tool.

Automation in wire path generation and manufacturing orientation determination is required to decrease the engineering time and planning costs for WEDM, thus rendering it a profitable and compatible RM tool. Wire path generation requires global visibility analysis. However, due to the nature of the WEDM process, existing visibility analysis is not applicable to the WEDM process. Furthermore, the building orientation determination techniques are not applicable for WEDM-RP, since WEDM-RP does not employ the layered manufacturing concept. A new manufacturing orientation determination technique needs to be developed to determine the optimal fabrication orientations for the WEDM process. These problems and methods to solve them are discussed in the following chapter. Chapter 3 discusses global tangent visibility and presents the algorithm to determine the global tangent visibility. Chapter 4 introduces the method to simplify triangles in STL file into planar surface format. Chapter 5 introduces the orientation determination algorithms. And, in Chapter 6, an automatic wire path generation algorithm is presented.
Chapter 3.
Global Tangent Visibility Analysis for Polyhedral CAD Models

WEDM is a conventional subtractive process. This process is capable of machining any electrically conductive material regardless of the material hardness. As this is an electro-thermal process, the cutting force for WEDM is typically ignorable. The five-axis WEDM machine has been the established industrial standard for years; however, a six-axis WEDM with improved cutting speed and accuracy is now available in the market. The extra axis on the six-axis WEDM allows the machine to rotate a given part while using the cutting wire to fabricate the part. This rotational capability makes it possible for the six-axis WEDM to fabricate more complex geometries. Significant development has been achieved in terms of the WEDM process during the past decade, yet the importance of visibility analysis for the WEDM process is not well recognized, especially given the introduction of the sixth axis. In this chapter, the tangent visibility analysis for WEDM processes is presented.

3.1 Importance of Visibility Analysis

The extent to which a product is manufacturable is the fundamental concern concerning in the design of manufacturing processes. Manufacturability covers many aspects, including product geometry, material, process restriction, and finished product requirements. These aspects can be summarized by answering two fundamental questions: 1) Can the object be manufactured by a particular process? 2) How can the
object be manufactured? Visibility analysis plays a key role in answering these questions, especially for conventional manufacturing processes, such as CNC machining, injection molding, and CMM inspection path planning (Yin, Ding et al. 2000). Based on visibility analysis, the fundamental questions can be stated as 1) What is the visible portion of the object from a given point? 2) How can the object be setup the manufacturing process so that a larger portion of the object can be fabricated?

Visibility analysis is broadly applied in manufacturing process planning. For example, an optimal set-up orientation has been established for CNC milling processes; tool path generation for the milling process has been automated (Woo 1994; Kang and Suh 1997; Yang, Ding et al. 1999; Chen, Dong et al. 2003); an automatic inspection plan can be determined for CMM (Kweon and Medeiros 1998; Spitz, Spyridi et al. 1999; Ainsworth, Ristic et al. 2000; Lin and Murugappan 2000; Zhang, Ajmal et al. 2000; Kramer, Huang et al. 2001; Elkott, Elmaraghy et al. 2002; ElKott and Veldhuis 2007), and a good mold part line can be calculated for molding processes (Chen, Chou et al. 1993; Chen, Chou et al. 1995).

Automation of production planning is a key step in manufacturing processes for RM applications. From the literature, visibility analysis is fundamental to achieving automation in manufacturing process planning. Solving problems in regard to the set-up orientation and tool path generation for a process automatically will make it possible to use the process for RM applications. Despite the importance of visibility analysis, very limited research has focused on WEDM processes. Due to the differences between the
contact model of the WEDM process and that of the milling process, earlier research results on visibility analysis cannot be applied directly to WEDM. In the next section, the differences between the common visibility problem and the tangent visibility problem is discussed.

3.2 Differences between Common Visibility and Tangent Visibility

Tangent visibility is different from the common visibility problem generally described in the literature. Common visibility is utilized to determine the visible portion of a given geometry in a given direction. To determine the common visibility of a point $v$ requires a source point $s$ and examines the intersection between the line segment $\overline{sv}$ and the given geometry. The point $v$ is visible to the source point $s$ if the straight line segment $\overline{sv}$ connecting $v$ and $s$ does not intersect with any other given geometry. On the other hand, tangent visibility does not require any source point $s$. A point $v$ is tangent visible if a straight line tangent to point $v$ does not intersect with the given geometry. Figure 3-1 illustrates the difference between common visibility and tangent visibility.
(a) Common visibility, point $v_2$ is visible from source $s$, point $v_1$ is not visible from source $s$.

(b) Tangent visibility, both point $v_1$ and $v_2$ are tangent visible.

Figure 3-1 Difference between common visibility and tangent visibility

Common visibility analysis can be used for milling processes or to determine if CMM probe can reach a surface. As illustrated in Figure 3-2 (a), if point $p$ is visible to point $s$, the direction from point $s$ to point $p$ could be used as the cutting tool orientation or the probe orientation on CMM. On the other hand, a point $p$ is tangential visible, the cutting wire orientation is same with the orientation of the straight line that passed the point $p$ and is tangent to the planar surface containing $p$, as illustrated in Figure 3-2 (b). Clearly, common visibility is different than tangent visibility. However, the straight line tangent to point $p$ is the visibility boundary for common visibility, but those boundary results have limited use for milling processes or CMM activities. Consequently, the boundary of common visibility problem is not well recognized in the literature for visibility analysis. A new visibility analysis method is required to perform the tangent visibility analysis. In next section, several fundamental definitions for tangent visibility are introduced.
3.3 Definitions of Tangent Visibility

For any point \( p \) on a free-form surface, the straight lines that are tangent to the point \( p \) are all located on the same planar surface. This planar surface is referred to as the tangent surface of point \( p \). Mathematically, if a free-form 3D surface could be expressed as:

\[
F(x, y, z) = C
\]

Equation 3-1

Then tangent surface of point \((x_0, y_0, z_0)\) can be expressed as:

\[
\frac{\partial F}{\partial x} (x - x_0) + \frac{\partial F}{\partial y} (y - y_0) + \frac{\partial F}{\partial z} (z - z_0) = 0
\]

Equation 3-2

All straight lines that pass the point and are located on the tangent surface are feasible solutions to tangent visibility. Figure 3-3 illustrates an example of a tangent
surface for a point. However, from a manufacturing process perspective, we are interested in finding a direction that is tangent to the point without collisions. In addition to the requirement that all straight lines be located on the tangent surface, the collision-free solutions also require global tangent visibility information. In this research, boundary representation geometry with planar surfaces—specifically, the polyhedral CAD models, such as the stereolithography (STL) format are utilized.

Figure 3-3 A point with its tangent surface
The geometry \( G \) is the boundary representation geometry in this research. It contains edges, surfaces, and points. Because \( G \) is a polyhedral model, all surfaces in the model are planar surfaces and all edges in \( G \) are straight lines.

**Definition 3-1—**convex edge: Edge \( e \) is the intersection of plane \( P_1 \) and plane \( P_2 \), and \( \alpha \) represents the angle formed by the two planes. If \( \alpha \in (180, 360) \), the edge \( e \) is defined as a *convex edge*. Figure 3-4 (a) illustrates an example of a convex edge. The set of convex edges is noted as \( E_{\text{vex}} \).

**Definition 3-2—**concave edge: Edge \( e \) is the intersection of plane \( P_1 \) and plane \( P_2 \). \( \alpha \) represents the angle formed by the two planes. If \( \alpha \in (0, 180) \), the edge \( e \) is defined as a *concave edge*. Figure 3-4 (b) illustrates an example of concave edge. The set of concave edges is noted as \( E_{\text{cave}} \).

(a) Convex edge  
(b) Concave edge

![Convex edge and concave edge examples](image)

**Definition 3-3—**convex point: If a point \( v \) is on a convex edge \( e \), this point \( v \) is a convex point. The set of convex points is noted as \( S_{\text{vex}} \). Point \( v_1 \) in Figure 3-5 is an
example of a convex point on a convex edge $e_1$. Hereafter $V(e)$ represents the set of all points located on edge $e$.

$$S_{vex} = \{ v : v \in V(e), e \in E_{vex}, e \notin E_{cave} \}$$  \hspace{1cm} \text{Equation 3-3}

![Diagram of convex and concave points on edges](image)

**Figure 3-5 Point edge examples**

**Definition 3-4—concave point:** If a point $v$ is on a concave edge $e$, this point $v$ is a concave point. The set of concave points is noted as $S_{cave}$. Point $v_2$ in Figure 3-5 is an example of concave point on concave edge $e_2$.

$$S_{cave} = \{ v : v \in V(e), e \in E_{cave}, e \notin E_{vex} \}$$  \hspace{1cm} \text{Equation 3-4}

**Definition 3-5—convex concave point:** If a point $v$ is both on a convex edge $e_{vex}$ and a concave edge $e_{cave}$, this point $v$ is a convex–concave point. In other words, this
point is the intersection point of a convex edge and a concave edge. The set of convex–concave points is noted as $S_{vc}$. Point $v_3$ in Figure 3-5 is an example of a convex–concave point. It is the intersection point of convex edge $e_3$ and concave edge $e_2$:

$$S_{vc} = \{ v: v \in V(e_{vex}), v \in V(e_{cave}), e_{cave} \in E_{cave}, e_{vex} \in E_{vex} \} \quad \text{Equation 3-5}$$

**Definition 3-6—interior point:** If a point $v$ is not on any convex or concave edges, this point $v$ is an interior point. The set of interior points is noted as $S_{in}$. Point $v_4$ in Figure 3-5 is an example of an interior point:

$$S_{in} = \{ v: v \in S, v \notin V(e), e \in E_{vex} \text{ or } e \in E_{cave} \} \quad \text{Equation 3-6}$$

**Definition 3-7—boundary point:** if a point is an interior point or a point on an edge of the graph, it is a boundary point. The set of boundary points is noted as $S_B$.

Based on the several definitions given above, the point sets have the following relationships:

1. The convex point set $S_{vex}$, the concave point set $S_{cave}$, the convex–concave point set $S_{vc}$, and the interior point set $S_{in}$ are pairwise mutually exclusive:

$$S_{vex} \cap S_{cave} = \Phi \quad S_{vex} \cap S_{vc} = \Phi \quad S_{vex} \cap S_{in} = \Phi$$
$$S_{cave} \cap S_{vc} = \Phi \quad S_{cave} \cap S_{in} = \Phi \quad S_{vc} \cap S_{in} = \Phi \quad \text{Equation 3-7}$$
2. The union of these four sets is the boundary point set $\mathbb{S}_B$:

$$\mathbb{S}_B = \mathbb{S}_{\text{vex}} \cup \mathbb{S}_{\text{cave}} \cup \mathbb{S}_{\text{vc}} \cup \mathbb{S}_{\text{in}} \quad \text{Equation 3-8}$$

**Definition 3-8—Tangent visibility cone:** For a given geometry $\mathcal{G}$ in boundary representation, the boundary point set of $\mathcal{G}$ is noted as $\mathbb{S}_B$, and the set of points inside the geometry $\mathcal{G}$ and not in set $\mathbb{S}_B$ is noted as $\mathbb{S}_{\text{inside}}$. Point $v$ is in the set $\mathbb{S}_B, v \in \mathbb{S}_B$; point $v$ is said to be tangent visible if there is a direction $d$ such that the straight line that passes $v$ and along $d$ does not intersect with any point $\mathbb{S}_{\text{inside}}$, given the geometry. The tangent visibility cone is denoted as $\text{TVC}(v, \mathbb{S}_B)$.

**Lemma 3-1.** If a point $v$ is an interior point, then the TVC of this point will degenerate into several triangles sharing the same apex and located on the same plane. The shared apex is the interior point $v$. The plane is the plane that contains the interior point $v$.

**Proof.** The tangent surface for an interior point will be the planar surface, so that the TVC of this interior point will degenerate into a 2D planar polygon; meanwhile, the cones of straight lines degenerate into triangles sharing the same apex.

**Lemma 3-2.** If a point $v$ is a concave point, and the TVC of the point $v$ is not an empty set, then the straight line is along the edge that contains $v$.

**Proof.** (Proof by Contradiction) Assume that the concave point $v$ is on the concave edge $e$. The concave edge is shared by plane $P_1$ and plane $P_2$, as illustrated in Figure 3-6. Because the TVC of the point $v$ is not an empty set, point $v$ is tangent visible. Assume to the contrary that there is a direction $d'$ other than the edge $e$, such that the
straight line along $d'$ passes the concave point $v$ and this straight line does not intersect with the inside point set $S_{\text{inside}}$.

This straight line is noted as $L$. A point $v_2$ on $L$ can be found, such that this point is located at the negative side of either plane $P_1$ or plane $P_2$ or both of them and $v_2 \notin S_B$ and $v_2 \in S_{\text{inside}}$. However, the direction of TVC cannot construct a straight line intersecting with the inside point set $S_{\text{inside}}$. This is in contradiction to the initial assumption. This completes the proof. From the proof, we can also observe that the only feasible direction $d$ of $TVC(v, S)$ is along the concave edge $e$.

![Figure 3-6 Lemma proof.](image)

From Lemma 3-1 and Lemma 3-2, we can provide the complete definition of the tangent visibility cone mathematically. The tangent visibility cone (TVC) of $v$, denoted as $TVC(v, S)$, can be defined as:
\[ TVC(v, S) = \]
\[
\begin{cases}
    (v - \lambda d) \cap (S - (S_{v-co} - S_{cave})) = \Phi & \text{if } v \in S_{in} \text{ or } v \in S_{vex} \\
    (v - \lambda d) \cap (S - S_{v-line}) = \Phi & \text{if } v \in S_{cave}, \forall \lambda \\
    (v - \lambda d) \cap (S - (S_{vc} \cup S_{v-line})) = \Phi & \text{if } v \in S_{vc}
\end{cases}
\]

\[ \in \mathbb{R}, \lambda \neq 0 \]  

where:  
\( S \) is the set of points  
\( S_{v-co} \) is the set of co-plane points with \( v \)  
\( S_{v-line} \) is the set of concave points such that the straight line is tangent to point \( v \).  
\( S_{v-line} = \{ v: v, p \in e, e \in E_{cave}, v \neq p \} \)  
\( \mathbb{R} \) is the set of real numbers  

To understand the concept of \( TVC(v, S) \), we must try to find all the possible straight lines that pass point \( v \) and satisfy the following rules:  

1. If point \( v \) is a convex point or an interior point, then the straight line cannot intersect with any concave point in \( S \) or any point that is not co-plane with it.  
2. If point \( v \) is a concave point, then the straight line can only intersect with a point in \( S \) that is on the same concave edge with \( v \).
3. If the point \( v \) is a convex–concave point, then the straight line can only intersect with another convex–concave point or any point in \( S \) that is on the same concave edge with \( v \).

Observation 3-1. If point \( v \) is a concave point or an interior point, \( v \in S_{in}, \text{or } v \in S_{cave} \), for any straight line \( L = v - \lambda d (\lambda \in \mathbb{R}) \) that passes \( v \), then the direction \( d \) is a feasible solution in \( TVC(v, S) \). The straight line \( L \) must contain at least 2 points that are either convex or convex concave points.

**Proof.** A straight line can only get in and out of the plane boundary through a convex edge. The points on the convex edges are convex or convex concave points. In conclusion, we need at least 2 points that are either convex or convex concave.

Observation 3-2. If \( v \in S_{in}, \text{d } \in TVC(v, S_{B}) \), the straight line \( L \) passes \( v \) along direction \( d \), \( L \cap S_{cave} = \emptyset \).

**Proof.** According to Lemma 3-2, the TCV of a concave point is a straight line along the concave edge contains this point, and the straight line \( L \) cannot pass any interior point; therefore, if the straight line \( L \) passes the interior point \( v \), it cannot intersect with a concave point. This completes the proof.

Based on the TCV definition, Lemma 3-1, and Lemma 3-2, the TCV for the concave points and interior points are straight lines or geometries on a planar plane. On the other hand, based on Observation 3-1, the TVCs for the concave points and interior points must cover the convex points and convex concave points. The problem of
calculating the global tangent visibility for a given 3D geometry can be simplified to a 2D problem. In the next section, the algorithm for computing the tangent visibility for a given geometry is presented.

3.4 Calculating Global Tangent Visibility

As discussed in Section 3.3, the global tangent visibility problem for the polyhedral model can be simplified into a 2D problem. In order to find the global tangent visibility information, each planar surface in the given geometry must be expanded to obtain the intersection plane with the given geometry. Because the geometry is a polyhedral model, the intersection plane is on the tangent surface of each planar surface. Based on the intersection plane results, a tangent visibility analysis is performed and the tangent visible percentage for the planar surface is calculated.

3.4.1 Polygon Plane Intersection

The first step to computing tangent visibility is to calculate the polygon plane intersection graph. The convex-concave edge property plays a key role in the tangent visibility definition. In order to compute the tangent visibility, the edge property must be correctly assigned. Before providing the rules for assigning the edge property, several important definitions are presented first.

Definition 3-9—original edges: original edges are the edges in the original geometry $G = \{F, E, P\}$, where $F$ is the set of surfaces in geometry $G$, $E$ is the set of edges
in geometry $\mathbb{G}$, and $\mathbb{P}$ is the set of points in geometry $\mathbb{G}$. These edges are shared by
connected polygons in the given geometry. The set of original edges is noted as $E_O$:

$$E_O = \{ e : e \in E, e = P_1 \cap P_2, P_1, P_2 \in \mathbb{P} \}$$  \hspace{1cm} \text{Equation 3-10}

**Definition 3-10—extended surface for the polygon:** The extended surface is used to calculate the intersection graph for the polygon. Polygon $P = \{ \text{norm}, \mathbb{P} \}$, where norm is the polygon norm, $\mathbb{P}$ is the set of points on the polygon boundary. The extended surface for the polygon is noted as $F_e$ and can be expressed as:

$$\text{norm} \cdot \mathbf{v} + d_0 = 0$$  \hspace{1cm} \text{Equation 3-11}

where $\mathbf{v} = [x, y, z]$

$$d_0 = -\text{norm} \cdot \mathbf{v}_i, \mathbf{v}_i \in \mathbb{P}$$

**Definition 3-11—intersection graph for the polygon:** the intersection graph for polygon $P$ is denoted as $\mathbb{G}_{PI}$. $\mathbb{G}_{PI} = \{ E, V(E) \}$. Where $E$ is the set of all edges of the

**Definition 3-12—Non-original edges:** Non-original edges are the edges that are not the original geometry $\mathbb{G} = \{ F, E, \mathbb{P} \}$. It is the intersection result of the extended surface $F_e$ and the other polygons. The set of non-original edges is noted as $E_N$:

$$E_N = \{ e : e \in E, e = F_e \cap P_2, P_1 \cap P_2 = \Phi, P_1, P_2 \in \mathbb{F} \}$$  \hspace{1cm} \text{Equation 3-12}

**Definition 3-13—special original edges:** special original edges are the edges in the original geometry $\mathbb{G} = \{ F, E, \mathbb{P} \}$. However, it is not shared directly by the polygons. It
is the intersection result of the extended surface $F_e$ of $P_1$ and the other polygon $P_2$. The set of non-original edges is noted as $E_{sp}$:

$$E_{sp} = \{ e : e \in E, e = F_e \cap P_2, P_1 \cap P_2 = \emptyset, P_1, P_2 \in \mathbb{F} \}$$  \hspace{1cm} \text{Equation 3-13}$$

Figure 3-7 illustrates an example of an intersection graph for polygon $P_1$. Edge $e_1$ is shared by polygon $P_1$ and $P_2$, and it is an original edge. According to the relationship of $P_1$ and $P_2$, $e_1$ is a convex edge. Edge $e_2$ is shared by Polygon $P_5$ and $P_1$, and the relationship between $P_5$ and $P_1$ is concave, so that $e_2$ is a concave edge. Edge $e_3$ is the intersection result of the extended surface $F_e$ and the polygon $P_4$. Edge $e_4$ is the intersection result of $F_e$ and polygon $P_3$. Both edge $e_3$ and edge $e_4$ are original edges; therefore, they are special original edges considering that they do not belong to polygon $P_1$. A non-original edge, edge $e_5$ is the intersection result of polygon $P_6$ and extended surface $F_e$ and it is a non-original edge.
Figure 3-7 Illustration of an intersection graph

The following rules are used to assign the property of the edges:

1. The convex concave property of the original edge is defined by the edge definition presented in Section 3.3.

2. All non-original edges are treated as concave edges.

3. Special original edges need further inspection to determine the convex concave property. Assume that the special edge is shared by polygon $P_{s1}$ and polygon $P_{s2}$, and the original polygon is $P_1$:
   
   a. If $P_1$ locates on the same plane with $P_{s1}$ or $P_{s2}$, the convex concave property of the special edge is the same as the convex concave
property of the original edge. In Figure 3-7, edge $e_3$ is a special edge, and because polygon $P_4$ and polygon $P_1$ are located on the same plane, edge $e_3$ is a convex edge.

b. If $P_1$ is not co-plane with $P_{s1}$ and $P_{s2}$, the convex concave property is determine as following:

i. Except the points on shared edge between $P_{s1}$ and $P_{s2}$, all points on $P_{s1}$ is located on different side of $P_1$ with $P_{s2}$, the edge is assigned as a concave edge. In Figure 3-7, edge $e_4$ is shared by polygon $P_3$ and polygon $P_5$. Point $v_2$ on polygon $P_5$ is located on positive side of polygon $P_1$. Point $v_1$ on polygon $P_3$ is located on negative side of polygon $P_1$, so the edge $e_4$ is a concave edge on the intersection graph for polygon $P_5$.

ii. Except the points on shared edge between $P_{s1}$ and $P_{s2}$, all points on $P_{s1}$ is located on same side of $P_1$ with $P_{s2}$, the edge is a convex edge.

The algorithm for calculating the intersection graphs is illustrated in Figure 3-8.

<table>
<thead>
<tr>
<th>Algorithm 3-1 Computing the intersection graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>For</strong> (each polygon $F_i$ in the given Geometry $G$) {</td>
</tr>
<tr>
<td>$E_i = \Phi$</td>
</tr>
<tr>
<td><strong>For</strong> (each polygon $F_j \neq F_i$ in the given Geometry $G$) {</td>
</tr>
<tr>
<td>$e = F_i \cap F_j$</td>
</tr>
<tr>
<td>Assign edge property to $e$ /* convex or concave */</td>
</tr>
<tr>
<td>$E_i = E_i \cup { e }$</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>/* $E_i$ are used to compute the tangent visibility for each polygon*/</td>
</tr>
</tbody>
</table>

Figure 3-8 Algorithm for computing the intersection graph
3.4.2 Computing the Tangent Visible Percentage

The algorithm to calculate the tangent visibility percentage is presented in this section. The result of the intersection graph and the properly defined edge properties are used as an input for this algorithm.

The input of the algorithm is summarized as follows:

1. **Intersection Polygon** $P_1$ is the polygon we are interested in.
2. **Intersection graph**: $\mathcal{G}_{P_1}$ is the intersection graph for polygon $P_1$. $\mathcal{G}_{P_1} = \{ E, V (E) \}$.
3. **Convex edge sets**: $E_{vex}$ is all convex edges in the intersection graph $E_{vex} \subseteq E$.
4. **Concave edge sets**: $E_{cave}$ is all concave edges in the intersection graph $E_{cave} \subseteq E$.

The algorithm for computing the tangent visibility can be described as follows (see Figure 3-9):

*Step 1*: Determine whether the edges in $\mathcal{G}_{P_1}$ intersect with any concave edges.

*Step 2*: If all edges in $\mathcal{G}_{in}$ are convex edges, then use convex edges to form rectangle with points in $\mathcal{G}_{in}$ in order to cover the intersection polygon $P_1$.

*Step 3*: If $E_{vex} \neq \Phi$ and $E_{cave} \neq \Phi$, then form rectangles and triangles to cover the polygon $P_1$. 
Figure 3-9 Flowchart of the computing tangent visibility for a polygon

Figure 3-10 illustrates the algorithm to calculate the tangent visibility. The sub-functions of the tangent visibility algorithms are illustrated in Figure 3-11, Figure 3-12, Figure 3-13, Figure 3-14, and Figure 3-15.

<table>
<thead>
<tr>
<th>Algorithm 3-2 Compute tangent visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>$VIS_{List} = \Phi \quad <em>/\text{tangent visibility}</em>/$</td>
</tr>
<tr>
<td>For each intersection graph $G_{Pr} = {E, V(E)}$ for polygon $P$ {</td>
</tr>
<tr>
<td>$TVC = \Phi \quad <em>/\text{the tangent visible region}</em>/$</td>
</tr>
<tr>
<td>For each edge $e \in E$ {</td>
</tr>
<tr>
<td>If ($Edge_{Intersect_Test}(e, E)$) {</td>
</tr>
<tr>
<td>$v = ClosestPoint(e, V(E))$</td>
</tr>
<tr>
<td>$TVC = TVC \cup Form_Region(e, v, P) \quad <em>/\text{polygon clipping}</em>/$</td>
</tr>
<tr>
<td>}</td>
</tr>
<tr>
<td>If ($e \in E_{cone}$) {</td>
</tr>
<tr>
<td>$TVC = TVC \cup Form_Cones(e, V(E), P) \quad <em>/\text{polygon clipping}</em>/$</td>
</tr>
<tr>
<td>$TVC = TVC \cup Form_Rects(e, V(E), P) \quad <em>/\text{polygon clipping}</em>/$</td>
</tr>
<tr>
<td>}</td>
</tr>
</tbody>
</table>
Figure 3-10 Algorithm for computing the tangent visibility

```plaintext
vis = Area(TVC) + Area(P)
VisList = VisList + [(P, vis)]
```

Figure 3-11 Algorithm for the sub-function Edge-Intersect_Test

**Sub-function 3-1: Edge_Intersect_Test (e,E)**

$L$ is the straight line that is coincident with edge $e$

**For** (each edge $e_i \in E$ and $e_i \neq e$)

- Flag = true
- $v = L \cap e_i$

If ($v \neq \emptyset$) and ($e \in E_{concave}$) {
  If ($v \notin V(E)$) {
    /*intersects with concave edge in the middle*/
    Flag = false
  } else {
    If ($v \in V(E_{vex})$) {
      /*intersects at convex concave points*/
      $E_v = \{ e: e \in E, v = e \cdot v(0) \text{ or } v = e \cdot v(1) \}$
      If ($\forall V(E_v) \text{ is located on same side of } L$) {
        /*intersects at convex concave point*/
        /*Line L runs into solid*/
        Flag = false
      } else {
        /*intersects with the concave point*/
        Flag = false
      }
    } else {
      /*intersects with the convex point*/
      Flag = false
    }
  }
}

return Flag
### Sub-function 3-2: FormRegion \((e,v,P)\)

\[ INF = 10^6 \]

/*build a “large enough” rectangle \(R = (v_1,v_2,v_3,v_4)*\)*/
/* it has an edge that is coicident to edge \(e\)*/
/* the other edge passes point \(v\) and is parallel to edge \(e\)*/

\[
\begin{align*}
v_1 &= e.v(0) - INF * (e.v(0) - e.v(1)) \\
v_2 &= e.v(0) + INF * (e.v(1) - e.v(0)) \\
v_3 &= v + INF * (e.v(1) - e.v(0)) \\
v_4 &= v - INF * (e.v(0) - e.v(1))
\end{align*}
\]

\(Region = P \cap R\) /*get the intersection of the original polygon \(P\) and the rectangle \(R\)*
return \(Region\)

---

**Figure 3-12 Algorithm for sub-function FormRegion**

### Sub-function 3-3 FormRects \((e,V(\mathbb{E})),P\)

\(Region = \emptyset\)

**For** (each \(v \in V(\mathbb{E})\)) {
/*build edge \(e_1\) passes the point \(v\) and parallel to edge \(e\)*/
\[
e_1.v(0) = v + (e.v(1) - e.v(0)) \\
e_1.v(1) = v + (e.v(0) - e.v(1))
\]
If (EdgeIntersectTest \((e_1,\mathbb{E})\)) {
\[
v' = \text{ClosestPoint}(e_1,V(\mathbb{E}))
\]
\(Region = Region \cup \text{FormRegion}(e_1,v',P)\) /*polygon clipping*/
}
/*build edge \(e_2\) start at the point \(v\) and end at point \(e.v(0)\)*/
\[
e_2.v(0) = v \\
e_2.v(1) = e.v(0)
\]
If (EdgeIntersectTest \((e_2,\mathbb{E})\)) {
\[
v' = \text{ClosestPoint}(e_2,V(\mathbb{E}))
\]
\(Region = Region \cup \text{FormRegion}(e_2,v',P)\) /*polygon clipping*/
}
/*build edge \(e_3\) start at the point \(v\) and end at point \(e.v(1)\)*/
\[
e_2.v(0) = v \\
e_2.v(1) = e.v(1)
\]
If (EdgeIntersectTest \((e_3,\mathbb{E})\)) {
\[
v' = \text{ClosestPoint}(e_3,V(\mathbb{E}))
\]
\(Region = Region \cup \text{FormRegion}(e_3,v',P)\) /*polygon clipping*/
}
}
return \(Region\)

---

**Figure 3-13 Algorithm for the sub-function FormRects**
Sub-function 3-4: Form_Triangle \((v_1, v_2, v_3, V(E), P)\)

\[\text{Region} = \Phi\]
\[INF = 10^6\]

/*build a “large enough” triangle \(T_{r_1} = \{ v_a, v_b, v_c \} */\]

\[v_b = v_1 + INF \times (v_2 - v_1)\]
\[v_c = v_1 + INF \times (v_3 - v_1)\]
\[v_a = v_1\]

/*build a “large enough” triangle \(T_{r_2} = \{ v'_a, v'_b, v'_c \} */\]

\[v'_b = v_1 + INF \times (v'_1 - v_2)\]
\[v'_c = v_1 + INF \times (v'_1 - v_3)\]
\[v'_a = v_1\]

/*Triangle \(T_{r_1}\) and \(T_{r_2}\) share a vertex, they are vertical angle triangles*/

Flag = false

For (each \(v_p \in E_{cave}\)) {
  If (\((v_p \text{ inside } T_{r_1}) \text{ or } (v_p \text{ inside } T_{r_2})\)) {
    Flag = true
    Break
  }
}

If (! Flag) {
  Region = P \cap (T_{r_1} \cup T_{r_2})
}

return Region

Figure 3-14 Algorithm for the sub-function Form_Triangle

Sub-function 3-5: Form_Cones \((e, V(E), P)\)

\[\text{Region} = \Phi\]

For (each \(v \in V(E)\)) {
  Region = Region \cup Form_Triangle(e, v(0), e, v(1), v, V(E), P)
  Region = Region \cup Form_Triangle(e, v(1), e, v(0), v, V(E), P)
  Region = Region \cup Form_Triangle(v, e, v(1), e, v(0), V(E), P)
}

Return Region

Figure 3-15 Algorithm for the sub-function Form_Cone

3.4.3 Complexity Analysis

The global tangent visibility algorithm has three major steps, as illustrated in Figure 3-9. Let the input intersection graph \(\mathcal{G}_{P_{T}} = \{E, V(E)\}\) have \(m\) concave edges, \(n\) convex edges, and \(k\) points, where \(k \leq 2(m + n)\). The first major step is the edge intersection test. In this step, an intersection test is performed for the edges in the
intersection graph \( \mathcal{G}_{pi} \) with all concave edges. The complexity of this operation is \( O(m \times (m + n)) \). The second step in the global tangent visibility algorithm is that of performing the polygon-clipping operations for polygons without concave edges. In this research, the polygon-clipping operation is finished using the third-party open source package, Computational Geometry Algorithms Library (CGAL). The general polygon-clipping operation has the complexity \( O(xy) \) with \( x \) and \( y \) being the edge numbers of the arbitrary polygons (Greiner and Hormann 1998). Consequently, the complexity of the second step is \( O(n^2) \), given that these polygons do not have concave edges. The third step in the tangent visibility algorithm is that of forming visible regions and performing polygon clippings. The complexity is \( O((m + n) \times (k + (m + n)^2)) \). Because \( k \leq 2(m + n) \), \( O((m + n) \times (k + (m + n)^2)) = O((m + n)^3) \). The second and third steps in the algorithm are not executed simultaneously; therefore, when polygons contain both convex and concave edges, the worst case size complexity is \( O((m + n)^3) \) when polygons contain both convex and concave edges. Let the total number of polygons be \( p \); there are \( p \) intersection graphs to perform tangent visibility analysis. Therefore, the overall complexity of the tangent visibility algorithm is \( O(p \times (m + n)^3) \).

### 3.5 Example of Tangent Visibility

In this section, several examples are given to show the effectiveness of the proposed global tangent visibility algorithm. In each model, the polygon surface is colored in red if the whole polygon surface is not complete tangent visible. Figure 3-16 illustrates an example with one polygon surface that is not complete tangent visible. The
polygon surface is colored in red, as illustrated in Figure 3-16 (b). In Figure 3-16 (c), the detailed tangent visibility result is illustrated. The three circled triangle regions are not tangent visible.

Figure 3-16 Example part 1

Figure 3-17 illustrates an example part with multiple polygon surfaces that are not complete tangent visible. The detail tangent visibility results are provided for each of those polygon surfaces.

(a) original part geometry with polygon index  (b) polygon surfaces that are not completely tangent visible
(c) intersection graph for polygon 40

(d) circled rectangle region is not tangent visible for polygon 40

(e) intersection graph for polygon 52

(f) circled triangle region is not tangent visible

(j) intersection graph for polygon 56

(k) circled triangle region is not tangent visible
Figure 3-17 Example part II

Figure 3-18 illustrates a model pagoda. All planar surfaces on this model pagoda are tangent visible. Figure 3-19 shows a human femur model. The tangent invisible triangles are colored in red and illustrated in Figure 3-20.
Figure 3-18 Example part III

Figure 3-19 Example part IV
STL files, generated from commercial software, are used to demonstrate the computation results. Table 3-1 summarizes the computational time of the four example parts. The results show that with the increase in the number of facets, the total computation time increases.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of facets</th>
<th>Computation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Part I</td>
<td>52</td>
<td>12 sec</td>
</tr>
<tr>
<td>Example Part II</td>
<td>60</td>
<td>19 sec</td>
</tr>
<tr>
<td>Example Part III</td>
<td>1382</td>
<td>2218 sec</td>
</tr>
<tr>
<td>Example Part IV</td>
<td>2158</td>
<td>3857 sec</td>
</tr>
</tbody>
</table>

### 3.6 Conclusion

This chapter presented a method for computing global tangent visibility for polyhedral CAD models. A definition of tangent visibility was introduced and the difference between the common visibility and tangent visibility and the importance of
performing tangent visibility analysis for in-line cutting systems were explained. An algorithm to calculate the global tangent visibility was also developed. Several numerical examples are given to illustrate the effectiveness of this algorithm. In conclusion, the method of computing tangent visibility presented in this chapter has useful applications for in-line cutting manufacturing system analysis, such as determining setup orientation and wire path generation. A detailed approach to using tangent visibility results to guide these applications is introduced in the following chapters.
Chapter 4.

Polyhedral Model for WEDM-RP

4.1 Introduction

Polyhedral models are boundary representation models with planar surfaces and straight edges. Polyhedral models are frequently used in computer-aided design and manufacturing, rapid prototyping, and finite element analysis (Zhu and Lee 2005). These models can be generated from a set of points using a triangulation process, or generated directly by various geometric modelers (Veron and Leon 2001). Because of the planar surface and straight-edge characteristics, polyhedral models are approximation models for geometries containing curved lines or surfaces.

The most commonly used polyhedral model is the Stereolithography (STL) model, which represents the object boundary geometry with triangular facets and their facet normals. In an STL file, each triangle facet has three vertices ordered by the right-hand rule and a unit normal vector indicating the positive direction of the triangle. These vertices and vectors are represented in the three-dimensional coordinates. The STL polyhedral surface model representation is platform-independent and accepted by different commercial CAD/CAM software systems. The advantages of universal acceptance and platform-independence make STL the general input for additive RP process. STL representation is an approximation model for geometries containing curved lines or surfaces. More triangles are required for better approximation, and as
such more triangles generally result in a longer processing time for CAD/CAM software and RP processes.

For WEDM-RP, the STL format is used as the polyhedral model input. Based on the complexity analysis presented in Chapter 3, more polygons in the input geometry result in significant increase of computation time. In order to achieve more efficient computation, an algorithm was developed to decrease the number of polygons in the input by combining the co-planar triangles in the STL file. In Section 4.2, the attributed adjacent graph is introduced. In Section 4.3, the algorithm to simplify STL polyhedral model is presented. In Section 4.4, the existing problems with using this approach are discussed along with our proposed solution.

4.2 Attributed Adjacency Graph

The Attributed Adjacency Graph (AAG), proposed by Joshi and Chang, is developed for use in a graph-based feature-recognition method (1988). In the present research, AAG is used to organize and simplify the relationship of the triangulated surfaces in the STL file.

An AAG in WEDM-RP can be defined as graph $G$:

- $G = (N, E, \alpha)$.
- $N$ is the set of nodes. Each node $n_i$ represents a triangular surface in the STL file.
- $E$ is the set of edges. Each edge $e_i$ is the edge shared by two adjacent nodes, $n_i$ and $n_j$.

$E = \{e_i : e_i = n_i \cap n_j, n_i, n_j \in N\}$. 
• $\alpha$ is the set of relations between adjacent edges. The relations are defined as follows:

- If the edge $e_i$ is a concave edge, $\alpha_i = 0$;
- If the edge $e_i$ is a convex edge, $\alpha_i = 1$;
- If the edge $e_i$ is a co-plane edge, $\alpha_i = 2$; and
- The relationships are represented in Figure 4-1.

![Diagram of relationships between planes](image)

**Figure 4-1 Relationships of 2 planes**

Because each triangle in the STL file has three edges, the AAG obtained from the STL file is a 3-regular graph with triangles making up the basic patterns found in those graphs. Figure 4-2 illustrates the cross-slot geometry in the STL format, and its nodes index the details for the triangles in the STL file. Figure 4-3 shows the complete AAG built upon the input STL file. The cross-slot geometry has 18 planar surfaces; however, in the STL format, it has 60 triangle surfaces. Without properly preparing the STL format input, the number of surfaces increases more than 3 times. This significant increase in the number of facets will increase the execution time for the global tangent visibility calculation. In WEDM-RP, an algorithm was developed to combine the co-planar
triangles together and decrease the total number of planar surfaces used to determine the global tangent visibility.

Figure 4-2 Cross-slot facet index illustration
4.3 Algorithm to Simplify the STL Polyhedral Model for WEDM-RP

The AAG in WEDM-RP is presented as edges. Each edge is a triplet with two nodes and an edge attribute. The idea behind this simplification is to combine all co-planar nodes together, for each group of co-plane nodes, a representative node index will be used to represent the whole group of nodes. This representative node index is used to substitute its co-plane node indexes in AAG. The data structures used in this simplification procedure is illustrated in Figure 4-4. All co-planar nodes are stored in NodeLists list. Each element in NodeList is a doublet \((NList, index)\). Each \(NList\) records all node indexes on a same planar surface where \(index\) represents the index.
number for the $NList$. $HeadList$ is an array that records the first element in each $NList$ in $NodeLists$. The relationships between $NodeList$, $NList$, and $HeadList$ are illustrated in and Figure 4-5. The algorithm for simplifying the AAG is presented in Figure 4-6.

![Data Structures](Image)

Figure 4-4. Data structures for $NodeList$, $NList$, and $HeadList$.

![Relationships](Image)

Figure 4-5 Relationships between $NodeList$, $NList$, and $HeadList$
Algorithm 4-1: SimplifyAAG

Input: AAG edge set $E$ and nodes set $N$, co-planeSets ($NodeLists$)
Output: simplified AAG

For (each $e_i \in E$) {
    find $e_i, n_i$ in NodeLists with index number ($List, index$)
    $e_i, n_i = \text{HeadList}(index)$
    find $e_i, n_j$ in NodeLists with index number ($List, index$)
    $e_i, n_j = \text{HeadList}(index)$
    SimplifyAAG = SimplifyAAG + $\{ e_i \}$
}

Figure 4-6 Algorithm to simplify AAG

An algorithm is developed to prepare the co-planar set data to simplify the AAG algorithm. To achieve this simplification, a modified depth search is performed first to find the co-planar nodes set. The algorithm detail is illustrated in Figure 4-7.

Algorithm 4-2: Find_CoPlane_Sets

Input: AAG edge set $E$ and nodes set $N$
Output: sets of co-planar nodes, $HeadList$
$NodeLists = \Phi$ /*List of co-planar node set*/
$Visited = \Phi$ /*List of visited node number*/
$HeadList = \Phi$ /*node index will be used in simplified AAG*/
index = 0

For (each $n_i \in N$) {
    If ($n_i \not\in Visited$) {
        $OpenList = \Phi$
        $CloseList = \Phi$
        $OpenList = OpenList + \{ n_i \}$
        $CloseList = CloseList + \{ n_i \}$
        $Visited = Visited + \{ n_i \}$
        counter + +
        $HeadList = HeadList + \{(n_i, index)\}$
        While ($OpenList \neq \Phi$) {
            $tempN = \text{first node in OpenList}$
            $OpenList = OpenList - \{ tempN \}$
            $\text{Find_neighbor}(tempN,E,neighbors)$
            Add neighbors nodes at beginning of $OpenList$
            $CloseList = CloseList + \{ tempN \}$
            $Visited = Visited + \{ tempN \}$
        }
        $NodeLists = NodeLists + \{(closList, index)\}$
    }
}
$\text{Find_neighbor}(tempN,E,neighbors)$
$neighbors = \Phi$
For (each edge $e_i \in E$) |
    If ($e_i$, alpha = 2) |
        /*this is a co-planar edge*/
        If ($e_i$, $n_i = n_j$) |
            If (co – planeCheck($tempN$, $n_i$, $e_i$)) |
                neighbors = neighbors + {$e_i$, $n_j$}
            
        
    |
    If ($e_i$, $n_j = n_i$) |
        If (co – planeCheck($tempN$, $n_i$, $e_i$)) |
            neighbors = neighbors + {$e_i$, $n_j$}
    |
/*co-planeCheck is introduced in Section 4.4 */

Figure 4-7 Algorithm to find co-planar set

Applying the Algorithm 4-2 to find all the co-planar nodes set in the cross-slot example; the NodeLists result is illustrated in Table 4-1. Figure 4-8 illustrates the surface indexes for the cross-slot STL after simplification. Based on the index information and the relationships in the simplified AAG, the new AAG for the cross-slot geometry is illustrated in Figure 4-9. When this simplification procedure is applied, the total number of planar surfaces decreases significantly. This decrease in planar surfaces will result in significant improvement in computation speed.

This simplification strategy works very well for prismatic geometries. A common triangulation procedure divides a planar surface into several smaller triangles. Combining the triangles into one polygon will decrease the number of planar surfaces and improve computation speed. However, for non-prismatic geometries, the STL accuracy problem affects the simplification result. This accuracy problem in the STL is addressed in next section.
Table 4-1 Cross slot example result

<table>
<thead>
<tr>
<th>Index</th>
<th>Nlist</th>
<th>HeadList</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1, 2}</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>{4, 3}</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>{6, 5}</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>{8, 7}</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>{13, 9, 10, 11, 12, 14}</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>{19, 15, 16, 17, 18, 20}</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>{25, 21, 22, 23, 24, 26}</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>{29, 27, 28, 30, 31, 32}</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>{34, 33}</td>
<td>34</td>
</tr>
<tr>
<td>10</td>
<td>{36, 35}</td>
<td>36</td>
</tr>
<tr>
<td>11</td>
<td>{40, 37, 38, 39, 41, 42}</td>
<td>40</td>
</tr>
<tr>
<td>12</td>
<td>{45, 43, 44, 46, 47, 48}</td>
<td>45</td>
</tr>
<tr>
<td>13</td>
<td>{50, 49}</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>{52, 51}</td>
<td>52</td>
</tr>
<tr>
<td>15</td>
<td>{54, 53}</td>
<td>54</td>
</tr>
<tr>
<td>16</td>
<td>{56, 55}</td>
<td>56</td>
</tr>
<tr>
<td>17</td>
<td>{58, 57}</td>
<td>58</td>
</tr>
<tr>
<td>18</td>
<td>{60, 59}</td>
<td>60</td>
</tr>
</tbody>
</table>

Figure 4-8 Surface index after simplification
4.4 Accuracy Problems in the STL file

4.4.1 Common Errors in the STL File

Due to the fact that the STL file is an approximate polyhedral model, especially for non-prismatic geometries, it can have the following types of errors: gaps, degenerate facets, overlapping facets, and non-manifold topology (Leong, Chua et al. 1996).

The gap error is generally caused by missing facets. This type of error can often exist at the intersections between large curvature surfaces. An example of a missing facet is illustrated in Figure 4-10. Degenerate facets occur when the edges of a triangle facet
are collinear. Because the triangulation is an approximation procedure, the curved surface or edge are replaced by straight-line edges and planar surfaces. This procedure leads to the formation of very narrow triangles in the triangulation procedure. Note that in Figure 4-11, the condensed line segments are in fact narrow triangles. The approximation process results in these narrow triangles. The edges in these triangles are difficult to distinguish from each other in computation due to the round-off errors.

Figure 4-10 Gaps due to missing facets(Leong, Chua et al. 1996)

Figure 4-11 Example of degenerate facet

Overlapping facets may be caused by the numerical round-off errors in triangulation. An overlapping facet example is illustrated in Figure 4-12. Non-manifold
errors include non-manifold-edge, non-manifold-point, and non-manifold-face errors. A non-manifold edge is one that is shared by more than two surfaces. An example of a non-manifold edge is illustrated in Figure 4-13.

Figure 4-12 Overlapping facets example

Figure 4-13 Non-manifold edge example (Leong, Chua et al. 1996)
4.4.2 STL Errors and Solutions in WEDM-RP

In industry and academia, there is a high level of awareness of the stated errors; therefore, many algorithms and heuristics have been developed to solve and avoid them. However, besides the four well-known common errors, WEDM-RP faces a different type of error in STL.

The edge attribute in AAG is assigned based on the angle formed by the two facets sharing this edge. An edge is concave, if the angle formed by the two facets sharing this edge is \([0^0, 180^0]\). An edge is convex, if the angle formed by the two facets sharing this edge is \((180^0, 360^0)\). An edge is a co-planar, if the angle formed by the two facets sharing this edge is \(180^0\). Because the global tangent visibility analysis and STL file simplification will use this information about the convex, concave, and co-plane edges to perform computation, the correctness of the edge classification is very important. Clearly, the correctness of the edge classification depends on the correctness of the facet relationship in the STL file. However, the STL files generated from commercial software packages contain errors in the face relationship. Such face relationship errors result in incorrectness in classifying convex, concave, and/or the co-plane relationship between adjacent facets. This new type of error is named as \textit{incorrectness in edge convexity}. In this research, the STL files generated from Solidworks® 2008 and Pro Engineering® Wildfire® 3.0 are tested; both software packages produce some errors in the face relationship.
In order to better understand this new type of STL error in WEDM-RP, two important factors in STL generation: chord height and angle control, must be addressed first. Known as deviation control factors, these factors will affect the accuracy of the model and the size of its file. Chord height, also known as “chordal tolerance,” specifies the maximum distance between the surface of the original design and the tessellated surface of the STL triangle (the chord). Therefore, the chord height controls the degree of tessellation of the model surface. A smaller chord height will result in less deviation from the actual model and a bigger STL file. Angle control is used to regulate how much additional tessellation occurs along surfaces with small radii. Smaller radii result in more triangles being used in the STL fine. Unless a higher setting is necessary, a smaller angle control setting will achieve a smoother surface.

To illustrate the facet relationship error, a frustum of a cone is created. The part and its dimensional information are shown in Figure 4-14. By changing the chord height and angle control, totally five different STL file are generated. A program is also developed to calculate the angles between adjacent facets in the STL files.
Table 4-2 summarizes the facet relationship errors calculated from the five different STL files. The chord-height column and angle-control column provide the triangulation accuracy control information. Clearly, the number of facets and number of edges increase when the accuracy level increases. The frustum is a convex geometry; the triangulation result should retain the convexity of the frustum. Consequently, every edge on the surface of the frustum should be convex as well. The number of edges column in Table 4-2 indicates the number of concave edges in the corresponding STL file. The number of concave edges increases while the STL file accuracy increases. These concave edges will mislead the global tangent visibility result and make it difficult to properly perform subsequent analysis, such as wire path generation and setup orientation determination. In order to minimize the convexity error effect on WEDM-RP, a new convexity determination method is required—a method that can tolerate the convexity error and generate acceptable adjacent facet relationships.
<table>
<thead>
<tr>
<th>Tolerance (inch)</th>
<th>Angle (chord height)</th>
<th># facets</th>
<th># edges</th>
<th># Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solidwork case 1</td>
<td>0.033727</td>
<td>30</td>
<td>152</td>
<td>228</td>
</tr>
<tr>
<td>Solidwork case 2</td>
<td>0.01299</td>
<td>10</td>
<td>244</td>
<td>366</td>
</tr>
<tr>
<td>Solidwork case 3</td>
<td>0.0014052</td>
<td>0.5</td>
<td>2876</td>
<td>4314</td>
</tr>
<tr>
<td>Pro E case 1</td>
<td>0.094444</td>
<td>0.5</td>
<td>84</td>
<td>126</td>
</tr>
<tr>
<td>Pro E case 2</td>
<td>0.0028</td>
<td>0.1</td>
<td>528</td>
<td>792</td>
</tr>
</tbody>
</table>

In WEDM-RP, the angle between two adjacent facets is calculated following a widely accepted method proposed by Ye et al. (2001). Assume that facet $F_1$ contains three points $\{v_1, v_2, v_3\}$ and its normal vector is $N_1$. Facet $F_2$ contains three points $\{v_1, v_2, v_4\}$ and its normal vector is $N_2$. The shared points of $F_1$ and $F_2$ are $v_1$ and $v_2$, respectively. Let the angle between $N_1$ and $N_2$ be as $\beta$. If the dot product of vector $\overrightarrow{v_3v_4}$ and $N_2$ is larger than zero, then the angle between the two facets $\alpha = 180 + \beta$. If the dot product is less than zero, then the angle is $\alpha = 180 - \beta$. The details are illustrated in Figure 4-15.
Besides using angle to determine the relationship of adjacent facets, the distance from the non-sharing point to the facet is also a good benchmark to determine the relationship between two adjacent facets. Distance $d_1$ and $d_2$ in Figure 4-15 are examples of distances from the non-sharing point to the facet. If both distances from non-sharing point to the facet are less than an allowance value, those two facets could be treated as co-planar. However, a fixed allowance is not a good approach considering that the accuracy level varies in STL file. In WEDM-RP, the allowance value is assigned to chord height value such that the computation error can match up with the input accuracy level. An auto correction program is built to automatically fix the close-to-co-plane facets as co-plane facets. In WEDM-RP, the angle is used as the main factor to determine the facet angle. Figure 4-16 is used to determine the final relationship between two adjacent facets.
Determine facet angle

Input: \( F_1 = \{N_1, v_1, v_2, v_3\}, F_2 = \{N_2, v_1, v_2, v_4\}, \) allowance \( \epsilon \)

Output: Angle \( \alpha \)

\[ \beta = \arccos(N_1 \cdot N_2) \]

\( d_1 \) is the distance from \( v_3 \) to \( F_2 \) (\( d_1 < 0 \) if \( d_1 \) or negative side of \( F_2 \))

\( d_2 \) is the distance from \( v_4 \) to \( F_1 \) (\( d_2 < 0 \) if \( d_2 \) or negative side of \( F_1 \))

\[ \text{If } (\overline{v_3v_4} \cdot N_2 > 0) \{ \alpha = 180 + \beta \} \]

\[ \text{If } (\overline{v_3v_4} \cdot N_2 < 0) \{ \alpha = 180 - \beta \} \]

\[ \text{If } (\overline{v_3v_4} \cdot N_2 \approx 0) \{ \]

\[ \quad \text{If } (|d_1| < \epsilon) \text{ and } (|d_2| \leq \epsilon) \{ \]

\[ \quad \quad \alpha = 180 \]

\[ \}

\]

Figure 4-16 Procedure to determine the facet angle

A pagoda model, shown in Figure 4-17, is tested using a fixed allowance and the result is illustrated in Table 4-3.

Figure 4-17 A pagoda model
Table 4-3 Auto-correction result for the pagoda model

<table>
<thead>
<tr>
<th>Facet #</th>
<th>Chord height</th>
<th>Angel control</th>
<th>Corrected</th>
<th>Left unchanged</th>
<th>Maximum distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>492</td>
<td>0.01052</td>
<td>0.5</td>
<td>10</td>
<td>12</td>
<td>0.03111</td>
</tr>
<tr>
<td>1382</td>
<td>0.005</td>
<td>0</td>
<td>347</td>
<td>2</td>
<td>0.005013</td>
</tr>
<tr>
<td>1488</td>
<td>0.004</td>
<td>0</td>
<td>349</td>
<td>0</td>
<td>0.003548</td>
</tr>
<tr>
<td>11970</td>
<td>0.0003</td>
<td>0</td>
<td>5607</td>
<td>0</td>
<td>0.000295</td>
</tr>
</tbody>
</table>

In Table 4-3, when comparing the chord height with the maximum distance from a non-sharing point to the other facet, the chord height is generally larger than the maximum distance. As a result, chord height is a good benchmark by which to determine the co-planar relationship between two adjacent facets. For the “left unchanged” edges, WEDM-RP outputs those questionable edges and let the user define the convex, concave, and co-planar relationships between the facets sharing those edges.

Because of the operations to eliminate round-off error effect, the original strategy to determine co-planarity will not be applicable for some special cases. Figure 4-18 illustrates a cumulative error example. The co-plane facets are surrounded by red segments. The reasons for curved co-plane phenomenon are caused by the co-planarity determination strategy used in WEDM-RP.
Figure 4-18 Cumulative error in co-plane determination

Figure 4-19 illustrates the details of the cumulative co-planar errors. Assume that facets $f_1, f_2, f_3,$ and $f_4$ are treated as co-planar facets based on their co-planar relationships. If $f_1$ is a co-planar with $f_2$, and $f_2$ is a co-planar with $f_3$, then $f_1$ should be a co-planar with $f_3$. However, when these facets are co-planar because they satisfy the co-planar testing criteria mathematically but are not exactly co-planar, there is a strong
possibility that errors will accumulate. For example, in the given hip-implant example, $f_1$ is a co-planar with $f_2$, $f_2$ is a co-planar with $f_3$, and $f_3$ is a co-planar with $f_4$. Clearly, $f_4$ is not a co-planar with $f_1$; this contradicts the common derivation result. As mentioned in Section 4.2, there is a special co-plane check procedure to eliminate the cumulative errors in the co-planar operation.

Figure 4-19 Cumulative error detail example

In Algorithm 4-2, the function “co-planeCheck” is used to prevent cumulative errors. The purpose of this function is to test the distances from the points on the potential co-plane facet to the original facet. If the distances are smaller than the
allowance $\epsilon$, then the facet will be treated as a co-plane facet; otherwise, the edge attribute will be set to its original attribute. The details of the co-planeCheck function are illustrated in Figure 4-20. After the co-planeCheck function has been applied, the cumulative error is eliminated. For the example shown in Figure 4-19, $f_1$ is a co-plane only with $f_2$, and $f_3$ is a co-plane with $f_4$.

<table>
<thead>
<tr>
<th>Co-planeCheck function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: Original Node $tempN$, new Nodes $newN =* (v_1, v_2, v_3, N) *$, edge $e_i$</td>
</tr>
<tr>
<td>Output: true / false</td>
</tr>
<tr>
<td>$d_1$ is the distance between $v_1$ and facet $tempN$</td>
</tr>
<tr>
<td>$d_2$ is the distance between $v_2$ and facet $tempN$</td>
</tr>
<tr>
<td>$d_3$ is the distance between $v_3$ and facet $tempN$</td>
</tr>
<tr>
<td>If $((d_1 \leq \epsilon) \ and \ (d_2 \leq \epsilon) \ and \ (d_3 \leq \epsilon))$</td>
</tr>
<tr>
<td>return true</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>$e_i, \alpha = 1$ /<em>change the edge attribute to convex</em>/</td>
</tr>
<tr>
<td>return false</td>
</tr>
</tbody>
</table>
| }

Figure 4-20 Detail of Co-planeCheck

4.5 Summary

In this chapter, a method was developed to simplify the STL polyhedral model and combine co-plane triangles into polygons. By this simplification procedure, the total computation time will decrease significantly because the total number of planar surfaces decreases. The Attributed Adjacency Graph, the graphic-based feature-recognition technique, is used to build the relationships between the triangles in the STL files. When the graph is analyzed and simplified, the triangles on the same plane are combined into one polygon. Furthermore, the existing problems in the STL format and a new STL error problem in convexity are also discussed. Algorithms are developed to eliminate the
convexity problem in the STL file by introducing an error allowance during the co-plane determination procedure. The cumulative error for co-plane determination is defined and discussed. A co-plane check function used to prevent cumulative error is also provided in detail.
Chapter 5.
Manufacturing Coordinates Determination and Setup Optimization

In Chapter 3, the algorithm to compute global tangent visibility for polyhedral models was presented. In this chapter, the methodology to use the global tangent visibility results to determine favorable setup orientation and manufacturing coordinates was illustrated.

Section 5.1 explains the interpretation of tangent visibility results. Section 5.2 defines the manufacturing coordinate system for WEDM machines. In Section 5.3, several lemmas are proved to support the classification of tangent visibility results. In Section 5.4, the algorithm to determine the number of intermediate coordinate systems for WEDM-RP is presented in detail.

5.1 Interpretation of Tangent Visibility Results

In order to make WEDM process capable for RM applications, automation of setup orientation determination and wire path generation is the most important obstacle to overcome. After obtaining the global tangent visibility result, the tangent visibility result need to be translated into valuable manufacturing information. Global tangent visibility region for a polygon \( P \) is formed by a polygon combining procedure. This polygon combining procedure was presented in detail in Algorithm 3-2 in Chapter 3. In the algorithm, three different polygon operation functions are used. In each of these
three functions, a “large enough” triangle or rectangular region, named as region $R$, is built upon the input polygon $P$. These functions will generate results in the intersection of polygon $P$ and region $R$. For example, Figure 5-1 provides a demonstration of the intersection of polygon $P$ and “big enough” triangles. The “big enough” triangles are formed by a concave edge $\overline{v_{10}v_{11}}$ and a point $v_{1}$ on the polygon $P$, as shown in Figure 5-1 (b). The shaded area in Figure 5-1 (c) is the tangent visible region covered by the “big enough” triangle. This will be used with other tangent visible regions to form the final global tangent visibility region for polygon clipping operations. However, these polygons cannot be used directly to inform manufacturing operations. The truly useful information for manufacturing comes from the “big enough” triangles. The two straight lines which form those triangles, $v_{10}v_{11}$ and $v_{1}v_{11}$ (as illustrated Figure 5-1 (b)) , can in fact be utilized as orientations for the cutting wire. To cover the tangent visible region, the cutting wire needs to be rotated around point $v_{11}$ on polygon $P$ from $v_{1}v_{11}$ to $v_{11}v_{10}$.
A demonstration of the intersection of polygon $P$ and the “big enough” rectangle is illustrated in Figure 5-2. The “big enough” rectangle is formed by a concave edge $v_{11}v_{12}$ and a point $v_8$ on the polygon $P$, as shown in Figure 5-2 (a). The shaded area in Figure 5-2 (b) is the tangent visible region covered by the “big enough” rectangle and will be used with other tangent visible regions to form the final global tangent visibility region for polygon union operations. The two straight lines, $v_{11}v_{12}$ and $v_8v_8'$, as illustrated in Figure 5-2 (a), which form this rectangle, can in fact be utilized as orientations for the cutting wire. To cover the tangent visible region, the cutting wire needs to move from $v_{11}v_{12}$ to $v_8v_8'$.

---

1 The red lines in the polygon represent concave edges; the black lines represent convex edges.
A special group of polygons does not require global tangent visibility analysis due to the absence of concave edges in their intersection graph. For example, Polygon 25 does not contain any concave edges on its intersection graph, as shown in Figure 5-3. In order to cover the whole area of Polygon 25, several possible combinations can be used. Two possible combinations are illustrated in Figure 5-3. Due to the absence of concave edges, any straight line on the surface of Polygon 25 can serve as the cutting wire orientation. Both line set 1 and line set 2 (as shown in Figure 5-3), are capable of covering the whole area of Polygon 25. In order to simplify the translation procedure from visibility results into manufacturing information, the “big rectangle” approach is used by finding any two parallel straight lines that are capable of covering the whole area of the given polygon without concave edges. For example, line set 1 in Figure 5-3 is formed by two parallel straight lines, and serves as a feasible solution, covering polygon 25 in Figure 5-3.
In summary, manufacturing information that is translated from global tangent visibility results only contains two types of basic shapes: triangles and rectangles. Based on these two types of shapes, covering the tangent visible region requires following only two rules:

1. To cover a triangle shape: the tangent visibility result will provide the middle point, starting cutting line and end cutting line. The cutting wire will rotate around the middle point on polygon $P$ from the starting cutting line to the end cutting line. An example is illustrated in Figure 5-4.

Figure 5-3 Example of polygon without concave edges
2. To cover a rectangle shape: the tangent visibility result will provide the starting cutting line and end cutting line information. The cutting wire will move from the starting cutting line to the end cutting line on the surface of polygon $P$. An example is illustrated in Figure 5-5.
5.2 Manufacturing Coordinates in WEDM-RP

The coordinate system for global tangent visibility results is based on the coordinate system of the input geometry. Therefore, the coordinates used to define the starting cutting lines, end cutting lines and middle points are the same as the coordinates of the input geometry. However, the coordinates required to build part should be based on the WEDM machines. In this section, the definition of a coordinate system in WEDM-RP is presented, and a method of converting the coordinate information into a final manufacturing coordinate system for WEDM machines is introduced.
The cutting tool for WEDM machines is a straight cutting wire. The rotational capability on six-axis WEDM machine comes from the rotational axis for the workpiece instead of the cutting wire. Due to the machine’s physical limitations, the cutting wire movement range is restricted. In order to access all tangent visible regions, the movement of the cutting wire and the rotational movement of the producing part must be properly coordinate.

In this research, a series of manufacturing coordinates are defined to classify the tangent visibility results into several categories. Under each set of manufacturing coordinates, the cutting wire can access all tangent visible regions without the movement of the rotational axis. Each coordinate system is defined by three vectors: (1) rotational norm, (2) rotational orientation and (3) orientation product. Detailed definitions of these vectors are provided as follows.

**Definition 5-1: Rotational Norm** \( R_N \): Rotational Norm is a unit vector that starts at the center of mass of the given geometry.

**Definition 5-2: Rotational Orientation** \( R_o \): Rotational Orientation is a unit vector that is perpendicular to the rotational norm \( R_N \). It is also a vector that starts at the center of mass of the given geometry.

**Definition 5-3: Orientation Product** \( O_p \): Orientation product is the cross product of rotational norm \( R_N \) and rotational orientation \( R_O \). \( O_p = R_N \times R_O \)
Definition 5-4: Manufacturing Coordinates: *Manufacturing coordinates* are the coordinates used on WEDM machines. All wire path information should be generated using manufacturing coordinates so the input geometry can be fabricated on WEDM machines.

Definition 5-5: Intermediate Coordinates: *Intermediate coordinates* make up the coordinate system that is defined by rotational orientation $R_O$, rotational norm $R_N$, and orientation product $O_p$. In WEDM-RP, a series of intermediate coordinate systems are created to classify the cutting wire path information. These intermediate coordinates are then translated into manufacturing coordinates along with the cutting wire path information.

Figure 5-6 illustrates the relationships among rotational norm $R_N$, rotational orientation $R_O$, and orientation product $O_N$. Each of these vectors has a physical meaning. Rotational Norm $R_N$ is same as the rotational axis on a six-axis WEDM or other rotary axis. The direction of the rotation is defined using right hand rule: the fingers of the right hand are curled to match the rotation motion, and the thumb indicates the direction of the vector. For example, Figure 5-7 illustrates an example of a rotational norm. For each given rotational norm $R_N$, a series of $R_O$'s is defined, as illustrated Figure 5-7. Each rotational orientation $R_O$ is used to indicate the current position of the rotary axis under a certain coordinate. Orientation product $O_p$ represents the neutral position of the cutting wire. Based on the physical limitations of a WEDM machine, the cutting wire can only rotate in a restricted area around the orientation product $O_p$. 
Figure 5-6 A coordinate system for WEDM-RP

Figure 5-7 Definitions of $R_N$ and $R_0$

Figure 5-8 provides an example of intermediate coordinates and manufacturing coordinates. The neutral wire position is determined by the manufacturing coordinate.
Surface 1 is tangent visible and can be accessed using the manufacturing coordinate. Surface 2 is also tangent visible, but the wire orientation is under the intermediate coordinate. In order to access Surface 2 using a physical cutting wire, the rotary chuck is required to rotate so $R_o$ of the intermediate coordinate is horizontal to the manufacturing coordinate. Mathematically, in order to transform the wire orientation into the neutral wire position, the intermediate coordinate system needs to be translated into a manufacturing coordinate system. In WEDM-RP, the translation of tangent visibility results into manufacturing information is a procedure which classifies the cutting wire orientation and transforms the intermediate coordinate system into a manufacturing coordinate system.

![Diagram of intermediate and manufacturing coordinates](image)

**Figure 5-8** Demonstration of intermediate coordinate and manufacturing coordinate
5.3 Classification of Tangent Visibility Results

Based on the discussion in Section 5.1, the global tangent visibility result contains only triangle and rectangle shapes. Due to the fact that the coordinates of the tangent visibility result are based partly on geometry coordinates, those shapes needs to be classified into intermediate coordinate systems before they can be translated into manufacturing coordinates. In this section, the methods of classifying edges and facets into different intermediate coordinate systems are discussed.

5.3.1 Classification of Edges

All triangle or rectangle shapes require edges to cover them. For example, each triangle shape, as illustrated in Figure 5-4, requires two edges to cover the whole triangle. The two edges are the starting cutting edge and the end cutting edge. Several lemmas are introduced in order to classify those edges under a certain intermediate coordinate system.

Lemma 5-1: An edge is coverable for a certain $R_N$ if the edge forms an angle with $R_N$ between $[90 - \alpha, 90 + \alpha]$, where $\alpha$ is the taper angle limitation for WEDM machines.

Proof: Due to the physical limitation of WEDM machines, the maximum angle between the edge and the neutral wire position is $\alpha$. Under a certain intermediate coordinate system, the angle range between the vector Orientation Product $O_P$ and the edge is $[-\alpha, \alpha]$. If the edges are normalized and the start point of the edges are moved to center of the mass (as illustrated in Figure 5-9), all edges that are located in the cone...
space are coverable under the current intermediate coordinate system. Because $O_p = R_N \times R_0$, $O_p$ is perpendicular to $R_N$. As the result, if any edge forms an angle between $[90 - \alpha, 90 + \alpha]$ with $R_N$, then the edge is coverable under the current intermediate coordinate system.

![Figure 5-9 Edge and $R_N$ relationship](image)

**Lemma 5-2:** In 3D space, only $[90 \div \alpha]$ number of $R_N$s are required to cover all possible edges in 3D space, where $\alpha$ is taper angle limitation for WEDM machines.

**Proof:** If we normalize all edges and translate their starting point to center of mass, then all edges in 3D will form a unit globe with center at center of mass. According to Lemma 5-1, all coverable edges for a given $R_N$ are shown in Figure 5-10 (a). Each $R_N$ is capable of covering $2\alpha$ degree of edges, as illustrated in Figure 5-10 (b). Clearly, if a whole globe of edges is required to cover, we only need $[360 \div (2 \times 2 \times \alpha)] = [90 \div \alpha]$ number of $R_N$s.
5.3.2 Classification of Polygons without Concave Edge

In Section 5.1, the tangent visibility coverage for polygons without concave edges is discussed. Rectangle shapes are used to cover those polygons. However, the number of feasible solutions for those rectangle shapes is infinite. A certain edge orientation for the rectangle shape still needs to be determined in order to translate the tangent visibility result into manufacturing information. Several lemmas are introduced in order to classify those polygons without concave edges under a certain intermediate coordinate system.

**Lemma 5-3**: A polygon without concave edges is coverable for a certain $R_N R_O$ intermediate coordinate system if the polygon norm forms an angle with $O_N = R_N \times R_O$ between $[90 - \alpha, 90 + \alpha]$, where $\alpha$ is the taper angle limitation for WEDM machines.

**Proof**: For a given polygon norm, a valid wire orientation is perpendicular to the facet norm. So, if the angle between the polygon norm and the orientation product $O_P$ is
between \([-\alpha, \alpha]\), then as illustrated in Figure 5-11, the polygon is reachable under the current intermediate coordinate system.

![Figure 5-11 Facet Norm and \(O_p\) relationship](image)

**Lemma 5-4:** Only \([90 \div \alpha]\) number of \(R_Os\) under the any \(R_N\) is required to cover all possible polygons in 3D space, where \(\alpha\) is the taper angle limitation for WEDM machines.

**Proof:** In 3D, if all polygon norms are normalized and their starting points are translated to the center of a mass, then all polygon norms will form a unit globe with its center at the center of that mass. According to Lemma 5-3, all coverable polygon norms for a given \(O_p\) are shown in Figure 5-12. Each \(O_p\) is capable of covering \(2\alpha\) degrees of an edge. Clearly, if covering a whole globe of edges is required, then we only need \([360 \div (2 \times 2 \times \alpha)] = [90 \div \alpha]\) number of \(O_p\)s.
5.4 Determination of Intermediate Coordinates for WEDM-RP

Based on the discussion in Section 5.3, the tangent visibility results can be classified into several intermediate coordinates and then translated into manufacturing coordinates. Based on Lemma 5-1 and Lemma 5-2, \([90 \div a]\) number of \(R_N\)s are capable of covering all possible edges in 3D space. Based on Lemma 5-3 and Lemma 5-4, \([90 \div a]\) number of \(O_p\)s under any \(R_N\) is enough to cover all polygons without concave edges. However, even though any set of intermediate coordinate systems can be a feasible solution for WEDM-RP, finding the optimal solution for setup is not guaranteed. In this section, an algorithm to determine the optimal set of intermediate coordinate systems is presented such that the number of trial setups is minimized.
Based on the lemmas in Section 5.3, in order to find the optimal set of intermediate coordinates, the most important vectors to define are the rotational norm $R_N$s. With properly defined $R_N$s, all possible edges in 3D space and polygons without concave edges can be covered. Based on the definitions of rotational norm, rotational orientation, and orientation product, an intermediate coordinate system requires at least two out of the three vectors, and the third vector is the cross product resulting from the two known vectors. An overall flowchart on how to determine the intermediate coordinate system by determining the rotational norm $R_N$ and rotational orientation $R_O$ is presented. The flowchart is illustrated in Figure 5-13.

![Flowchart to determine intermediate coordinate systems](image)

**Figure 5-13 Flowchart to determine intermediate coordinate systems**

In this calculation procedure, a greedy algorithm is applied to calculate two initial vectors, initial $R_N$ and initial $R_O$. The intermediate coordinates formed by initial $R_N$ and initial $R_O$ will cover most of the tangent visible areas of the given
geometry. Based on the two initial vectors, $[90 \div \alpha]$ norms are calculated by rotating initial $R_N$ around initial $R_O$. Each rotational norm forms a $2\alpha$ angle with its neighbor rotational norms. For example, in Figure 5-14, the taper angle $\alpha = 20^0$, $[90 \div \alpha] = [90 \div 20] = 5R_N's$ is defined around the initial $R_O$. Each $R_N$ forms a $40^0$ angle with its neighbor $R_N's$.

![Initial $R_O$ with its $R_N's$](image)

**Figure 5-14 An example of initial $R_O$ with its $R_N's$**

Based on Lemma 5-4, for any given $R_N$, only $[90 \div \alpha]$ intermediate coordinate systems are required to cover all polygons without concave edges. Consequently, we build $[90 \div \alpha] R_O$ evenly distributed vectors around initial $R_N$. Each rotational orientation $R_O$ forms a $2\alpha$ angle with its neighbor rotational orientations. For example, in
Figure 5-15, the taper angle $\alpha = 20^\circ$, $[90 \div \alpha] = [90 \div 20] = 5 \, R_o$’s is defined around the initial $R_N$. Each $R_o$ forms a $40^\circ$ angle with its neighbor $R_o$’s.

![Figure 5-15 An example of initial $R_N$ with its $R_o$'s](image)

The flowchart for calculating initial $R_N$ is illustrated in Figure 5-16. In this procedure, a list of initial $R_N$ candidates is generated. For each facet or polygon with concave edges, the facet or polygon norm is used as a potential initial $R_N$ and saved into the $RnList$. The total area that is reachable under each initial $R_N$ candidate is calculated, and the final initial $R_N$ is the one that has the maximum reachable total area. Furthermore, the initial $R_N$ affects the cut off plane, as discussed in Chapter 6. Consequently, any initial $R_N$ candidate must have at least one polygon without a
concave edge that is perpendicular to the initial \( R_N \). This perpendicular polygon is used as the cut off plane.

![Flowchart of calculating initial \( R_N \)](image)

**Figure 5-16** Flowchart of calculating initial \( R_N \)

The initial rotational orientation \( R_0 \) is a unit vector that is perpendicular to the initial \( R_N \). The flowchart for calculating initial \( R_0 \) is presented in Figure 5-17. In this procedure, all polygon norms which are perpendicular to the initial \( R_N \), are candidates for initial \( R_0 \). Those candidates are grouped together if the norms coincide. For each group of polygons, the total area is calculated. The final initial \( R_0 \) is the group norm with the maximum total area. If no polygon has a norm that is perpendicular to the initial \( R_N \), the initial \( R_0 \) will be an arbitrary vector that is perpendicular to the initial \( R_N \).
After calculating \([90 \div \alpha]\) number of \(R_N\)'s and \([90 \div \alpha]\) number of \(R_O\)'s under initial \(R_N\), the resulting intermediate coordinate systems are able to cover all polygons without concave edges. More intermediate coordinate systems may need to be defined using the \([90 \div \alpha]\) number of \(R_N\)'s calculation in order to cover all possible edges from the tangent visibility results. A classification procedure is used to classify the tangent visibility results under different intermediate coordinate systems. Figure 5-18 illustrates the flowchart for classifying tangent visibility results into different coordinate systems.
There are three types of tangent visibility results, as discussed in Section 5.1. For a rectangle result, if the edge defines the rectangle shape it can be covered under a certain intermediate coordinate; the whole rectangle shape can be covered under the intermediate coordinate. For a triangle result, each shape is formed by two edges, and an intermediate coordinate can be determined for each edge. A procedure called FindRnRoEdge is defined to determine the intermediate coordinate system for a given edge. The flowchart of FindRnRoEdge is illustrated in Figure 5-19. The third type of tangent visibility result is a polygon without concave edges. As discussed in Section 5.2, there are an infinite number of feasible solutions to cover these polygons. Based on Lemma 5-3, we need only to find one intermediate coordinate system for these polygons. Due to the fact that any $[90 + \alpha]$ number of $R_o$ under any given $R_N$ is capable of covering all polygons without concave edges, the initial $R_N$ and its first pre-defined $[90 + \alpha]$
number of $R_o$ will be used to cover all polygons without concave edges. A function named “FindRnRoFct” is defined to classify the polygon without concave edges into different intermediate coordinates.

![Flowchart for FindRnRoEdge procedure](image)

In “FindRnRoEdge” procedure, the angle between the input edge and pre-defined $R_N$ are calculated. Based on Lemma 5-1, the $R_N$ forms an angle with the input edge between $[90 - \alpha, 90 + \alpha]$ will be selected. The angle between the input edge and existing orientation product $O_N$ under the current $R_N$ is calculated; the $O_N$ that forms an angle with the input edge between $(0, \alpha)$ or $(180 - \alpha, 180)$ is select. In case no existing $O_N$ satisfies this requirement, a new $O_N$ is calculated by the cross-product of the
input edge and the rotational norm \( R_N \). This procedure returns the \( R_N \) and \( R_o \) that form the intermediate coordinate system to cover the edge.

In “FindRnRoFct” procedure, the angle between the input polygon norm and pre-defined \( O_P \) under initial \( R_N \) is calculated. Based on Lemma 5-3, the \( O_P \) forms an angle with the input edge between \([90 - \alpha, 90 + \alpha]\) will be selected. The detailed information of the procedure is shown in Figure 5-20.

<table>
<thead>
<tr>
<th>FindRnRoFct Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> initial ( R_N ), all pre-defined ( R_o ) under initial ( R_N ), the polygon ( P )</td>
</tr>
<tr>
<td><strong>Output:</strong> intermediate coordinate system ( R_{N-out} ) ( R_{O-out} )</td>
</tr>
<tr>
<td>( R_{N-out} = \text{initial } R_N )</td>
</tr>
<tr>
<td>( \text{find an } O_P(i) = R_N(i) \times R_O(i), \text{ forming an angle with norm of } P \text{ between } [90 - \alpha, 90 + \alpha] )</td>
</tr>
<tr>
<td>( R_{O-out} = R_O(i) )</td>
</tr>
<tr>
<td><strong>Result is</strong> ( R_{N-out} ) ( R_{O-out} )</td>
</tr>
</tbody>
</table>

Figure 5-20 Detail for FindRnRoFct

### 5.5 Summary

In this chapter, a method of translating tangent visibility results into manufacturing information was presented. An algorithm to determine the optimal set of intermediate coordinate systems was also developed in this chapter. The tangent visibility result is classified into the optimal set of intermediate coordinates such that the total number of trial setups is minimized. Several lemmas are also proved to support the classification procedure. After the result classification, all tangent visibility results can be transformed from intermediate coordinate systems into manufacturing coordinate systems. The transformed information is then used in wire path generation.
Chapter 6.

Wire Path Planning Methodology

This chapter introduces a methodology for automating the tasks involved in wire path planning for WEDM-RP. Section 6.1 presents a method of generating the wire path and transforming it into its final manufacturing orientation. In Section 6.2, wire cutting sequence problems are discussed in detail. In Section 6.3, the cut-off plane for WEDM-RP is presented.

6.1 Wire Path Generation and Transformation

6.1.1 Wire Path Generation

In Chapter 5, tangent visibility results are classified into several intermediate coordinate systems. Under each intermediate coordinate system, the tangent visibility results are stored as rectangle shapes, triangle shapes and polygons without concave edges. In this section, a detailed wire path generation method is presented for all three result types.

Figure 6-1 illustrates an example of a cutting wire in its intermediate coordinate system. In this example, the UV plane and the XY plane are perpendicular to vector $O_p$ in the intermediate coordinate system. The distance between the UV plane and the XY plane is $D$, which is a parameter related to the physical restrictions of a WEDM machine. The distance $D$ represents the physical distance between the upper wire guide and lower wire guide on a WEDM machine. The distances from the origin of the intermediate
coordinate system to the UV plane and the XY plane are both equal to $\frac{D}{2}$. In Figure 6-1, the cutting edge from the tangent visibility result is an edge which covers the polygon in the graph. The extension of the cutting edge will intersect with the UV plane and the XY plane to create the cutting wire trajectory points.

![Diagram](image)

**Figure 6-1 A cutting wire in its intermediate coordinate system**

A flowchart for wire path generation is shown in Figure 6-2. The tangent visibility results are classified into rectangle results, triangle results, and results for polygons without concave edges. For each result, corresponding intermediate coordinate systems and wire path trajectories are found.
A rectangle result has two cutting edges that have the exact same orientation. Consequently, the rectangle result can be finished using the same intermediate coordinate system. The cutting wire trajectories on the XY plane and the UV plane are used to guide the cutting wire and finish the rectangle coverage.

A triangle result has two cutting wire orientations; the orientation differences may result in a need for different intermediate coordinate systems to cover the triangle result. There are three possible scenarios for the intermediate coordinate system for triangle coverage: the cutting wires are in the same coordinate system; the two cutting wires share the same rotational orientation $R_N$; or the two cutting wires share different
rotational orientations \( R_N \)'s. The flowchart for determining the intermediate coordinate system for triangle results is illustrated in Figure 6-3. Based on the result, one or multiple intermediate coordinate systems are generated to cover the whole triangle area.

Figure 6-3 Flowchart for determining an intermediate coordinate system for a triangle result

As discussed in Chapter 5, a polygon without concave edges can be covered using an infinite number of rectangle regions. The straight lines parallel to vector \( O_P \) are used as the rectangle boundary, as illustrated in Figure 6-4. Because the straight lines covering the polygon have the exact same orientation, the coverage for a polygon without concave edges can be finished using the same intermediate coordinate system as well.
6.1.2 Wire Path Transformation

After the wire path trajectories are generated for each tangent visibility result, those trajectories are transformed into a manufacturing coordinate system. The manufacturing coordinate system is illustrated in Figure 6-5:

1) The origin of the manufacturing system is at the rotational center of the B-axis on a six-axis WEDM machine.

2) The UV plane and the XY plane are perpendicular to the Z-axis.
In order to transform the intermediate coordinate system into the final manufacturing coordinates, the center of mass (the origin of the intermediate coordinates), needs to be translated. The rotational norm $R_N$ needs to be translated to the $X$-axis. Assuming that the center of mass is $P_{CM}$, the trajectory point in the intermediate coordinate is $P$, the rotational norm of the intermediate coordinate is $R_N$, and the rotational orientation is $R_O$. The new trajectory point $P_{new}$ in the manufacturing coordinate can be calculated by:

$$P_{new} = P 	imes T - P_{CM}$$

Equation 6-1

where $T$ is the transformation matrix and can be calculated by:

$$T = \begin{bmatrix}
\cos \theta + \omega_z^2 (1 - \cos \theta) & \omega_x \omega_z (1 - \cos \theta) - \omega_z \sin \theta & \omega_x \omega_y (1 - \cos \theta) + \omega_y \sin \theta \\
\omega_z \sin \theta + \omega_x \omega_y (1 - \cos \theta) & \cos \theta + \omega_y^2 (1 - \cos \theta) & \omega_z \omega_y (1 - \cos \theta) - \omega_x \sin \theta \\
\omega_x \omega_z (1 - \cos \theta) - \omega_z \sin \theta & \omega_x \sin \theta + \omega_z \omega_y (1 - \cos \theta) & \cos \theta + \omega_x^2 (1 - \cos \theta)
\end{bmatrix}$$
where $\omega = R_N \times [1,0,0]$, $\theta = \cos(R_N \cdot [1,0,0])$

### 6.2 Sequence of the Wire Cutting Operation

Each shape in the tangent visibility result has a trajectory segment on the UV plane and the XY plane. The wire path generated by this method is isolated. In order to minimize the rapid movement between the cutting segments, the separate trajectories should be translated into one connected trajectory. In this section, a method is presented to combine separate trajectories into one trajectory and generating the wire path plan automatically. The flowchart representing the generation of the cutting sequence is illustrated in Figure 6-6. In this operation, the wire trajectories are classified into trajectories generated by triangle coverage and trajectories generated by rectangle coverage. For all trajectories generated by triangle coverage, a function named “GOTOZero” is used to generate the lead-in wire path before the trajectory and lead-out wire path after the trajectory. For all rectangle coverage generated trajectories, the connected trajectories form one cutting chain, and the “GOTOZero” function is used to generate the lead-in wire path before the newly-formed cutting chain and the lead-out wire path after the cutting chain.
Before presenting the detailed methodologies related to generating lead-in/lead-out wire paths and cutting chain trajectories generated for rectangle coverage, several definitions used in those methods are presented. Six surfaces are defined as follows and illustrated in Figure 6-7:
**UV plane** is a plane parallel to \( R_N R_O \) plane, and in the area corresponding to negative direction of vector \( O_p \). The distance from the origin to the UV plane is \( \frac{D}{2} \). This plane is used for wire path projection.

**XY plane** is a plane parallel to \( R_N R_O \) plane, and in the area corresponding to the positive direction of vector \( O_p \). The distance from the origin to the UV plane is \( \frac{D}{2} \). This plane is used for wire path projection.

**Right Plane** is a plane parallel to \( O_p R_N \) plane, and in an area corresponding to the positive direction of vector \( R_O \). The distance from origin to the right plane is \( \frac{\text{width}}{2} \).

**Left Plane** is a plane parallel to \( O_p R_N \) plane, and in an area corresponding to the negative direction of vector \( R_O \). The distance from the origin to the left plane is \( \frac{\text{width}}{2} \).

**Top plane** is a plane parallel to \( R_O O_p \) plane, and in an area corresponding to the positive direction of vector \( R_N \). The distance from the origin to the top plane is \( \frac{\text{height}}{2} \).

**Bottom plane** is a plane parallel to \( R_O O_p \) plane, and in an area corresponding to the negative direction of vector \( R_N \). The distance from the origin to the top plane is \( \frac{\text{height}}{2} \).
In these definitions, parameter $D$ is the fixed distance from the UV plane to the XY plane. This is a distance related to machine dimensions. Parameter $height$ and $width$ are related to input machine dimension limitations. In addition, four edges are defined and illustrated in Figure 6-7: (1) **Top Right Edge** is the intersection edge of the top plane and right plane; (2) **Top Left Edge** is the intersection edge of the top plane and left plane; (3) **Bottom Right Edge** is the intersection edge of the bottom plane and right plane; (4) **Bottom Left Edge** is the intersection edge of the bottom plane and left plane.
The flowchart representing the “GOTOZero” function is illustrated in Figure 6-8. This function intends to create a lead-in for the wire path from an idle position to the initial cutting point or to create the lead-out for the wire path from the last cutting point to an idle position.

There are three possible scenarios for this wire path generation process:

**Scenario 1:** When the starting point of the cutting chain trajectory is formed by coverage of a polygon without concave edges, the last segment in the cutting chain will be extended to intersect with one of the six surfaces defined above. When the starting point of the cutting chain trajectory is not formed by coverage of a polygon without concave edges, the last segment on the cutting chain will also be extended to test any intersection with the six surfaces defined above or any existing cutting chain trajectory.

**Scenario 2:** If the intersection with the surfaces is closer than the intersection with the cutting chain trajectory, then the lead-in/lead-out wire path will follow the rules used for Scenario 1.

**Scenario 3:** If the intersection with the cutting chain trajectory is closer than the intersection with the surfaces, the lead-in/lead-out wire path will follow the intersected cutting chain trajectory.
Figure 6-9 illustrates an example of Scenario 1. The cutting chain “Chain 1” and “Chain 2” are on the UV plane. The extension of the first segment of “Chain 1” intersects with Right Plane on edge E1. The extension of the last segment of “Chain 1” intersects with Right Plane on edge E2. The extension of the first segment of “Chain 2” intersects with Left Plane on edge E3. The extension of the last segment of “Chain 2” intersects with Left Plane on edge E4. Consequently, the cutting chain sequence for “Chain 1” is: Top Right Edge → E1 → Chain 1 → E2 → Top Right Edge. The cutting chain sequence for “Chain 2” is: Top Left Edge → E3 → Chain 2 → E4 → Top Left Edge.
Figure 6-10 illustrates another example of Scenario 1. The cutting chain “Chain 1” is on UV plane. The extension of the first segment of “Chain 1” intersects with Bottom Plane on edge $E_1$. The extension of the last segment of “Chain 1” intersects with Bottom Plane on edge $E_2$. Consequently, the cutting chain sequence for “Chain 1” is: TopRight Edge $\rightarrow$ BottomRight Edge $\rightarrow$ $E_1$ $\rightarrow$ Chain 1 $\rightarrow$ $E_2$ $\rightarrow$ BottomLeft Edge $\rightarrow$ TopLeft Edge.
Figure 6-10 Cutting path example 2

Figure 6-102 illustrates an example of Scenario 2. The projection of a cutting chain, “Chain 1”, is on the UV plane. The projection of the first segment of “Chain 1” intersects with Top Plane on edge $E_1$. The projection of the last segment of “Chain 1” intersects with Top Plane on edge $E_2$. Consequently, the cutting chain sequence for “Chain 1” is: Top Right Edge $\rightarrow$ $E_1$ $\rightarrow$ Chain 1 $\rightarrow$ $E_2$ $\rightarrow$ Bottom Left Edge $\rightarrow$ Top Right Edge.
6.3 Cut off Plane

The last operation of wire cutting is to cut off the part. In order to accomplish this automatically, a cut off plane is required. In Chapter 5, intermediate coordinate systems are defined to translate tangent visibility results into manufacturing information. During the process of finding the initial rotational norm $R_N$, the cut off plane is the plane that is perpendicular to the initial $R_N$ and is a polygon without concave edges. Because the rotational norm is the initial $R_N$ and the cut off plane is a polygon without concave edges, the intermediate coordinate could be formed by initial $R_N$ and initial $R_\theta$. The cut off plane coverage is rectangle coverage.
Figure 6-12 illustrates the example of a cut off plane for a model pagoda. Because the cut off plane is obtained from a polygon without concave edges, initial $R_N$ and initial $R_O$ are used to build the intermediate coordinate system. The bottom part of the pagoda is perpendicular to the initial $R_N$ and is a polygon without concave edges; it is selected as the cut off plane, as illustrated in Figure 6-12. The cutting wire orientations will be exactly same with the vector $O_p = initial \; R_N \times initial \; R_O$.

![Figure 6-12 An example of cut off plane](image1)

![Figure 6-13 Cut off plane for the model pagoda](image2)
6.4 Summary

In this chapter, wire path generation methods are introduced. Cutting wire trajectories are generated under each intermediate coordinate system based on tangent visibility results. Those cutting trajectories are transformed into manufacturing coordinates. In order to accomplish the lead-in/lead-out wire path generation automation, an algorithm is developed to find paths for different types of tangent visibility results. A cut off plane is introduced to generate the cut off wire path as the final step of machine operation.
Chapter 7.

Implementation and Results

The general concept of using WEDM as a rapid manufacturing tool was introduced in Chapters 3 through 6. This chapter presents the implementation results based on the proposed method. Section 7.1 explains the software used to implement WEDM-RP, and Section 7.2 introduces the model verification system.

7.1 Software Implementation

The WEDM-RP algorithms were implemented in VC++.net 2005 and tested on an Inter® Core™2 Duo 2.60GHZ PC, running Windows XP. The software accepts ASCII STL files as input and outputs the Numerical Control (NC) code. The following are also considered as input parameters for the six-axis WEDM: taper angle, the distance between UV and XY planes, STL file accuracy level, maximum height and maximum width, and wire path incremental accuracy. Recall that the taper angle is the maximum angle to which the cutting wire can rotate from its neutral position. The STL file accuracy level is used to deal with round-up errors in the co-planar triangle combination procedure. The distance between the UV and XY planes, and the maximum height and maximum width of the six-axis WEDM are the parameters used to generate wire path.

Due to the requirement of a polygon-clipping operation in the tangent visibility algorithm, the polygon libraries in Computational Geometry Algorithms Library (CGAL) were used to improve computation speed. Figure 7-1 illustrates the software structure
for the WEDM-RP system. Seven major modules were implemented for this WEDM-RP. The file organization module was used to organize the input ACSII STL file into the data structure used in WEDM-RP. The AAG simplification module was used to simplify the input polyhedral geometry into planar polygons (see Chapter 4 for detailed information). The intersection graph module was used to calculate the intersection graph for each planar polygon in the input geometry (see Chapter 3 for detailed information on the intersection graph module). The tangent visibility module solved the global tangent visibility problem based on the intersection graphs generated from the intersection graph module. In this module, CGAL libraries were used to finish the polygon-clipping operations (see Chapter 3 for detailed information on the tangent visibility module). The manufacturing orientation module determines the optimal manufacturing orientation for the input geometry. The algorithms used in this module were provided in Chapter 5. The numerical control code generation module was discussed in Chapter 6. NC codes drive the WEDM machines to fabricate the final product. The visualization module was built to help visualize the intermediate and final result in the WEDM-RP system. This module generated Visual Basic for Application (VBA) codes, which were then executed in Solidwork® 2008 to generate drawings automatically. For example, the drawings for the intersection graphs and tangent visibility results were all generated automatically by the VBA codes produced by the WEDM-RP system. This visualization function is used to help the user understand and verify the algorithms’ output.
7.2 Examined Prototypes

Several CAD models were created and STL files generated to verify the algorithms in the WEDM-RP system. The tangent visibility algorithms were verified by a group of similar geometries shown in Figure 7-2. The testing polygons are highlighted in blue, the polygon drawings and the visible portion of the polygon drawings are also provided in Figure 7-2. For each polygon graph, the visible percentage was first calculated manually. The visible percentage was calculated by the visible polygon area divided by the original polygon area. The results were identical with the results generated from the tangent visibility algorithms.
Part orientation algorithms are used to determine the optimal setup orientations for the given geometry and calculate number of setup required to finish the final product. Figure 7-3 illustrates a slot part with its coordinate system. The part orientation algorithm determines that this slot part requires one setup and two rotational operations.
Recall that the number of rotational norms represents the number of setups, and under each rotational norm, the number of rotational orientations indicates the rotational axis movement (see Chapter 5 for detailed information). Figure 7-4 provides the setup orientation and rotational orientation results for the slot part. The slot part will be setup along $R_N(1)$, and the rotational axis will have two positions, $R_0(1)$ and $R_0(2)$ to finish all fabrication operations. Figure 7-5 illustrates the wire trajectory result for $R_N(1)R_0(1)$ coordinate. The wire trajectory information will be transformed into final manufacturing coordinates in NC generation module.

Figure 7-3 Slot part with its design coordinate system
Figure 7-4 Setup orientation and rotational orientations for the slot part

Figure 7-5 Wire trajectory under $R_N(1)R_D(1)$ coordinates

Figure 7-6 illustrates an inner feature part with its coordinate system. The part orientation algorithm determined that this part required one setup and two rotational operations. Figure 7-7 provides the setup orientation and rotational orientation results
for the inner feature part. The part was set up along $R_N(1)$, and the rotational axis had two positions, $R_O(1)$ and $R_O(2)$, whereby all the fabrication operations were finished.

Figure 7-6 Inner feature example with its design coordinate system
Figure 7-7 Setup orientation and rotational orientation for inner feature example

Figure 7-8 illustrates a model pagoda, a more complex part, with its coordinate system. The part orientation algorithm determined that this part required one setup and four rotational operations. Figure 7-9 provides the setup orientation and rotational orientation results for the pagoda. The pagoda was set up along \( R_N(1) \), and the rotational axis had four positions, \( R_o(1), R_o(2), R_o(3) \) and \( R_o(4) \), whereby all the fabrication operations were finished. Note that though the pagoda as designed along the X-axis in its design coordinate and the NC code generation required the part to rotate around the Y-axis, the WEDM-RP program found the correct design orientation and found the good setup orientation corresponding to the rotational orientations for fabricating the pagoda.
Figure 7-8 Model pagoda example with its design coordinate system

Figure 7-9 Setup orientation and rotational orientation for model pagoda example
Besides the prismatic parts shown in Figure 7-3, Figure 7-6 and Figure 7-8, several non-prismatic geometries are examined as well. Figure 7-10 illustrates a model hourglass. The global tangent visibility analysis shows that all triangles on the model hourglass are globally tangent visible. However, due to existence of the incorrectness in edge convexity in the input STL file, it requires 17 setups to finish manufacturing the whole hourglass model.

![Figure 7-10 A model hourglass](image)

Figure 7-11 (a) illustrates a complex model pagoda with curved surfaces. The global tangent visibility analysis shows that only two small triangle areas on the model pagoda are not globally tangent visible, as illustrated in Figure 7-11 (b). This tangent invisibility is also caused by the incorrectness in edge convexity in the input STL file. It requires 28 setups to finish manufacturing the whole pagoda model.
a) model pagoda b) two unreachable small triangles

Figure 7-11 A complex model pagoda with unreachable triangles

Table 7-1 shows the total computation time for different model geometries. Clearly, the prismatic geometries require less unit computation per facet than non-prismatic geometries. Because the prismatic geometries do not require the error elimination procedure to deal with incorrectness in edge convexity error in STL file (see Chapter 4 for detail), the unit computation time is much less than non-prismatic geometries.
Table 7-1 Computation results of five CAD models

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of facets</th>
<th>Computation Time</th>
<th>Unit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prismatic geometries</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The slot part</td>
<td>28</td>
<td>6 sec</td>
<td>0.2143 sec/facet</td>
</tr>
<tr>
<td>Inner feature example</td>
<td>44</td>
<td>13 sec</td>
<td>0.2955 sec/facet</td>
</tr>
<tr>
<td>Model pagoda</td>
<td>140</td>
<td>35 sec</td>
<td>0.2500 sec/facet</td>
</tr>
<tr>
<td><strong>Non-Prismatic geometries</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hourglass</td>
<td>228</td>
<td>694 sec</td>
<td>3.0439 sec/facet</td>
</tr>
<tr>
<td>Complex model pagoda</td>
<td>1382</td>
<td>2218 sec</td>
<td>2.8640 sec/facet</td>
</tr>
</tbody>
</table>

7.3 Model Verification System

The setup used for the model verification is illustrated in Figure 7-12. A four-axis hot wire foam cutter was used to simulate the UV axis and XY axis movement on the WEDM machine. A rotary device was added to the system in order to provide an additional axis of revolution to simulate the rotational axis on the six-axis WEDM.

![Figure 7-12 Four-axis foam cutter with a rotary device](image-url)
Due to the fact that the rotary device was manually controlled, the numerical control codes were generated for each $RNR_O$ combination. The verification system did not have the same manufacturing coordinate system as a six-axis WEDM machines. For the verification system, the rotational axis was around the $Y$-axis. The $X$-axis and $Z$-axis were used to control the wire movement on the four-axis foam cutter. For example, the model pagoda required one setup and four rotational orientations; therefore, a total of four NC code files were created for fabricating the model pagoda. A finished model pagoda is illustrated in Figure 7-13.

![Finished model pagoda](image)

**Figure 7-13 Finished model pagoda**

One of the perennial problems that prevent fully automatic fabrication process for wire EDM is the slug removal problem. Without human intervention, the slug generated during the fabrication procedure may fall onto the manufacturing part or even damage the cutting wire.
Slug removal problem is not considered in current research. During the fabrication process, the slug removal task is currently accomplished manually. Due to the fact that the current verification system employs styrene foam as cutting material, the light weight of the styrene foam will not cause damage on hot cutting wire and the final product. The slug removal problem is a good extension for current research.

7.4 Summary

This chapter presented the current implementation of the algorithms described in Chapters 3 through 6. The results of the tangent visibility algorithms, setup orientation algorithms, and numerical control code generation algorithms were used to create and compare some sample prototypes. In order to verify the correctness of this research, a four-axis foam cutter with a rotary device was built to simulate the six-axis WEDM machine operation. Several parts were fabricated on this verification system to demonstrate the correctness and capability of the proposed approach.
Chapter 8.
Conclusion, Contributions, and Future Research Directions

8.1 Research Contributions

In this research, a new method for using the six-axis WEDM as a rapid manufacturing process is developed and demonstrated. In order to make WEDM a viable alternative for rapid manufacturing applications, a system referred to as the WEDM-RP was created. This WEDM-RP system is capable of analyzing the global tangent visibility for input STL geometries and automatically generating the setup plan and wire path. This system makes it possible to create functional prototypes in a variety of electric conductive materials and can do so with virtually no manual process engineering.

The algorithms developed to analyze the global tangent visibility for polyhedral geometries in this research are not limited to the six-axis WEDM. It can be applied to other in-line cutting systems, such as laser, water jet, and hot wire foam cutter. Tangent visibility is different from traditional visibility; the former requires straight-line tangent accessibility analysis instead of point-to-point visibility analysis. This tangent visibility algorithm generates the fundamental information for automations in the tool path generation of in-line cutting systems.

Based on the global tangent visibility results, an algorithm to determine the number of setup orientations was developed. The setup planning for the rotational axis
on the six-axis WEDM machine was also calculated automatically. Complex geometries can be fabricated on the six-axis WEDM, yet due to the limited attention given to wire path generation for the six-axis WEDM, commercial software and WEDM manufacturing do not have the capacity to generate a production plan for WEDM with a rotational axis. This orientation algorithm makes it possible to generate setup plans for the six-axis WEDM machine automatically.

Furthermore, an automatic wire path generation procedure for the six-axis WEDM was successfully developed. This procedure can not only provide an automatically functioning wire path to cut the user input geometry, but it can also provide an automatic lead-in/lead-out wire path. This automation capability in wire path generation makes WEDM-RP a viable choice for rapid manufacturing applications.

8.2 Restrictions and Limitations

The method developed in this research is not capable of fabricating all models with respect to geometry complexity. Due to the inherent limitations of WEDM, geometries with concave shapes cannot be fabricated entirely on the WEDM machine. The WEDM-RP system can only generate a wire path for the tangent visible portion of a given geometry. An example of a prismatic part with a blind hole is shown in Figure 8-1. The blind hole cannot be fabricated on WEDM machines; the WEDM-RP's result indicated that the blind hole cannot be fabricated.
The other problem in WEDM-RP is the round-up errors generated from the STL file. Tangent visibility analysis results play the key role in the WEDM-RP system. The manufacturing orientation and wire path generation were all based on tangent visibility results. However, due to the round-up errors generated from the STL file, the correctness of the edge convex/concave/co-plane property currently cannot be guaranteed. Though a method to deal with the round-up errors was developed, several problems remain.

Due to its approximate nature, the STL file had edges and triangles that contained round-up errors. A fully convex geometry, such as the frustum of a cone
illustrated in Figure 8-2, should not contain any concave edges. However, the STL files generated by commercial software contain several concave edges (see Table 4-2). The procedure developed in this research can compensate for the round-up error; for example, some concave edges can be treated as co-plane edges. However, some other concave edges cannot be converted into co-plane edges automatically.

![Figure 8-2 A frustum of a cone](image)

Furthermore, the incorrectness of the edge convexity and the changes in the edge property may result in different tangent visibility results and different edge coverage methods. In Figure 8-3, triangles a, b, and c are co-plane triangles after some of the concave edges have been converted into co-plane edges. It is clear that edges $e_1$ and $e_2$, the two parallel edges, should be a feasible solution to cover the three triangle areas. However, the intersection graph obtained by extending the new co-plane surface makes this coverage infeasible. Due to the existence of some concave edges, edge $e_3$ is a feasible coverage edge instead of edge $e_2$; and $e_3$ and $e_1$ are not parallel to each other. This difference changed the rectangle coverage to triangle coverage. In addition, this
coverage change will affect the setup orientation determination and wire path generation as well.

Figure 8-3 Round-up errors cause an edge coverage problem

For example, the STL file for a model pagoda is illustrated in Figure 8-4. The user may determine that the pagoda requires a setup and several rotations to finish all the wire-cutting operations on a six-axis WEDM machine. However, due to the round-up error in the STL file and the coverage change result in the round-up error elimination procedure, the result generated from the WEDM-RP system may require more than one setup to fabricate the final pagoda part.
8.3 Future Research Directions

Although a system to automate the wire path and setup planning for WEDM and other in-line cutting systems was successfully created, and though the work outlined in this research was successfully demonstrated for a four-axis foam cutting operation with a manual rotary table, many enhancements for WEDM-RP are still possible. Specific directions for future work are discussed below.

8.3.1 Efficient Wire Path

During the tangent visibility analysis procedure, polygon-clipping operations were performed to combine visible polygon regions together. However, these polygon-
clipping operations may result in duplication in coverage for a certain area on the polygon. A more efficient polygon-clipping procedure is required to eliminate the duplication coverage in the tangent visibility result and thereby improve cutting efficiency. Furthermore, the wire path generated by WEDM-RP for the lead-in/lead-out procedure requires a long rapid travel distance from a certain neutral position to the real cutting spots; an improved algorithm decrease the lead-in/lead-out travel distance in order to decrease overall manufacturing time.

8.3.2 Edge Convexity Problem in the STL file

The edge convexity problem is the biggest challenge for WEDM-RP. The current round-up error elimination method can solve part of the problem, but may result in unwanted setup orientation and user interference. The WEDM-RP system requires polyhedral geometry as input, besides the STL file, other polyhedral geometry formats may be investigated as input geometry format. A polyhedral geometry format that has error-free edge convexity will make WEDM-RP more powerful when dealing with complex non-prismatic parts.

8.3.3 Integration of Tangent Visibility and Common Visibility

One of the advantages of in-line cutting systems is their capability to cut a big chunk of material in one tool pass. Instead of cutting the material restricted by the cutting-depth limitation for different cutting tools, all in-line cutting systems are capable of cutting material without being curtailed by a cutting-depth limitation. This advantage makes the in-line cutting system a great rough cutting process for removing more
material in less time. On the other hand, due to the inherent limitations of in-line cutting systems, the WEDM and the wire foam cutter cannot fabricate cavity geometries.

In my view, the combination of tangent visibility and common visibility analysis can generate a more efficient manufacturing approach. Based on tangent visibility analysis, the in-line cutting system will be used to remove more material tangentially around the part outline in less time. In addition, a common visibility analysis would be applied, and a point contact system, such as a milling process, would be used to finish the unreachable features and regions of the in-line cutting system. This hybrid manufacturing procedure would make subtractive processes more suitable and profitable for rapid manufacturing applications.

8.4 Conclusions

In conclusion, it is possible to use the six-axis WEDM or other in-line cutting system for rapid manufacturing applications. Further, the method developed in this research can eliminate the need for an expert process planner. The tangent visibility analysis can be performed for polyhedral geometries, and based on the tangent visibility result, the process planning for the six-axis WEDM can be automated. Moreover, the time necessary to create plans can be reduced significantly in comparison with traditional approaches.
References


Vita

Zhi Yang

Zhi Yang was born on November 19, 1978 in Jinan, Shandong, China. He received his Bachelor of Science degree with honors in Manufacturing Engineering from SouthEast University in China in 2001, and his Master of Science degree is in Mechatronics Engineering from SouthEast University in China in 2004. He gained a lot of working experience during his graduate study in China. He worked for Qidong Snapdragon Ceramics Co and helped them build a manufacturing information system in year 1999. He also joined the ERP research team and worked for Jiangsu Changjiang Electronics Technology Co, one of the largest public electronics company in China, from 2001 to 2003. In 2004, He was employed by Augmentum Inc. as a software engineer. He was enrolled in Ph.D program at Industrial and Manufacturing Engineering Department at Penn State in Fall of 2005. Besides working on his research, Zhi also participated in teaching practice. He was the teaching assistant for IE330, information technology for IE, and IE470, manufacturing system design and analysis. He received best teaching assistant award in 2009.