

# Surface Partitioning for 3+2-axis Machining

by

Armando Roman Flores

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Doctor of Philosophy  
in  
Mechanical Engineering

Waterloo, Ontario, Canada, 2007

©Armando Roman Flores, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## **Abstract**

Despite the inbuilt advantages offered by 5-axis machining, the manufacturing industry has not widely adopted this technology due to the high cost of machines and insufficient support from CAD/CAM systems. Companies are used to 3-axis machining and the operators are in many cases not yet ready for 5-axis machining in terms of training and programming. An effective solution for this 5-axis problem is a graduated migration through the use of 3+2-axis machining.

The objective of this research is to develop and implement a machining technique that uses the simplicity of 3-axis tool positioning and the flexibility of 5-axis tool orientation, to machine complex surfaces. This technique, 3+2-axis machining, divides a surface into patches and then machines each patch using a fixed tool orientation. The tool orientation and section boundaries are determined to minimize the overall machining time. For each section the tool orientation is different but remains constant while machining this section.

The number of patches selected for machining has a direct impact on the machining time. If the number of patches is small, the shape of the tool may vary greatly from that of the surface, which can result in smaller side-step distances. In contrast, a large number of patches leads to a better match between the tool and the workpiece, but it also leads to many re-orientations of the part as the tool moves between patches. Also, if the number of patches is large, the size of the patches will be reduced which will result in shorter tool passes that limit the tools ability to achieve the commanded feed rate. The optimum number of patches is a compromise between increasing the side step associated with large patches and the increase in time due to re-orientation of part and tool movement between patches. To find the optimal partition, a series of simulation tests are conducted to find the partition that would lead to the smallest machining time.

This work presents the application of well known methods from Pattern Recognition and newly developed methods by the current author that were adapted for surface machining and boundary

identification. This work also presents the methodology required to generate tool paths for 3+2-axis machining, which includes an explanation of the procedures required to determine an appropriate tool orientation, feed direction, tool path trajectory and tool parameters for patch-by-patch machining. These parameters are determined independently for each patch and aim at reducing the time required to machine a surface while maintaining the surface specifications.

This work presents the surface partitioning scheme and the method of selecting optimum number of partitions along with actual machining experiments. Machining tests on four different surfaces were conducted to demonstrate the efficiency of the proposed technique. The results show that 3+2-axis machine reduced machining times over 3-axis ball nose machining and 5-axis machining using the “*Sturz*” method. Also, since the tool axis remains fixed during cutting, the tool offers constant feed rates and a better surface finish compared to simultaneous 5-axis.

## **Acknowledgements**

I want to thank my family for all their support. I cannot imagine having gone through it without them.

I want to express my gratitude to my supervisors, Dr. Bedi and Dr. Ismail. Thanks for having faith in me, and for all your support and guidance through all these years.

I want to thank Dr. Mann for reviewing my thesis. I would also want to thank Dr. Glinka, Dr. Huissoon and Dr. Dong for agreeing to be on my thesis committee.

I thank Robert Wagner for helping me conduct the machining tests.

Funding for the research described in this thesis was provided by the Consejo Nacional de Ciencia y Tecnologia (CONACYT).

*To Maria,*

# Table of Contents

<b>Abstract</b> .....	<b>iii</b>
<b>Acknowledgements</b> .....	<b>v</b>
<b>Table of Contents</b> .....	<b>vii</b>
<b>List of Figures</b> .....	<b>x</b>
<b>List of Tables</b> .....	<b>xiii</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
1.1 Research goals.....	3
1.2 Proposed strategy.....	5
1.3 Thesis layout.....	6
<b>Chapter 2 Multi-axis machining</b> .....	<b>8</b>
2.1 3-axis machining.....	8
2.2 High-speed machining.....	10
2.3 5-axis machining.....	11
2.3.1 Tool positioning.....	12
2.3.2 Rolling Ball Method (RBM).....	13
2.3.3 Tool path planning.....	15
2.3.4 Gouging.....	17
2.4 3+2-axis machining.....	18
2.5 Summary.....	21
<b>Chapter 3 Surface partitioning</b> .....	<b>24</b>
3.1 Test surfaces.....	25
3.2 Clustering algorithms.....	28
3.2.1 k-means clustering algorithm.....	28
3.2.2 Fuzzy c-means.....	32
3.2.3 Hierarchical clustering.....	33
3.2.4 Clustering algorithms for surface partitioning.....	34
3.3 Feature extraction.....	36
3.3.1 Proximity group.....	37
3.3.2 Orientation group.....	37

3.3.3 Shape group .....	38
3.4 Feature selection.....	41
3.4.1 Evaluation of features.....	42
3.5 Feature vector .....	46
3.6 Side-step subdivision method.....	54
3.7 Summary .....	61
<b>Chapter 4 Patch boundaries.....</b>	<b>63</b>
4.1 Patch representation.....	65
4.1.1 Entire group of sample points.....	67
4.1.2 Corner points .....	67
4.1.3 Boundary points.....	68
4.1.4 Random selection of points .....	69
4.1.5 Clusters method.....	69
4.1.6 Statistical methods.....	70
4.1.7 Results of patch representation.....	72
4.2 Classification distances .....	73
4.3 Classification methods.....	74
4.3.1 Voronoi diagrams .....	74
4.3.2 <i>k</i> -Nearest Neighbour algorithm ( <i>k</i> -NN).....	75
4.3.3 MICD.....	77
4.4 Implementation of classification methods.....	79
4.4.1 Nearest neighbour algorithm .....	80
4.4.2 MICD method.....	85
4.5 Comparison between the nearest neighbour and the MICD .....	87
4.6 Summary .....	90
<b>Chapter 5 Tool path generation .....</b>	<b>91</b>
5.1 Projected normals plane .....	91
5.2 Effective radius of the tool .....	99
5.3 Feed direction .....	100
5.3.1 Eigenvector Method .....	101

5.3.2 Exhaustive Method.....	102
5.4 Tool orientation .....	103
5.4.1 Tool positioning.....	106
5.5 Tool parameters.....	106
5.5.1 Types of tools .....	106
5.5.2 Tool size .....	107
5.6 Side-step distance .....	109
5.7 Types of tool paths .....	111
5.8 Summary .....	112
<b>Chapter 6 Implementation of the patch-by-patch 3+2-axis machining method .....</b>	<b>113</b>
6.1 Sampling points.....	115
6.2 Identify common shapes.....	118
6.2.1 Hybrid Methods.....	119
6.2.2 Experiments.....	122
6.3 Surface partitioning .....	124
6.4 Patch boundaries.....	126
6.5 Tool path generation.....	128
6.6 Number of patches.....	129
6.7 Machining time.....	130
6.8 Boundary Marks .....	132
6.9 Summary .....	132
<b>Chapter 7 Machining tests.....</b>	<b>134</b>
7.1 Surface 1.....	137
7.2 Surface 2.....	141
7.3 Surface 3.....	144
7.4 Surface 4.....	147
7.5 Summary .....	151
<b>Chapter 8 Conclusion.....</b>	<b>153</b>
8.1 Future work .....	155

## List of Figures

Figure 2-1 Scallops left from machining.....	10
Figure 2-2 5-axis tilt-rotary table milling machine .....	12
Figure 2-3 Cutting profile of a ball nose cutter and an inclined toroidal cutter .....	13
Figure 2-4 Positioning the tool inside the sphere .....	15
Figure 2-5 Positioning the tool outside the sphere .....	15
Figure 2-6 Broken passes resulting from variable side-step distances .....	16
Figure 2-7 Tool orientation determined by the mean surface normal .....	19
Figure 3-1 Test surfaces .....	27
Figure 3-2 Clustering data and initial cluster centers .....	30
Figure 3-3 1 <sup>st</sup> . iteration .....	30
Figure 3-4 3 <sup>rd</sup> . iteration.....	31
Figure 3-5 10 <sup>th</sup> . Iteration.....	31
Figure 3-6 Dendrogram .....	34
Figure 3-7 Time comparison using multi-dimensional data.....	36
Figure 3-8 Shadow area.....	39
Figure 3-9 Pseudo-curvature circle .....	40
Figure 3-10 Complex shapes .....	43
Figure 3-11 Surface partitioning process .....	47
Figure 3-12 Half-toroidal surface divided into four patches .....	48
Figure 3-13 Side-step distance (cm) for surface 1.....	55
Figure 3-14 Side-step distance .....	55
Figure 3-15 Side-step subdivision method .....	57
Figure 4-1 Boundary identification .....	64
Figure 4-2 Sample surface partitioned into 4 patches .....	66
Figure 4-3 Corner points selected by the user .....	67
Figure 4-4 Patch representation using boundary points .....	68
Figure 4-5 Patch representation using random selection of points.....	69
Figure 4-6 Patch representation using the clusters method .....	70
Figure 4-7 Patch representation using statistical tools .....	71
Figure 4-8 Parameters of the ellipse .....	72
Figure 4-9 Voronoi Diagrams .....	75

Figure 4-10 Sample points partitioned into two classes .....	77
Figure 4-11 Test points (*) are classified into their corresponding classes using $k=1$ .....	77
Figure 4-12 Minimum Intra-Class Distance (MICD) method.....	79
Figure 4-13 Classification process using the nearest neighbour method .....	81
Figure 4-14 Surface patches used for tests. a) Patch 1. b) Patch 2.....	84
Figure 4-15 Classification process using the Minimum Intra-Class Distance (MICD) method .....	86
Figure 5-1 Projected normals plane.....	93
Figure 5-2 Spherical coordinates.....	94
Figure 5-3 Projected Normals Plane using spherical coordinates (degrees) .....	95
Figure 5-4 Projected normals plane for tool path generation .....	97
Figure 5-5 Projected normals plane for Surface Partitioning .....	98
Figure 5-6 Inclined toroidal endmill and its projected effective radius.....	99
Figure 5-7 Comparison of the effective radius of a toroidal and a ball nose cutter.....	100
Figure 5-8 Axes of the ellipse determine feed direction .....	102
Figure 5-9 Exhaustive Method .....	103
Figure 5-10 Tool axis inclination with respect to $N_{max}$ .....	104
Figure 5-11 Tool orientation to prevent gouging .....	104
Figure 5-12 Tool orientation .....	106
Figure 5-13 Cutting tools.....	107
Figure 5-14 Test to determine the tool size .....	108
Figure 5-15 Tool's inclination angle .....	109
Figure 5-16 Side-step distance comparison.....	110
Figure 5-17 Types of tool paths.....	111
Figure 6-1 Process diagram for 3+2-axis machining.....	114
Figure 6-2 Sequential classification process .....	120
Figure 6-3 Neural Network structure for feature recognition.....	121
Figure 6-4 Training data.....	121
Figure 6-5 Classification results.....	123
Figure 6-6 Surface partitioning process .....	124
Figure 6-7 Complicated shapes .....	127
Figure 6-8 Classification of cutter contact points.....	127
Figure 6-9 Boundary marks.....	132

Figure 7-1 Deckel Maho 5-axis machining center .....	135
Figure 7-2 Test surfaces .....	135
Figure 7-3 Surface properties of Surface 1.....	138
Figure 7-4 Machining parameters for surface 1 .....	139
Figure 7-5 Cutting test surface 1 .....	140
Figure 7-6 Surface properties of Surface 2.....	142
Figure 7-7 Machining parameters for surface 2 .....	143
Figure 7-8 Cutting test surface 2 .....	143
Figure 7-9 Surface properties of Surface 3.....	145
Figure 7-10 Machining parameters for surface 3 .....	146
Figure 7-11 Cutting test surface 3 .....	146
Figure 7-12 Surface properties of Surface 4.....	148
Figure 7-13 Machining parameters for surface 4 .....	149
Figure 7-14 Cutting test surface 4 .....	150

## List of Tables

Table 2-1 Comparison of multi-axis machining methods .....	22
Table 3-1 Control points for surface 1 .....	26
Table 3-2 Control points for surface 2 .....	26
Table 3-3 Control points for surface 3 .....	26
Table 3-4 Hierarchical clustering .....	34
Table 3-5 Partitioning tests.....	44
Table 3-6 Partitioning tests.....	45
Table 3-7 Partitioning results for surface 1 .....	50
Table 3-8 Partitioning results for surface 2 .....	51
Table 3-9 Partitioning results for surface 3 .....	52
Table 3-10 Partitioning results for surface 4 .....	53
Table 3-11 Side-step subdivision method applied to surface 1 .....	58
Table 3-12 Side-step subdivision method applied to surface 4 .....	58
Table 3-13 Surface partitioning using the modified side-step method.....	60
Table 3-14 Number of unsuitable patches (12 tests in total).....	62
Table 4-1 List of methods applied for the identification of boundaries .....	65
Table 4-2 Sample labeled matrix.....	66
Table 4-3 Comparison of methods to represent the surface patches .....	73
Table 4-4 Comparison using different percentages of points on sample patch 1 .....	83
Table 4-5 Comparison using different percentages of points on sample patch 2.....	83
Table 4-6 Accuracy of the two methods used for classification.....	85
Table 4-7 Comparison of different classification methods .....	88
Table 6-1 Sample size tests for Surface 1.....	116
Table 6-2 Sample size tests for Surface 4.....	116
Table 6-3 Comparison of the effectiveness of classification for patch 1 .....	118
Table 6-4 Comparison of the effectiveness of classification for patch 2 .....	118
Table 6-5 Tool path length comparison (mm).....	125
Table 6-6 Tool path length comparison (mm).....	128
Table 6-7 Tool path length comparison (mm).....	129
Table 6-8 Estimated and machining time comparison .....	131
Table 6-9 Time test using maximum Feed Rate.....	131

Table 7-1 Numerical simulation tests for Surface 1 .....	139
Table 7-2 Machining time comparison for Surface 1 .....	140
Table 7-3 Numerical simulation tests for Surface 2 .....	142
Table 7-4 Machining time comparison for Surface 2 .....	144
Table 7-5 Numerical simulation tests for Surface 3 .....	145
Table 7-6 Machining time comparison for Surface 3 .....	147
Table 7-7 Numerical simulation tests for Surface 4 .....	149
Table 7-8 Machining time comparison for Surface 4 .....	150

# Chapter 1

## Introduction

Complex surfaces are conventionally machined on 3-axis milling machines. These machines owe their popularity to their relative low cost and simplicity of tool path generation. Machining is usually performed with ball nose cutters because tool positions and side-step distances are easy to determine. However, 3-axis machining generally requires longer machining times because the width of the strip machined by a tool is typically smaller than those obtained with 5-axis machines. Thus, newer technologies aimed at improving machining efficiency, beyond the capabilities offered by regular 3-axis machining methods are still needed.

Compared to 3-axis machines where the tool orientation is fixed, 5-axis machines use two rotary axes to dynamically change the tool orientation during machining. 5-axis machines use simultaneous movements to change the tool or the workpiece orientation. Dynamic adjustment of the tool orientation allows a better match of the tools's geometry to the part surface, which results in a larger machined side-step between passes, fewer tool passes, and a shorter tool path length over 3-axis machining [1].

The reduction in tool path length typically leads one to think that 5-axis machining would result in reduced machining time compared to 3-axis machining. In reality, this is not always the case, because the rotary axes cannot turn fast enough to keep up with linear axes and thus slow down the actual feed rate of the tool. Gray [2] noted that saturation of the rotary axes servo drives frequently results in slower than programmed feed rates and longer than expected machining times. Furthermore, the slow down is further accentuated by the singularity point associated with the kinematics of these machines [3], where a small change in the tool axis direction can result in large rotations of the rotary axes.

There are several factors that lead industry to hesitate to switch to 5-axis machining. Although shape-matching-tool-positioning methods in five-axis machining have shown considerable reductions in tool path length [4], these methods are relatively new and have not been adopted by CAM companies within their software. The application of 5-axis machining in industry is difficult because of limited software support, the complexity in programming and the high cost of investment and training. The kinematic complexity of 5-axis machines introduces many questions regarding accuracy and accessibility for tool positioning and adds to the concerns of industry [5].

An effective solution for the 5-axis problem is a graduated migration through the use of 3+2-axis machining. 3+2-axis machining is a practical and economical alternative that addresses some of the issues encountered in 5-axis machining. In 3+2-axis machining the tool orientation is changed in discrete steps, which facilitates the transition from 3-axis machining. Since the method uses the 3-axis method for tool positioning, but uses the 5-axis capability to orient the tool, it will be easier for the operators to develop the necessary knowledge for an eventual migration to simultaneous 5-axis machining.

In 3+2-axis machining the tool orientation is selected to match the tool geometry to a region of the part. Each region is machined using a different tool orientation that remains locked during cutting. The fixed tool orientation provides more rigidity that allows operating at higher cutting speeds without loss of accuracy or fluctuations in feed rate. Also, it allows predicting more easily the trajectory of the tool and checking for gouging.

3+2-axis machining strategies can be performed on 3-axis machines with the addition of a rotary/tilt table or on indexable 5-axis machines. In reality, the machining is conducted in three continuous axes. The tilting and rotation are conducted in discrete steps that are independent of the

other axes, thus they are labelled as  $\frac{1}{2}$ - axes. As such, the proper designation for this method should be  $3\frac{1}{2}\frac{1}{2}$ -axis machining. For brevity, however, it is denoted in this work as 3+2-axis machining.

Many industries have invested in machines that can change the tool orientation in discrete steps either manually or automatically. Such machines are commonly used for five sided machining where orientation facilitates accessibility. The various sides of the part can be machined in one setup. Such machines are much less expensive than simultaneous 5-axis machines and do not require excessive training because the tool trajectory on these machines is calculated using three-axis methods and software, which are well known on the shop floor. The availability of these machines provides the motivation of this work; to combine the flexibility of orientation offered by true 5-axis machines, while maintaining low cost and ease of programming offered by 3-axis machines.

The efficiency of 3+2-axis machining can approach that of 5-axis machining but it requires special considerations to be carried out successfully. 3+2-axis machining has inherent limitations as the tool orientation is determined using discrete rotations of two additional axes. If the surface cannot be machined using a single tool orientation, the surface has to be machined in sections. Commonly, the partitioning process and the determination of the angles of orientation depend on the skills and intuition of the operator, which may be prone to errors and variability. The lack of guidelines to determine these two parameters impedes producing consistent results and correctly assessing the reliability of the process. As long as these intuition based procedures remain, the problems of 3+2-axis machining will remain unsolved.

## **1.1 Research goals**

The purpose of this research is to develop an efficient method for 3+2-axis machining that can be competitive with current machining methods. The goal of the current research is to expand and

improve the current methodologies developed for 3+2-axis machining. In particular, the objectives of this work are

1. To conduct a complete clustering analysis that will define the criteria for surface partitioning required for patch-by-patch surface machining.
2. To conduct studies that will help identify the best selection of surface properties for the clustering algorithm. The goal is to provide the user with an approach to selecting the parameters that incorporate relevant properties that can be used automatically to partition a surface with consistently good results.
3. To design an algorithm capable of identifying the boundaries of the surface patches. This method should be able to work with different types of surfaces and allow the partitioning algorithm to be applied to unevenly spaced data as may be produced from 3D scanning or triangulated surface data.
4. To develop a robust methodology to generate tool paths for patch-by-patch surface machining. An improved technique for determining the tool inclination that avoids gouging and improves machining times will be investigated. Effects on varying the direction of cut and tool path trajectories to reduce machining times will be investigated. A proper method to obtain more accurate calculations of the side-step distance will be developed.
5. To develop a method to identify common shapes such as planes, fillets and other form surfaces that can reduce the data that needs to be processed.
6. To develop an automatic method capable of conducting surface partitioning and tool path generation for patch-by-patch machining. The final goal of this work is to simplify the process of generating tool paths and reduce the dependence on human input.

7. To conduct machining tests and comparisons with other machining strategies to demonstrate the efficiency of the proposed 3+2-axis machining methodology.

The proposed strategy to address the above objectives is discussed in detail next.

## **1.2 Proposed strategy**

A method capable of guiding the CNC-operator through the process of making better tool paths for 3+2-axis machining is proposed in this work. The proposed methodology is based on the division of the surface into patches. Partitioning the surface offers the operator the opportunity to visualize individual patches as distinct entities and to generate a tool path based on the characteristics of the patch. Each patch can be machined independently using an appropriate set-up that includes a particular tool orientation, feed direction and side-step distance.

3+2-axis machining combines the flexibility of orientation offered by 5-axis machines with the ease of programming offered by 3-axis machines. Both 5-axis and 3-axis machining offer advantages. In 3-axis machining higher feed rate can be achieved and in 5-axis machining a wider side-step can be realized because of the better match between the tool and the surface shapes. The proposed strategy is designed to take advantage of these combined traits by dividing the surface into patches and then determining a proper tool orientation for each patch, followed by machining each patch with a fixed tool orientation.

Since 3+2-axis machining requires optimizing the tool orientation for a region, it is necessary to develop a surface partitioning method to identify those regions where the surface properties within the patch do not vary significantly. Complex surfaces usually have irregular curvature distribution that can cause difficulty in machining. These surfaces can be subdivided into smaller connected patches that share similar surface properties. The partitioning strategy developed in this work must

guarantee that within a patch the shape of the surface does not vary greatly from the shape of the tool in a particular orientation.

3+2-axis machining requires determining a tool positioning strategy that can be appropriate for the entire patch. It is known that each contact point is best machined when the tool is in a specific orientation. In 3-axis machining, the tool orientation does not change. Consequently, the tool machines each point in an inefficient way. In 3+2-axis machining the tool orientation is optimal at least in one point in each patch. Since the variation in surface properties inside a patch is not large, the shape of the tool can closely match the geometry of the majority of the points inside the patch.

The patch-by-patch machining method introduces additional tool travel when the tool has to move from one patch to another and requires workpiece re-orientation which can take some time. If the number of patches is large, the overhead due to the movement between patches and due to re-orientation can be larger than the gains of the method. Alternatively, if the number of patches is small, the benefit of the method is not fully realized since the shape of the tool may vary greatly from that of the surface. Accordingly, a technique for selecting the optimum number of patches is required and one such method is presented in this work.

Current CAD/CAM systems provide limited support to optimize the procedures in 3+2-axis machining. In general, these systems require the user to make decisions based on prior experience, general rules of thumb and subjective judgement. Considering that this approach does not guarantee the efficiency and reliability of the decisions, it is desirable to have tools to support planning the machining configuration. Such tools are proposed in the current work.

### **1.3 Thesis layout**

This thesis is divided into eight chapters.

In Chapter 1, a general introduction is given mainly to highlight the need for 3+2-axis machining and outlines the objectives of this thesis.

In Chapter 2, an examination of various strategies and procedures related to multi-axis surface machining are presented. A literature survey is presented to highlight the necessity for a robust and efficient method for 3+2-axis machining.

In Chapter 3, a study of the parameters that effect surface partitioning is conducted. A study of different clustering algorithms is presented. Such a comprehensive study has not been done by other researchers working in the field.

In Chapter 4, an evaluation of different methods to determine the patch boundaries is presented. A series of graphical and numerical tests are included to validate the proposed methodology.

In Chapter 5, the methodology to generate tool paths is presented. This chapter includes an explanation of the procedures required to determine an appropriate tool orientation, feed direction, tool path trajectory and tool parameters for patch-by-patch machining.

In Chapter 6, the proposed methodology for 3+2-axis machining is presented in detail. The procedures required to machine a complex surface are explained. Machining tests will be presented to validate the numerical results.

In Chapter 7, a comparison between the proposed 3+2-axis machining methodology and other multi-axis machining strategies is presented.

In Chapter 8, final conclusions and considerations about further developments to this work are presented.

## Chapter 2

### Multi-axis machining

Complex surfaces are commonly used in the design of aeronautical, automotive and consumer goods. These surfaces are made by machining the components directly or by making moulds and dies to create them. Either way, machining plays an important role in realizing curved surfaces in engineering components. Increased global competition in the manufacturing industry has forced companies to substantially increase their productivity, thus creating a demand for more efficient machining methods [5]. The application of advanced technologies such as 5-axis and high-speed machining increase the chances to meet these demands. However, these technologies require the development of economical and reliable strategies that result in efficient implementation and increased productivity.

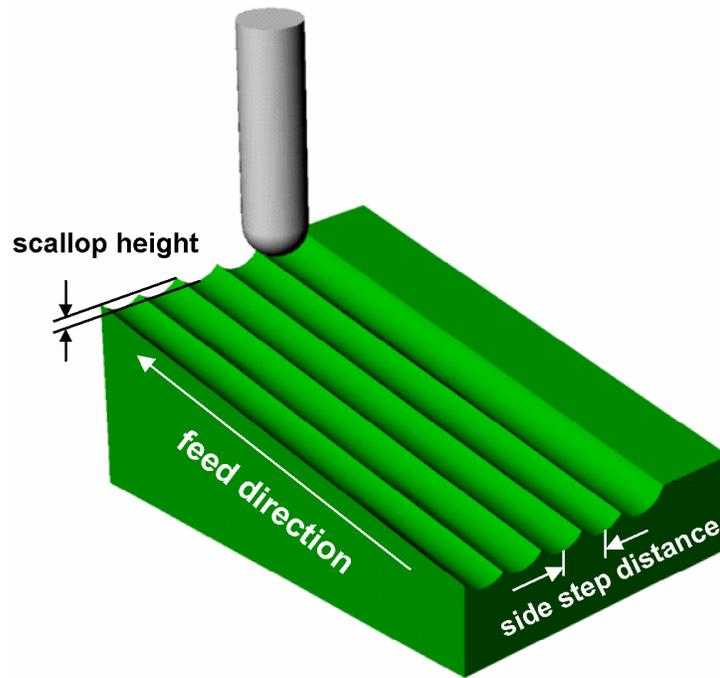
#### 2.1 3-axis machining

Complex surfaces are conventionally machined with 3-axis milling machines, where the tool orientation is fixed typically in the vertical or horizontal direction and the tool is moved along pre-planned trajectories to shape the stock into the desired surface. Positioning and determining the trajectory of the tool requires models of the surface and of the tool and can be determined with relative ease. The trajectory of the tool is determined by offsetting the design surface. The tool center moves from point to point along curves that lie on the offset surface. The spacing between the curves is called side-step, while the spacing between subsequent points along the pass is called feed forward step. The side-step determines the number of passes required to machine a surface. The smaller the side-step is, the larger the machining time will be. Typically, ball nosed endmill cutters are used for machining complex surfaces or curved elements in a workpiece. A ball nose endmill will usually not

gouge a surface, provided that the radius of the ball is smaller than the minimum radius of curvature of the surface [2].

In 3-axis machining, material is left between tool passes in the form of scallops because the tool geometry does not exactly match the surface geometry, as shown in Figure 2-1. Low scallop height requirements in 3-axis machining may require long machining times as demonstrated by Cho *et al.* [6]. The side-step distance is often defined by finding the shortest distance between passes so that the largest scallop height is equal to a user-specified tolerance.

3-axis machining is carried out using simple algorithms to compute the tool paths and simulate the surface resulting from a tool path. However, 3-axis machining has some limitations related to the fixed orientation of the cutter during machining. The entire surface may not be accessible to the cutter in one setup. Furthermore, the use of radiused corner endmills is still restrictive, due to the absence of efficient tool position strategies in commercial CAM packages [7]. The limitations of 3-axis machining have opened new areas for research and development in surface machining to the development of flexible and efficient machining methods that implement new strategies.



**Figure 2-1** Scallops left from machining

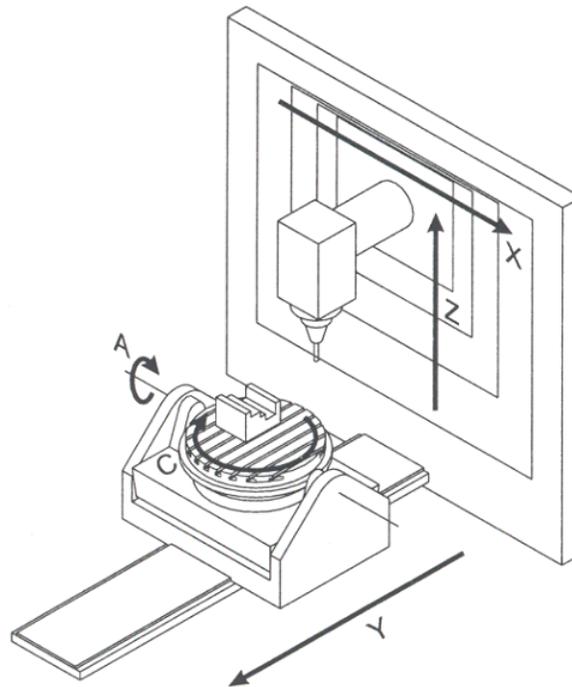
## **2.2 High-speed machining**

The classical definition of High-speed machining as described by Shmoll [8] involves a cutting speed that is five times or more than conventional machining methods. Compared to traditional 3-axis machining, the volume of material removed per pass is much less but the use of higher cutting speeds permits higher feed rates that result in increased material removal rates. Smaller side-step distances result in lower cutting forces, and smaller scallops, which reduces the time needed for a subsequent polishing process. However, machine operators need to be trained for different conditions during machining that require determining an appropriate cutting tool and feasible cutting parameters according to the material and workpiece specifications as shown by Kaldos *et al.* [9]. King and Vaughn [10] demonstrated that as the cutting speed increases above the conventional speed range, new dynamic effects are encountered in the machining process. Programming requires special

considerations to prevent high cutting depths that could result in tool breakage. Also, the use of ball nose cutters requires developing appropriate methods to prevent machining with the bottom of the tool where the cutting speed is zero.

### **2.3 5-axis machining**

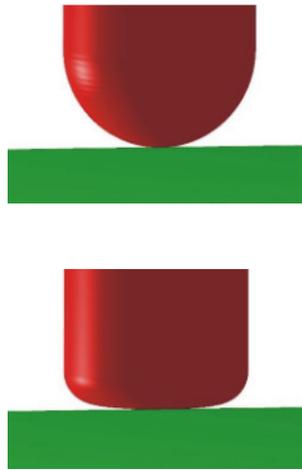
Five-axis machines change the tool or the workpiece orientation by using two additional axes (*A* and *C*). While there are many 5-axis machine configurations, the one used in this work is shown in Figure 2-2; however, the results should be generalizable to other configurations. In 5-axis machining the tool is oriented to match its shape to the geometry of the surface being machined. Sheltami *et al.* [4] showed reductions of up to 80% in tool path length for machining complex surfaces compared to 3-axis machining. Though two additional axes have brought more flexibility and accessibility of the cutter, 5-axis tool paths are difficult to program. The best use of 5-axis machines requires strategies that guarantee appropriate gouge-free tool positioning at every point along the tool path.



**Figure 2-2 5-axis tilt-rotary table milling machine**

### **2.3.1 Tool positioning**

Most 5-axis tool positioning methods are designed to maximize the effective radius of the tool. The effective radius is defined as the radius of curvature of the tool at the point of contact. Tilting a tool by an inclination angle results in a larger effective radius, this provides a wider machined strip width around the contact point and thus results in fewer tool passes and a shorter tool path length. An inclined toroidal endmill can be used to machine a surface as effectively as a much larger ball endmill [11], as shown in Figure 2-3, where the cutting profile of two cutters of the same radius is compared.



**Figure 2-3 Cutting profile of a ball nose cutter and an inclined toroidal cutter**

Tool positioning strategies aim at determining optimal tool orientations for the cutter at the contact point. A simple tool positioning strategy is the “*Sturz*” method, where a user-selected angle is used to incline the tool axis with respect to the surface normal at the cutter contact point. The difficulty of this method is determining the angle of inclination and the adjustments necessary to prevent gouging. Rao *et al.* [12] [13] developed the Principle Axis Method, a method for tool positioning where the tool is inclined to match the minimum radius of curvature at the contact points by inclining the tool about the maximum curvature direction. This method does not guarantee gouge-free tool paths. Warkentin *et al.* [11] proposed the Multi-Point Machining Method. This tool positioning method orients the tool such that it contacts the surface at more than one point simultaneously. The tool is forced to maintain tangential contact at the first point and then is rotated about two independent axes until an optimal position is achieved. However, this approach is mathematically complex and difficult to implement for general surfaces.

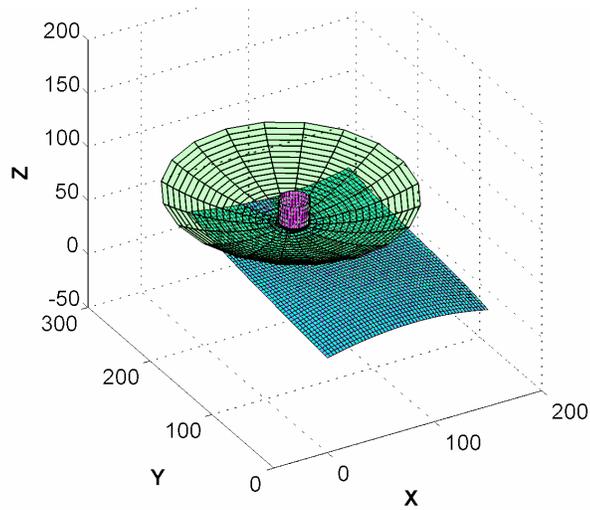
### **2.3.2 Rolling Ball Method (RBM)**

Most methods of tool positioning use the properties of the surface at the contact point to determine tool trajectories. However, a tool is not a point entity and occupies a volume. Thus, tool positioning

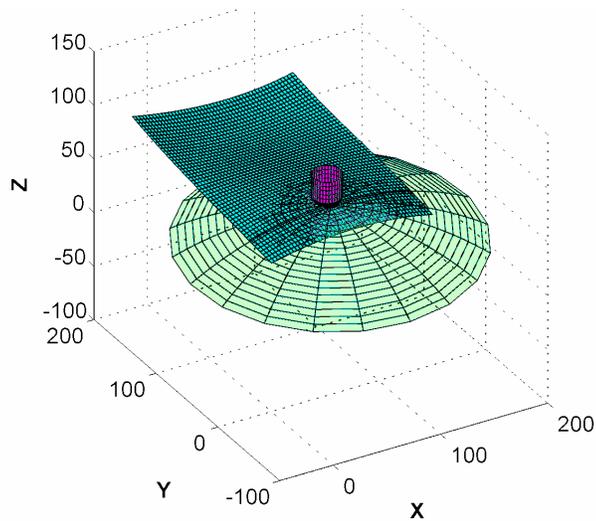
cannot be considered just as a local issue. Both the tool surface and part surface must be considered for optimal tool positioning, as demonstrated by Warkentin *et al.* [11]. The method developed by Warkentin *et al.*, called the Multi Point Machining (MPM), exploited the regional nature of tool position to find multiple contact points. However, the MPM is too complex. To address this issue, the Rolling Ball Radius method was developed to take into account additional information for each contact point and evolved into a more robust method for tool positioning.

The Rolling Ball Radius concept developed by Gray *et al.* [14] [15] is based on positioning a spherical ball at the point of interest and inflating it until it touches another point on the surface. This is the largest ball that can sit on that point. The method is closely related to MPM and results in a variable radius of the ball at points along the tool path. The ball approximates a portion of the surface in the vicinity of the cutting tool. This ball encapsulates the tool and is used to determine its position and orientation at each point of the tool path.

Warkentin *et al.* [11] showed that any cylindrical tool, be it toroidal or flat-ended can be positioned inside a sphere such that it forms a circular line of contact. This concept is used in the Rolling Ball Method (RBM). Since the sphere is positioned such that it does not gouge the surface. Positioning the tool inside it guarantees a gouge free tool position. However, there are infinite ways to position the tool in the sphere. Since there is only one contact point between the sphere and the surface, it is set to be the cutter contact point so the cutting tool is ensured to contact the surface at one point. Tool position also depends on the sign of the curvature. For a positive curvature the tool is positioned inside the ball and for a negative curvature the tool is placed outside the sphere, as shown in Figure 2-4 and Figure 2-5, respectively.



**Figure 2-4 Positioning the tool inside the sphere**

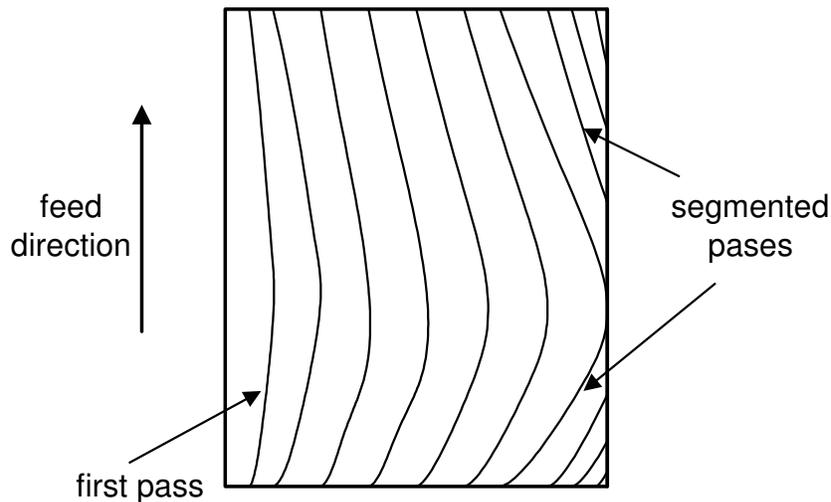


**Figure 2-5 Positioning the tool outside the sphere**

### 2.3.3 Tool path planning

Optimal tool path planning aims at reducing the time required to completely machine a surface to the user-specified tolerance. Tool paths are usually generated using parallel passes and the distance between tool passes is kept constant over the entire surface. Iso-parametric tool paths can be

generated by holding one surface parameter constant and traversing the range of the other as shown by Boomhead and Edkins [16]. Examples of non-parametric tool paths can be based on parallel Cartesian planes as demonstrated by Li and Gerard [17] or Huang and Oliver [18]. For low curvature surfaces, constant side-step distances may produce consistent scallop height. However, in some cases selecting a constant side-step for the entire surface may result in redundant machining between two adjacent passes. To address this problem, Jensen and Anderson [19] and Pi *et al.* [20] calculated a variable side-step for every pair of adjacent passes that is controlled by the largest scallop height produced between the two passes. A constant cusp height method was also proposed by Lee [21]. This method varies the side-step at every point to maintain a constant scallop height along the tool pass. Even though this method can reduce redundant machining, it still lacks efficiency due to the dependency on previous passes which often result in broken tool paths as shown in Figure 2-6.



**Figure 2-6 Broken passes resulting from variable side-step distances**

Several researchers have focused on determining an appropriate feed direction that gives the widest side-step at any point along the tool path. Rao *et al.* [13] and Lauwers *et al.* [22] addressed this problem by calculating the feed direction along the minimum curvature direction. Kim and Sarma [23] developed an approach that seeks the directions of maximum sweep rate using a greedy direction

field. Chiou and Lee [24] constructed a Machining Potential Field that considers the surface geometry, the cutter geometry and the tool orientation, and generated a tool path that uses an iterative searching algorithm. In general, these methods have different feed directions for every point, which complicates the generation of tool paths.

### **2.3.4 Gouging**

Preventing gouging is a critical problem in 5-axis surface machining. Gouging results from any type of overcut caused by non-tangential contact of the tool to the surface. Tool positioning must consider the area surrounding the contact point in the shadow of the tool. Gouge detection is usually performed after the tool path is generated, and requires an iterative and time-consuming process to correct tool positions until the tool path is gouge free. This approach, found in some CAM systems, requires validating the corrected tool positions and generating a smooth transition for the modified tool path [24]. Also, gouging can be detected during tool positioning using geometric properties of the surface around the cutter contact point. Redonnet *et al.* [25] and Lauwers *et al.* [26] developed similar strategies that involve adjusting the original tool position until no gouging is detected. Lee [27] extended the method to identify gouging by the back side of the tool. Space-search methods identify a region in the vicinity of the contact point that is gouge-free. Jun *et al.* [28] proposed a space-searching method that generates a set of feasible tool orientations, which can be used to determine an optimal tool orientation that minimizes the machined surface error. Gray *et al.* [14] [15] developed a graphical technique based on the rolling ball positioned at the point being machined, as discussed earlier. In this method, a ball is inflated until it touches another point on the surface and provides the information to generate gouge-free tool positions and orientations.

The criterion for an optimal tool position is to minimize the machined surface errors and to maximize material removal rate. Compared to 3-axis machining, the simultaneous motion of the tool

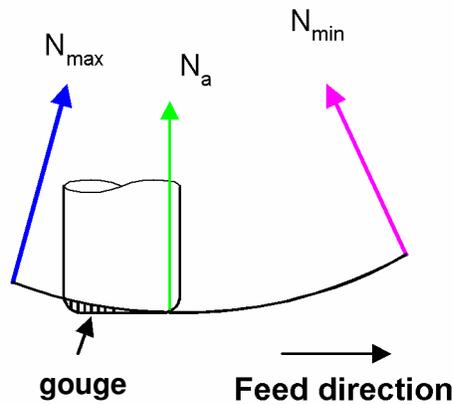
axis complicates the process of determining the projected cutting profile of the cutter. Since geometric modeling of the instantaneous cutting profile of the tool can only estimate the cusp height that will be generated, the side-step has to be set conservatively, or calculated more appropriately using simulation or swept surfaces as demonstrated by Gray et al [29]. Machining errors can be detected and corrected early by the intersection of the swept surfaces and the workpiece. While the use of the swept profile in 3-axis machining has provided exact analytical results as demonstrated by Yun *et al.* [30], this approach cannot be directly used for 5-axis machining due to the complex motion of the tool.

## **2.4 3+2-axis machining**

Researchers have attempted to take advantage of 5-axis methods without using the expensive 5-axis machines. Ralph and Loftus [31] introduced the idea of machining using discrete rotations of two additional axes. Ralph and Loftus designed an inclined end mill machining strategy for 3-axis machining centres. The method calculates the cutter orientation using an iterative process of inspection based on the cusp height. The application is suitable for low curvature surfaces, but it does not guarantee that the tool positioning strategy is suitable for the entire surface, which could result in inconsistent cusp profiles.

Suh *et al.* [32] [33] developed a CAM method by which 5-axis machining can be carried out on a 3-axis CNC machine together with a tilt/rotary table. The part surface and the machining environment are converted into a digitized workspace map. All the possible part setups that satisfy the machinability conditions are identified. The part surface is divided in a way that minimizes the number of part setups and the surface ridges where multiple tool paths join. While this approach is feasible, it requires extensive computation and experimentation to determine the partitions.

Chen *et al.* [34] proposed a technique to bridge the gap between 3- and 5-axis machining. In their work the subtractive fuzzy clustering method is used to identify the number of clusters. The partitioning is done on the basis of a collection of local geometric parameters such as the curvature, the normal and other surface parameters. The average normal for each patch is identified to determine the rotations of the part required to make the “average normal” vertical. The part is held in this orientation and the patch is machined using 3-axis methods. Since simultaneous movements of the rotary axes are not required, this technique can be implemented on indexable 5-axis machines. Chen *et al.* [34] partitioned an example surface into fourteen patches, but they did not conduct machining tests to validate the concept. The method is also limited to ball nose cutters since other tools such as toroidal cutters could result in gouging as shown in Figure 2-7. Furthermore, the proposed method may increase the machining time in comparison to machining the surface as a single patch using a ball nose tool because, for a surface partitioned in fourteen patches the time required for surface re-orientation and non-cutting rapid traverse time can be significant; this re-orientation time was not considered by the Chen *et al.*



**Figure 2-7 Tool orientation determined by the mean surface normal**

Gray [2] developed the 3+2-axis Arc-Intersect Method (AIM) for simultaneous 5-axis machines. In this approach the tool orientation is optimized for each tool pass and not for a specified region. The optimal tool position for each point is confined to a plane defined by the feed direction and the average surface normal for each pass. The tool orientation for each pass is calculated using the largest tilt angle of the projected tool positions. Experiments carried out confirmed that the surface finish is more uniform and smooth compared to those obtained with 5-axis machining. This approach, however, requires a large number of orientation changes, which makes its use prohibitive for manual rotary-tilt tables or indexable 5-axis machines. A single tool orientation could be inadequate for an entire pass, which may require breaking the pass and further reduce the efficiency of the method. Partitioning the surface offers a more general solution for 3+2-axis machining.

The work presented in this thesis has evolved from the concept proposed by Chen *et al.* [34] and resulted in a patch-by-patch machining method for sculptured surfaces. A surface clustering analysis was conducted by Roman [35] [36] to validate the most appropriate clustering parameters required for partitioning and to provide an insight into the surface partitioning process. Experiments were conducted using the Fuzzy c-means algorithm and the number of patches was determined by selecting the number of partitions that gave the shortest machining time. The effect of various geometric properties was studied on sample surfaces and a list of properties belonging to three categories namely proximity, orientation and curvature were identified. The Proximity group relates special proximity of surface points and is formed by the surface coordinates  $[S_x S_y S_z]$  and the parameter duo  $[u v]$ . The Orientation group contains the surface normal  $[N_x N_y N_z]$ , and the Curvature group includes parameters related to the curvature of the surface. It was shown that although these properties can be grouped in various combinations and with varying weights, the combination of the parametric duo and the normal vector consistently results in good partitions. Finally, the experiments

showed that partitioning a surface has an impact on the machining time in two different ways: it introduces additional time required for rapid traverse when the tool moves from one patch to another, and time to re-orient the workpiece. Secondly, if the variations within a patch are considerable, the benefits of the method are not fully realized since the shape of the tool may vary greatly from that of the surface.

## **2.5 Summary**

Table 2-1 presents a comparison of different multi-axis machining methods. This table includes some of the main advantages and disadvantages of each method. In some parameters, 3-axis and 5-axis machining have opposite characteristics. Whereas 3-axis is a cheaper alternative, with vast CAM support and rigidity, 5-axis distinguishes for its flexibility, shorter tool path lengths and the ability to machine complex surfaces. High-speed machining has similar characteristics as 3-axis machining but its productivity is higher due to its ability to remove material at higher rates. Finally, 3+2-axis machining combines the flexibility of orienting the tool offered by simultaneously 5-axis and the low cost and ease of programming offered by 3-axis machining.

**Table 2-1 Comparison of multi-axis machining methods**

	<b>Advantages</b>	<b>Disadvantages</b>
<b>3-axis</b>	<ul style="list-style-type: none"> <li>• Lower cost</li> <li>• Operators are used to this technology</li> <li>• Significant CAM support</li> <li>• Rigid (constant feed rate/consistent surface finish)</li> <li>• Simple tool path strategies</li> </ul>	<ul style="list-style-type: none"> <li>• Longer tool paths</li> <li>• May require multiple set-ups</li> <li>• Inefficient or difficult to machine complex parts</li> <li>• Limited use of toroidal and flat endmills</li> </ul>
<b>High-speed</b>	<ul style="list-style-type: none"> <li>• Smaller cutting forces</li> <li>• High accuracy (reduces polishing)</li> </ul>	<ul style="list-style-type: none"> <li>• Large number of passes, longer tool paths</li> <li>• Technology constraints, imposed by tool and workpiece materials and tool wear</li> </ul>
<b>5-axis</b>	<ul style="list-style-type: none"> <li>• Flexibility</li> <li>• Ability to machine complex and difficult geometry</li> <li>• Optimal orientation of the tool for each contact point</li> <li>• Wider machining strip → fewer tool passes → shorter tool paths</li> </ul>	<ul style="list-style-type: none"> <li>• Complexity</li> <li>• Lower and inconsistent feed rate (longer than expected machining times)</li> <li>• High cost of equipment and training</li> <li>• Difficult to check for accessibility/collisions/gouging</li> <li>• Effects of machine kinematics</li> </ul>
<b>3+2-axis</b>	<ul style="list-style-type: none"> <li>• Cheaper and stiffer alternative to 5-axis</li> <li>• Constant feed rate</li> <li>• Consistent surface finish</li> <li>• Tool orientation is locked during cutting allowing to use higher feed rates</li> <li>• Motion is in the 3 linear axes</li> </ul>	<ul style="list-style-type: none"> <li>• Research is still in its infancy</li> <li>• May require surface sub-division</li> <li>• Sub-optimal orientation of the tool for a region (smaller machining strip compared to 5-axis)</li> <li>• Tool orientation is achieved by discrete rotations that could increase the setup time</li> </ul>

The literature review presented in this chapter points out the necessity for a more robust methodology for 3+2-axis machining. 3+2-axis machining is a viable alternative for surface machining, but the lack of research in the field has limited this methodology to reach its maximum potential.

This work presents a new approach for 3+2-axis machining. This work is based on the identification of regions that have similar characteristics that can be machined using a single tool orientation. This surface partitioning strategy is presented next in Chapter 3.

## Chapter 3

### Surface partitioning

In 3+2-axis machining the tool moves only in linear motion and its orientation remains constant while machining. If a surface cannot be machined using one tool orientation, the surface needs to be partitioned. Although the partitioning of a surface is a crucial step in 3+2-axis machining, the lack of reliable and robust methods for subdivision forces the operator to subjectively conduct this task.

Partitioning a surface helps to improve the efficiency of 3+2-axis machining by identifying regions with similar surface properties that can be machined using a single tool orientation. The strategy ensures that within a patch the shape of the surface does not vary greatly from the shape of the tool in a particular orientation, which can result in a wider strip width and shorter tool paths.

The objective of developing a partitioning scheme is to provide guidelines and methods that can support the decisions taken by the operator to optimize the machining procedure. This objective actually involves two separate issues:

- How the partitioning should be conducted?
- How to determine the appropriate parameters for the surface patches?

To answer the first question, a comprehensive surface partitioning analysis is conducted and presented in this chapter. This analysis includes an evaluation of surface properties and a comparison of clustering algorithms. To address the second question, a study to identify the most relevant clustering parameters is conducted. Different surface properties are used to test the partitioning of known surfaces. Initial tests examine the surface properties individually and eliminate those with little or misleading influence. Further tests examine a combination of surface properties and the application

of different weights. Finally, tests to identify a set of surface properties that consistently result in good partitions are presented in this work.

### 3.1 Test surfaces

Six sample surfaces are used for tests in this work. The surfaces are plotted using a grid of 60x60 points and are shown in Figure 3-1. The first two surfaces considered in this study include a half-sphere and a half-torus selected because of their predictable partitions. Four parametric surfaces are selected because they resemble some of the characteristics found in dies and moulds.

The parametric equation for the half sphere is given by

$$\begin{aligned}x &= r \cos(\theta) \sin(\phi) \\y &= r \sin(\theta) \sin(\phi) \\z &= r \cos(\phi)\end{aligned}\tag{3.1}$$

where  $\theta$  runs from 0 to  $2\pi$ ,  $\phi$  runs from 0 to  $\pi/2$ , and  $r=100$ .

The toroidal surface with center at the origin is defined parametrically by

$$\begin{aligned}x &= (c + a \cos(\phi)) \cos(\theta) \\y &= (c + a \cos(\phi)) \sin(\theta) \\z &= a \sin(\phi)\end{aligned}\tag{3.2}$$

where  $\theta$  runs from 0 to  $2\pi$ ,  $\phi$  runs from 0 to  $\pi$ ,  $a = 30$ , and  $c = 80$  ( $c$  is the radius from the center of the hole to the center of the torus, and  $a$  is the radius of the tube).

The control points used for generation for the Bézier surfaces are presented in Table 3-1, Table 3-2 and Table 3-3 for surface 1, surface 2 and surface 3, respectively. The last surface, surface 4, is defined using equation (3.3), where  $x$  runs from 0 to 100 and  $y$  runs from 0 to 40.

$$z = 50 - 40 \left( \frac{x}{60} \right) e^{\left[ -\left( \frac{x}{60} \right)^2 - \left( \frac{y}{30} \right)^2 \right]}\tag{3.3}$$

**Table 3-1 Control points for surface 1**

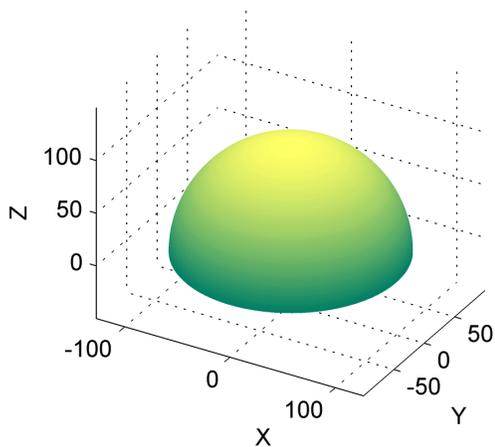
$p_{0,0} = [0, 0, -47]$	$p_{0,1} = [0, 75, -52]$	$p_{0,2} = [0, 150, -42]$	$p_{0,3} = [0, 225, -5]$
$p_{1,0} = [50, 0, -35]$	$p_{1,1} = [50, 75, -99]$	$p_{1,2} = [50, 150, -56]$	$p_{1,3} = [50, 225, 0]$
$p_{2,0} = [100, 0, -65]$	$p_{2,1} = [100, 75, -79]$	$p_{2,2} = [100, 150, -28]$	$p_{2,3} = [100, 225, -37]$
$p_{3,0} = [150, 0, -17]$	$p_{3,1} = [150, 75, -49]$	$p_{3,2} = [150, 150, -50]$	$p_{3,3} = [150, 225, -53]$

**Table 3-2 Control points for surface 2**

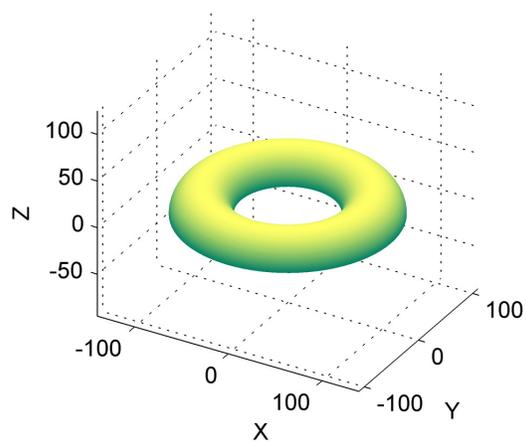
$p_{0,0} = [0, 0, 50]$	$p_{0,1} = [0, 75, 80]$	$p_{0,2} = [0, 150, 110]$	$p_{0,3} = [0, 225, 30]$
$p_{1,0} = [50, 0, 80]$	$p_{1,1} = [50, 75, 110]$	$p_{1,2} = [50, 150, 150]$	$p_{1,3} = [50, 225, 60]$
$p_{2,0} = [100, 0, 70]$	$p_{2,1} = [100, 75, 100]$	$p_{2,2} = [100, 150, 110]$	$p_{2,3} = [100, 225, 40]$
$p_{3,0} = [150, 0, 50]$	$p_{3,1} = [150, 75, 60]$	$p_{3,2} = [150, 150, 130]$	$p_{3,3} = [150, 225, 50]$

**Table 3-3 Control points for surface 3**

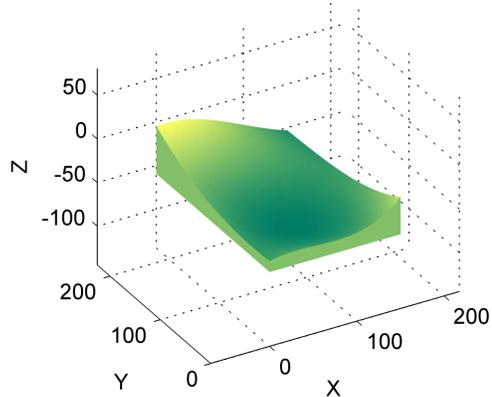
$p_{0,0} = [0, 0, 38]$	$p_{0,1} = [0, 75, 45]$	$p_{0,2} = [0, 150, 48]$	$p_{0,3} = [0, 225, 62]$
$p_{1,0} = [50, 0, 50]$	$p_{1,1} = [50, 75, 75]$	$p_{1,2} = [50, 150, 26]$	$p_{1,3} = [50, 225, 52]$
$p_{2,0} = [100, 0, 52]$	$p_{2,1} = [100, 75, 73]$	$p_{2,2} = [100, 150, 32]$	$p_{2,3} = [100, 225, 48]$
$p_{3,0} = [150, 0, 40]$	$p_{3,1} = [150, 75, 52]$	$p_{3,2} = [150, 150, 48]$	$p_{3,3} = [150, 225, 60]$



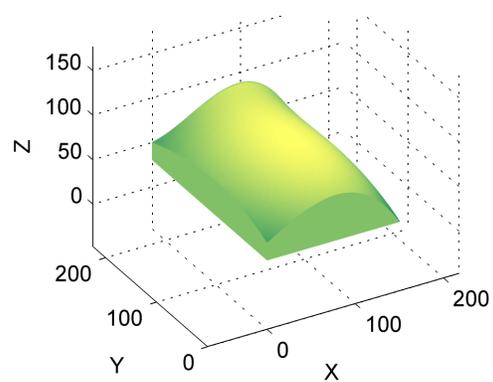
**(a) Half-sphere**



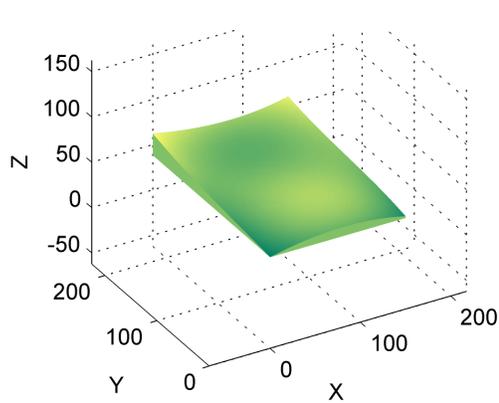
**(b) Half-torus**



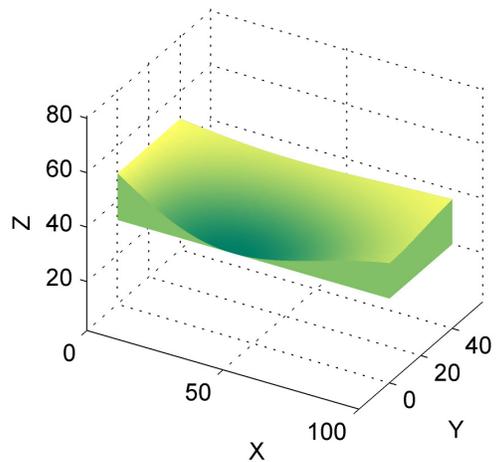
**(c) Surface 1**



**(d) Surface 2**



**(e) Surface 3**



**(f) Surface 4**

**Figure 3-1 Test surfaces**

## 3.2 Clustering algorithms

Clustering can be considered one of the most important unsupervised learning problems [37]. Procedures that use unlabelled samples, data of unknown characteristics, are said to be unsupervised. Unsupervised problems are related to identifying the number and locations of classes from a collection of samples. Once a measure of similarity is chosen, the next step is determining a procedure that will create the groupings. The conclusions derived by these algorithms have to be backed by intuition and a significant quantity of experimentation [38].

Clustering is a process of partitioning a set of data into group of elements (clusters) that are similar. The effectiveness of clustering depends on the specific algorithm and the criteria used. The clustering criteria include the number of clusters and the parameters to measure the similarity between clusters. There is no absolute best criterion that would be independent of the final aim of the clustering [39]. It is the user who must supply the clustering parameters in such a way that the clustering results will fit his/her requirements.

This section explores three different approaches used for clustering: the k-means algorithm, the Fuzzy c-means algorithm, and the hierarchical clustering method.

### 3.2.1 k-means clustering algorithm

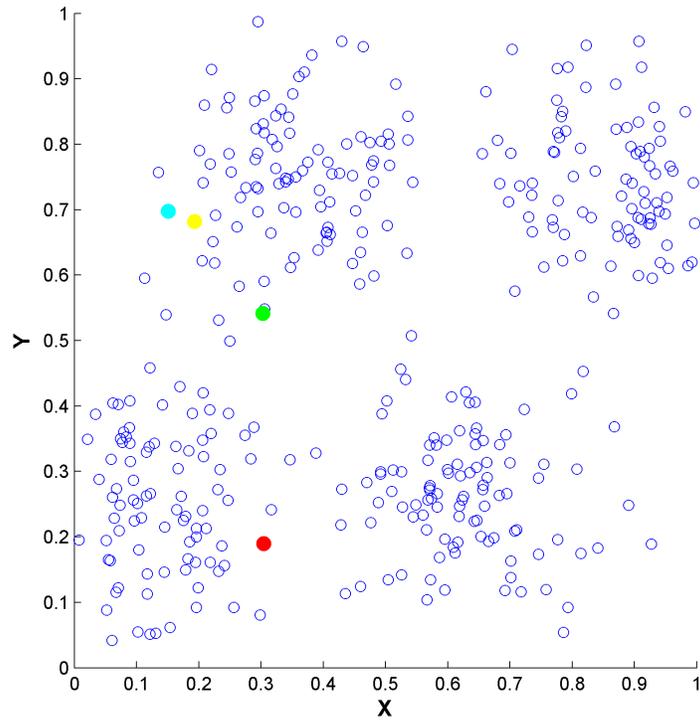
The *k*-means algorithm is well known for its efficiency in clustering large data sets [40]. This algorithm is used to divide the data set into a predetermined number of patches and uses an objective function that is based on the square-error distance. This distance is defined as

$$J = \sum_{j=1}^k \sum_{i=1}^n \|p_i - C_j\|^2 \quad (3.4)$$

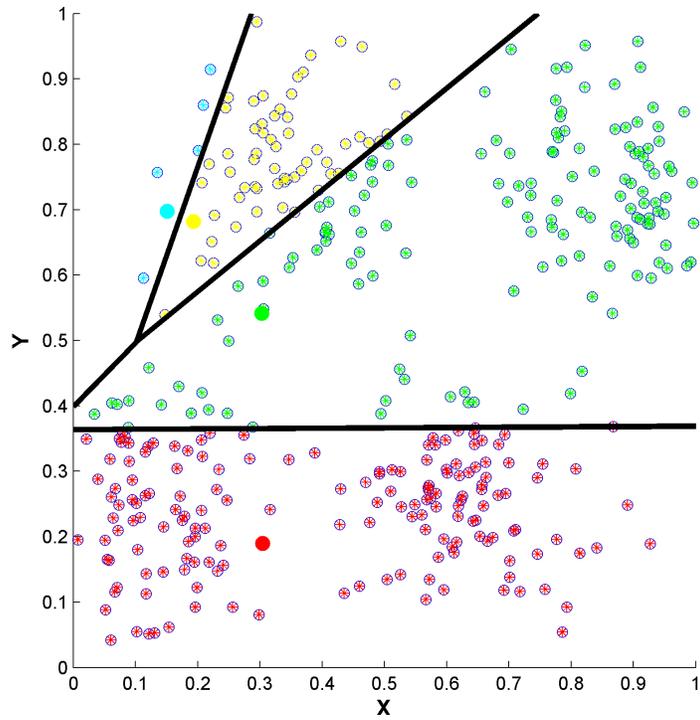
where *k* is the number of patches, *n* is the number of sample points, *p<sub>i</sub>* are called the sample points and *C<sub>j</sub>* are the centroids of the partitions.

The k-means can be easily applied for multi-dimensional data. The inputs for the algorithm are the multidimensional data and the number of clusters. The output is a one-dimensional vector (of the same length of the input vector) that indicates which cluster the point belongs to.

A 2-dimensional example to illustrate the process of partitioning using the k-means is shown in Figure 3-2. In this figure the 2-dimensional sample points that need to be sub-divided are represented with blue circles. The k-means algorithm starts by randomly positioning  $k$  number of centroids (coloured dots). The centroid (or cluster center) is also a 2-dimensional vector comprised of  $X$ - and  $Y$ -coordinates. After the centroids are calculated, each sample point in the data set is associated to its nearest centroid using the objective function given above (Figure 3-3), which is an indicator of the distance of the sample points from their respective centroids. After each data point is associated to a centroid, it is necessary to recalculate a new position for the centroid using the mean of all the points inside the cluster. The change in centroid alters the distance to the cluster points, which begins an iterative process of associating the points to the (repositioned) centroids (Figure 3-4). The process continues until the centroids do not move, or the objective function is minimized to a user specified value (Figure 3-5).



**Figure 3-2 Clustering data and initial cluster centers**



**Figure 3-3 1<sup>st</sup>. iteration**

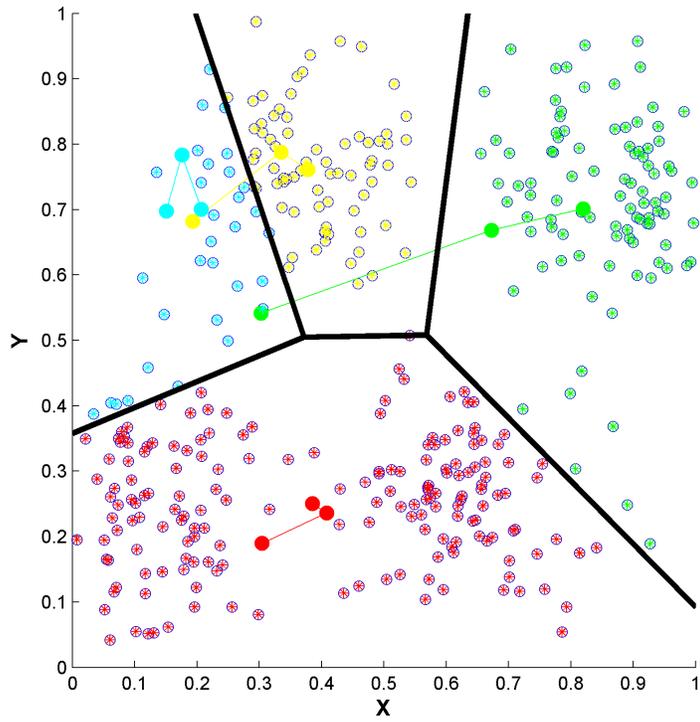


Figure 3-4 3<sup>rd</sup>. iteration

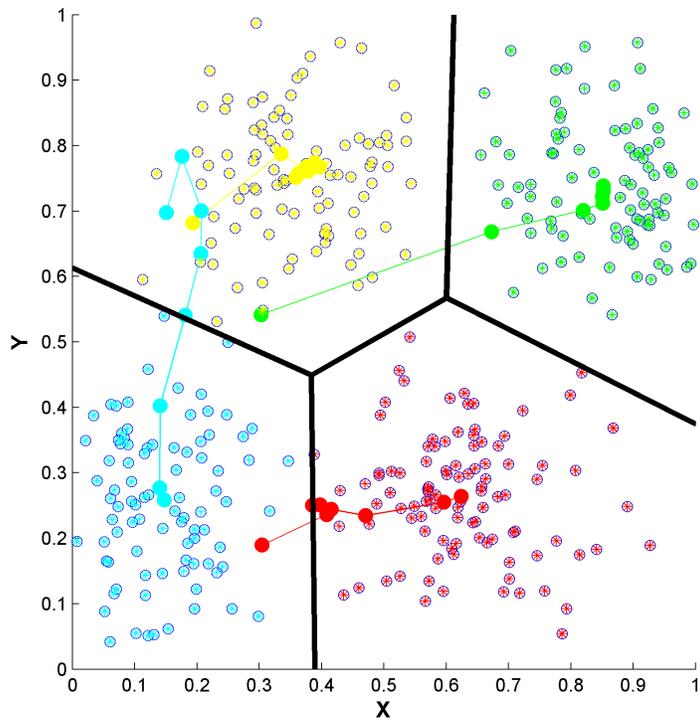


Figure 3-5 10<sup>th</sup>. Iteration

### 3.2.2 Fuzzy c-means

The Fuzzy c-means clustering algorithm is based on a generalization of the sum of square error function [41]. The Fuzzy c-means is a technique where each data point belongs to a cluster to some degree that is specified by a membership grade [37]. Multidimensional space points can be grouped into a specific number of different clusters based on this grade. The output arguments of this function are the cluster centers and the partition matrix. The cluster center is the element that represents all the points in a cluster.

In the k-means clustering algorithm each data point is assumed to be in exactly one cluster. The Fuzzy c-means relax this condition and assume that each sample has some “fuzzy” membership in a cluster [37], which allows one piece of data to belong to two or more clusters [39].

The objective function for the Fuzzy c-means is given by

$$J = \sum_{j=1}^m \sum_{i=1}^n u_{ij}^m \|p_i - C_j\|^2 \quad (3.5)$$

where  $m$  is the number of patches,  $n$  is the number of sample points,  $u_{ij}$  is the degree of membership of  $x_i$  in the cluster  $j$ ,  $p_i$  are the sample points and  $C_j$  are the centroids of the partitions.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership  $u_{ij}$  and the cluster centers  $c_j$  by

$$u_{ij} = \frac{1}{\sum_{k=1}^m \left( \frac{\|x_i - C_j\|}{\|x_i - C_k\|} \right)^{\frac{2}{m-1}}} \quad (3.6)$$

$$c_j = \frac{\sum_{i=1}^n u_{ij} p_i}{\sum_{i=1}^n u_{ij}} \quad (3.7)$$

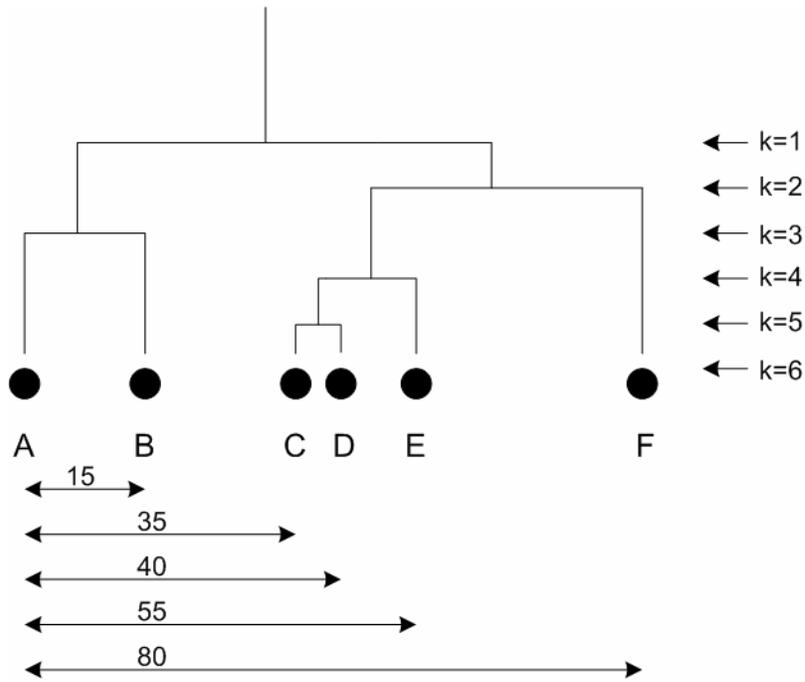
The iterations stop when  $\max_{ij} \left\{ \left| u_{ij}^{k+1} - u_{ij}^k \right| \right\} \leq E$ , where  $E$  is a termination criterion between 0 and 1 [39].

### 3.2.3 Hierarchical clustering

The hierarchical clustering is an exhaustive technique that groups data according to a specified distance. The hierarchical clustering method starts by calculating the similarity between every point in the data set. Different types of distance measurements can be used and the list includes list includes Euclidean, Standardized Euclidean, City Block, and Correlation distance [42]. Based on the distance between the points, a hierarchical tree (or dendrogram) is created. The hierarchical tree represents how the clusters are joined in a multi level hierarchy.

The hierarchical clustering method starts by assigning each sample point into a cluster. Next, the closest pair of elements is merged into a cluster. The iteration process begins by calculating the distance between the new cluster and the old set of sample points. The process is finished when all the points are grouped into the predefined number of clusters.

Figure 3-6 shows a dendrogram for a simple 1-D problem involving 6 samples. At  $k = 6$  the six samples are singleton clusters. At the next level, samples C and D have been grouped to form one cluster. Once a cluster is formed, the samples stay together at all subsequent levels. The grouping of points continues until the specified number of clusters is calculated, as shown in Table 3-4. This table show the groupings formed using the hierarchical clustering algorithm.



**Figure 3-6 Dendrogram**

**Table 3-4 Hierarchical clustering**

# of clusters (k)	Clusters
6	[A] [B] [C] [D] [E] [F]
5	[A] [B] [C D] [E] [F]
4	[A] [B] [C D E] [F]
3	[A B] [C D E] [F]
2	[A B] [C D E F]
1	[A B C D E F]

### 3.2.4 Clustering algorithms for surface partitioning

The main requirements that a clustering algorithm should satisfy for surface partitioning include dealing with different types of surface properties; identify clusters with arbitrary shape; high dimensionality; and computational speed.

The k-means and the Fuzzy c-means use a similar process of clustering and yield similar partitions but offer different types of classification. Whereas the k-means executes a sharp classification in which each object is either assigned to a cluster (exclusive clustering), the Fuzzy c-means classification function causes the sample points to become a relative one and an object can belong to several classes at the same time but with different degrees (overlapping clustering) [43]. The Fuzzy c-means admits the possibility of partial membership. Each point may belong to two or more clusters with different degrees of membership. The degree an object belongs to a fuzzy set is denoted by a membership value between 0 and 1 [44].

Hierarchical clustering is a method that determines the clusters based on a distance-based hierarchical tree. While it is possible to obtain different numbers of partitions in the same calculation, this method normally requires a considerable number of calculations that can result in large computation times and impractical for surface partitioning. For a one-dimensional problem using a 2500 sample points (50x50 grid) the computational time consumed was close to 87 minutes using MATLAB ® on a Pentium 4 running at 1600 MHz.

Since the k-means and the Fuzzy c-means yield to similar partitions, it is necessary to compare the computational requirements for both methods. A comparison of the computational time required for these methods is presented below. The test is conducted using different number of sample points using the test surface 1. In this test both methods were used for partitioning the surface into four patches using  $n$ -dimensional inputs. The results for the estimated time of computation are shown in Figure 3-7. This comparison is conducted on a 1-dimensional (1-D) vector using the Rolling Ball Radius (RBR) and a 5-dimensional (5-D) vector formed by the surface normals ( $X$ - and  $Y$ -coordinates) and the surface normal. While the k-means showed faster calculations, the results obtained with both algorithms showed that a solution for a 100x100 grid can be calculated in less than

10 seconds. Given that the k-means simplifies the calculations and accelerates convergence, this algorithm will be used in the remainder of this work.

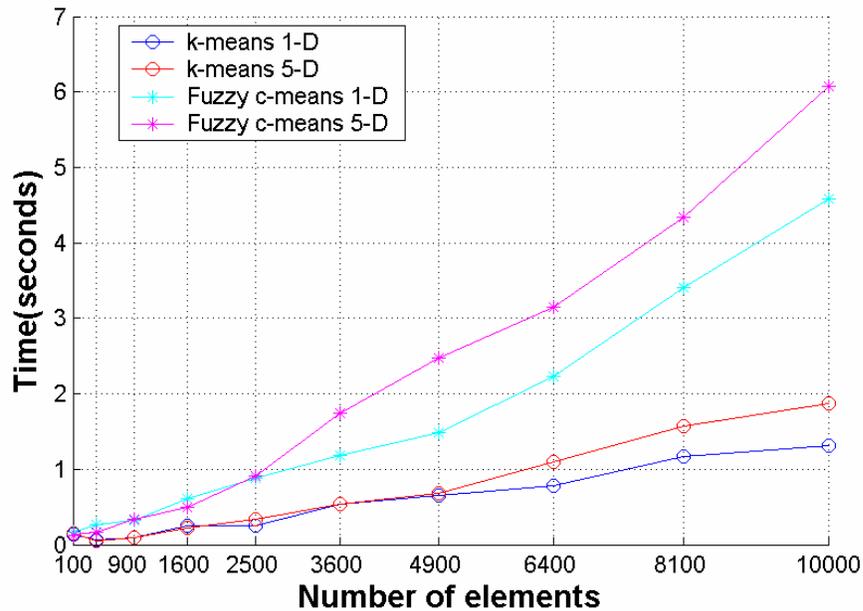


Figure 3-7 Time comparison using multi-dimensional data

### 3.3 Feature extraction

This section presents the methodology to extract the data from the sample points and proceeds to conduct a classification of these surface properties. In pattern recognition, the set of surface properties to be used for partitioning are referred to as “features”. The features can be applied to a clustering algorithm to define the surface patches.

The process of surface partitioning starts by extracting the features for a set of sample points. The features are classified into three major groups: the proximity group, the orientation group, and the shape group. The proximity group contains geometric parameters that are related to the spatial location of the point. The orientation group is comprised of parameters associated to the orientation of

the tool or the workpiece. The shape group includes features related to the shape and curvature of the surface.

### 3.3.1 Proximity group

The proximity group is formed by parameters that provide information about the location of the sample point in space. The parameters that are associated to this group are the surface coordinates ( $S$ ) and the parameter duo ( $u, v$ ). For a Bézier surface the coordinates [ $S_x S_y S_z$ ] corresponding to the parameters  $u$  and  $v$  can be calculated from Equation (3.8), where  $0 \leq u \leq 1$ ,  $0 \leq v \leq 1$ ,  $px_{i,j}$ ,  $py_{i,j}$  and  $pz_{i,j}$  are the control points, and  $n$  and  $m$  define the degrees of the Bézier surface. The surface is based on a basis function that is given in Equation (3.9).

$$S = \begin{bmatrix} S_x(u, v) \\ S_y(u, v) \\ S_z(u, v) \end{bmatrix} = \begin{pmatrix} \sum_{i=1}^n \sum_{j=1}^m ((px_{i,j})(B(i, n, u)(B(j, m, v))) \\ \sum_{i=1}^n \sum_{j=1}^m ((py_{i,j})(B(i, n, u)(B(j, m, v))) \\ \sum_{i=1}^n \sum_{j=1}^m ((pz_{i,j})(B(i, n, u)(B(j, m, v))) \end{pmatrix} \quad (3.8)$$

$$B(i, n, u) = \frac{n!}{(n-i)!i!} (1-u)^{n-i} u^i \quad (3.9)$$

### 3.3.2 Orientation group

The orientation group relates parameters associated to the orientation of the tool. These parameters provide information that can help identify points that share a common tool orientation. This group consists of the surface normals ( $N$ ) and the rotation angles ( $A, C$ ).

The surface normal, [ $N_x N_y N_z$ ], is used to determine the tool orientation and is defined in Equation (3.10).

$$N = \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = \frac{\frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}}{\left| \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v} \right|} \quad (3.10)$$

$A$  and  $C$  represent the tilting and rotating angles, respectively; they are used for the two additional axes in 3+2-axis machining. These angles can be calculated from the normal vector using equations (3.11) and (3.12).

$$A = \arctan \frac{\sqrt{N_x^2 + N_y^2}}{N_z} \quad (3.11)$$

$$C = \arctan \frac{N_x}{N_y} \quad (3.12)$$

### 3.3.3 Shape group

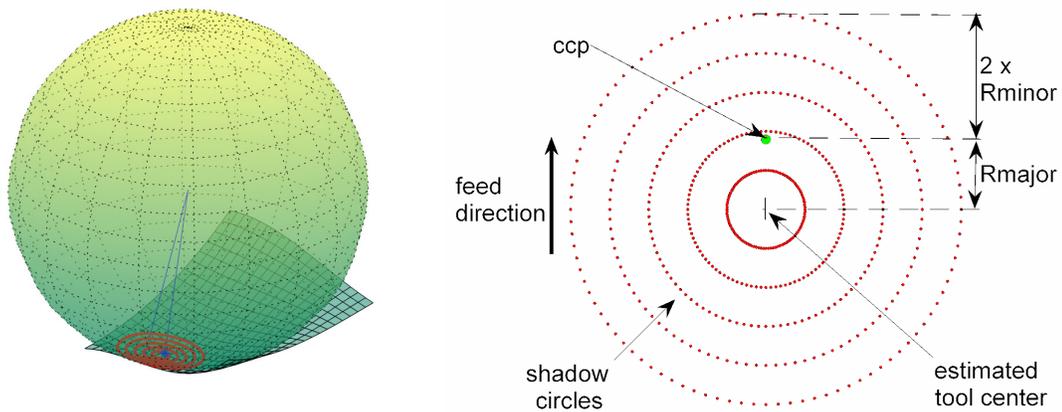
The shape group consists of parameters related to the geometry surrounding the sample points and includes the Rolling Ball Radius and various types of curvature.

#### Rolling Ball Radius

The regional nature of the Rolling Ball Method (RBM) coupled with simplicity of calculation made it an ideal choice for use in 3+2-axis machining. The basic idea of the RBM consists of locally approximating the surface around the machining point with a portion of a sphere. A sphere can be used as an interface between the surface and the cutting tool to provide a support for tool positioning. The tool is placed inside this sphere such that the cutter contact point coincides with the machining point. To guarantee a gouge-free tool positioning, the radius of the sphere is determined by the largest ball that can be located at the point to be machined without gouging points on the surface in the vicinity of the machining point. At the contact point, the sphere fits the local surface without gouging

the surface. The portion of surface, where the gouging risk exists, is the shadow area or the region that the tool casts a shadow on. For each point on this shadow area, the radius of the ball represents a pseudo-radius of curvature.

The area in the shadow of the tool represents the region of the tool and surface that must be taken into account to compute a gouge-free tool position. This region is not initially known because it depends on the orientation and final position. Thus, the tool projection onto the surface is over-estimated by a circular region and this larger circular zone is used in the computation. This shadow area of the tool is divided into five concentric circles comprised of 100 points each and for these points the pseudo-radius of curvature is computed. The points are referred to as the shadow-check points. The center of the shadow area is located at  $R_{\text{major}}$  from the cutter contact point along the direction opposed to the feed-direction. The radius of the shadow area is set to be equal to  $R_{\text{major}} + 2 \times R_{\text{minor}}$ , as shown in Figure 3-8.

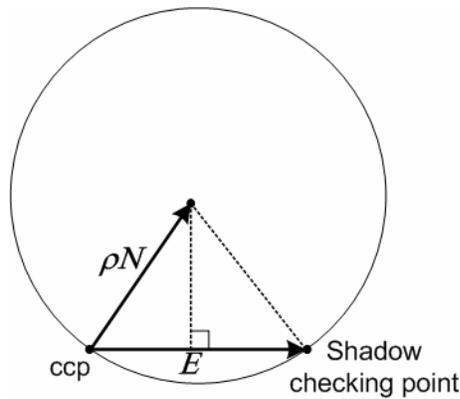


**Figure 3-8 Shadow area**

The Pseudo-radius of curvature is defined as the radius of the circle whose center lies along the surface normal at the cutter contact point where the circle passes through the cutter contact point and the shadow checking point [2]. The computed pseudo-radii at all the shadow-check points for a cutter contact point are compared to identify the ‘most concave’ radius. This radius is selected to be the pseudo-radius of the sphere used for partitioning. The pseudo-radius of curvature is shown in Figure 3-9 and is given in Equation (3.13).

$$\rho = \frac{-\vec{E} \cdot \vec{E}}{2 \times N \cdot \vec{E}} \quad (3.13)$$

$$\vec{E} = \text{Shadow\_checking\_point} - \text{ccp} \quad (3.14)$$



**Figure 3-9 Pseudo-curvature circle**

### **Types of Curvatures**

The curvature is an indicator of the rate of change of the surface normal over the surface at a point. Four geometric parameters, Gaussian, mean, minimum and maximum curvatures, can be used to define the curvature at a point. Gaussian curvature ( $K$ ) in equation (3.15), describes the local shape of the surface, while ( $H$ ) in equation (3.16) represents the Mean curvature.

$$K = \frac{LN - M^2}{EG - F^2} \quad (3.15)$$

$$H = \frac{1}{2} \left( \frac{EN - 2FM + GL}{EG - F^2} \right) \quad (3.16)$$

$$E = \frac{\partial S}{\partial u} \cdot \frac{\partial S}{\partial u} \quad F = \frac{\partial S}{\partial u} \cdot \frac{\partial S}{\partial v} \quad G = \frac{\partial S}{\partial v} \cdot \frac{\partial S}{\partial v} \quad (3.17)$$

$$L = \frac{\partial^2 S}{\partial u^2} \cdot \frac{\partial^2 S}{\partial u^2} \quad M = \frac{\partial^2 S}{\partial u^2} \cdot \frac{\partial^2 S}{\partial v^2} \quad N = \frac{\partial^2 S}{\partial v^2} \cdot \frac{\partial^2 S}{\partial v^2} \quad (3.18)$$

$K_{min}$  and  $K_{max}$  in equations (3.18) and (3.19), respectively are called the principal curvatures and serve as bounds on the components of curvature not in the tangent plane [27].

$$K_{min} = H - \sqrt{H^2 - K} \quad (3.19)$$

$$K_{max} = H + \sqrt{H^2 - K} \quad (3.20)$$

### 3.4 Feature selection

Feature selection is the procedure of selecting the most important features so as to reduce their number and at the same time retain as much as possible of their class discriminatory information [45]. The number of features is usually large, but there is more than one reason to reduce the number of features. Besides computational complexity, a related reason is that although two features may carry good classification information when treated separately, there is little gain if they are combined together in a feature vector [37].

One of the main challenges in clustering is the selection of the properties that best describe the members of the data to be used, and the proper identification of the clusters in the data. Bezdek [41] concluded that:

*“the variety of structures is without bound because each observation can easily have several dimensions. There are no principles or universal criteria for clustering. The selection of the*

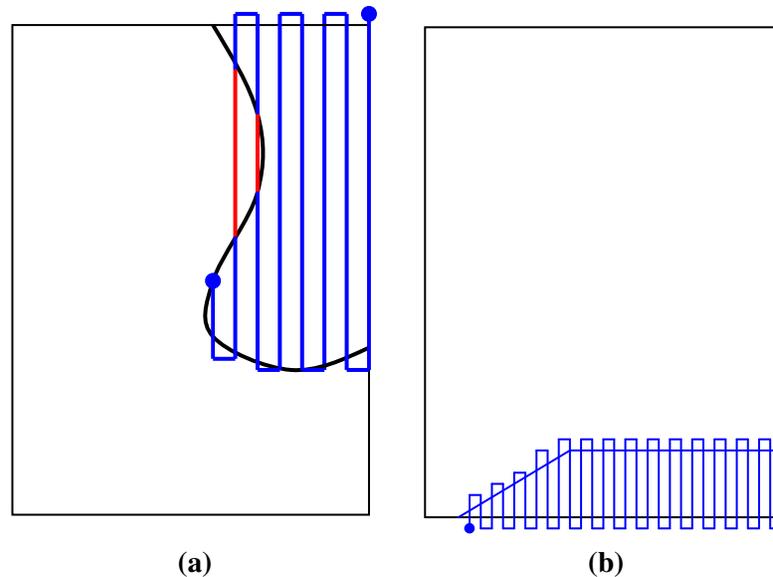
*parameters is partially subjective and open to discussion. Moreover, it is not very common to find separated and equally proportioned clusters in real data. Data sets may embody a mixture of shapes, sizes, and geometries. Based on this, a successful cluster analysis is a challenging assignment for the investigator, and the solutions should be based on experience and practical perception”.*

Chen *et al.* [34] used the geometric properties of the surface to form a multi-dimensional vector, which was used to partition a surface. In their work the parameter values, surface normal, Gaussian, mean, maximum and minimum curvatures were all lumped together into one vector for partitioning. However, having several forms of curvature could result in redundancy and inefficient partitioning. A previous work by the present author [35] analyzed the influence of surface parameters on partitioning. It was determined that the multi-dimensional vector describing the geometry of a point must include information about the spatial location of the point and the orientation of the surface normal. The location of the point indicated by the surface coordinates helps keep neighbouring points in one cluster to avoid creating disjointed patches. The surface normal vector provides information to cluster points that could be machined with the same tool orientation. The study also demonstrated that the use of various curvature parameters may result in redundancy as the surface shape is implied by variations in the normal vector.

### **3.4.1 Evaluation of features**

The effect of using different parameters for surface partitioning is presented here. The surfaces are represented by a sample of points and the properties at these points are used as the input to the k-means algorithm. The output is a grouping of these points into clusters (patches), if however, the points did not lie in one closed region it would represent two or more disjointed sub-patches. Disjointed patches require additional tool travel that complicates tool path generation and increases the machining time. If the number of disjointed sub-patches is large then the movement of the tool between patches will increase significantly and thus the partitioning is not considered useful. Another

problem with irregular patches is illustrated in Figure 3-10 (a). Depending on the feed direction, non-convex patches can force the tool to be lifted up and down during machining. Another problem that must be avoided is shown in Figure 3-10 (b). In this case, the tool passes are not long enough to allow the tool to reach its maximum feed rate, which can result in longer than expected machining times.



**Figure 3-10 Complex shapes**

The results of the partitioning tests on four sample surfaces are shown in Table 3-5 and Table 3-6. The first two surfaces were selected because of their predictable partitions. The last two surfaces were selected because of they resembles typical surfaces in tools and dies. These tables list the properties employed, in column one, and shows the partitioning results for the surfaces in columns two and three.

The clustering tests using single parameters show that in the majority of the cases the shape of the patches is irregular, which can complicate the tool path generation process. The use of a single parameter for partitioning shows that it might be difficult to find a parameter that can be applied successfully in all types of surfaces. Therefore, the clustering data should be a combination of parameters that can provide information related to the location and the shape of the patch.

Table 3-5 Partitioning tests

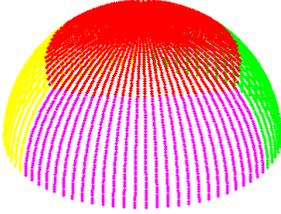
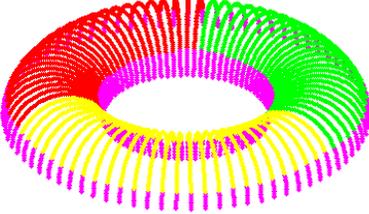
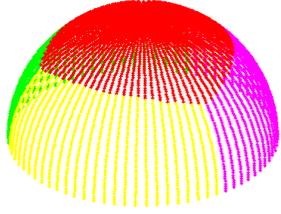
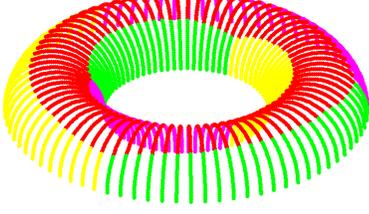
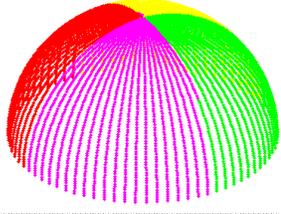
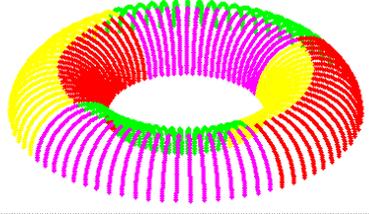
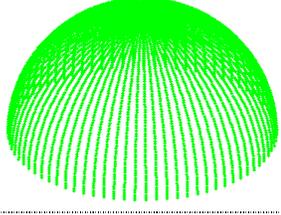
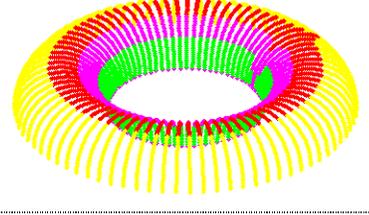
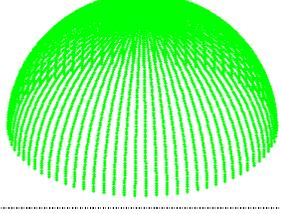
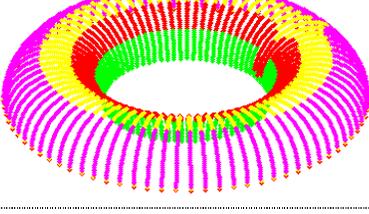
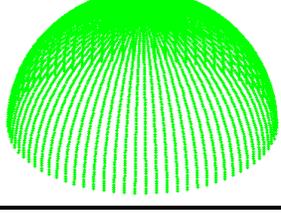
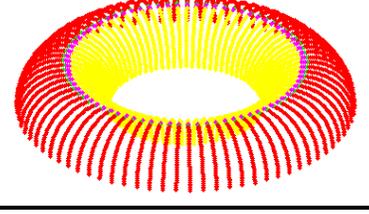
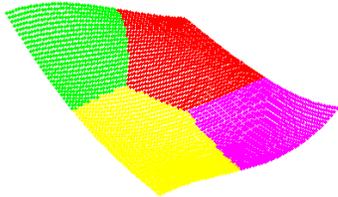
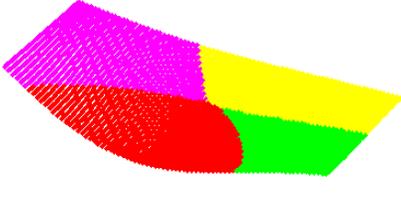
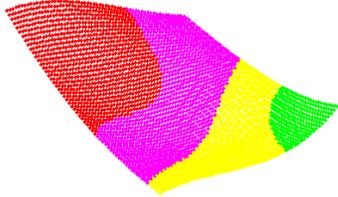
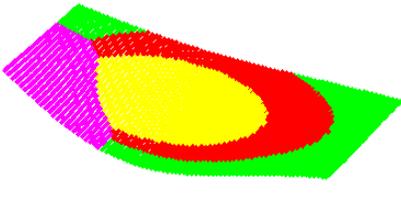
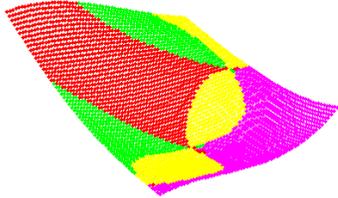
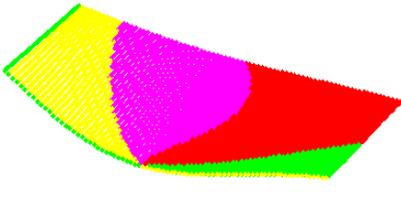
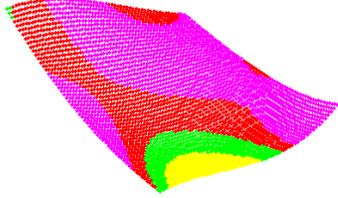
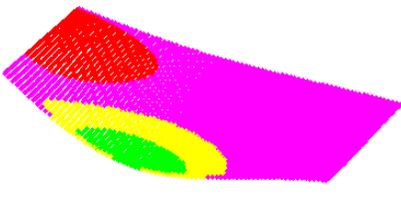
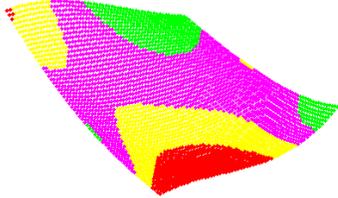
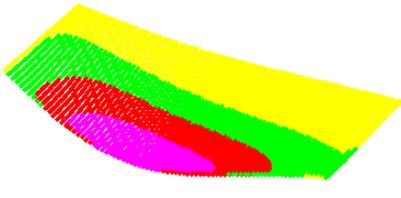
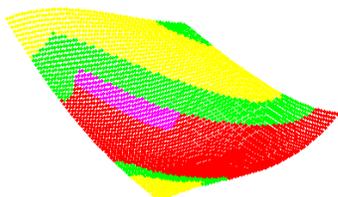
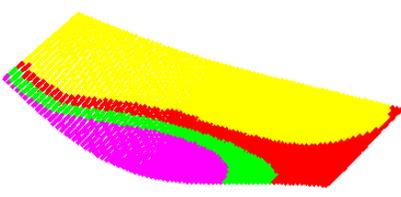
	Half-sphere	Half-torus
[S]		
[N]		
[AC]		
[K]		
[H]		
[RBR]		

Table 3-6 Partitioning tests

	Surface 1	Surface 4
[S]		
[N]		
[AC]		
[K]		
[H]		
[RBR]		

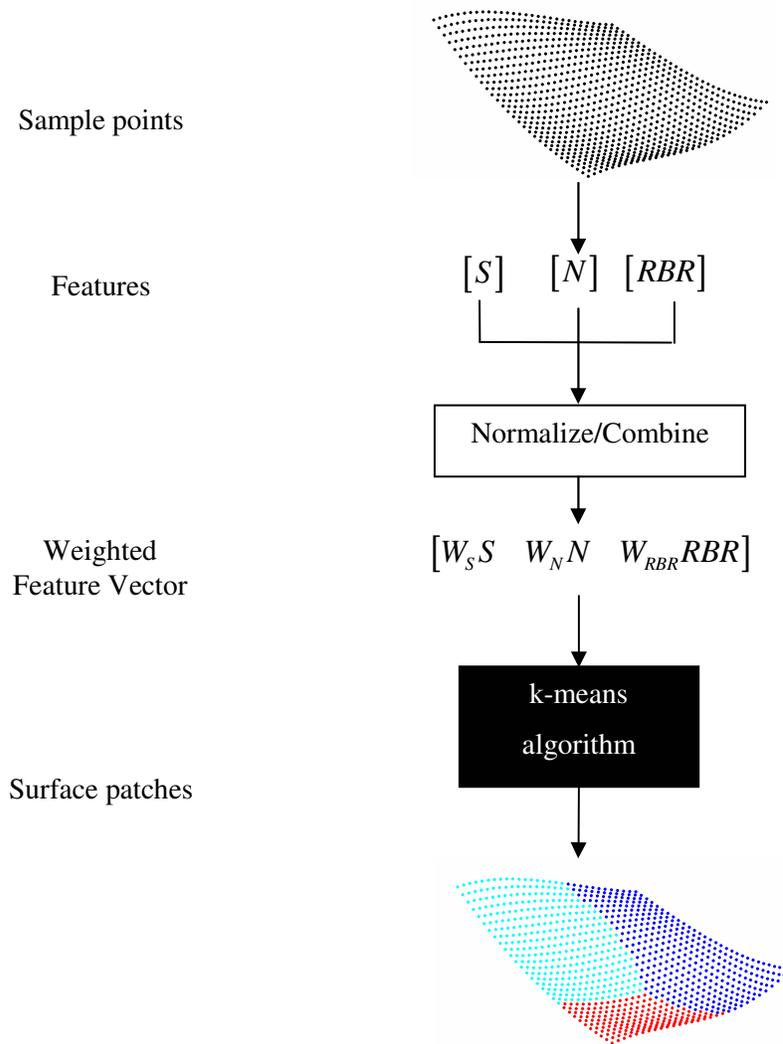
The results obtained using the surface coordinates ( $S$ ) show regular and more consistent patches. These coordinates group points that are close to each other and can be used in combination with other parameters to prevent the creation of disjointed patches. The normal vector ( $N$ ) shows better results than the use of the angles  $A$  and  $C$ . The use of the normal vector provides more regular patches, which takes into account the shape of the surface. The different curvature parameters lead to similar results; to prevent redundancy, however, only one parameter should be considered from each group of clustering parameters.

### **3.5 Feature vector**

The previous experiments showed that the use of individual parameters do not provide adequate patches for 3+2-axis machining. To address this issue, the surface properties can be grouped into a variety of combinations. The combinations of surface properties form a feature vector that comprises one or more entities. This feature vector can be used for a better representation of each sample point.

The surface partitioning diagram using a feature vector is presented in Figure 3-11. The process starts by extracting the surface properties from a group of sample points that represent the surface to be machined. The multi-dimensional vector describing the geometry at a point on the surface must include information from the three groups identified earlier. The location of the point, defined by the surface coordinates, helps keeping neighbouring points in one cluster and preventing disjointed patches. The orientation parameters provide information on the required orientation of the workpiece for machining. If the variations in the surface normal are minimal, the tool can match the surface geometry more closely, which results in larger side-step distances and fewer tool passes. The orientation parameter is represented by the surface normal. The shape parameters reflect the rate of change of the surface in the vicinity of the point of interest and are represented by the Rolling Ball Radius ( $RBR$ ). All the parameters have to be normalized for the clustering experiments and then

different weights can be assigned to these parameters, which further increase the possibilities. This weighted feature vector is applied to the partitioning algorithm to divide the data set into the specified number of patches, as shown at the bottom of Figure 3-11.

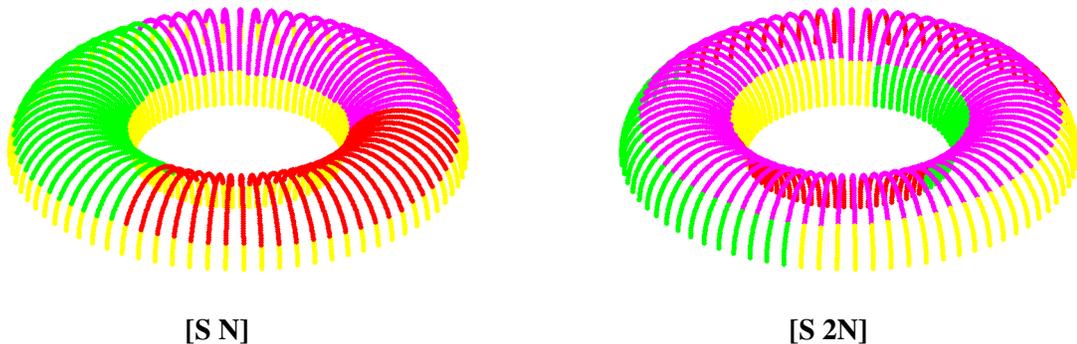


**Figure 3-11 Surface partitioning process**

The features can be normalized using Equation (3.21), where  $\min(\ )$  and  $\max(\ )$  are functions that find the minimum and maximum of a set of data.

$$normalized\_feature = \frac{feature - \min(feature)}{\max(feature - \min(feature))} \quad (3.21)$$

A partitioning test using a half-toroidal surface was conducted to evaluate the suitability of the weighted feature vector. Although, a torus should be machined as an independent entity with a form tool, this test allows understanding the partitioning process and the effect of applying different weights to the features. The first partitioning test was conducted applying equal weights for the surface coordinates and the normal vector (Figure 3-12). This partitioning is not appropriate because the entire top section of the part should be machined using a single orientation. Also, the sides and the interior of the torus cannot be machined using a single orientation. By applying a double weight for the surface normals, this problem could be eliminated. The result, shown on the right side of Figure 3-12, presents a more reasonable partitioning for this surface. The top portion of the surface can be machined using a single tool orientation and the sides and the interior part of the toroidal are divided into three sections. Although the interior section of the half-toroidal may present some accessibility problems, this partition is a more logical solution.



**Figure 3-12 Half-toroidal surface divided into four patches**

Modifying the weight of the features offers more control on the partitioning but also increases the number of possible solutions. Incrementing the weight of the surface normals can help group points with similar surface normals that can be machined more appropriately using a single tool orientation. Overemphasizing the weight of the surface normal, however, can result in disjointed patches. An optimal weighted feature vector should result in large joined regions that are geometrically similar.

Five feature vectors were applied to parametric surfaces. These vectors include combinations of weighted features from the three partitioning groups identified earlier. These groups were selected because they consistently provided good partitions on preliminary tests. Also included in these tests is the use of the ‘cosine’ distance (from the Matlab library). This function treats the clustering data as vectors and results in similar results to those obtained by applying a higher weight for the surface normals. The five combinations proposed are as follows:

1. [*S N*] Equal weight for both parameters using ‘Euclidean’ distance.
2. [*S 2N*] Double weight for the normal vectors using ‘Euclidean’ distance.
3. [*S N*] Equal weight for both parameters using the ‘cosine’ distance.
4. [*S N RBR*] Equal weight for all parameters using ‘Euclidean’ distance.
5. [*S RBR*] Equal weight for both parameters using ‘Euclidean’ distance.

The partitioning results for four test surfaces are shown in Table 3-7, Table 3-8, Table 3-9, and Table 3-10. These tests were conducted to compare the different feature vectors and identify groups that result in good partitions. A good partitioning should facilitate the generation of tool paths and minimize the occurrence of irregular patches. Irregular patches can reduce the efficiency of patch-by-patch machining and may result in longer than expected machining times.

Table 3-7 Partitioning results for surface 1

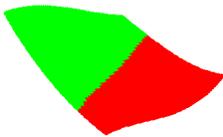
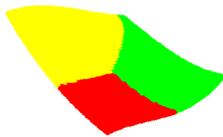
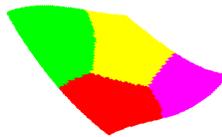
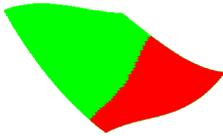
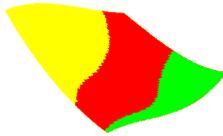
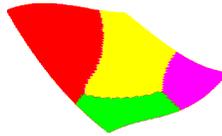
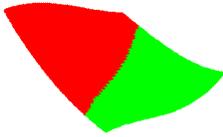
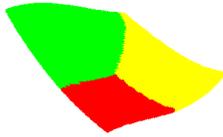
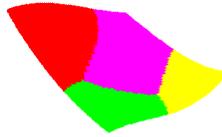
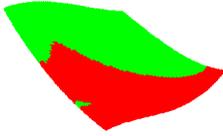
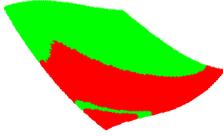
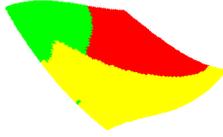
	2 patches	3 patches	4 patches
[S N] Squared Euclidean			
[S2 N] Squared Euclidean			
[S N] cosine			
[S N RBR] Squared Euclidean			
[S RBM] Squared Euclidean			

Table 3-8 Partitioning results for surface 2

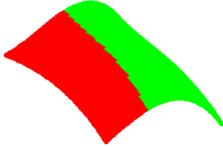
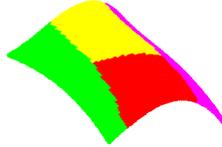
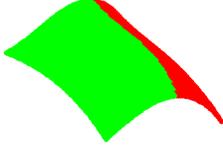
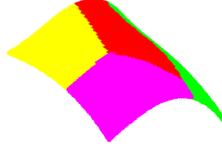
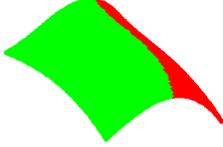
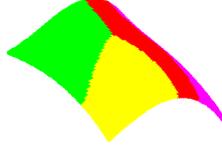
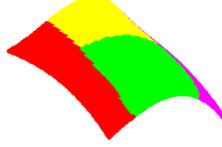
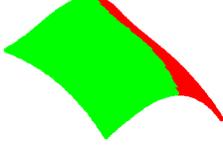
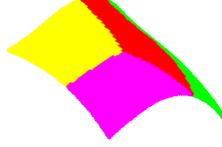
	2 patches	3 patches	4 patches
[S N] Squared Euclidean			
[S 2N] Squared Euclidean			
[S N] cosine			
[S N RBR] Squared Euclidean			
[S RBM] Squared Euclidean			

Table 3-9 Partitioning results for surface 3

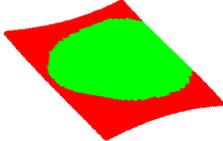
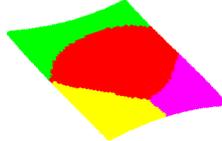
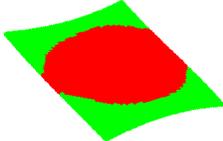
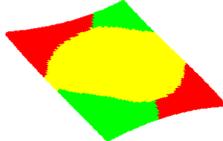
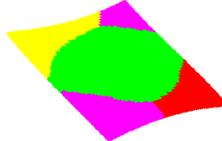
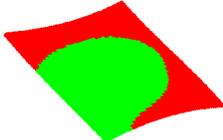
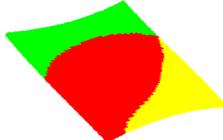
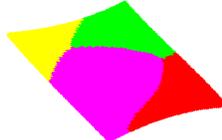
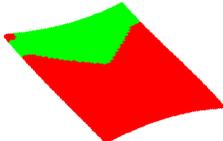
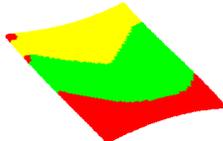
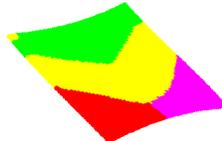
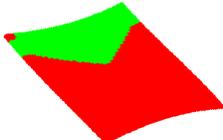
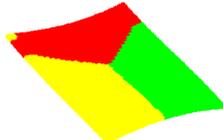
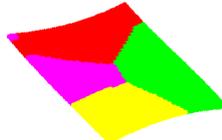
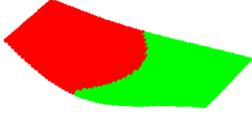
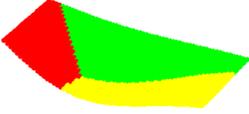
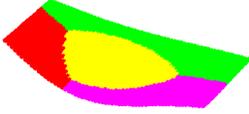
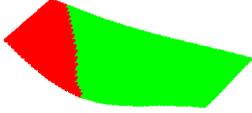
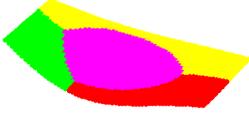
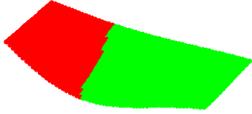
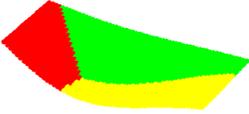
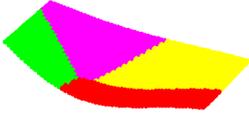
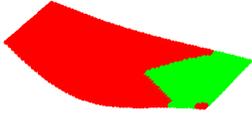
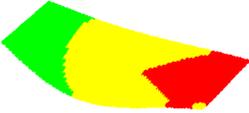
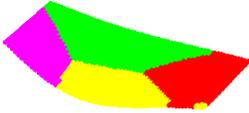
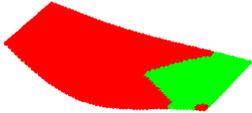
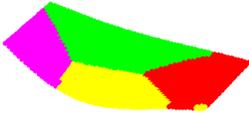
	2 patches	3 patches	4 patches
[S N] Squared Euclidean			
[S 2N] Squared Euclidean			
[S N] cosine			
[S N RBR] Squared Euclidean			
[S RBM] Squared Euclidean			

Table 3-10 Partitioning results for surface 4

	2 patches	3 patches	4 patches
[S N] Squared Euclidean			
[S 2N] Squared Euclidean			
[S N] cosine			
[S N RBR] Squared Euclidean			
[S RBM] Squared Euclidean			

The results obtained using the different feature vectors show that not all the partitioning groupings produce surface patches that can be machined easily. The results obtained in these tests show that the inclusion of the Rolling Ball Radius in the feature vector has a negative impact on the partitions. The results obtained using this feature present irregular patches that are not appropriate for machining, thus, this feature is no longer considered for further tests. The different variations of the feature vector formed by the surface coordinates and normals constantly result in regular shapes and good partitions. These variations will be evaluated in simulations tests to determine an appropriate approach for 3+2-axis machining. These tests are presented in Chapter 6.

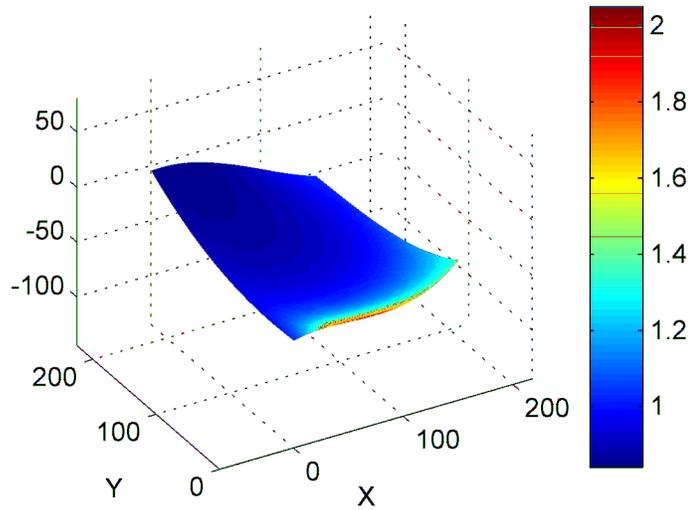
### **3.6 Side-step subdivision method**

While implementing the clustering algorithms it came to light that there are certain parameters, such as the side-step distance, that are function of some of the properties considered for partitioning. The side-step depends on the curvature of surface, the effective radius of the tool and the surface finish. The effective tool radius is a function of surface normal and tool axis. If the side-step for a particular tool inclination is calculated at all points on the surface, every point should have a different side-step as the normal varies at these points, as shown in Figure 3-13. By processing this side-step, surface points that have similar surface normals can be identified. This is the basic idea behind using the side-step as a feature for partitioning.

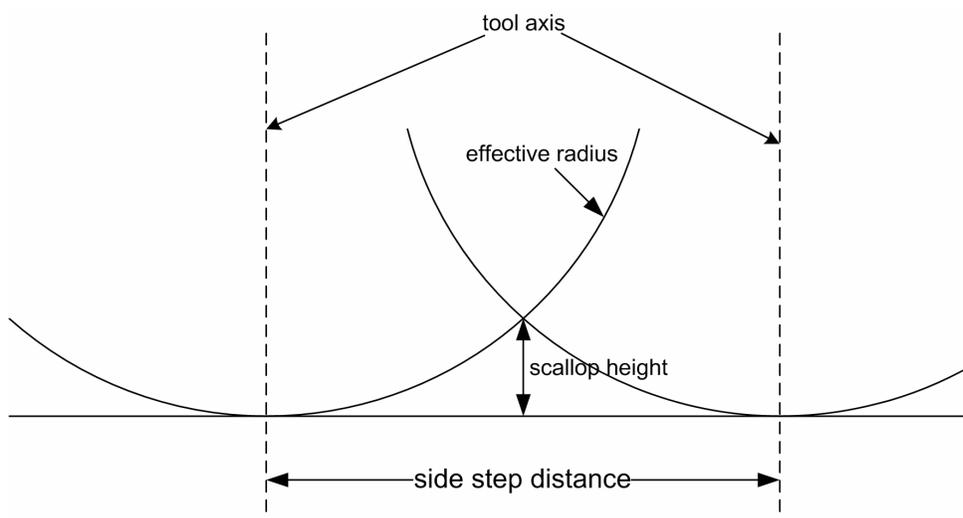
The side-step distance is defined by finding the shortest distance between passes so that the largest scallop height is equal to a user-specified tolerance, as shown in Figure 3-14. An approximation of the side-step distance can be obtained using the effective radius of the tool and the scallop height (Equation (3.22) and (3.23)).

$$effective\_radius = \frac{R}{\sin(\theta)} + R_{insert} \quad (3.22)$$

$$sidestep\_dist \approx 2 * \sqrt{2 * effective\_radius * scallop\_height} \quad (3.23)$$



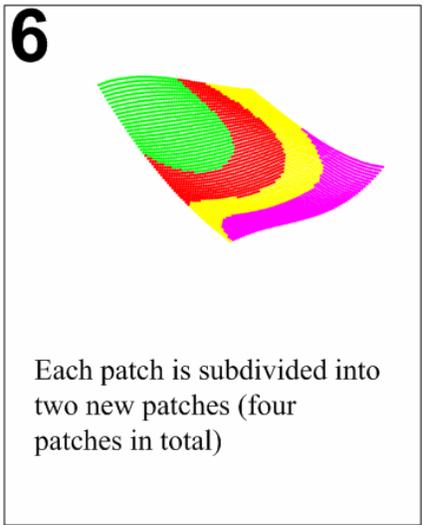
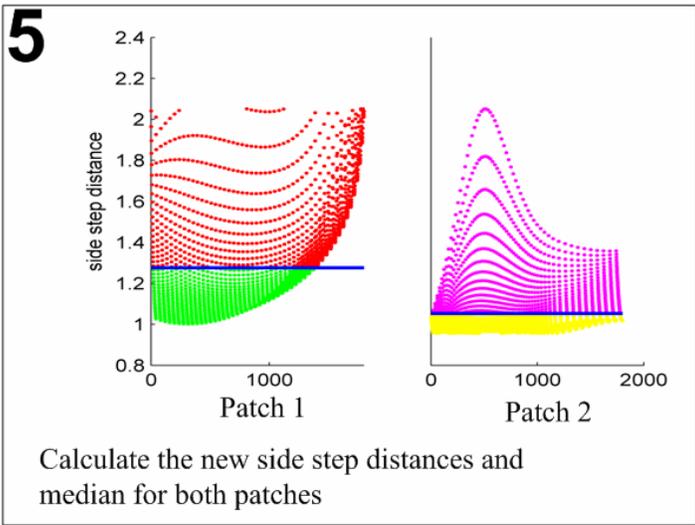
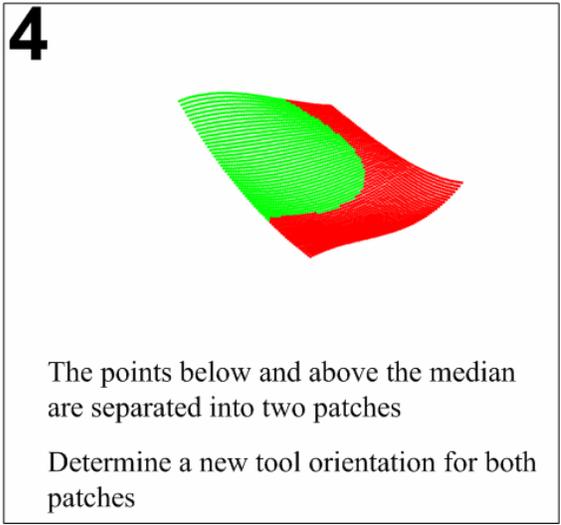
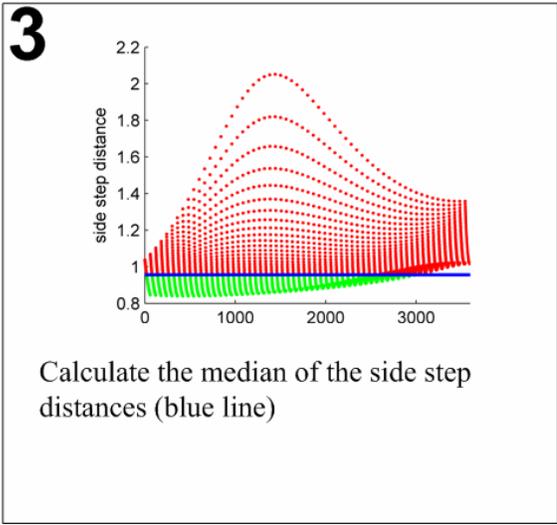
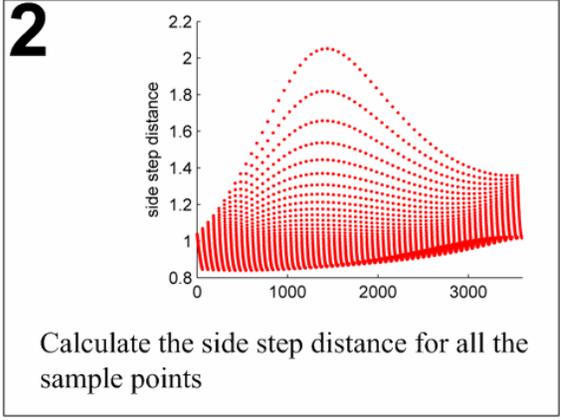
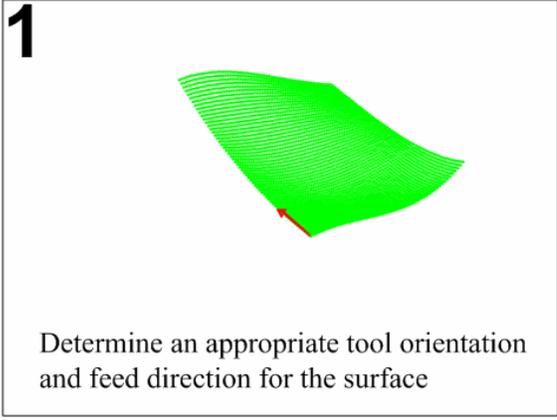
**Figure 3-13 Side-step distance (cm) for surface 1**



**Figure 3-14 Side-step distance**

The process of subdivision starts by calculating the side-step distance for all the surface points for a specified tolerance or scallop height. Then, the median of the side step is calculated, which divides the higher half of the samples from the lower half. All the points above and below the median are divided into two separated patches. Since the side-step calculation depends on the inclination angle between the surface normal and the tool axis, this new grouping of points has normal vectors that form a tighter cluster. The new groupings of points are used to define new patch boundaries and a new tool inclination angle is calculated for each patch. The subdivision process is then repeated until the user-defined number of patches is obtained.

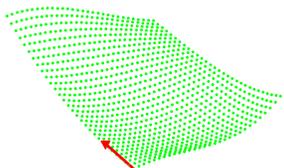
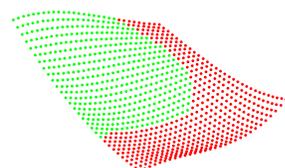
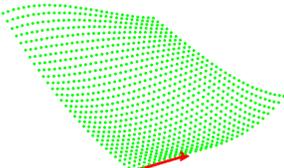
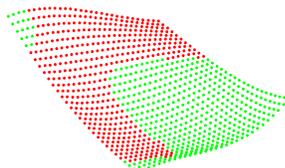
The best way to explain the side-step subdivision method is with a partitioning example. This process is explained graphically in Figure 3-15, where a complex surface is partitioned into four patches. The subdivision process starts by determining the tool orientation and feed direction for the surface, as shown in step 1. Using these parameters the side-step distance for all the sample points is calculated. The first subdivision is conducted by grouping all the points below and above the median as illustrated in the fourth step. Once the two patches are identified, a new set of tool orientations and feed directions are calculated for each patch. Then, the side-step distance is calculated for all the points inside a patch and the subdivision process is conducted again. This process is repeated until the surface is partitioned into the specified number of patches.



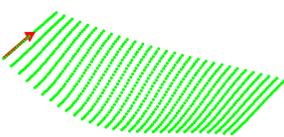
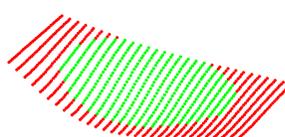
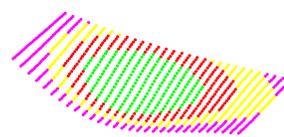
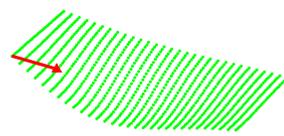
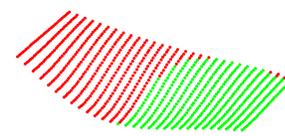
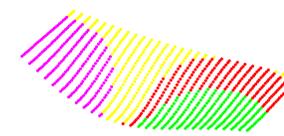
**Figure 3-15 Side-step subdivision method**

Additional tests were conducted to evaluate the efficiency of the proposed method for the partitioning of surfaces. Table 3-11 and Table 3-12 show the results obtained using the side-step subdivision method for surface 1 and surface 4, respectively. Although the results obtained in the previous example showed reasonable partitions, the results obtained in the new tests showed irregular shapes that are impractical from a machining perspective. Examples of these irregular shapes include long but not wide patches, patches inside other patches, isolated points and disjointed patches.

**Table 3-11 Side-step subdivision method applied to surface 1**

Feed direction	1 patch	2 patches	4 patches
$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$			
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$			

**Table 3-12 Side-step subdivision method applied to surface 4**

Feed direction	1 patch	2 patches	4 patches
$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$			
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$			

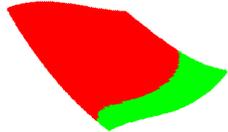
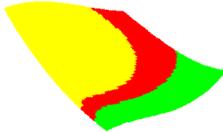
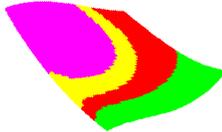
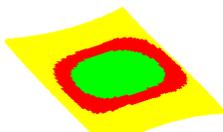
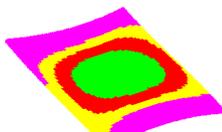
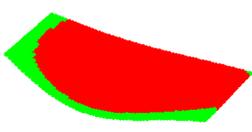
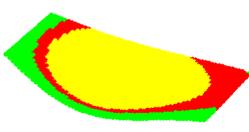
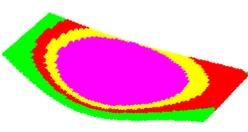
To reduce the probability of obtaining irregular patches using the side-step subdivision method, a modified version of the method was developed. Two changes were implemented in this modified version. The first change is the implementation of a feature vector combining the side-step distance and the surface coordinates. By including the surface coordinates the feature vector carries information related to the position of the point in space that can help to reduce the occurrence of irregular patches. The second change consists of using the k-means clustering algorithm instead of the median. Compared with the previous method that used a fixed number of points to subdivide the surface, the use of the k-means helps to identify points that have a similar side-step using a distance function.

Just as in the previous version, the partitioning process in the modified side-step subdivision method starts by calculating the side-step distance for all the sample points of the surface to be machined. The side-step distance and the surface coordinates are grouped into the feature vector that is used as the input for the k-means clustering algorithm. The surface is partitioned into two patches (the green and red patches in the examples shown in Table 3-13) using this feature vector. In this modified version, the points with a large side-step are grouped into the first patch (the green patch), which is kept together for the remainder of the process. The sample points from the second patch (the red patch) are used to determine a new tool orientation and side-step distances. This patch can then be partitioned into two new patches to form three patches in total. Patch 1 (green) remains the same, but patch 2 (red) is subdivided into two new patches; patch 2 (red) and patch 3 (yellow). The points with the larger side-step are kept together (patch 2) and further partitions must be made on the other patch (patch 3). The process of sub-division can be repeated until a user-defined number of patches are generated.

Table 3-13 shows the results of the improved version of the side-step subdivision method in four test surfaces. These surfaces are partitioned into 2, 3 and 4 patches. The results show that the addition

of the surface coordinates into the feature vector reduced the occurrence of disjointed patches, yet, it did not eliminate the problem of generating irregular shapes. Irregular shapes such as the green patch in surface 4 are impractical from the machining point of view, as this patch requires several short tool passes that limits achieving the full feed rate. Another example of irregular shapes that are difficult to machine includes patches inside other patches, curved boundaries, and long strips. Although the idea and the fundamentals of this technique were promising in the beginning, the occurrence of irregular patches in the tests proved that this method is not viable for 3+2-axis machining.

**Table 3-13 Surface partitioning using the modified side-step method**

	2 patches	3 patches	4 patches
Surface 1			
Surface 2			
Surface 3			
Surface 4			

### 3.7 Summary

In this chapter a comprehensive analysis for surface partitioning was conducted. Partitioning tests demonstrated that the k-means clustering method can successfully subdivide a surface and is the foundation of the partitioning scheme proposed in this work. Additional tests were conducted to evaluate different surface properties available for clustering. These tests showed that partitioning depends on the geometric properties that form the multi-dimensional vector used applied to the clustering algorithm. The effect of various geometric properties was studied on sample surfaces and a list of properties belonging to three categories were identified, namely proximity, orientation and shape. It was shown that although these properties can be grouped in various combinations and with varying weights, the combination of the surface coordinates and the normal vector consistently results in good partitions.

Table 3-14 summarizes the partitioning results obtained using different feature vectors. This table examines the frequency of occurrence of patches that can complicate the generation of tool paths. This study shows that the best results were obtained using the  $[S N]$  feature vector. The tests showed that the application of weights on the features complicates the partitioning process due to the increased number of solutions and the necessity to conduct calibration tests to prevent overemphasized features. Additional features such as the Rolling Ball Radius (*RBR*) and the side-step distance resulted in irregular and impractical patches, and accordingly will not be considered in further tests.

**Table 3-14 Number of unsuitable patches (12 tests in total)**

	[S N]	[S 2N]	[S N RBR]	[S RBR]	[S side-step]
<b>Disjointed</b>	0	3	0	1	1
<b>Isolated points</b>	0	0	8	6	0
<b>Long strips</b>	1	4	0	1	10
<b>Non-convex</b>	2	2	8	6	9
<b>Sharp edges</b>	0	0	9	8	1

The tests conducted in this chapter identified features that are appropriate for clustering, they are: the surface coordinates and the surface normals. It also identified the k-means clustering algorithm as the effective method for partitioning. However, one parameter still remains to be identified, namely, the number of sub-divisions or patches. In this work a partitioning is considered good if it leads to the least amount of time required to machine the surface. This principle is used as the basis for developing a method to determine the optimal number of sub-divisions. This process is described later in Chapter 6.

The results obtained from the clustering algorithm are in the form of a labelled matrix that contains only the number of patches that the sample point belongs to. However, this table does not provide information related to the boundaries of the patches. To generate tool paths, it is necessary to develop a method that can determine if a contact point belongs to the patch to be machined. In the next chapter different methods to classify the contact points are evaluated. This study compares the classification methods and various methods to represent the surface patches to develop an appropriate approach for the identification of the boundaries of the patches.

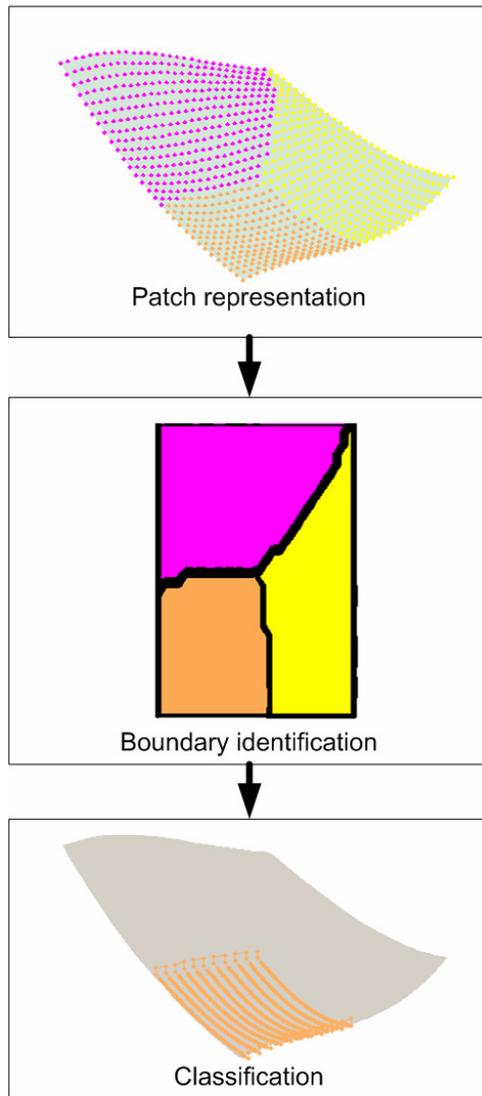
## Chapter 4

### Patch boundaries

In the surface partitioning process a surface is divided into patches so that in each patch the surface variation is minimum. After partitioning a surface, a tool path must be generated for each patch. As will be seen in section 5.7, this step requires identifying the boundaries of each patch and a classification of the cutter contact points to determine if they belong to the patch to be machined.

The process required to identify the boundaries of the patches is presented in Figure 4-1. This process starts by determining a method to represent the surface patches. Usually the shapes of the patches are complex, thus, it is necessary to find a method that can represent the shape of the patch adequately. A surface patch can be represented using the entire group of sample points, a random group of sample points, the mean or other more sophisticated methods. Once a proper way to represent the surface patches is determined, the boundaries of the patches are calculated. These boundaries are used to classify the cutter contact points in the tool path generation phase. The tool path is generated by calculating cutter contact points that define the trajectory of the tool. The contact points need to be evaluated to determine whether they belong to the patch to be machined or not. The correct identification of boundaries allows machining the surface properly by assigning the contact points to the correct patch.

This chapter presents the application of well known methods from the Pattern Recognition field and their adaptation to surface machining. It also presents newly developed methods by the current author. For completeness, a short description of each method is presented. A list of these methods is shown in Table 4-1. Each method is evaluated numerically and graphically to determine an appropriate approach. This assessment is conducted using the evaluation criteria also listed in Table 4-1.



**Figure 4-1 Boundary identification**

**Table 4-1 List of methods applied for the identification of boundaries**

	<b>Patch Representation</b>	<b>Boundary Identification</b>	<b>Classification Methods</b>
<b>Known methods</b>	1. Sample points 2. Random points 3. Mean 4. Mean + covariance	1. Voronoi 2. k-NN (nearest neighbour) 3. MICD	1. MICD 2. Nearest neighbour (sample points)
<b>Developed</b>	1. Corner points 2. Boundary points 3. Clusters		1. Nearest neighbour (random) 2. Nearest neighbour (clusters)
<b>Evaluation Criteria</b>	1. Ability to represent complex shapes 2. Memory Requirements 3. Complexity of algorithm	1. Speed of computation 2. Accuracy 3. Complexity of algorithm	1. Speed of computation 2. Accuracy

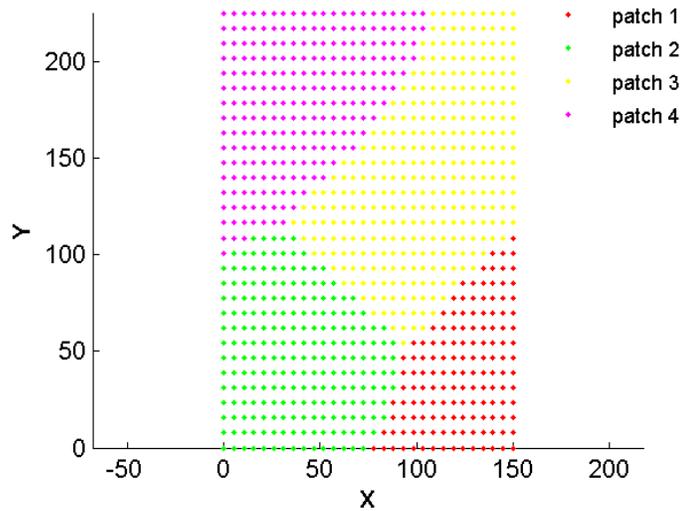
This chapter presents a series of tests to evaluate various methods to represent a surface patch, and classification techniques to define the boundaries. The tests are applied on the sample test surface 1, which was selected because it resembles typical surfaces in tools and dies.

#### 4.1 Patch representation

In the partitioning process the sample points are grouped into clusters. Each sample point is unique and can only belong to one patch, as shown in Figure 4-2. A sample partitioning output file is shown in Table 4-2. Each row in the table represents the point coordinates and the patch containing it.

**Table 4-2 Sample labeled matrix**

$S_x$	$S_y$	$S_z$	Patch
0	0	-47	2
0	9.3	-52.36	2
0	18.75	-56.52	2
⋮			
⋮			
150	225	-52	3



**Figure 4-2 Sample surface partitioned into 4 patches**

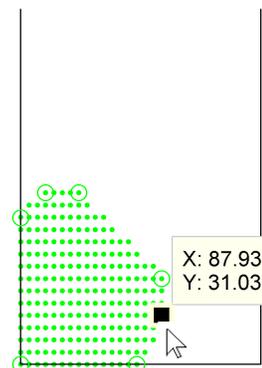
This section presents different approaches to representing surface patches, as well as a short description of each method. A comparison of these methods is presented at the end of this section.

### 4.1.1 Entire group of sample points

This is a simple and accurate method to represent patches. Despite the fact that the use of all the sample points or a large portion of these points can represent adequately the surface patch, this often results in a computationally expensive classification. An alternative to address this issue could aim at reducing the number of points while trying to maintain the integrity of the patch, but it is often difficult to find a middle ground between these two criteria.

### 4.1.2 Corner points

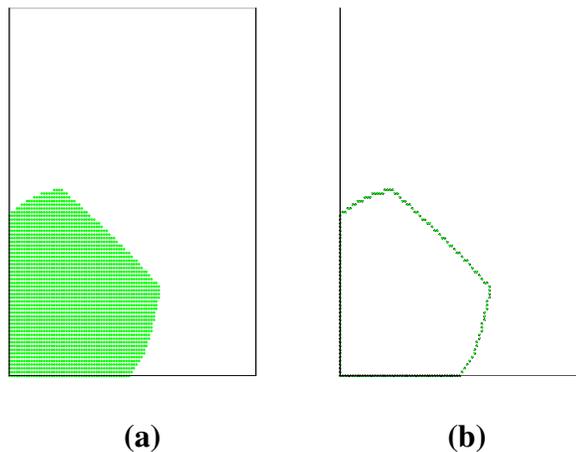
In this method the surface patch is represented using a selection of points determined by the user. Initially, the user is limited to selecting four points from the sample points. These points are used to define a quadrilateral that approximates the shape of the patch. This method worked well in the initial tests, but it was limited to simple shapes. An improvement of this method allowed the user to select more points, as shown in Figure 4-3. In this example, the user selected six points that lie on the boundary and can represent adequately the surface patch. Although, increasing the number of points can improve the performance, this approach is limited as the process requires complex algorithms to be applied without user interaction.



**Figure 4-3 Corner points selected by the user**

### 4.1.3 Boundary points

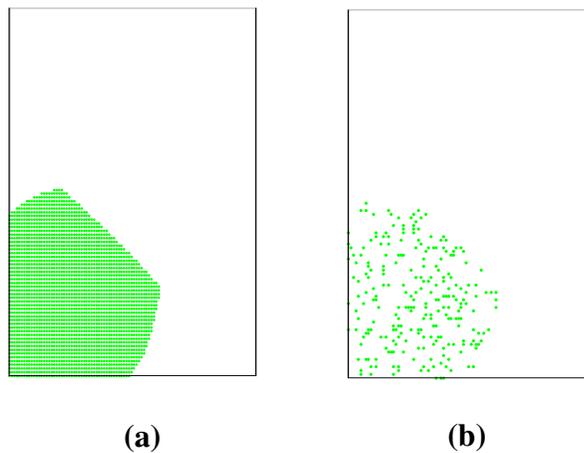
The method developed by this author to extract the patch boundaries uses the fact that the sample points used for clustering lie along a rectangular grid in the parametric space. The process starts by scanning the points defining a patch. The points belonging to a patch are processed to identify whether a point is inside or on the boundary. If a point has four neighbours that all belong to the same patch, it is classified as lying inside the patch otherwise it is classified as lying *on the boundary*. The boundary points are extracted and grouped into a boundary group, as illustrated in Figure 4-4. Once the boundary groups are identified, the test points are processed to identify which patch they belong to using the nearest neighbour method. While this method maintains a good accuracy for classification and reduces the time required for calculations, its implementation is complicated due to the fact that not all the surfaces have a regular grid. For surfaces that are not defined on an even grid (e.g. scanned data) the boundaries cannot be easily identified and the complexity of the method increases.



**Figure 4-4 Patch representation using boundary points**

#### 4.1.4 Random selection of points

One simple option to reduce the number of points is the random selection of a percentage of the original sample points. This can be easily implemented and does not require complicated algorithms. This method only requires determining a percentage of points that will be used to represent the patch. Figure 4-5 shows an example of a patch represented using five percent of the original points selected randomly.

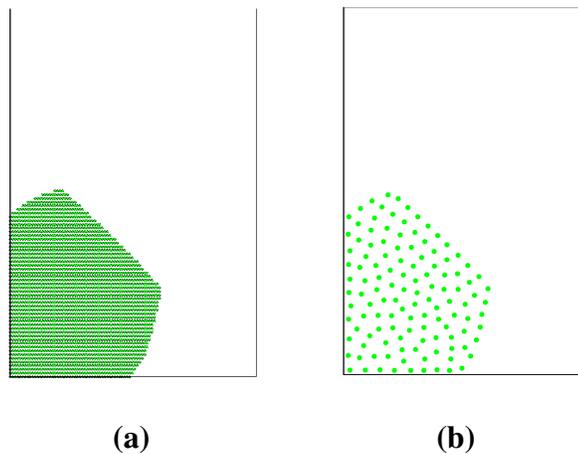


**Figure 4-5 Patch representation using random selection of points**

#### 4.1.5 Clusters method

Clustering is defined as “*the process of organizing objects into groups (clusters) whose members are similar in some way*” [39]. Using the cluster centers, a large amount of sample points can be represented using a reduced number of sample points. Based on the characteristics of the clustering algorithms described in the previous chapter, it was decided that the k-means is a potential method as it is easy to implement and has a fast convergence. The k-means can handle large number of sample points and provides a cluster center that can be used to represent a group of points.

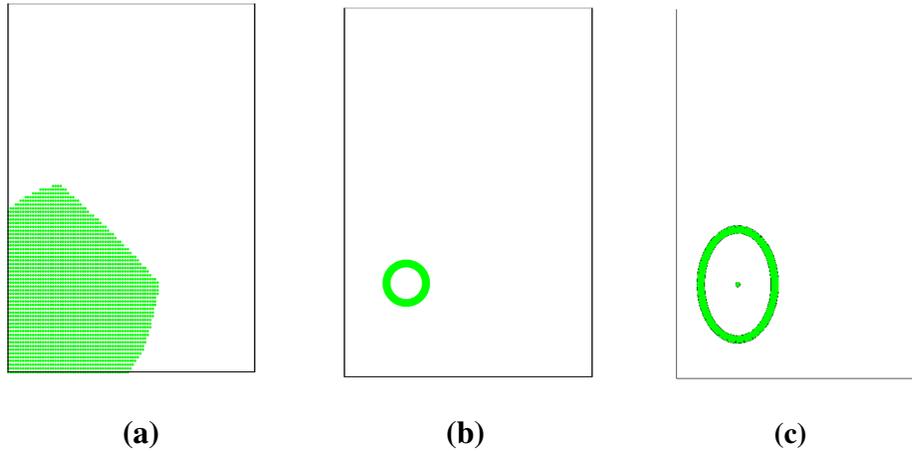
An example of a patch represented using the k-means is shown in Figure 4-6. This process starts by defining the number of clusters that will be generated. This number can be defined using a small percentage of the original number of points, two percent for this test. Once the number of clusters is defined, a partitioning test is conducted. The feature vector for this test is formed by the surface coordinates of the sample points for each patch. The cluster centers obtained from the partitioning test are used as the new points used to represent the surface patch.



**Figure 4-6 Patch representation using the clusters method**

#### **4.1.6 Statistical methods**

The mean and the covariance are two common statistics used to represent data. Using the mean and the covariance, the distribution of sample points for each class can be approximated by a circle or an ellipse (for a 2-dimensional example), respectively. Figure 4-7 shows the patch representation using these two statistics. The mean is represented by a point in the middle of the given set. The position and size of the ellipse are defined using the mean and covariance of the class, respectively. The covariance is the region that represents all the points that fall within one standard deviation around the centroid.



**Figure 4-7 Patch representation using statistical tools**

For a parametric surface the mean vector of class  $i$  is given by

$$\mu_i = \begin{bmatrix} \overline{Sx}_i \\ \overline{Sy}_i \end{bmatrix} \quad (4.1)$$

where  $\overline{Sx}_i$  and  $\overline{Sy}_i$  are given by

$$\overline{Sx}_i = \frac{1}{n} \sum_{i=1}^n Sx_i \quad (4.2)$$

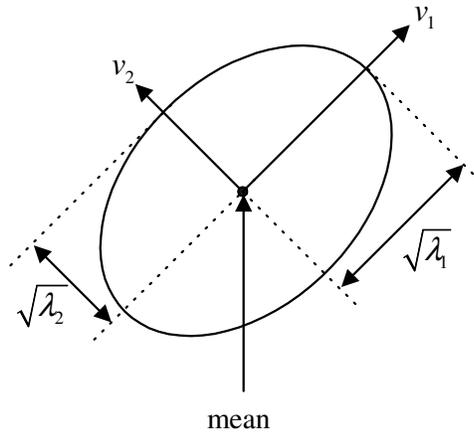
$$\overline{Sy}_i = \frac{1}{n} \sum_{i=1}^n Sy_i \quad (4.3)$$

The covariance is calculated from

$$\Sigma_i = \sum_{i=1}^n \frac{(Sx_i - \overline{Sx}_i)(Sy_i - \overline{Sy}_i)}{n} \quad (4.4)$$

To represent the ellipse graphically, the parameters are defined using the eigenvalues and eigenvectors of the covariance of the sample points, as shown in Figure 4-8. The center of the ellipse

is positioned using the mean vector of the cluster. The direction and length of the major axis are defined by the eigenvector and eigenvalues, respectively.



**Figure 4-8 Parameters of the ellipse**

Eigenvalues ( $\lambda$ ) are natural frequencies associated with linear transformations [46]. The real number  $\lambda$  is called an eigenvalue of  $\Sigma_i$  if there exists a nonzero vector  $v$  such that

$$\Sigma v = \lambda v \tag{4.5}$$

Every nonzero vector  $v$  satisfying the equation is an eigenvector of  $\Sigma_i$  associated with the eigenvalue  $\lambda$  [47].

#### 4.1.7 Results of patch representation

In Table 4-3 a comparison of the methods to represent the surface patches of the test surface is given. This table lists the different methods presented in this section and a briefly description of the main advantages and disadvantages. Based on the analysis presented in this table, three methods will no longer be considered for the representation of patches. The remaining methods will be evaluated in

terms of accuracy and computational speed to determine an appropriate approach for the classification of cutter contact points.

**Table 4-3 Comparison of methods to represent the surface patches**

<b>Method</b>	<b>Advantages</b>	<b>Disadvantages</b>	<b>Feasible?</b>
Sample points	<ul style="list-style-type: none"> <li>• Easy to implement, accurate</li> </ul>	<ul style="list-style-type: none"> <li>• Computationally expensive</li> </ul>	<i>Yes</i>
Corner points	<ul style="list-style-type: none"> <li>• Patch can be represented with few points</li> </ul>	<ul style="list-style-type: none"> <li>• Complex algorithms to automate the process</li> </ul>	<i>No</i>
Boundary points	<ul style="list-style-type: none"> <li>• Accurate, reduced number of points to represent the patch</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult to implement</li> </ul>	<i>No</i>
Random points	<ul style="list-style-type: none"> <li>• Simple algorithm, easy to implement</li> </ul>	<ul style="list-style-type: none"> <li>• May result in misclassified points on the boundaries</li> </ul>	<i>Yes</i>
Clusters	<ul style="list-style-type: none"> <li>• Accurate, reduced number of points to represent the patch</li> </ul>	<ul style="list-style-type: none"> <li>• Requires implementation of a clustering algorithm</li> </ul>	<i>Yes</i>
Mean	<ul style="list-style-type: none"> <li>• Simple algorithm, minimum memory requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Works only for simple shapes</li> </ul>	<i>No</i>
Mean + covariance	<ul style="list-style-type: none"> <li>• Simple algorithm, minimum memory requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Approximates the shapes</li> </ul>	<i>Yes</i>

## 4.2 Classification distances

The distance measure for features is of critical importance for all classification methods [48]. A variety of distance measures exists for different purposes [49] [50]. Two widely used measures are the Euclidean distance and Mahalanobis distance. These distances are widely used in cluster analysis and classification techniques [45]. The Euclidean distance is a distance metric based on the Pythagorean Theorem [51]. The Euclidean distance between two points is the length of the path connecting them. The Mahalanobis distance is a type of weighted Euclidean distance where the weighting is determined by the range of variability of the sample points (expressed by the covariance matrix) [48]. The basics of these distance measures are presented below. A more detailed introduction can be found in [37] [52].

The Euclidean distance is given by

$$d_i^{Euclidean} = \sqrt{(x - \mu_i)^T (x - \mu_i)} \quad (4.6)$$

where  $\mu_i$  represents the mean vector of class  $i$  and  $x$  represents the sample vector to classify. The

Mahalanobis distance is given by

$$d_i^{Mahalanobis} = \sqrt{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)} \quad (4.7)$$

where  $\Sigma_i^{-1}$  represents the inverse of the covariance matrix of class  $i$ .

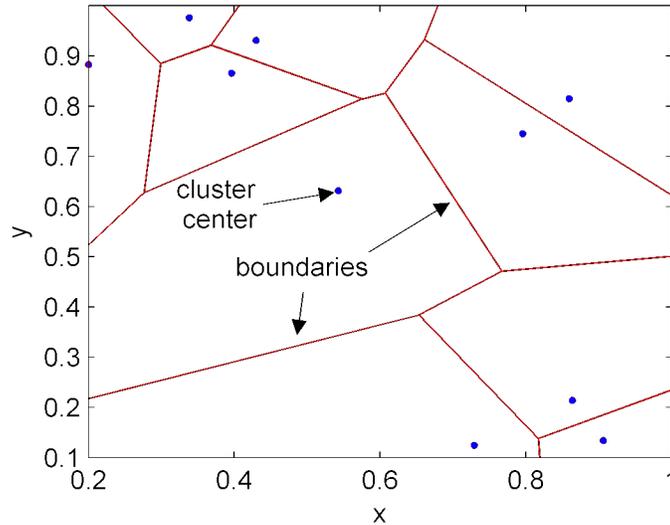
## 4.3 Classification methods

### 4.3.1 Voronoi diagrams

Voronoi diagrams are widely used in diverse fields. Eppstein [53] noted that Voronoi diagrams tend to be involved in situations where a space should be partitioned into "*spheres of influence*". Voronoi diagrams represent the region of influence for a given set of points [54]. For each point  $P$  in a set  $S$ , a boundary enclosing all the intermediate points lying closer to  $P$  than to any other point  $Q$  in the set  $S$  is called a Voronoi polygon. The set of all Voronoi polygons for a given point set is called a Voronoi diagram [55]. In the Voronoi diagram, a Voronoi region consists of all points that are at least as close to a site as to any other site [56]. An example of a Voronoi diagram with 10 points is illustrated in Figure 4-9. Details of this method can be found in [57], [58] and [59].

In the 3+2-axis machining strategy developed by Chen *et al.* [34], the boundaries of the patches were defined using Voronoi diagrams. Chen *et al.* used the cluster center location to represent the surface patches. This cluster center is calculated using the mean of the surface coordinates. This method assumes that the sample points are distributed about the cluster center in a circular (spherical)

manner. While the method is fast and simple to implement, its major drawback is that it only works for simple or regular shapes.



**Figure 4-9 Voronoi Diagrams**

### 4.3.2 $k$ -Nearest Neighbour algorithm ( $k$ -NN)

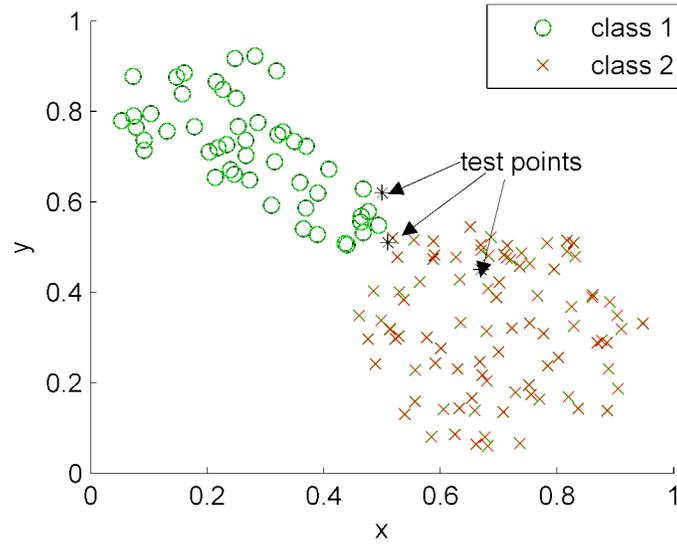
The  $k$ -Nearest Neighbour algorithm is a simple yet effective method for classification in the areas of pattern recognition, machine learning, data mining, and information retrieval [60]. The  $k$ -NN uses a distance or similarity function to find  $k$  nearest neighbours of a data point in question and classifies this data point by, usually, a majority voting over the known class labels of the nearest neighbours [61].

The  $k$ -NN has been successfully used in a variety of applications and as shown by Elkan [62] and Hayashi *et al.* [63] it can be competitive with state-of-the-art classification methods. The  $k$ -NN is defined as an instance-based learning or lazy-learning because it defers processing of training data until a test point in question needs to be classified [61], which usually involves storing data in memory and then finding a solution for a particular query.

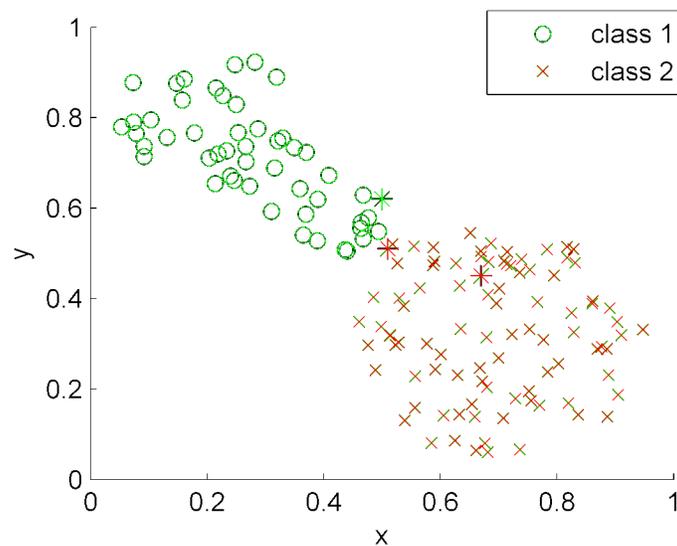
In the classification phase, a set of  $k$  data points nearest to the test point are selected. The special case where the class is predicted to be the class of the closest training sample ( $k= 1$ ) is called the nearest neighbour algorithm [64]. Conventionally, a majority rule is used in the classification phase, where the test point is assigned to the class represented by a majority of its  $k$  nearest neighbours. Other rules for classification have been proposed, such as the Distance Weighted Rule [65], where a higher weight is assigned to the nearest neighbours.

The accuracy of the  $k$ -NN algorithm depends on the selection of the proper size of  $k$ . This number is usually determined based on experimentation and user experience. While a small value of  $k$  can speed the calculations, larger values of  $k$  reduce the effect of noise in the classification, but make boundaries between classes less distinct [64]. Other considerations, such as the presence of noisy or irrelevant data, must also be considered in the selection of an optimal size of  $k$ .

Figure 4-10 presents a 2-dimensional example to illustrate the procedure required to classify points using the nearest neighbour. The process starts by identifying the sample points and their correspondent classes. In this case, three test points indicated with black asterisks need to be classified into one of the two classes. The sample points from class 1 and class 2 are represented by green circles and red crosses, respectively. To determine in which class the test points belong, the Euclidean distance between the test points and the sample points is calculated. Each test point is classified into the class where the nearest neighbour belongs, as shown in Figure 4-11.



**Figure 4-10 Sample points partitioned into two classes**



**Figure 4-11 Test points (\*) are classified into their corresponding classes using  $k=1$ .**

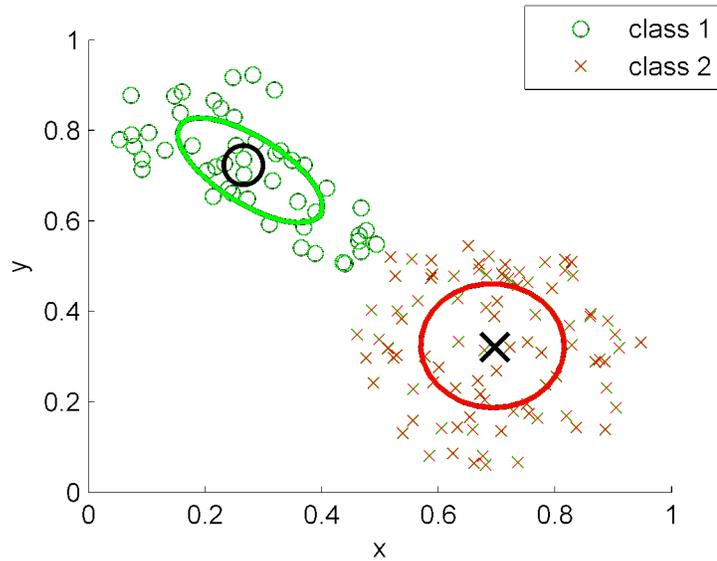
### 4.3.3 MICD

The Minimum Intra-Class Distance (MICD) method is a statistical approach used for classification. In this method the covariance matrix is computed from the sample points belonging to each class. Once the covariance of each class is calculated, the Mahalanobis distance with respect to the test point is

computed. The test point is classified into the class for which the Mahalanobis distance is minimal. Compared to the MED where the shape of the patch is approximated in a circular (spherical) manner, in the MICD method the distribution of the sample points belonging to each class is assumed to be ellipsoidal. The MICD method considers not only the distance from the test point to the cluster center, but it also considers the size and shape (distribution of sample points) for each cluster.

In the MICD method the approach is to estimate the standard deviation of the distances of the sample points from the cluster center (mean vector) [46]. If the distance between the test point and the cluster center is less than one standard deviation, it is highly probable that the test point belongs to the class. The farther away it is, the more likely that the test point does not belong to the class.

In the MICD method the distribution of sample points for each class is approximated with an ellipse. While the generation of the ellipse is not required in the classification process, it is presented as it helps to understand the procedure (Figure 4-12). For a given point to belong to a class, the test point should be closer to the center in the direction where the ellipse has a short axis, while in those where the test point is along the major axis the test point can be farther away from the center. From a mathematical basis, the Mahalanobis distance is the distance of the test point from the cluster center divided by the width of the ellipsoid in the direction of the test point [46].



**Figure 4-12 Minimum Intra-Class Distance (MICD) method**

To summarize, three classification methods were presented in this section. Because of their limitations related to the use of the mean to represent the surface patches, the Voronoi diagrams are not considered for further tests. The remaining two methods are implemented for the classification of points in the following section and are compared numerically and graphically to determine a feasible solution.

#### **4.4 Implementation of classification methods**

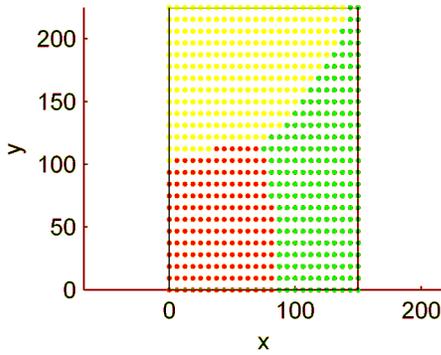
This section presents the application of two methods for the classification of contact points, namely the nearest neighbour and the MICD method. It presents the methodology required to classify the cutter contact points. In the nearest neighbour method, three methods to represent the surface patches are tested: the entire group of sample points, random points and the clusters. In the MICD method the mean and covariance are used to represent the surface patches.

#### 4.4.1 Nearest neighbour algorithm

Figure 4-13 shows the procedure required to classify contact points using the nearest neighbour algorithm. In this method the distances between the test point and the sample points are calculated. The smallest distance is then stored as the classification distance for that point. The test point is classified into the group that contains the point that is closest to the test point.

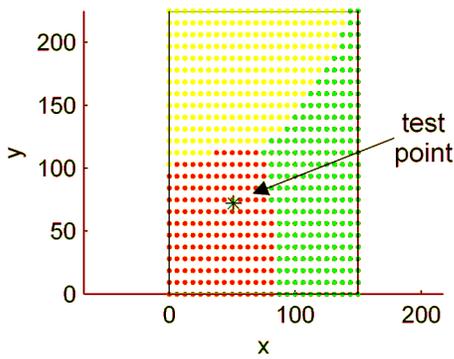
The selection of the parameter  $k$  has an influence on the accuracy of the  $k$ -NN method. Based on the characteristics of the classification data obtained from the surface patches, it was determined that a value of one for  $k$  could result in good classifications. This method is also known as the nearest neighbour method. The selection of a small number for  $k$  can speed the calculation, which is an important requirement in this work, and can work with data that has minimum noise or outliers, as in the case of the classification data obtained from the partitioning tests.

1. Every patch is represented by labelled sample points.



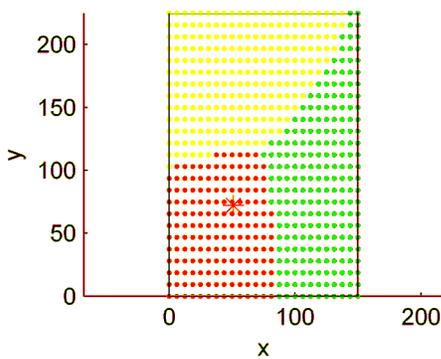
x	y	patch
0	75	2
0	84.375	2
0	93.75	2
0	103.125	3
0	112.5	3
0	121.875	3

2. The distance from the test point and all the sample points is calculated.



x	y	patch	dist
0	0	2	88.23265
0	9.375	2	80.76441
0	18.75	2	73.73305
0	28.125	2	67.27567
0	37.5	2	61.57313
0	46.875	2	56.85302

3. The sample point that results in the smallest distance is used to identify the patch that contains the test point.



x	y	patch	dist
50	46.875	2	25.14489
50	56.25	2	15.78171
50	65.625	2	6.452955
50	75	2	3.162278
50	84.375	2	12.41534
50	93.75	2	21.77298
50	103.125	2	31.14106

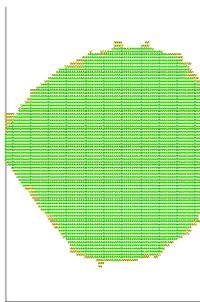
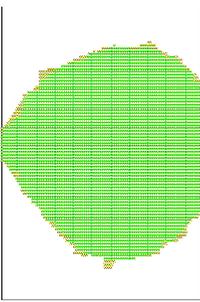
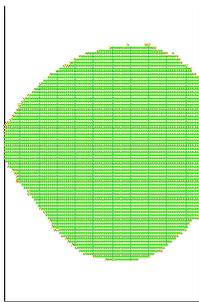
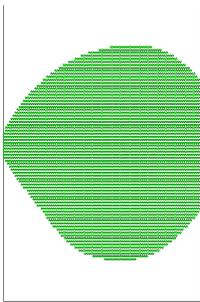
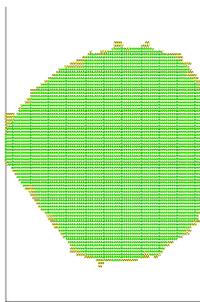
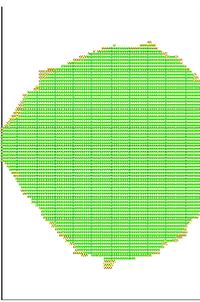
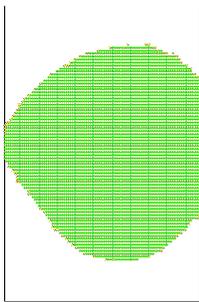
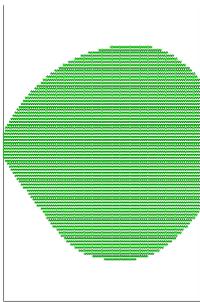
Figure 4-13 Classification process using the nearest neighbour method

The nearest neighbour algorithm is easy to implement, but it is computationally expensive, especially when the size of the classification data grows. This method uses an exhaustive process that searches for the nearest neighbour among the classification samplers. This type of brute-search method could become severe in high-dimensional feature spaces (the number of operations is proportional to  $k \cdot \text{number of samples}$ ) [45].

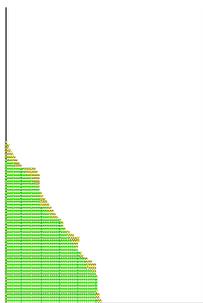
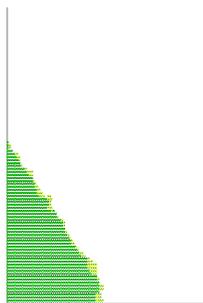
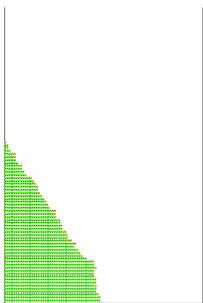
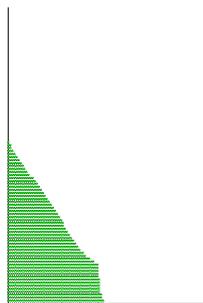
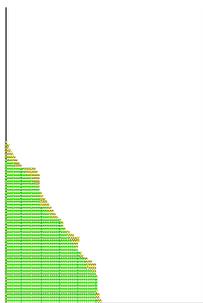
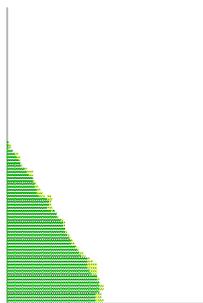
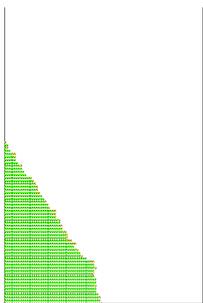
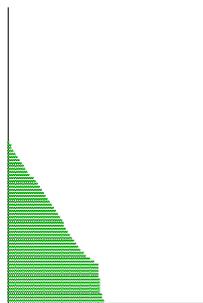
The efficiency of the nearest neighbour method depends on the points used to represent each patch. While the use of all the sample points or a large portion of these points can represent adequately the surface patch, this can result in a computationally expensive classification. An alternative to address this issue could aim at reducing the number of points while trying to maintain the integrity of the patch.

Tests were conducted to evaluate the effect of using a fraction of the original points. In these tests, shown in Table 4-4 and Table 4-5, different percentages were used to define the patches with the objective of establishing a guide to selecting a range of percentages that could be used in other tests. The results show a high accuracy and considerable time savings over the use of the entire classification data. These results validate the application of this modified version of the nearest neighbour and its implementation in further tests.

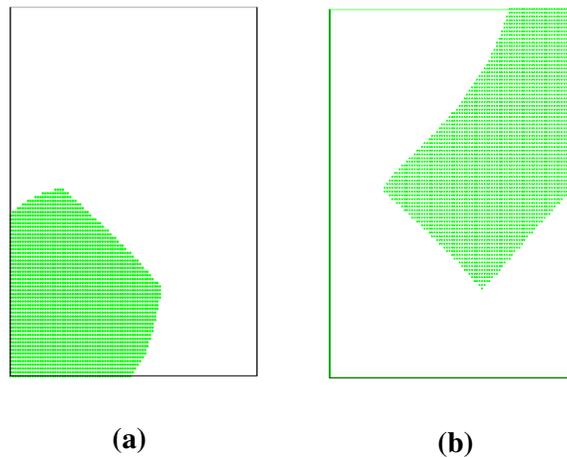
**Table 4-4 Comparison using different percentages of points on sample patch 1**

Percentage of points used	5 %	10 %	30 %	100%
Correctly classified (green)				
Incorrectly classified (yellow)				
Time	21.56 sec.	41.93 sec.	111.82 sec.	422 sec.
Number of misclassified (out of 5617)	257	211	93	0
Accuracy	95.4 %	96.2 %	98.34 %	100 %

**Table 4-5 Comparison using different percentages of points on sample patch 2**

Percentage of points used	5 %	10 %	30 %	100%
Correctly classified (green)				
Incorrectly classified (yellow)				
Time	21.03 sec.	39.30 sec.	124 sec.	411 sec.
Number of misclassified (out of 1490)	133	93	40	0
Accuracy	91.0 %	93.7 %	97.3 %	100 %

The previous tests showed the feasibility of using a fraction of the points to represent a patch. A second series of tests using the two surface patches shown in Figure 4-14 were conducted to test the method developed using the clusters method. Accordingly, a series of tests to determine the number of clusters were conducted using 2.5 %, 5 %, 10 %, 20 % and 30 % of the number of sample points for each patch.



**Figure 4-14 Surface patches used for tests. a) Patch 1. b) Patch 2**

Table 4-6 presents a comparison of two classification methods: the random selection and the clusters method. In these tests, it was observed that the time required to calculate the cluster centers does not exceed more than few seconds in the majority of cases. As the classification consumes a similar amount of time in both cases due to the fact that the same number of points is used, the computation time is not included in this comparison.

The numerical analysis shown in Table 4-6 confirmed the superiority of the k-means over the random selection of points. The results obtained with the k-means show a better performance even when a lower percentage is used. The results show that increasing the number of cluster centers does not have a significant influence on the accuracy of the method. The application of this method allows using lower percentage of points and minimizing the time required for computations. Further tests to

evaluate this method in terms of computational time and accuracy are presented next. The results are compared with other methods described in this section and are used to determine a feasible approach for the classification of points in the proposed 3+2-axis machining method presented in this work.

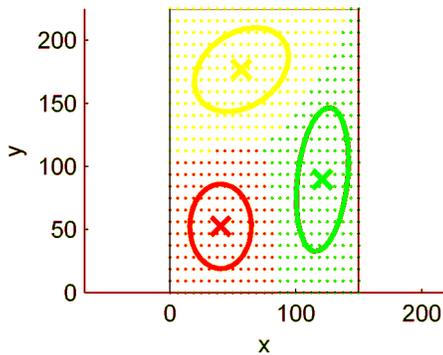
**Table 4-6 Accuracy of the two methods used for classification**

	Patch 1		Patch 2	
	Random	k-means	Random	k-means
<b>2.5%</b>	91 %	99 %	90 %	98 %
<b>5%</b>	92 %	99 %	94 %	98 %
<b>10%</b>	96 %	99 %	95 %	98 %
<b>20%</b>	97 %	99 %	97 %	98 %
<b>30%</b>	98 %	99 %	98 %	98 %

#### 4.4.2 MICD method

Figure 4-15 shows the process required to classify a contact point using the MICD method. The first step in this process is to calculate the mean and the covariance of each patch. The Mahalanobis distance (using the mean and covariance) is measured from the test point to all the surface patches. The surface patch that results in the smallest distance is used to classify the patch that contains the contact point.

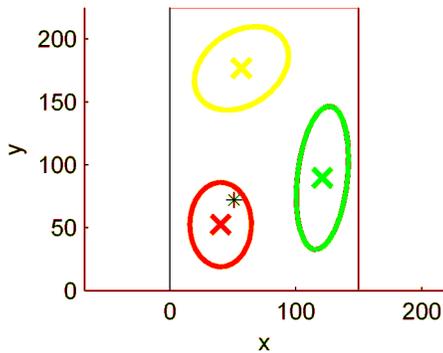
1. Every patch is represented by the cluster center and the covariance of each class.



x	y	patch
121.108	89.6591	1
40.3476	52.5333	2
56.8326	176.7346	3

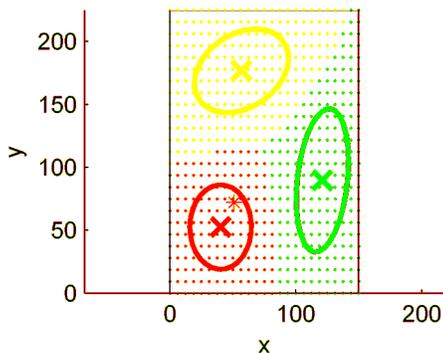
$$\Sigma_1 = \begin{bmatrix} 426.2426 & 308.5451 \\ 308.5451 & 3250.793 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 595.2188 & 3.6713 \\ 3.6713 & 1127.112 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 1394.273 & 373.3202 \\ 373.3202 & 1101.741 \end{bmatrix}$$

2. The Mahalanobis Distance from the test point and all the cluster centers points is calculated.



x	y	patch	dist
121.108	89.6591	1	3.4486
40.3476	52.5333	2	0.7243
56.8326	176.7346	3	3.2635

3. The cluster center that results in the smallest distance is used to identify the patch that contains the test point.



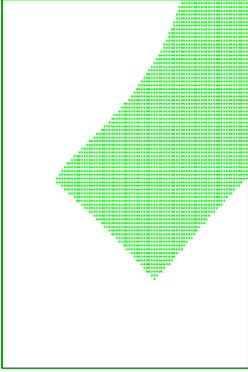
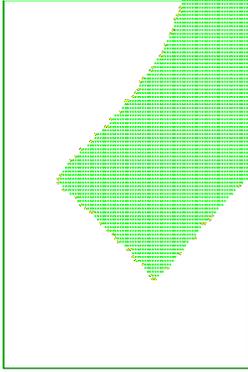
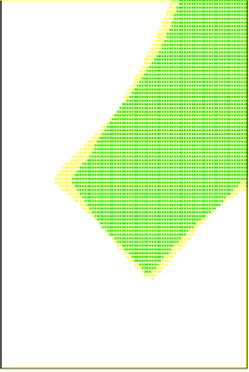
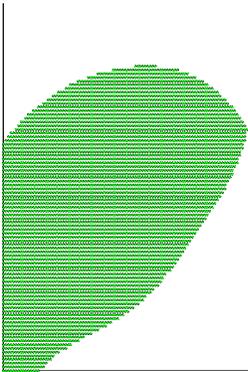
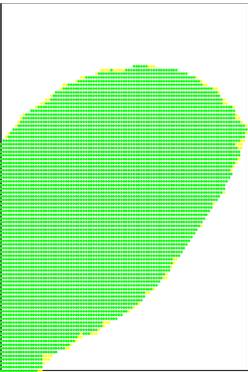
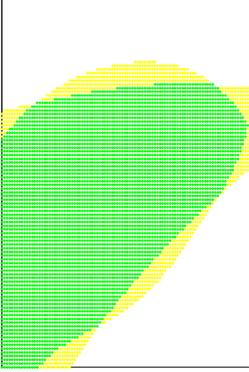
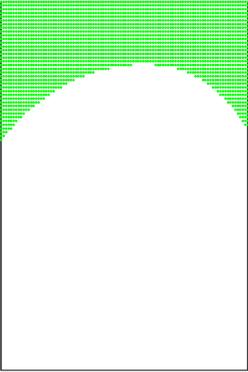
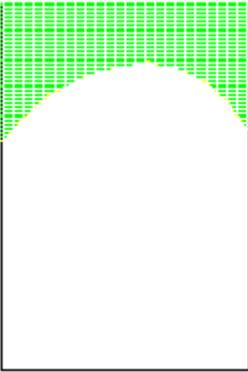
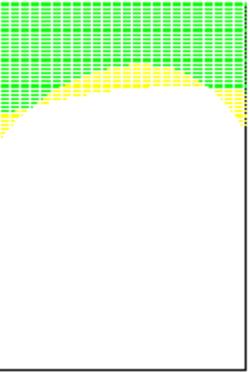
x	y	patch	dist
121.108	89.6591	1	3.4486
40.3476	52.5333	2	0.7243
56.8326	176.7346	3	3.2635

**Figure 4-15 Classification process using the Minimum Intra-Class Distance (MICD) method**

#### **4.5 Comparison between the nearest neighbour and the MICD**

The nearest neighbour and the MICD methods are evaluated here using numerical tests. In the first test the surface is divided into three patches. Using the sample points from the patch as test points, a classification is conducted using the three methods described before. The results obtained in this test are shown in Table 4-7. This table includes a graphical representation showing the correctly classified points in green and the misclassified points in yellow. Additionally, the time required for the computation and the accuracy of the method are shown below the patch. The three methods tested are: the nearest neighbour, the modified nearest neighbour (using the k-means algorithm) and the MICD method. These methods are evaluated on three surface patches, which have complicated shapes that can cause difficulties in the classification process.

**Table 4-7 Comparison of different classification methods**

Nearest Neighbour	Nearest neighbour clusters method 2.5 %	MICD
		
354 sec./100%	9.29 sec./98.3%	0.06 sec./90.3%
		
349 sec./100%	10.12 sec./98.12%	0.06 sec./83%
		
346 sec./100%	9.96 sec./98.6%	0.06 sec./80.5%

Some conclusions from this comparison are

- The MICD requires the least amount of time to classify points. The tests confirm that the MICD has a high accuracy and can be considered a good option for classification.
- Although the MICD method does not perform as well as the nearest neighbour in terms of accuracy, there are two advantages that make this method a good alternative for the classification process. The first one is the small time needed for computation, which is an essential requirement in this work due to the several tests required for each surface. The second advantage is the approximation of the patches obtained with the MICD. This approximation results in some misclassified points, but it also results in a reduction of sharp edges and curved boundaries that can lead to complications in the tool path generation process.
- The nearest neighbour has the best performance in terms of accuracy. However, it is the method that consumes the largest amount of time.
- A modified version of the nearest neighbour that reduced the number of sample points by applying a clustering algorithm was developed. This modified version performed with accuracy higher than 98% but required less than 5% of the time consumed by the original version. This method performed well in terms of accuracy and computation time, and is considered a good option for the classification process.

Two methods showed a good potential for the classification of contact points. These two methods are the modified version of the nearest neighbour using the k-means algorithm (clusters method) and the MICD method. Both methods will be considered for further tests in the remaining of this work. These two methods will be implemented in the proposed 3+2-axis machining method and will be evaluated in terms of machining time. Based on these results, a general solution for the classification process is proposed later in this work.

## 4.6 Summary

The proper identification of boundaries permits to machine surface patches adequately. The strategy ensures that the points are correctly classified and machined with appropriate tool parameters. This chapter presented a variety of approaches that were tested for the classification of points. These tests were conducted using different situations that are commonly encountered in patch-by-patch machining. Based on the results obtained in the simulation tests, two methods, the nearest neighbour using the clusters method and the MICD method proved to be effective for the identification of boundaries. The nearest neighbour is a method that showed a good performance in the classification of points. This method was modified to reduce the computation time required for the classification. The modifications reduced the number of calculations considerably, yet it did not affect its accuracy. The MICD, however, stands out for its speed of computation and its simple implementation.

The following chapter presents the methodology to generate tool paths for 3+2-axis machining. This methodology is used for machining tests in subsequent chapters. A comparison of the proposed methods for the identification of boundaries based on the machining time will be conducted in those tests.

## Chapter 5

### Tool path generation

Proper machining requires conditions that guarantee the correct part orientation and tool positioning with respect to the surface to be machined. In particular, the problem of minimizing the machining time is of significant interest to the machining industry. Thus, an optimal tool path planning strategy should aim at reducing the time required to completely machine a surface while maintaining the surface specifications.

Determining the machining setup for 3+2-axis machining is more complicated than for straight 3-axis machining. Even though the tool orientation is fixed as in 3-axis machining, the problem of selecting the tool orientation, calculating the side-step distance and determining the right tool size is similar to 5-axis machining. Although some strategies from 5-axis can be used, it is required to establish the conditions and methodologies required for its correct application in 3+2-axis machining.

This chapter presents the methodology required to generate tool paths for 3+2-axis machining. In 3+2-axis machining a surface can be partitioned and each patch can be machined using a particular feed direction, tool size, tool orientation and side-step distance. These parameters are determined independently for each patch and are generated to optimize the machining procedures. This methodology lays out the foundation for the proposed 3+2-axis machining strategy presented in this work.

#### 5.1 Projected normals plane

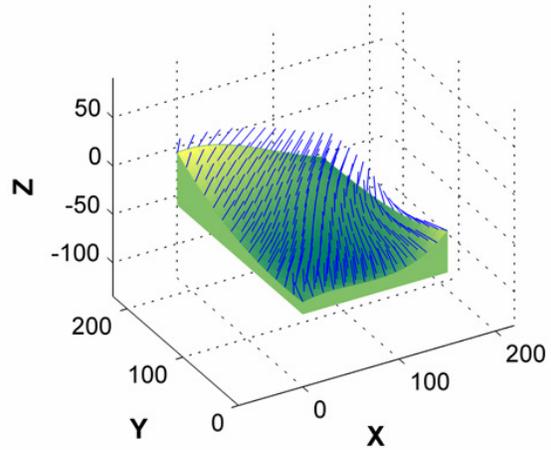
The visual representation of a surface is normally conducted by plotting the surface coordinates in a 2-D or 3-D plane that shows the location of the points in space. In this work, a new method is developed to provide a graphical representation of the surface normals in a 2-D plane. The surface

normals play an important role in the 3+2-axis machining strategy proposed in this work. Thus, the projected normals plane is developed to provide an additional graphical tool to the CNC operator for a better visual representation of the surface to be machined.

The process required to generate the projected normals plane is shown in Figure 5-1. In the first step, the surface normals are calculated for a group of sample points. Then, the surface normals are moved to a common point, in this case the origin. Depending on the orientation of the majority of the surface normals a plane can be determined. In this example the  $XY$ - plane is used and is represented by  $N_x$  and  $N_y$ . Other more convenient planes, however, could be used instead. In the last step, the tips of the projected normals are plotted in a 2-D plane.

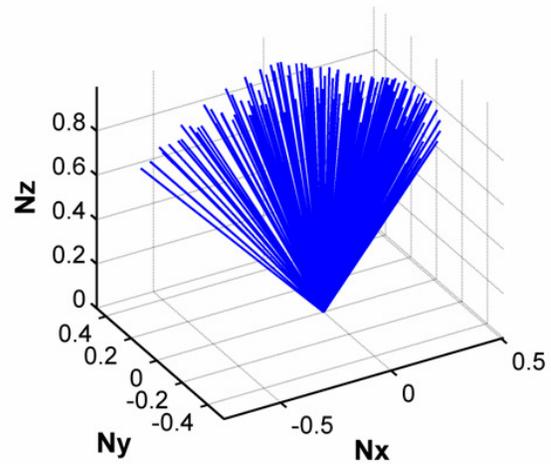
# 1

Calculate the surface normals for a surface or a patch using a group of sample points.



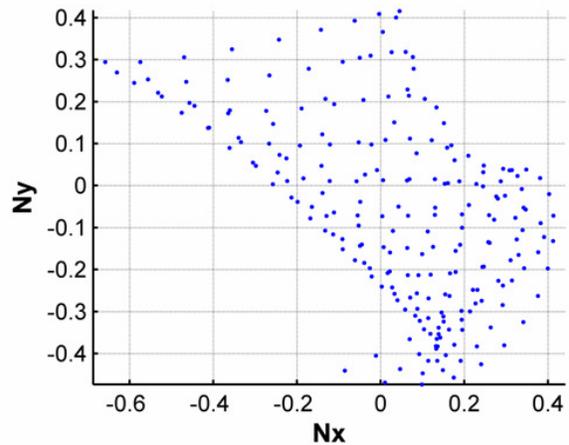
# 2

The surface normals are moved to the origin of a coordinate system.



# 3

The X- and Y- coordinates of the surface normal are plotted in a 2-D plane.



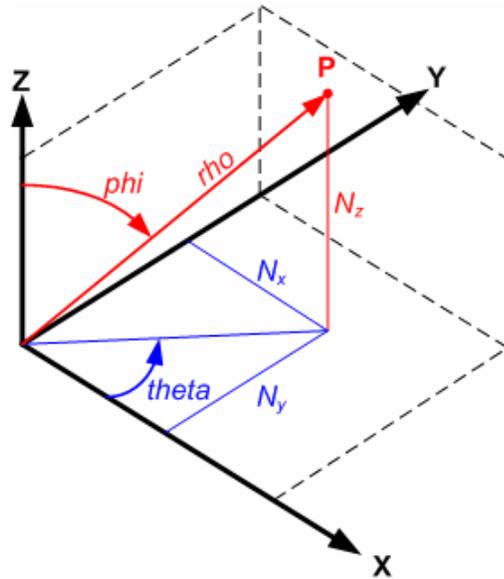
**Figure 5-1 Projected normals plane**

A modified version of the projected normals plane can be produced using the Spherical coordinate system. The Cartesian coordinates of a point  $P$ , which is the tip of the surface unit normals and is expressed in terms of  $N_x$ ,  $N_y$  and  $N_z$ , can be transformed into spherical coordinates using equations (5.1), (5.2) and (5.3). On a spherical coordinate system each point on a plane is determined by two angles ( $theta$  and  $phi$ ) and a distance ( $rho$ ), as shown in Figure 5-2. The angles  $theta$  (also known as azimuth) and  $phi$  (zenith) are angular displacements in radians measured from the positive  $X$ - axis, and the  $XY$ - plane, respectively; and  $rho$  is the distance from the origin to the point  $P$  [55].

$$theta = \arctan\left(\frac{N_y}{N_x}\right) \quad (5.1)$$

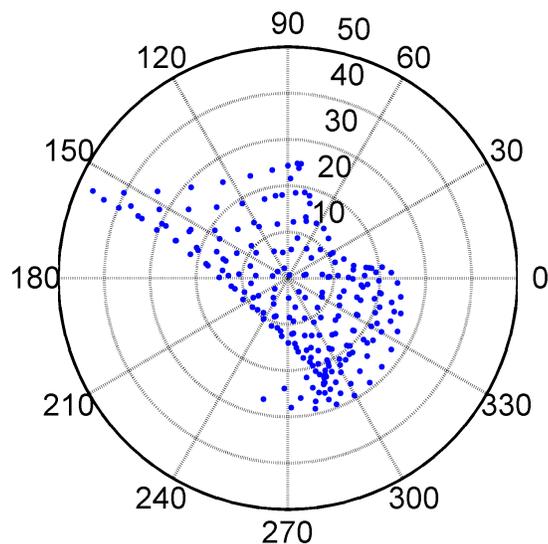
$$phi = \arccos\left(\frac{N_z}{\sqrt{N_x^2 + N_y^2 + N_z^2}}\right) \quad (5.2)$$

$$rho = \sqrt{N_x^2 + N_y^2 + N_z^2} \quad (5.3)$$



**Figure 5-2 Spherical coordinates**

This work adopts a similar approach to the one used in cartography to produce a 2-D spherical plot. As described in an article posted in “*The Casual Cartographer Newsletter*” [66], cartographers concentrate their attention on specific coordinates and have derived a two dimensional spherical coordinate system, commonly known as longitude-latitude. By assuming that most of the coordinates are on the surface of the sphere, a three dimensional coordinate system can be projected using only two coordinate values. Since all the normals in the projected surface normals plane have the same length, and lie on the surface of the sphere, the coordinate *rho* can be excluded from the coordinate system and a 2-D plot can be produced using the two angular coordinates. An example of this projection is shown in Figure 5-3. This projected plane is used to express the relationship between points in terms of angles. The two angular coordinates *theta* and *phi*, expressed in degrees, represent the angle from the positive *X*- axis and the positive *Z*- axis, respectively.

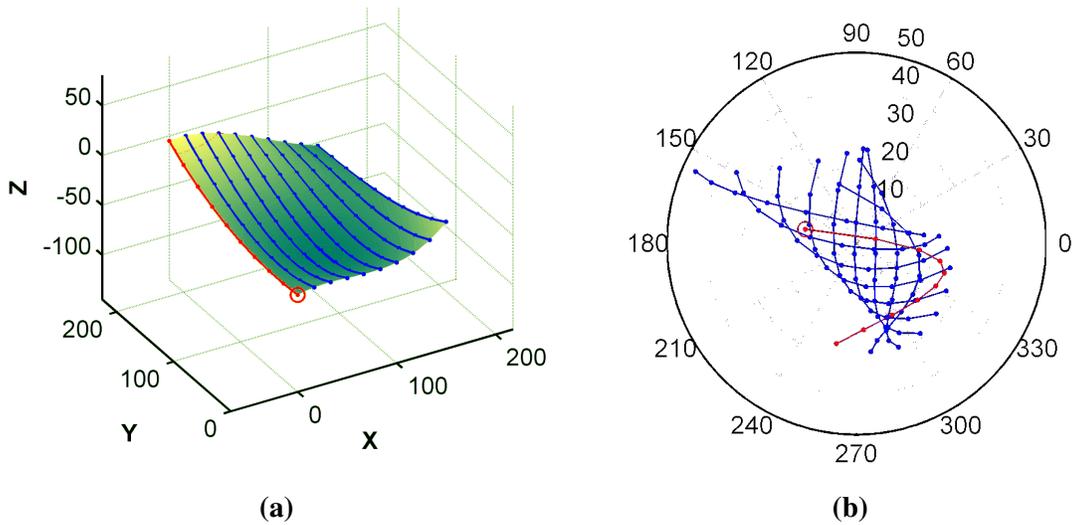


**Figure 5-3 Projected Normals Plane using spherical coordinates (degrees)**

The projected normals plane can be used to get a perspective of the angles required to machine a surface. The two angular coordinates are similar to the rotary-tilt angles used in 5-axis machining.

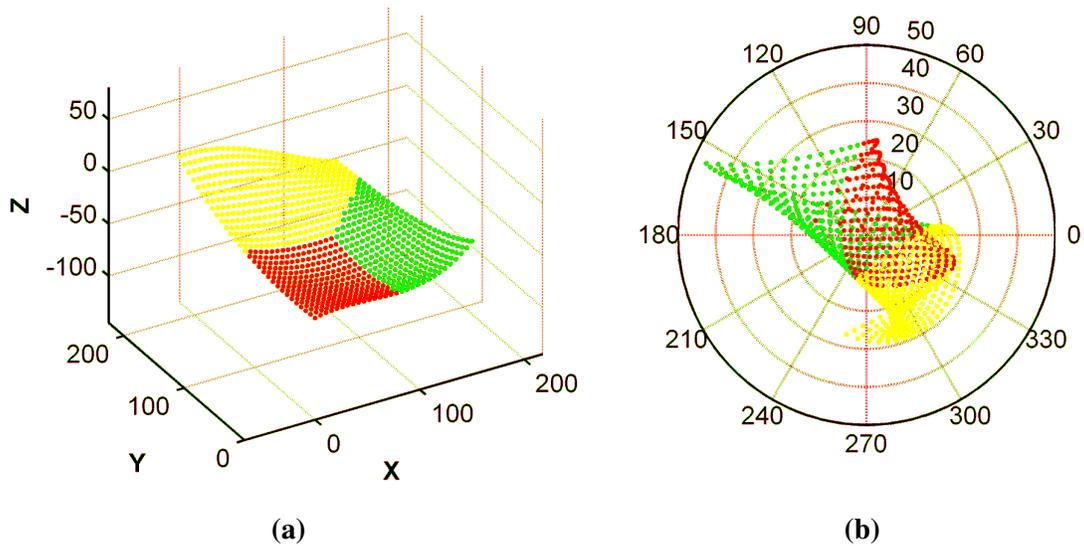
The angle  $\theta$ , which goes from 0 to 360 degrees, indicates the rotations of the table on which the workpiece is clamped, while the angle  $\phi$  is similar to the tilting angle required to orient the tool to the workpiece. By examining this plane, the operator can determine if the orientation angles are off the limits of the machine. For machining configurations that orient the tool in discrete steps, the plot can be used as a guide to determine the appropriate orientation of the workpiece and the number of setups required to machine the surface.

The projected normals plane is a graphical tool that can be helpful for the selection of other machining parameters, such as the direction of cut or the tool orientation. This plane provides a perspective of the distribution and the range of variation of the normals. The following example, shown in Figure 5-4, illustrates how the projected normals plane can be applied in a machining test. The first figure, shown on the left side, shows a 3-D plot of the surface to be machined. This plot shows the sample points and the tool passes along the surface. The size of the surface and changes of curvature can be appreciated from this figure. However, it is necessary to have more information to determine the best approach to machine this surface. Figure 5-4 (b) shows the projected normals plane using spherical coordinates. Assuming that the tool orientation in 5-axis is closely related to the surface normal, it can be inferred that this surface can be machined better with a 5-axis machine than with a 3-axis, due to the considerable variations in the surface normal. If the surface is machined using 5-axis machining it is important to note that there are some points that might require a tilting angle close to 45 degrees, which might be close to the limits of some machines.



**Figure 5-4 Projected normals plane for tool path generation**

The projected normals plane can be used to visualize the change of orientation from one point to the next. In the figure shown above, the red dotted line shows the change of angles for the first tool pass using a direction of cut along the positive Y- axis. The first point on the pass is shown with a red circle. If this tool pass is machined with a 5-axis machining method, there is a smooth transition between points at a majority of the points. In this case, there is only one dramatic change of orientation in the middle of the pass. If the tool pass is machined on a 3+2-axis machine, it may be convenient to divide this pass into two segments. The first segment, machined with a single tool orientation, should consist of the first points of the pass up to the point where the dramatic change in orientation occurs. The remaining points can be grouped in another segment and machined with another tool orientation. In this case, the feed direction chosen seems to be appropriate as the majority of the passes follow the direction of cut and show smooth transitions between points.



**Figure 5-5 Projected normals plane for Surface Partitioning**

The projected normals plane is also useful to evaluate the appropriateness of a partitioning test. As it is important to avoid disjointed or irregular patches, it is also important that the normals within a patch form a tight cluster. Clustering similar normals within a patch can reduce the deviation between the tool axis and the surface normals, which results in a closer match between the tool axis and the surface geometry.

A partitioning test is shown in Figure 5-5. This plot shows the 3-D plot of the surface and the projected normals plane, which can be used to determine if the partitioning test is appropriate for machining. To conduct this evaluation it is necessary to evaluate the shape of the patches and the distribution of the surface normals within each patch. The first figure shows well separated regions, simple boundaries and shapes that are not complex to machine. The second figure shows how the normals are grouped into three clusters. Even when some of points overlap other regions, in general, the distribution of the surface normal seems to be appropriate for each patch.

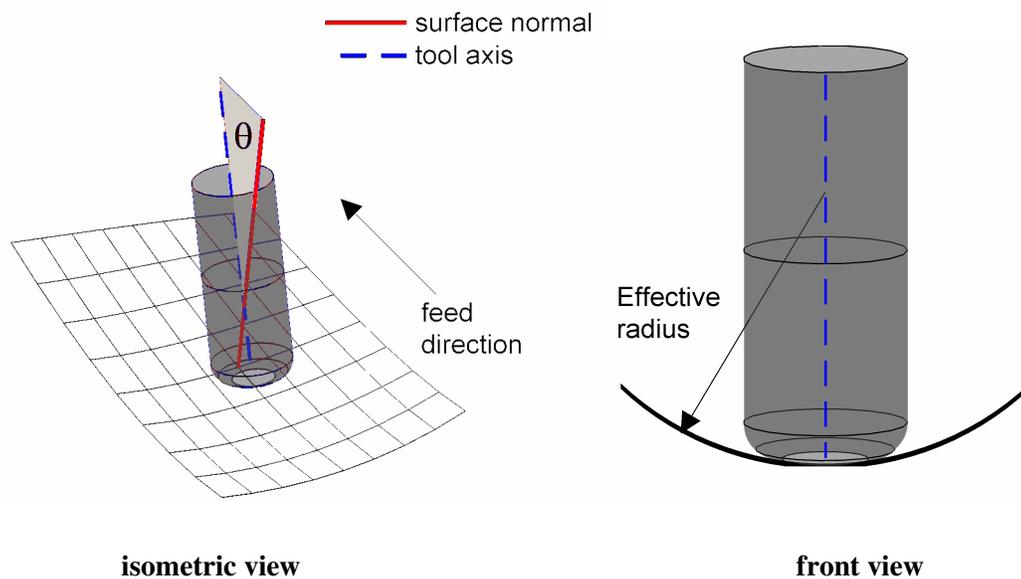
This section presented a graphical tool that can be used to get a better perspective of the surface to be machined. The projected normals plane offers a visual representation of the surface normals,

which can be used to optimize the selection of some machining parameters. The application of this tool is used later in this work and explained in more detail using other machining tests.

## 5.2 Effective radius of the tool

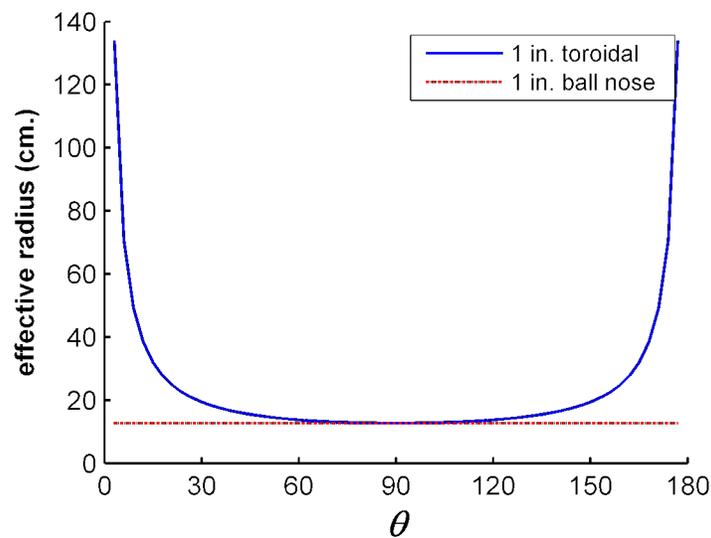
The effective radius of a cutter is defined as the radius of curvature of the tool at the point of contact. For a toroidal cutter the radius of curvature is infinity at the bottom of the insert and is equal to the radius of the insert at the side. An inclined tool results in a larger effective radius in a plane perpendicular to the tilt direction. The effective radius of a toroidal endmill at the cutter contact point can be approximated using equation (5.4), where an arbitrary inclination angle of the tool axis with respect to the surface normal is selected and applied to toroidal or flat endmill cutters at all surface points on the workpiece (Figure 5-6).

$$effective\_radius = \frac{R}{\sin(\theta)} + R_{insert} \quad (5.4)$$



**Figure 5-6 Inclined toroidal endmill and its projected effective radius**

Figure 5-7 shows the relationship between the inclination angle and the effective radius for toroidal cutter. Modifying the inclination angle ( $\theta$ ) can increase or decrease the effective radius of the tool. With this approach an inclined toroidal or flat endmill can be used to machine a surface as effectively as a much larger ball nose end mill [11]. If the inclination angle is too large, the gains are diminished; if the angle is too small, the tool may gouge the workpiece. Thus, maximum gains in the machining time will depend on the proper selection of the inclination angle.



**Figure 5-7 Comparison of the effective radius of a toroidal and a ball nose cutter**

### 5.3 Feed direction

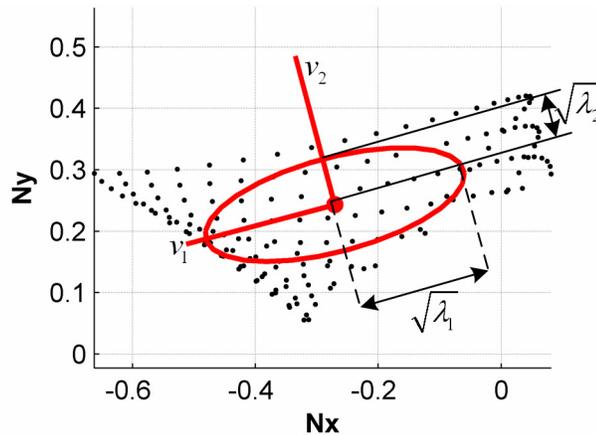
In 5-axis machining the tool axis is changed at every contact point. 3+2-axis machining, in contrast, uses a fixed tool orientation for an entire region or patch. Since the tool axis vector is kept fixed, the variations with respect to surface normals are larger than in 5-axis machining. If the surface normal deviates from the plane defined by tool axis and the feed direction by a large amount the contact point for machining moves from the optimal machining region in the front of the tool to the side, thereby losing much of the benefits.

Two methods were developed in this work to determine the direction of cut for each patch. The first method, named the Eigenvector Method, uses a statistical analysis to determine the feed direction. The second method is based on an exhaustive search to determine the feed direction that results in the shortest tool path length.

### **5.3.1 Eigenvector Method**

The Eigenvector Method uses the distribution of the surface normal vectors to determine the feed direction. Using the projected surface normals plane, the tips of the surface normals are approximated by an ellipse, which fits the data to within one sigma. The parameters of the ellipse are defined by the eigenvalues ( $\lambda$ ) and eigenvectors ( $v$ ) of the covariance of the surface normal vectors, as shown in Figure 5-8 (similarly to the MICD method explained in Chapter 4). The centre of the ellipse is positioned using the mean of the  $X$ - $Y$  coordinates of the projected surface normal vectors. The direction of the major and minor axis of the ellipse is defined using the eigenvectors  $v_1$  and  $v_2$ , respectively.

The feed direction is determined using one of the two axes of the ellipse. If the feed direction is determined by the minor axis ( $v_2$ ), the deviation of the surface normals and the tool axis will be large. This deviation is indicative that a large number of points will be machined by a region of the tool that does not lie in the front, thus minimizing gains due to shape matching. Thus, the feed direction should be along the major axis ( $v_1$ ).

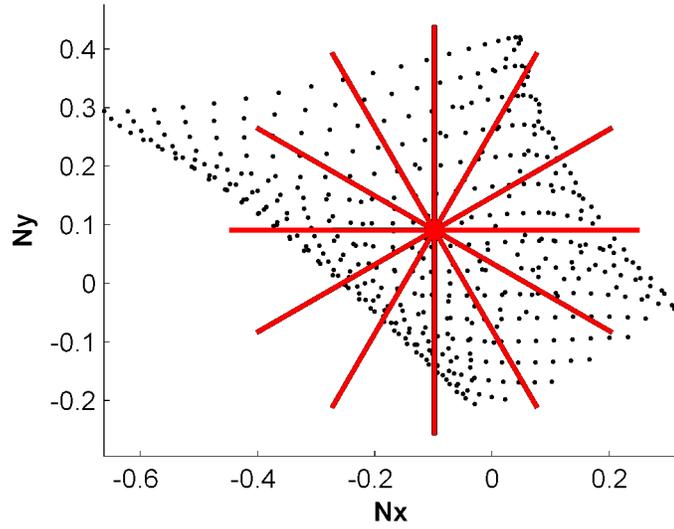


**Figure 5-8 Axes of the ellipse determine feed direction**

The vector  $v_1$  has positive and negative directions. To determine the feed direction, the surface normals are divided using the minor axes of the ellipse ( $v_2$  and  $-v_2$ ). The feed direction is determined by the side having more normal vectors. This feed direction is used to calculate the tool orientation and minimize the variations of the tool axis with respect to the surface normals.

### 5.3.2 Exhaustive Method

The exhaustive method was developed to have a reference to evaluate the appropriateness of the Eigenvector Method. This method consists of an exhaustive search of a feed direction for a specific surface or patch. For this method, the first direction of cut is determined to be along one of the coordinate axes, the X-axis for the example shown in Figure 5-9. Using this feed direction, a tool orientation is determined and a tool path is generated. The tool path length is stored for comparison purposes. The process is repeated using different feed directions, which are determined by rotating the feed direction by a predetermined angle. Finally, the feed direction that results in the shortest tool path length is chosen for machining.



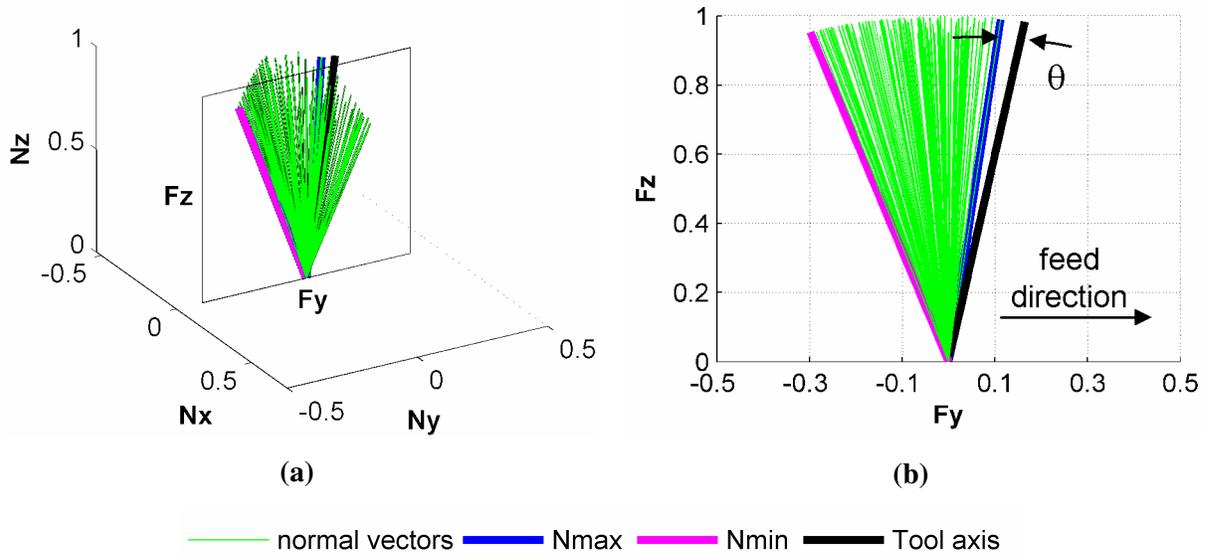
**Figure 5-9 Exhaustive Method**

#### 5.4 Tool orientation

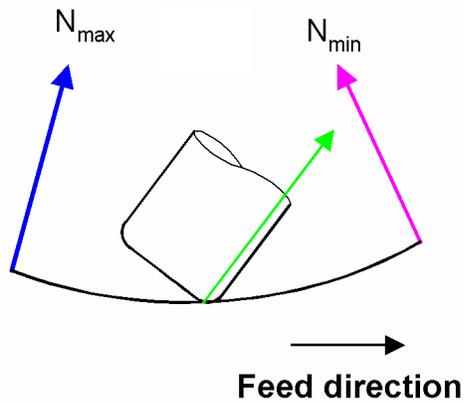
A method for determining the tool orientation for each patch is developed here. The tool orientation is calculated using the projected normals plane. The process starts by moving the surface normal vectors at all the points within a patch to a common origin of a coordinate system as shown in Figure 5-10(a). The surface normal vectors are projected onto a plane defined by the feed direction ( $F_y$ ) and the vertical axis ( $F_z$ ). This projected normal plane, shown in Figure 5-10(b), is used to determine the tool orientation. The first step to calculate the tool orientation is to determine the most inclined normal vector ( $N_{max}$ ) with respect to the feed direction ( $F$ ). The tool axis can be positioned safely by tilting the normal vector  $N_{max}$  by an angle  $\theta$  to determine the tool orientation using equation (5.5).

$$T = \frac{N_{max}}{|N_{max}|} * \cos(\theta) - \frac{(N_{max} \times F) \times N_{max}}{|(N_{max} \times F) \times N_{max}|} * \sin(\theta) \quad (5.5)$$

The projected normals are bounded in a small sector identified by the two bold lines ( $N_{max}$  and  $N_{min}$ ). If the tool axis is selected to lie inside the bold lines, i.e. using the average normal, then the tool will gouge the surface. This can be avoided by positioning the tool axis outside the bounded region as shown in Figure 5-11.



**Figure 5-10 Tool axis inclination with respect to  $N_{max}$**



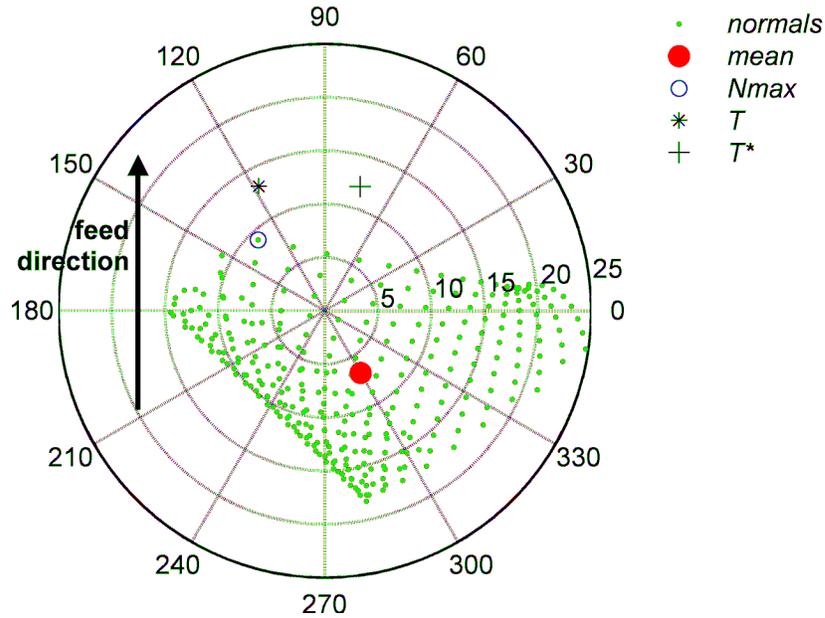
**Figure 5-11 Tool orientation to prevent gouging**

The inclination angle beyond the envelope is currently user selected and can be further optimized. If the tool inclination angle is large, the gains offered by shape matching will be minimized, whereas if the tool inclination is small gouging may occur. A small angle can be applied safely to concave surfaces. For convex surfaces  $\phi$  is selected to be zero for improved efficiency. If a patch has concave and convex regions it is treated as a concave surface for the purpose of determining the tool axis  $T$ .

Figure 5-12 shows the projected normals plane using spherical coordinates. This figure shows the distribution of the surface normals, and the most inclined normal vector ( $N_{max}$ ) with respect to the feed direction. The tool axis orientation for this patch is marked, and lies along the feed direction with respect to  $N_{max}$ . However, this tool orientation can be optimized by projecting the tool orientation into a feed direction vector that starts at the average normal vector. To guarantee that the tool axis minimizes the deviation within the tool axis and the surface normals, the tool axis ( $T$ ) can be projected using equation (5.2). This projected tool axis ( $T^*$ ) is then set to be the tool orientation for machining the patch.

$$Tp = T - normals\_mean \quad (5.6)$$

$$T^* = normals\_mean + F(Tp \cdot F) \quad (5.7)$$



**Figure 5-12 Tool orientation**

### 5.4.1 Tool positioning

If the tool axis is along vector  $T$  and the normal at the cutter contact point  $ccp_j$  is  $N_j$ , then the tool position  $P_j$  (bottom centre of the tool) is given by Equation (5.8), where  $R_1$  is the radius of the tool and  $R_2$  is the radius of the insert. The toroidal tool equation is convenient because it can model both the ball nose ( $R_1 = 0, R_2 = R$ ) and flat end milling cutter ( $R_1 = R, R_2 = 0$ ).

$$P_j = ccp_j + R_2 N_j + \frac{N_j - (N_j \cdot T) \cdot T}{|N_j - (N_j \cdot T) \cdot T|} * R_1 \quad (5.8)$$

## 5.5 Tool parameters

### 5.5.1 Types of tools

CNC machines are designed to move a tool relative to a workpiece. In industry ball nose cutters are commonly used for machining curved surfaces, whereas flat endmill cutters are used to machine flat or low curvature surfaces. Three different types of cutting tools are considered for the proposed

3+2-axis machining method: ball nose, toroidal and flat end-mills. These three cutters are shown in Figure 5-13.

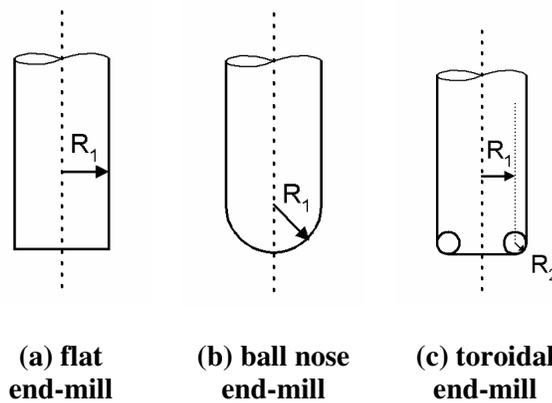


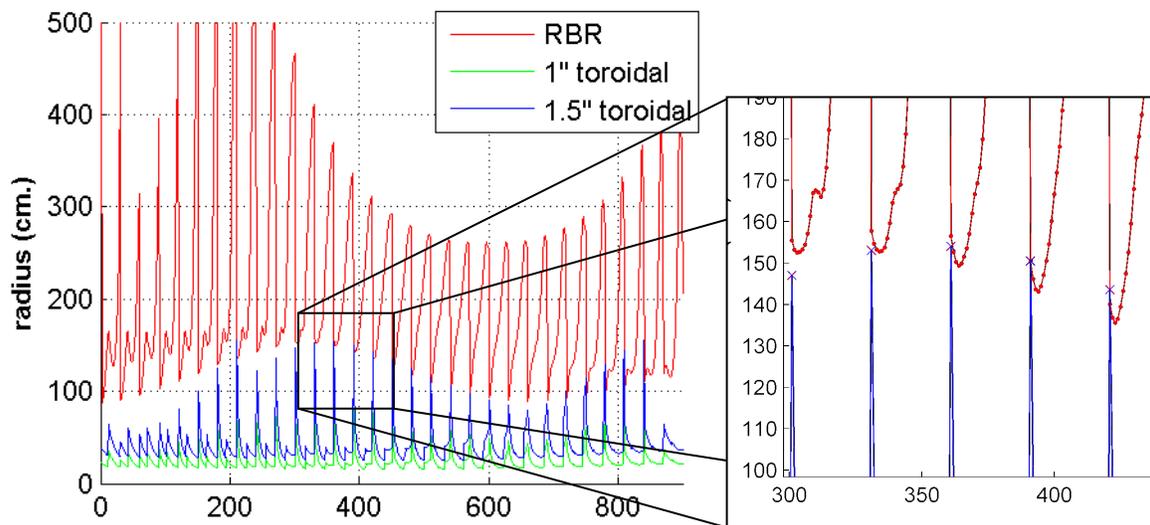
Figure 5-13 Cutting tools

### 5.5.2 Tool size

Tool path planning requires efficient tool size calculations. Calculating the maximum tool radius could be used to prevent gouging. Gouging is defined as the overcut caused when the cutter is moved along the tool path. If the tool radius is bigger than the minimum radius of curvature, gouging can occur on concave or saddle parts.

Inefficient machining may be a consequence if the selected tool is too small. In this work, a surface can be divided such that each portion is machined with the largest tool that guarantees no gouging. To determine the largest tool that can be used for each patch, a comparison between the radius of curvature of the surface and the effective radius of various tools at the contact point is conducted. The radius of curvature of the surface is calculated using the Rolling Ball Method explained in section 2.3.2 and the effective radius is calculated for all the tools available. The tool with the largest radius that does not exceed the radius of curvature at any point is selected for machining.

Figure 5-14 shows a sample test to determine the tool size. In this test, the sample test surface 1 is represented using a grid of 30x30 uniformly space points (900 sample points in total). The Rolling Ball Radius (RBR), red line, is calculated for all the sample points. The effective radius of the tool is calculated for the tools available at each contact point. In this test the radius is calculated for both a 1" toroidal and a 1.5" toroidal, shown in green and blue, respectively. The comparison starts using the largest tool available. If the effective radius of the tool is larger than the RBR at any point, the tool will gouge the surface and thus is not considered suitable for the surface. In this case there are some points where the effective radius of the tool is larger than the RBR, as shown in the exploded view. The comparison is then conducted with the next tool available. For this surface, there are no points where the effective radius of the 1" toroidal tool is larger than the RBR, therefore, this tool is selected as the largest tool that guarantees no gouging.

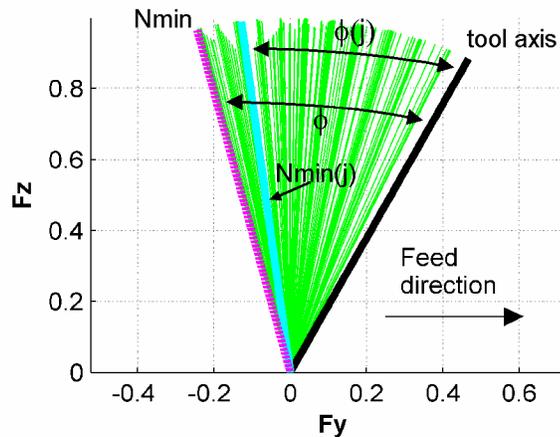


**Figure 5-14 Test to determine the tool size**

## 5.6 Side-step distance

The side-step distance is one of the key factors in 3+2-axis machining because of its impact on the actual machining time. The side-step is defined to be the largest allowable distance between two consecutive passes so that the largest scallop height does not exceed a user-specified tolerance. Cusps or scallops are left behind every time a curved surface is machined with a ball nose or toroidal endmill. Material is left between tool passes in the form of scallops because the tool geometry is not exactly matched to the surface geometry. These scallops must be removed in subsequent grinding and polishing operations.

The side-step distance depends on the inclination between the surface normals and the tool axis orientation. For a constant side-step the inclination angle ( $\phi$ ) is defined as the inclination angle between the tool axis and  $N_{min}$ , the furthest projected normal vector with respect to the tool axis vector, as shown in Figure 5-15. For a variable side-step the angle  $\phi(j)$  is calculated to find the largest allowable side-step distance for each pass for the maximum scallop height .



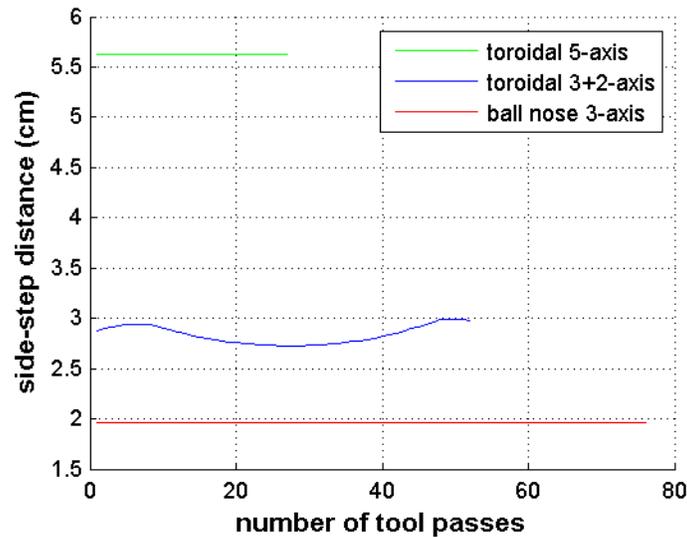
**Figure 5-15 Tool's inclination angle**

Although the tool is in general a toroidal cutter it can be approximated by a ball nose of radius equal to the radius of curvature of the torus at the contact point. The effective radius of the tool can be calculated from Equation (5.9). The side-step for each tool pass is given in equation (5.10).

$$effective\_radius = \frac{R_1}{\sqrt{1-(N \cdot T)^2}} + R_2 \quad (5.9)$$

$$side\_step = 2\sqrt{2(effective\_radius)(scallop\_height)} \quad (5.10)$$

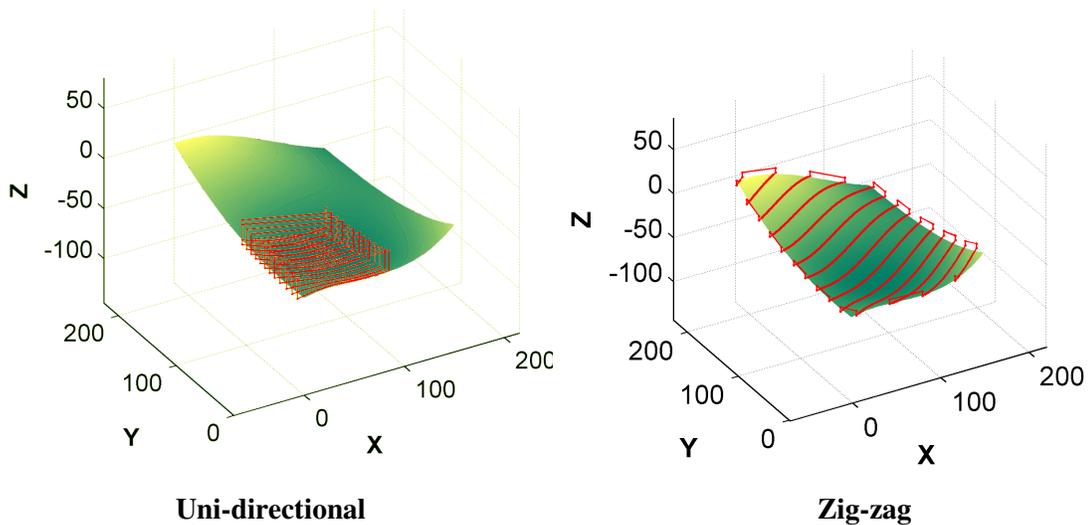
The side-step distance determines the number of tool passes required to machine a surface. The larger the side-step, the shorter the tool path length will be. Figure 5-16 shows a comparison of the number of tool passes required to machine the test surface 3 using 3-axis, 3+2-axis and 5-axis machining. For this comparison, the tool paths were generated using a tool of the same diameter. The 3+2-axis and the 5-axis machining experiments were conducted using a 1.5” toroidal tool with an insert of 6 mm, while the 3-axis machining tests were conducted using a 1.5” ball nose cutter.



**Figure 5-16 Side-step distance comparison**

## 5.7 Types of tool paths

The tool path is generated after partitioning the surface and determining the tool orientation for each patch. To machine the surface the user specifies the tool path pattern, namely zigzag or parallel (Figure 5-17). In the zigzag path the tool moves back and forth along the feed direction contacting the surface at all times. If the last point of the pass is inside the surface, i.e., when the surface is partitioned, the tool is lifted and moved in the air to the next tool pass to reduce boundary marks between patches. In the uni-directional path the tool cuts the surface as it moves in the feed direction, but lifts up and moves rapidly when returning.



**Figure 5-17 Types of tool paths**

Once the tool path pattern, zigzag or parallel, is known the exact path for each patch is determined. To machine the first patch the tool path is started from one end of the whole surface. The first point of the tool path is calculated from one of the corners of the surface. The first contact point is evaluated using the MICD or the nearest neighbour method to find if it belongs to patch one. If the contact point belongs to the patch to be machined, it is stored in a table. If it does not belong to the patch it is skipped and the next contact point is calculated. The next contact point is located at a user specified distance from the current point in the feed direction. Other methods of determining the next contact

point can also be used. Once a tool pass has finished, the side-step method is used to find the first point in the next pass and the process repeats until the entire surface is covered. Only those points belonging to the first patch are stored in the table and are already pre-ordered into parallel passes. The process continues until the entire surface is covered. Once the first patch is complete, the process is repeated for all the remaining patches.

## **5.8 Summary**

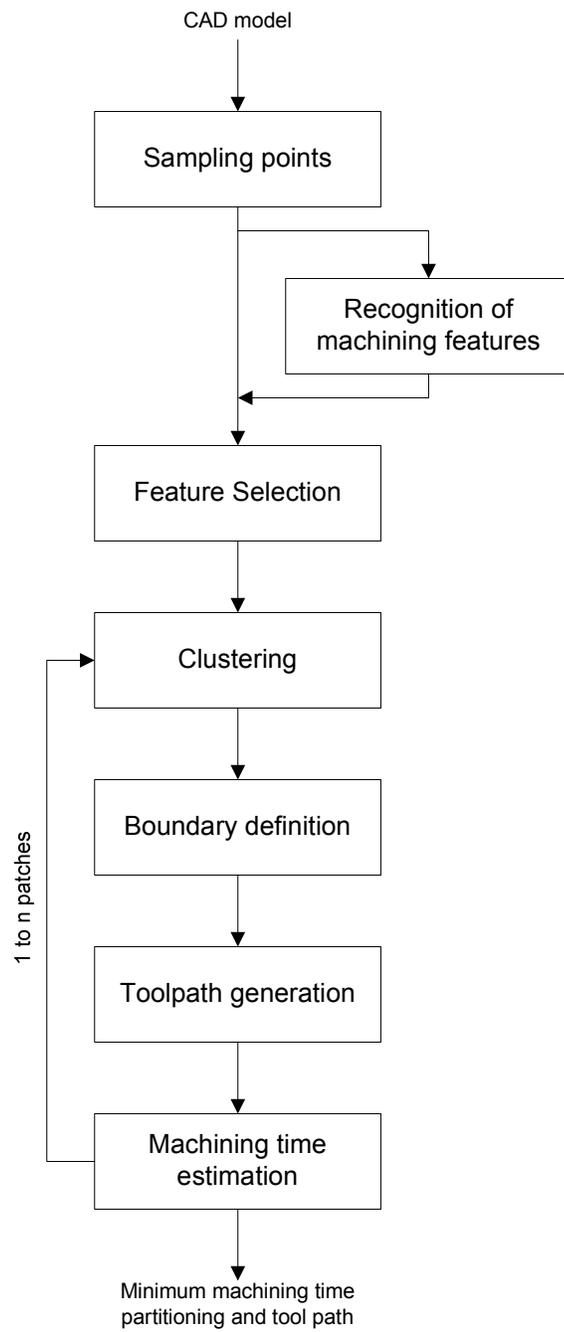
Guidelines to generate a tool path for patch-by-patch machining were presented in this chapter as well as a new method for the visual representation of a surface. This chapter presented a new method for determining a proper feed direction and tool orientation for each patch. It was found that these two parameters are closely related to the side-step distance, and their optimization should help to generate safer tool paths and reduce the machining time. A method to determine an appropriate tool size for each patch was also derived in this chapter. Finally, a new visualization tool that can be used for all types of multi-axis machines was developed in this chapter. This visualization tool provides insight into the surface normals distribution, which can be helpful in determining machining parameters.

An implementation of the methods developed, as well as simulation tests to verify the techniques proposed are presented next in Chapter 6. Later, in Chapter 7, a series of machining tests are conducted to validate the methodology proposed for 3+2-axis machining.

## Chapter 6

### Implementation of the patch-by-patch 3+2-axis machining method

In this chapter the 3+2-axis machining methodology, shown in Figure 6-1, is presented by describing the procedures required to machine a complex surface. This process starts by extracting surface properties for the sample points used to describe the surface geometry. In the first phase of the process, standard surfaces such as fillets and planes are identified. These surfaces can be machined separately from curved surfaces and should be eliminated from the clustering data. The remaining data is then used to form a feature vector that contains the set of surface properties that will be used for partitioning. The feature vector consists of properties that are significant indicators of the spatial location and the shape of the sample point. A clustering algorithm uses the feature vector to define the surface patches. A tool path is then generated both for machining within a patch and for rapid travel from one patch to the next. This step requires identifying the boundaries of each patch and a classification of the cutter contact points to determine if they belong to the patch to be machined. Although this method can sub-divide a surface into patches, it does not yield the optimal number of patches that results in the smallest machining time. To determine the number of patches, the surface is divided into different partitions that lie within a range defined by the user, and the machining time is calculated for each partition. The partitioning that results in the smallest machining time is selected for machining.



**Figure 6-1 Process diagram for 3+2-axis machining**

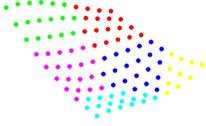
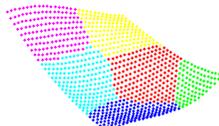
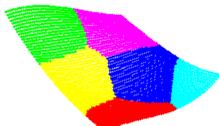
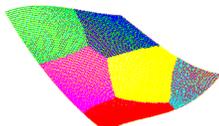
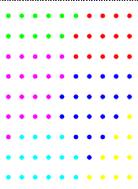
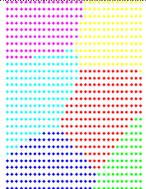
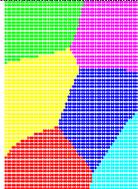
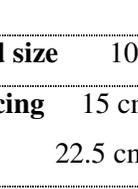
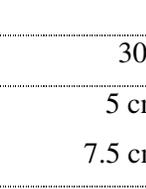
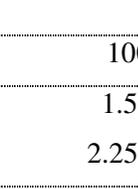
## 6.1 Sampling points

Defining the surface model accurately is an important factor for an efficient partitioning process. Each surface is different and experience plays an important role in the decision of determining the number of sample points. The number of sample points should be large enough to represent adequately the variations in shape of the surface, but at the same time it should be kept to a minimum to avoid the waste of computer resources associated with over refinement [67]. The geometry will dictate the areas where prominent changes in geometry occur, requiring a more detailed representation in that particular area [68].

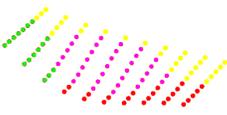
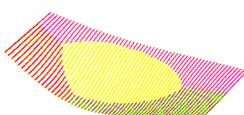
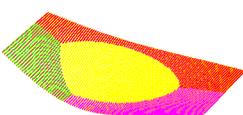
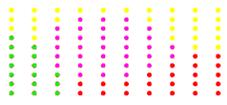
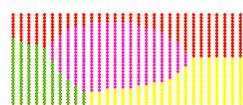
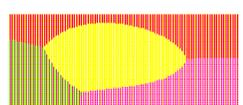
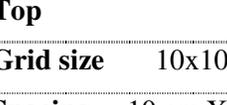
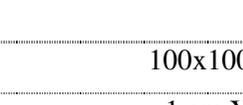
Determining the number of sample points to represent a surface is similar to the problem of determining the mesh size in Finite Element Methods (FEM). In FEM, there are no set rules for establishing the mesh size [69] and conventionally the process is conducted by experimentation. A straightforward check for accuracy of the model is to increase the number of elements by fifty percent and compare the results [68]. Another approach involves in repeating the analysis several times with successively refined meshes, and when acceptable convergence has occurred the last mesh is the standard one for all future computations for similar types of loadings [69].

This section presents experiments conducted to determine an appropriate number of sample points to represent a surface. In the first test, the number of sample points is determined by following the guidelines used in FEM to study the effects of using a denser grid. These tests are conducted on two surfaces with different sizes. The number of sample points is defined by a regular grid used to represent the surface. The tests are conducted using grids of 10 x 10, 30 x 30, 60 x 60 and 100 x 100 points, and the results are shown in Table 6-1, and Table 6-2. Also included in this table is the space between sample points and the computed time to conduct the partitioning tests.

**Table 6-1 Sample size tests for Surface 1**

				
<b>Isometric</b>				
<b>Top</b>				
<b>Grid size</b>	10x10	30x30	60x60	100x100
<b>Spacing</b>	15 cm X 22.5 cm Y	5 cm X 7.5 cm Y	2.5 cm X 3.75 cm Y	1.5 cm X 2.25 cm Y
<b>Time</b>	1.5 sec.	2.12 sec.	4.5 sec.	11.06 sec.

**Table 6-2 Sample size tests for Surface 4**

				
<b>Isometric</b>				
<b>Top</b>				
<b>Grid size</b>	10x10	30x30	60x60	100x100
<b>Spacing</b>	10 cm X 4 cm Y	3.33 cm X 1.33 cm Y	1.67 cm X 0.67 cm Y	1 cm X 0.4 cm Y
<b>Time</b>	0.70 sec.	0.75 sec.	1.34 sec.	2.45 sec.

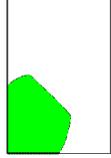
The effect of varying the number of sample points is first examined using visual inspection of the results. The tests show that the number of samples does not reflect drastic changes in the shape of the patches. In the majority of cases, there are only minor changes on the boundaries. The calculation time obtained for the different surfaces show that there are not substantial differences for surfaces represented with less than 60 by 60 grids. However, as the number of points is increased, the time required for calculations can grow exponentially.

To define the number of sample points, it is necessary to select a suitable parameter that can guide the operator in defining a surface accurately using a minimum number of sample points. One option is to use the size of the grid. However, this parameter will result in different resolutions depending on the size of the surface. A more appropriate solution is to use the space between the sample points, which provides a more general solution for different types of surfaces and is independent of the size of the surface. For the surfaces used in these tests, the space between sample points is defined in the *XY*- plane, but other planes can be easily used for these purposes.

Numerical tests on surface patches were conducted to determine an appropriate space between sample points. In these tests, a surface is partitioned using a grid of 200x200 points (for simplicity the tests were conducted using regular grids). The sample points are labelled based on the patch that they belong to, and are defined as test points. Once the test points are assigned to a patch, a new set of sample points is calculated using a grid with a larger space between points. These new points are used to partition the surface again and classify the test points. The misclassified points are used to determine the accuracy of the new grid. The results obtained on two sample patches are presented in Table 6-3 and Table 6-4.

**Table 6-3 Comparison of the effectiveness of classification for patch 1**

<b>Max spacing</b>	<b>Grid Size</b>	<b>Misclassified</b>	<b>Effectiveness</b>
9 cm.	25x25	540/9782	0.9448
4.5 cm.	50x50	233/9782	0.9762
2.25 cm.	100x100	86/9782	0.9912



**Table 6-4 Comparison of the effectiveness of classification for patch 2**

<b>Max spacing</b>	<b>Grid Size</b>	<b>Misclassified</b>	<b>Effectiveness</b>
9 cm.	25x25	968/13544	0.9285
4.5 cm.	50x50	371/13544	0.9726
2.25	100x100	141/13544	0.9896



The tests show that for a space of 4.5 cm between sample points the effectiveness is around 97 %. These results validate the numerical inspection presented earlier, where the partitions using spaces of 3 to 4 cm showed good results. The tests conducted show that the maximum space between sample points should be no larger than 4 cm. This space between points can be used as a reference, but it is recommendable to validate the proper number of sample points before conducting tests with new surfaces.

## **6.2 Identify common shapes**

Shapes such as fillets and planes commonly found in industrial parts are usually machined separately from curved surfaces. Standard surfaces can be machined as single entities or with form tools. Identifying and eliminating these particular shapes simplifies the partitioning process by reducing the amount of data that needs to be processed, as well as eliminating shapes that can result in outlier data.

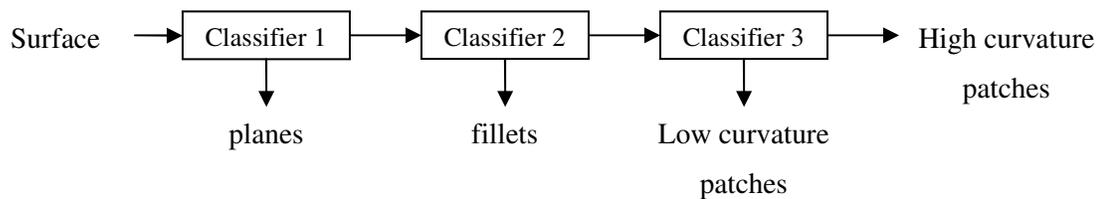
Shape recognition is an area in which extensive research has been done. Different methods, which include rule-based approaches, graph-based approaches, volume approaches and Neural Network-based approaches, have been developed. The rule-based approach use pattern-matching techniques and expert systems to develop recognition rules. The work presented by Vandenbrande and Requicha [70], and Chan and Case [71] is based on the rule-based approach. The graph-based approach (Joshi and Chang [72], de Floriani [73]) requires matching a part graph to the predefined shapes using graph manipulation algorithms. However, it is impossible to define all rules for all shapes, and in some cases this method requires new rules to be generated by slightly adjusting existing rules, which brings in subjectivity. The volume approach developed by Kim [74] and Sakurai [75] computes the removal volume from the solid model and decomposes the removal volume into cells for machining purposes. However, these approaches still suffer from extensive computation. Recently, some researchers such as Henderson and Prabhakar [76], Nezis and Vosniakos [40] have proposed to apply artificial neural networks in shape recognition. Neural Networks have proven to be effective in recognizing shapes due to their high degree of robustness and strong learning capability as shown by Li *et al.* [77]. In general, each approach is effective in identifying specific types of shapes; however, optimum efficiency cannot be achieved using a single approach.

### **6.2.1 Hybrid Methods**

Hybrid methods can achieve better results by benefiting from the diverse capabilities of different techniques through sequential sorting. Commonly, the first classifiers are simple methods used for filtering distinct groups of elements. Once the majority of the elements have been identified, more complex algorithms can be applied to the remaining data. Li *et al.* [77] developed a hybrid method based on feature hints, graph-based approach and Neural Networks. Lam and Wong [70] developed a method to recognize shapes from boundary representation (b-rep), which combines a graph-based

approach, a volume approach, and Neural Networks. However, these works cannot recognize fillets or rounded shapes that are commonly encountered in dies and moulds.

In this work, a hybrid method based on a sequential classification method that combines comparison rules and Neural Networks is proposed to solve the shape recognition problem. The classification process to recognize machining features that include planes, fillets and spherical surfaces is illustrated in Figure 6-2.

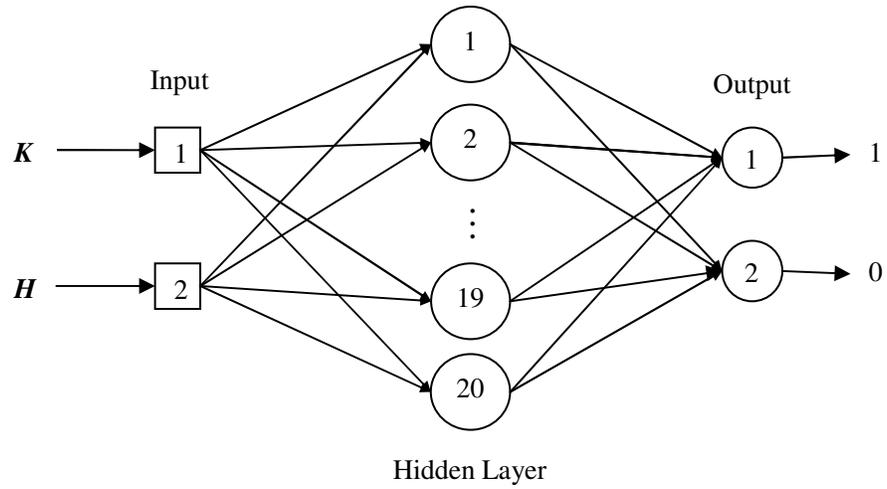


**Figure 6-2 Sequential classification process**

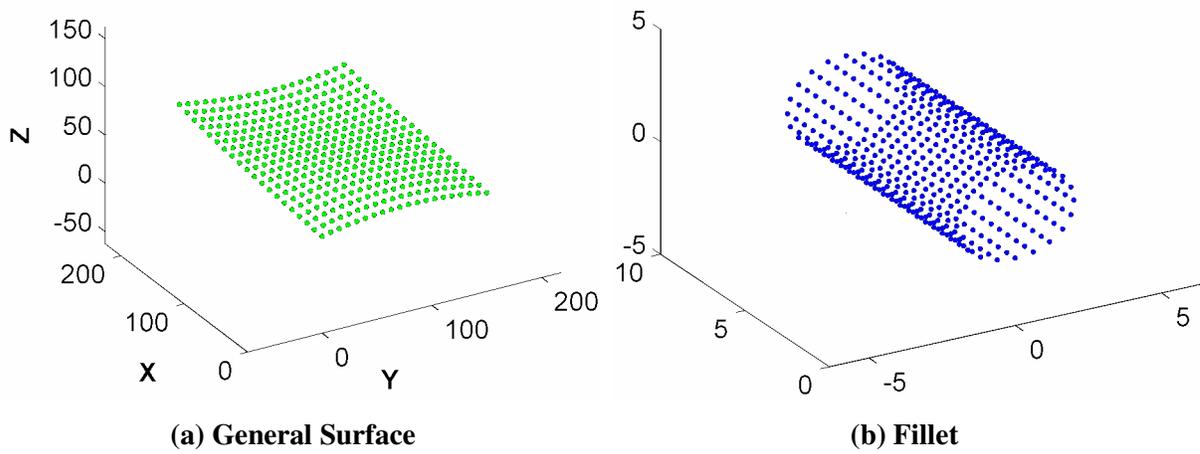
The first classifier identifies planar surfaces based on the shape relationship of the curvature parameters. If the Gaussian and Mean curvatures surrounding a point are zeros, then the point is identified as belonging to a plane [78]. The second classifier uses a Neural Network to identify radiused shapes. A Neural Network is a system of processing elements, called neurons, connected together to form a network that has the ability to learn from examples through repeated adjustments of their weights [79]. The classifier is a feed forward back propagation network, shown in Figure 6-3, with a twenty neuron hidden layer and a two neuron output layer. The training data for the Neural Network is presented in Figure 6-4, which includes a Bézier surface and a cylinder representing a shape with constant curvature. The last classifier is used to identify regions with low curvature. This classifier uses Equation (6.1) to identify low curvature regions, where the flatness radius,  $FR$ , is used for filtering and is adapted from the work presented by Lauwers *et al.* [22]. Based on experiments conducted by the current author, the flatness radius for these experiments was selected as  $10^{-5}$ .

$$K^2 + H^2 < FR$$

(6.1)



**Figure 6-3 Neural Network structure for feature recognition**



**Figure 6-4 Training data**

### **6.2.2 Experiments**

Six surfaces were tested using the hybrid classification method. The surfaces shown in Figure 6-5 were lumped into four different types of shapes including planes, fillets, low curvature patches and high curvature patches. The first two classifiers correctly identified the planar and radiused shapes. The third classifier provided information about the curvature of the surfaces. The surface shown in Figure 6-5 (d) was identified as a low curvature surface. However, the remaining three surfaces, shown in Figure 6-5 (a), Figure 6-5 (b) and Figure 6-5 (c), have small regions classified as fillets and low curvature patches. For these surfaces, it is more convenient to consider the surface as a high curvature patch, as the machining time can be increased if these isolated regions are machined separately.

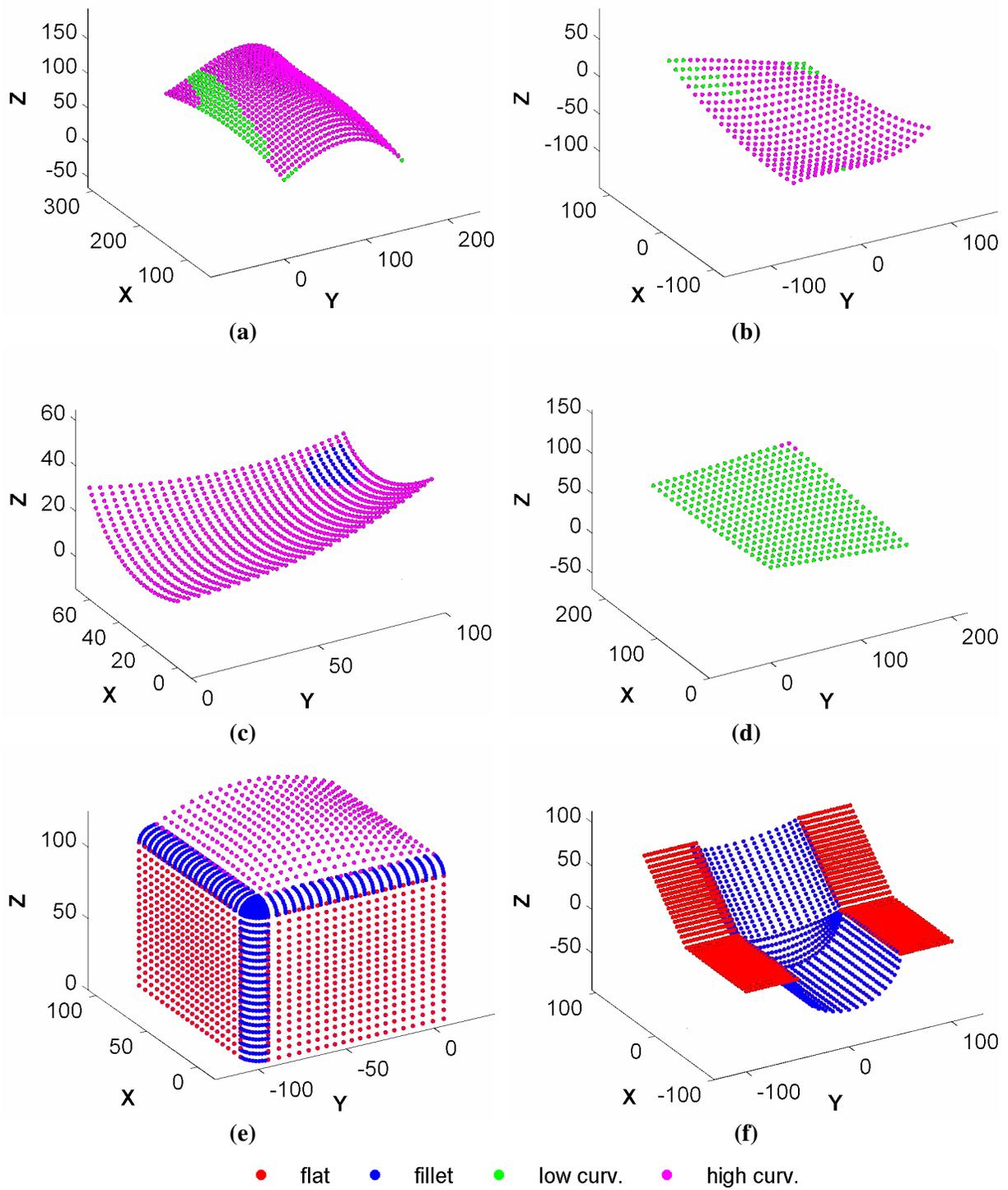


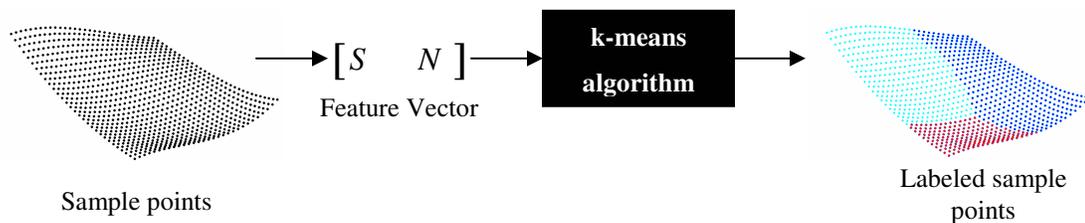
Figure 6-5 Classification results

The proposed shape recognition method was used to identify common shapes encountered in industrial parts. Identifying regions that can be machined with established methods facilitates the machining process and reduces the amount of data that needs to be processed.

### 6.3 Surface partitioning

In Chapter 3, a comprehensive analysis of different clustering algorithms and surface properties was conducted. The study conducted showed that the k-means clustering algorithm is an effective method for partitioning. Also, the tests showed that the use of the surface coordinates and normal vectors result in good partitions and can be applied for complex surfaces.

Figure 6-6 shows the proposed partitioning scheme for 3+2-axis machining. This process starts by extracting the surface properties for each sample point. These properties are used to form a feature vector that is applied to a clustering algorithm. The output of this process is a group of labelled sample points that can be used for the identification of the boundaries required to generate tool paths for each surface patch.



**Figure 6-6 Surface partitioning process**

Partitioning tests were conducted to evaluate variations of the feature vector  $[S \ N]$  formed by the surface coordinates and normals. These tests were conducted on the sample test surface number 1. This evaluation is conducted using the k-means clustering algorithm using two types of distances,

Euclidean and Cosine. These tests also include the feature vector  $[S\ 2N]$ , which uses a double weight for the surface normals.

Table 6-5 presents the results of the comparison conducted between the Euclidean and the Cosine distance. In this table the minimum tool path length for each partition is highlighted in bold. The results show that there is not a substantial difference in the tool path length results between these two distances. As the partitioning results conducted using these two distances do not reflect considerable changes in the shape of the patches either distance can be used for the partitioning tests.

Also included in this comparison is the use of a weighted feature vector. The use of a double weight for the surface normals shows larger tool paths than the normalized feature vector. Considering that the weighted feature vector may also need tests to avoid disjointed patches, this combination is no longer considered in further tests.

**Table 6-5 Tool path length comparison (mm)**

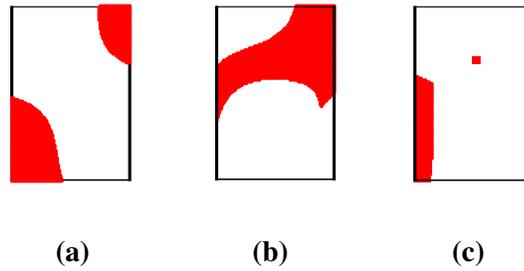
		k-means (Euclidean)		k-means (Cosine)	
		SN	S2N	SN	S2N
Surface 1	2 patches	17,337	17,539	<b>17,311</b>	17,571
	3 patches	16,295	16,408	<b>16,255</b>	16,592
	6 patches	<b>16,474</b>	16,495	16,513	16,668
Surface 4	2 patches	<b>2,936</b>	3,037	3,122	3,227
	4 patches	<b>4,348</b>	4,391	<b>4,347</b>	4,415

## 6.4 Patch boundaries

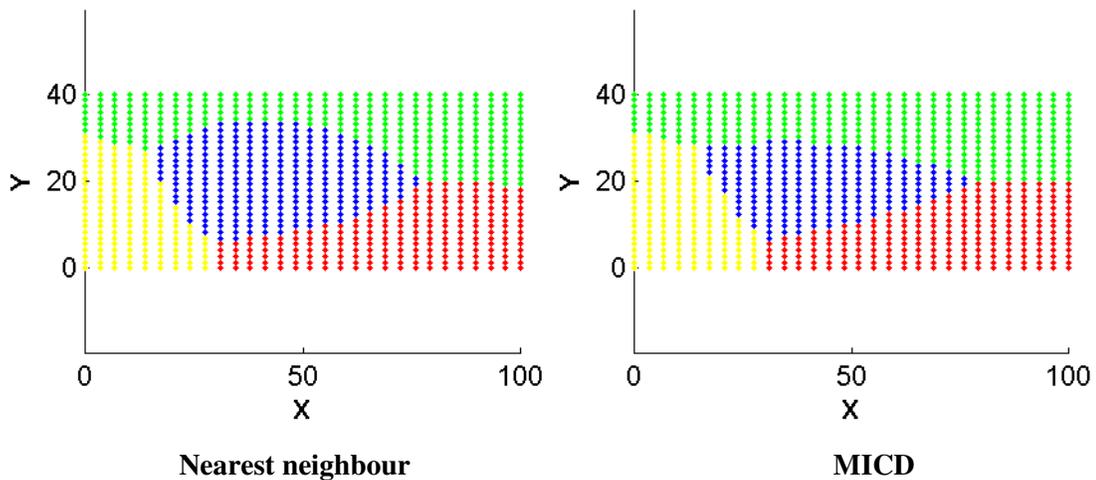
In Chapter 4, different approaches to determine the patch boundaries were examined. In the tests conducted two approaches showed a good performance for the classification of contact points: the nearest neighbour and the Minimum Intra-Class Distance (MICD) methods. To determine an appropriate approach for the machining tests, simulation tests are conducted to compare the tool path length using these two methods.

Although the two methods considered for the identification of boundaries worked well in the majority of cases, there are some particular situations where these methods could have difficulties or fail. Examples of shapes that can complicate the tool path generation process include isolated points, sharp corners and disjointed patches (Figure 6-7). Even when these problems are not common, it is important to recognize these situations. Knowing the capabilities and strengths of each method will allow making a correct assessment of the situation, which can be used to select the proper approach for each partition.

Figure 6-8 shows an example of a surface patch with a complex shape. This test is conducted on the sample test surface number 4. In the original partition, which is close to the one obtained with the nearest neighbour, the green patch may present some complexities for machining. For example, if the feed direction to machine the green patch is along the  $X$ - axis in some tool passes the pass will have to be split. If the tool passes are split, it forces the tool to lift to prevent contact with other patches and thereby increasing the machining time. If the feed direction is along the  $Y$ - axis, the tool passes in the middle of the patch will be short and will limit the tool to reach its full acceleration.



**Figure 6-7 Complicated shapes**



**Figure 6-8 Classification of cutter contact points**

The example shown above points to some of the problems associated with the nearest neighbour method. These problems, in contrast, are not encountered using the MICD method. In the classification of the points using the MICD some of the points are reassigned to an adjacent patch. However, this should not affect the efficiency of the machining process, as the tool inclination at boundary points is sub-optimal regardless of which neighbouring patch they are included in.

Other situations where the nearest neighbour can have problems include patches with isolated points and sharp corners. In these cases, the tool path will require short tool passes that can prevent reaching the programmed feed rate and thereby increasing the machining time.

Simulation tests were conducted to evaluate the MICD and the nearest neighbour methods. The results of this comparison are presented in Table 6-6. In the nearest neighbour method tests the surface patches are represented using the clusters method described in chapter 4. The results show that in most of the simulations the MICD performs better than the nearest neighbour. Since the MICD eliminates some of the problems encountered with the nearest neighbour and also results in shorter tool path lengths and faster computations, this method is considered the most appropriate for the classification of cutter contact points.

**Table 6-6 Tool path length comparison (mm)**

<b>Test surface</b>	<b>Number of patches</b>	<b>MICD</b>	<b><i>k</i>-NN</b>
Surface 1	2 patches	<b>17,372</b>	17,546
	3 patches	<b>17,197</b>	17,409
	6 patches	<b>16,484</b>	16,795
Surface 4	2 patches	<b>2,926</b>	2,996
	4 patches	4,312	<b>4,296</b>

## 6.5 Tool path generation

In the previous chapter, two methods to determine the feed direction were presented. The first one determines the direction of cut using the eigenvector of an ellipse that approximates the distribution of the surface normals. The second method uses an exhaustive search of feed directions to calculate the one that results in the shortest machining time.

The comparison between the Eigenvector and the Exhaustive methods is shown in Table 6-7. For these tests, a tool path is generated for each patch using both methods. The Exhaustive method is conducted for twelve feed directions. The one that results in the shortest tool path length is used to machine the patch. The results of the comparison show a better performance of the Exhaustive

method. This method is simple to implement and can be conducted automatically for each partition. Thus, this method is considered the most suitable approach for the proposed 3+2-axis machining method.

**Table 6-7 Tool path length comparison (mm)**

Test surface	Patch	Eigenvector	Tool path length	Exhaustive	Tool path length
		Vector		Vector	
Surface 1	Patch 1	[-0.53 -0.84]	10,211	[-1 0]	8430
Surface 1	Patch 2	[0.96 -0.26]	4456	[0 -1]	3700
Surface 1	Patch 3	[0.41 -0.91]	7446	[0 -1]	6908
Surface 3	Patch 1	[0.99 -0.10]	4737	[1 0]	4404
Surface 3	Patch 2	[0.79 0.60]	1349	[0 -1]	1073

## 6.6 Number of patches

In the proposed 3+2-axis machining method presented in this work, the number of patches that result in the smallest machining time is not known *a priori*. The process of selecting the number of patches is conducted by estimating the machining time for a range of patches selected by the user. This range is normally between 1 and 8 patches, but can be increased for large parts or complicated shapes. Based on the partitioning results, an estimated machining time for all the partitions is calculated and the one that results in the smallest machining time is chosen for machining. The estimation of the machining time is conducted by generating a tool path for each partition. The tool path is comprised of cutter contact points that follow a path determined by the direction of cut. If the surface is partitioned a tool path for each patch must be generated. The process to generate the tool path for a patch requires evaluating the contact points along the tool path to determine whether they belong to the patch or not.

## 6.7 Machining time

Since the optimal partitioning that results in the shortest machining time is not known *a priori*, the estimated times are calculated while increasing number of patches starting at one. The process is stopped when further partitioning results in increased machining time and the partition that results in the shortest time is chosen for machining.

The partitioning of a surface impacts the tool path length and the corresponding machining time. The transition from one patch to the next also requires time for re-orientation. Additional tool movement is required between patches, which adds to the total machining time. As the number of patches increases, the time spent between patches can have a negative impact on machining time.

The methodology developed in this work can be carried out in either discrete or simultaneous five-axis machines. From observations of the machine used in this study, a simultaneous five-axis machine, five seconds is added each time to account for time consumed when a tool changes its orientation. For machines that rotate the axes in discrete steps, one minute is added for each tool orientation change.

Knowing the tool path length and the feed rate, the time required to machine each patch can be calculated. For an accurate estimation of the machining time it is necessary to account for the effect of acceleration/deceleration of axes during tool movement. To account for this effect it was determined that the feed rate should be estimated at approximately 95 percent of the actual feed rate. This approximation was determined by observations of actual machining tests.

Machining tests were conducted to evaluate the appropriateness of the parameters used to estimate the machining time. These tests were conducted using the sample test surface number one and the results are shown in Table 6-8. The sample surface was partitioned into four patches and was also machined as a single patch to provide another reference. The results show a close relationship

between the tool path length and the machining time. In the simulations conducted to determine the machining time, the time estimates obtained for the zig-zag tool paths were close to the actual machining time.

The tool paths using unidirectional parallel passes are more difficult to predict because of the rapid traverse travel required. This estimation requires adding the time consumed by the tool to retract. A time test was conducted on the machine to evaluate the actual speed of the tool and the effect of acceleration and deceleration. The first test included four tool paths moving at maximum feed rate in linear motion and using different distances. The results presented in Table 6-9 show that there is a substantial effect of the acceleration and deceleration of the machine. For a more accurate estimation of the machining time, it is required to evaluate the actual acceleration rate of the machine for every machine axis

**Table 6-8 Estimated and machining time comparison**

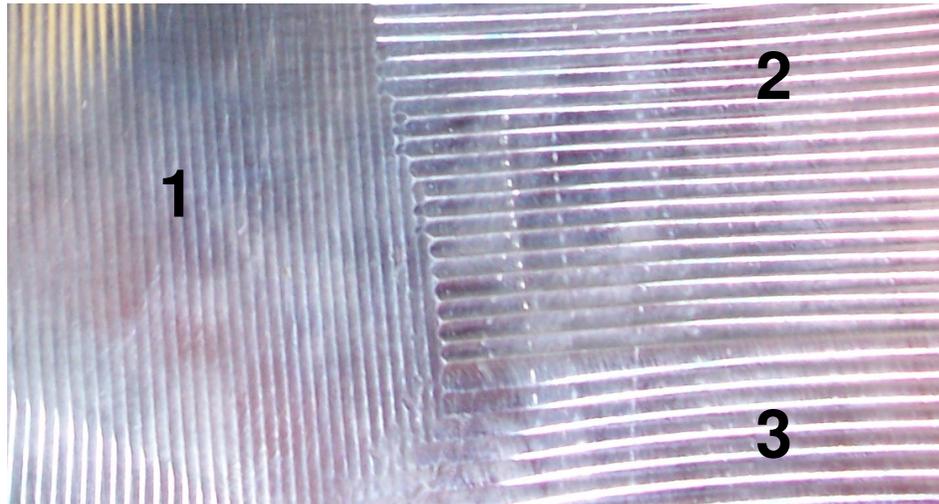
	<b>Number of patches</b>	<b>Type of tool path</b>	<b>Tool path length</b>	<b>Estimated time</b>	<b>Actual Mach. time</b>
Surface 1	1 patch	Zig-zag	22,535 mm	11.86 min	11.43 min
Surface 1	4 patches	Zig-zag	19,135 mm	10.52 min	10.23 min

<b>Tool pass length</b>	<b>Number of passes</b>	<b>Time</b>
450 mm.	50 passes	80 sec.
225 mm.	100 passes	94 sec.
112.5 mm.	200 passes	120 sec.
75 mm.	300 passes	150 sec.

**Table 6-9 Time test using maximum Feed Rate**

## 6.8 Boundary Marks

A machining test was conducted to evaluate the surface finish obtained with the proposed 3+2-axis machining strategy. For this test, the sample test surface 1 was partitioned into three patches and machined using a 1" toroidal cutter. The machined surface had a good surface finish, although the boundaries could be easily identified because of different side steps, as shown Figure 6-9. Small marks in the surface were generated between the patches' boundaries. The unevenness was still within tolerance and in cases when the patch boundary is parallel to the feed direction, i.e., between patch 2 and 3, the marks are almost negligible.



**Figure 6-9 Boundary marks**

## 6.9 Summary

This chapter described the methodologies required to machine a surface using the proposed 3+2-axis machining strategy. This chapter presented a series of tests to validate some of the methods developed in this work to improve the efficiency of 3+2-axis machining. Included in this chapter, were a method to define the number of sample points required to describe a surface. As well, simulation tests were conducted to determine appropriate parameters for the tool path generation phase.

Machining tests are conducted in the next chapter. These tests are conducted to validate the 3+2-axis machining strategy proposed in this work. For completeness, the results obtained in this work are compared with other common machining methods.

## Chapter 7

### Machining tests

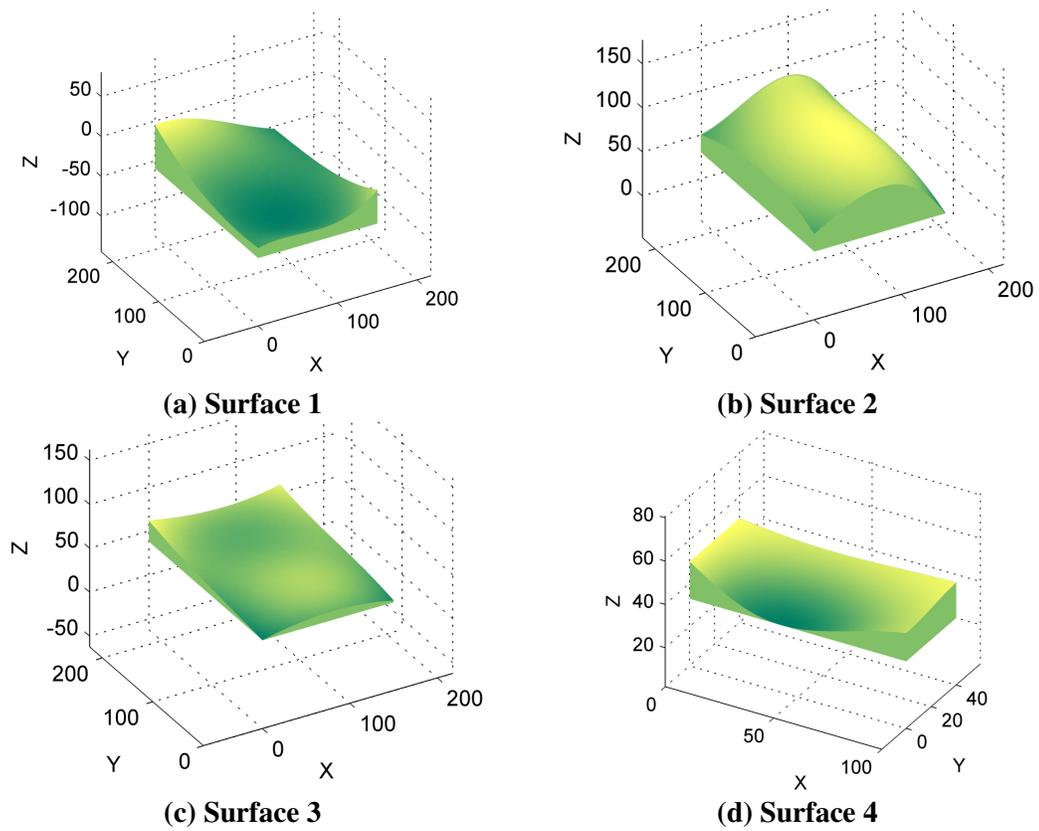
The method outlined in the previous chapter was applied to four surfaces. Actual machining tests of these surfaces were conducted to validate the 3+2-axis machining method and to verify the numerical estimations. The machining times obtained with the 3+2-axis machining method are compared with those using other common techniques described in literature for surface machining.

The machining tests were conducted on a Deckel Maho 80 P hi-dyn 5-axis machine. This five-axis machining center, shown in Figure 7-1, has a tilt-rotary type configuration and is capable of simultaneous movement in 3 translational and 2 rotational axes. The table moves linearly in the  $Y$ -direction while the head moves linearly in the  $X$ - and  $Z$ - directions. The table also tilts about the  $X$ -axis and rotates about the  $Z$ -axis (known as angles  $A$  and  $C$ , respectively). Although this machine can move all the five-axis simultaneously, for the 3+2-axis machining tests it was treated as an indexable machine and each patch was machined using only three axes,  $X$ ,  $Y$  and  $Z$ . The axes  $A$  and  $C$  were only used to set the inclination of each patch. In this way the machine, in effect, becomes a three-axis with a tilt/rotary fixture.

The machining tests were carried out on the four test surfaces shown in Figure 7-2. To represent each surface, a grid of 60x60 uniformly space points was used. The surface properties were calculated at the 3600 points and assembled into a feature vector. The control points and equations for these surfaces were given in section 3.1. These surfaces were selected because they resemble some of the characteristics found in dies and moulds.



**Figure 7-1 Deckel Maho 5-axis machining center**



**Figure 7-2 Test surfaces**

A comparison between the proposed 3+2-axis machining methodology and multi-axis machining strategies is presented in this chapter. This work will compare the results obtained with the 3+2-axis machining strategies developed with some of the most common methods used in the manufacturing industry. The comparison will be conducted using experimental cutting tests, which has not been the case in related work reported in literature. In particular, the configurations that will be included in this comparison are:

1. 3+2-axis (manual rotations)
2. 3+2-axis (automatic rotations)
3. 3-axis machining using a ball nose end mill
4. 5-axis machining using the "*Sturz*" method

The 3+2-axis machining strategy is developed to be carried out using 3-axis machines with the addition of a rotary/tilt table or on indexable 5-axis machines. In these machines the tilting and rotation are conducted in discrete steps that are independent of the other axes. The tool or workpiece orientation is conducted using manual rotations of the additional two axes. From observations on this work, each manual rotation of the axes was estimated to consume 1 minute. This estimation is used in the numerical simulations tests presented in this work required to determine the optimal partition of each surface.

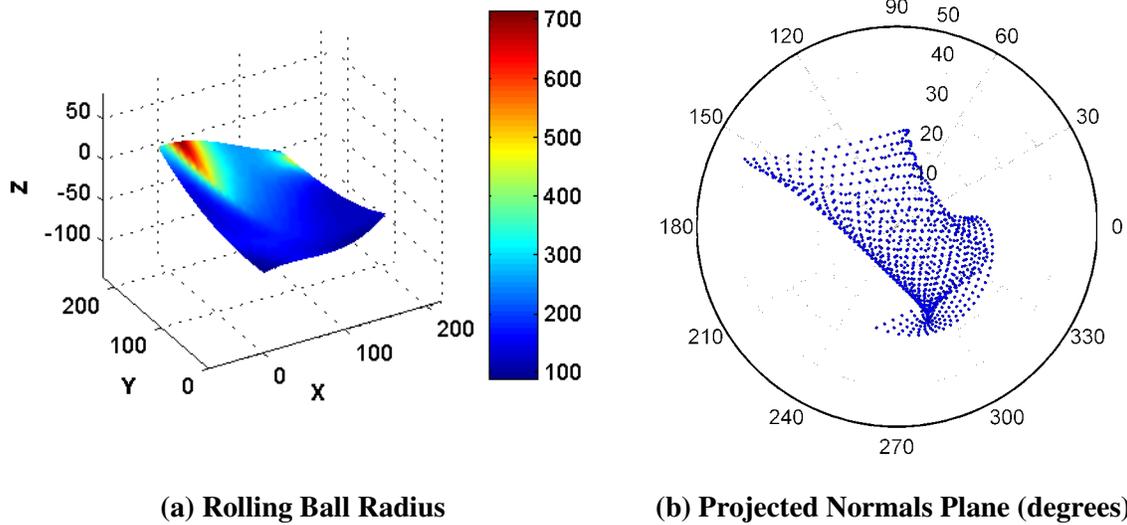
In a previous work by the author [36], it was shown that 3+2-axis machining results in competitive machining times compared to simultaneous 5-axis machining. Based on those results, it was considered that the proposed 3+2-axis machining strategy could be applied as a 5-axis machining technique. For those tests, each patch was machined using three axes and the tool orientation was set automatically using the axes *A* and *C*. This configuration consumes less time than the manual orientation and it was estimated to consume, on average, 5 seconds for each tool orientation.

For comparison purposes, true 3-axis and 5-axis machining tests are also conducted in this work. The 3-axis machining is carried out using a ball nose end mill and the 5-axis method is conducted using the "*Sturz*" method, where a fixed inclination angle of the tool with respect to the surface normal is used for tool positioning.

The cutting experiments will be conducted using the same machining parameters for all configurations. The toroidal tools available for the cutting tests were a 0.5", 0.75", 1", 1.25" and 1.5" diameter tools. The ball nose cutters available were a 0.5", 1", and 1.5" diameter tools. The maximum scallop height was defined at 0.0254 mm. The feed rate and spindle speed were specified as 2000 mm/min and 6000 RPM, respectively.

## **7.1 Surface 1**

The first machining test was conducted on a Bézier surface with convex, concave and saddle points. A major portion of this surface was classified as a high curvature surface using the classification method presented in section 6.2. The Rolling Ball Radius for this surface is shown on Figure 7-3 (a). The colour bar on the right of the surface represents the distribution of the radii along the surface. The Rolling Ball Radius is used to determine the maximum allowable tool radius that can be used to machine the surface. Also shown on this figure is the projected normals plane. This plot is used to get a perspective of the angles required to machine the surface and the distribution of the surface normals. This plot is also used to evaluate if there are points on the surface that may result in tool orientations off the limits of the machine.



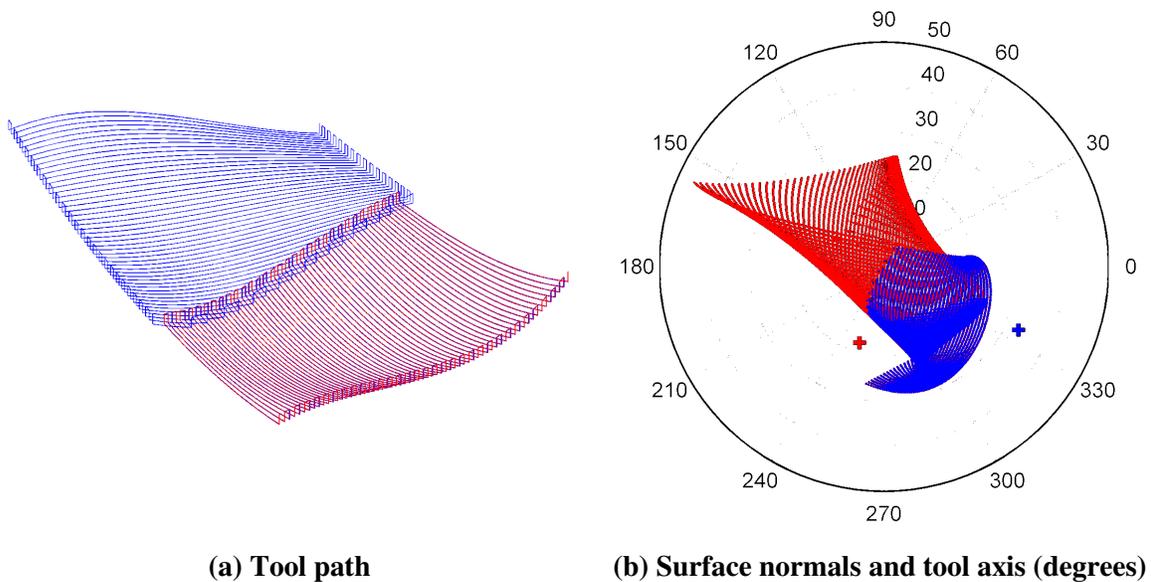
**Figure 7-3 Surface properties of Surface 1**

The numerical simulation tests for the first surface are shown in Table 7-1. This table includes the radii of the tool used for the cutting test, the tool path length and the machining times. The first machining time is estimated for a 3+2-axis machine using manual rotation of the axes, where the time for each workpiece orientation was estimated to be one minute. The second machining time is for a 3+2-axis machine using automatic rotation of the axes, where each orientation consumes five seconds.

Based on the numerical estimations, the surface, sub-divided into two patches, resulted in the minimum machining time. This partition and the correspondent tool path are shown on Figure 7-4(a). The feed direction for patch 1 (red points) is along the negative  $Y$  axis ( $[0 -1 0]$ ) and for patch 2 (blue points) is along the positive  $X$  axis ( $[1 0 0]$ ). Figure 7-4(b) shows the projected normals plane for this partition. These two plots are used to evaluate the appropriateness of the partition selected for machining. The first plot shows well separated regions and a simple boundary. The second plot shows that the distribution of the surface normals forms tight clusters. This graphical analysis proves that this partitioning is appropriate for machining.

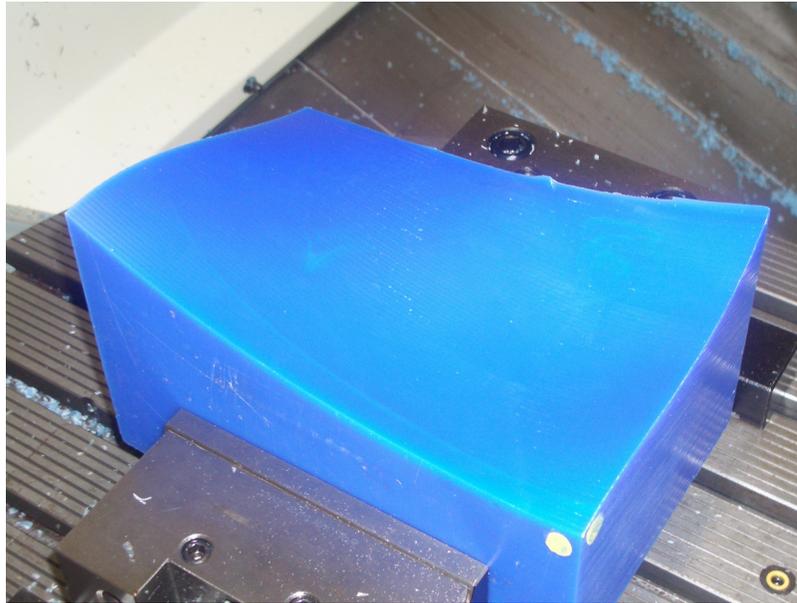
**Table 7-1 Numerical simulation tests for Surface 1**

	$R_1$ (mm)	$R_2$ (mm)	Tool path length	Mach. Time 3+2-axis (automatic)	Mach. time 3+2-axis (manual)
1 patch	13.05	6	17,098 mm	9.00 min	9.00 min
2 patches	13.05	6	16,314 mm	<b>8.74 min</b>	9.59 min
3 patches	13.05	6	16,402 mm	8.93 min	10.63 min
4 patches	13.05	6	15,904 mm	8.82 min	11.37 min
5 patches	13.05	6	15,606 mm	8.81 min	12.21 min
6 patches	13.05	6	15,588 mm	8.95 min	13.20 min



**Figure 7-4 Machining parameters for surface 1**

Figure 7-5 shows the machined surface photo. The actual machined time is listed in Table 7-2. The difference between the estimated time and the actual machining tests is small, which validates the estimations used in this work. The single patch using manual rotations is also included as a reference. This machining test also shows better machining times than the 3-axis and 5-axis machining. The optimal 3+2-axis machining time is shorter by approximately 13 % and 39 % compared to the 3-axis and 5-axis machining tests, respectively.



**Figure 7-5 Cutting test surface 1**

**Table 7-2 Machining time comparison for Surface 1**

	<b>Number of patches</b>	<b>R<sub>1</sub> (mm)</b>	<b>R<sub>2</sub> (mm)</b>	<b>θ (degrees)</b>	<b>Tool path length</b>	<b>Machining Time</b>
3+2-axis (manual)	1	13.05	6	5	17,098 mm	8.81 min
3+2-axis (automatic)	2	13.05	6	5	16,314 mm	<b>8.65 min</b>
3-axis	1	19.05	-	-	18,930 mm	9.96 min
5-axis	1	6.7	6	5	9,418 mm	14.39 min

Although the tool path length in the 5-axis machining test is lower than the other machining tests, this configuration resulted in the longest machining time. It was observed that the feed rate for this cutting test over most of the surface was around 500 mm/min ( $\frac{1}{4}$  of the programmed feed rate) and in some cases it went as low as 200 mm/min. These fluctuations in feed rate, also noted by Gray [2], result in longer than expected machining times and inconsistent surface finish.

Another problem encountered in the 5-axis machining time was that the original tool path had tool orientations off the limits of the machine (the tilting angle limit for the machine used in this

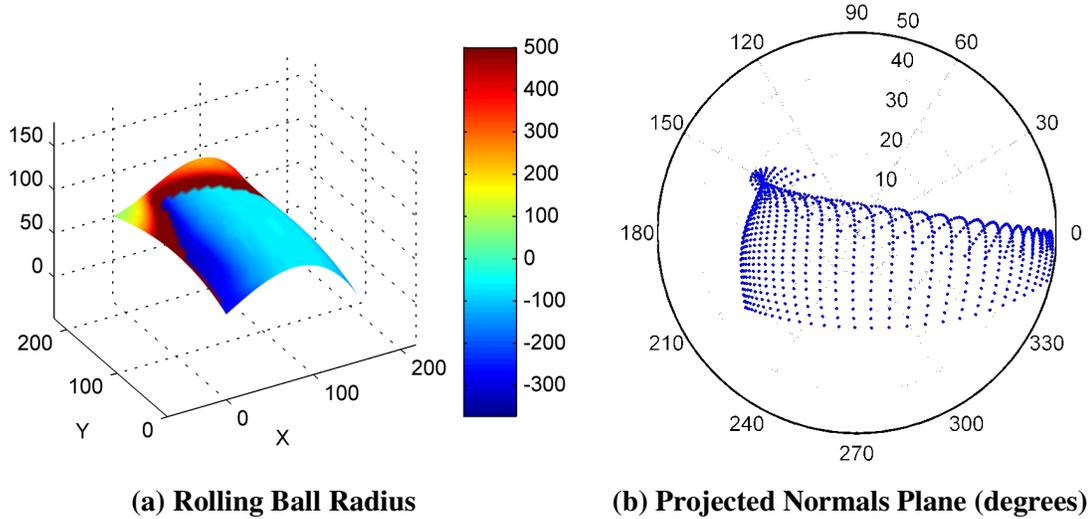
experiments is 30 degrees in one direction and 45 degrees in the other direction). To solve this problem, the tool axis was recalculated for these points that were off the limits.

Both the 3-axis and the 3+2-axis machining tests were conducted using 1.5” tools. However, for the 5-axis test this tool resulted in gouging and it was necessary to use a smaller radius. Alternatives to use a larger tool require recalculating appropriate tool orientations for the contact points that have a small radius of curvature. This can be achieved by using larger angles of inclination (between the tool axis and the normal) that result in smaller effective radius of the tool, or by orienting the tool to match the radius of curvature at the contact point. This process, however, requires the implementation of other 5-axis positioning methods or exhaustive searches of appropriate tool positions.

## **7.2 Surface 2**

The previous experiment showed some of the advantages of 3+2-axis machining and some of the difficulties encountered in 5-axis machining. Additional tests are conducted to validate the results and for a more comprehensive evaluation of 3+2-axis machining.

The second machining test is conducted on the surface shown in Figure 7-6(a). This surface, mainly convex, is also defined using Bézier equations and classified as a high curvature surface. The Rolling Ball Radius plot shows that this surface has a large region with a negative radius. For visualization purposes, all the radii larger than 500 are plotted using the same colour. In the majority of the points the radius of the points is large, which allows using larger tools. However, the projected normals plane, illustrated in Figure 7-6(b), shows some sample points with normals that are off the limits of the machine (< 45 degrees).

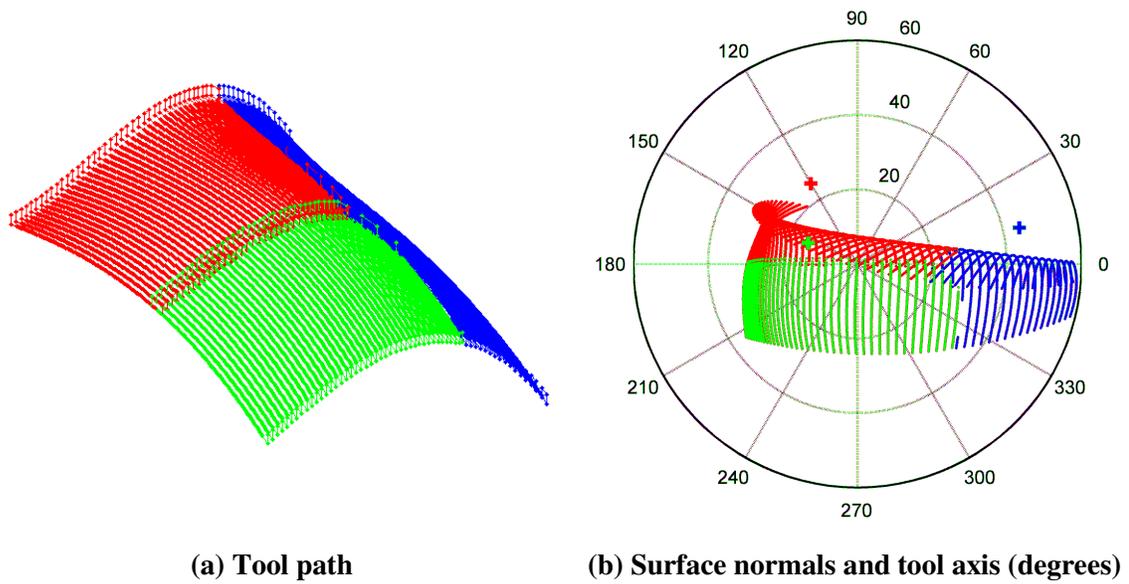


**Figure 7-6 Surface properties of Surface 2**

Table 7-3 presents the machining times for the sample test surface 2. This table shows that the minimum machining times are obtained with the 3 patch and 4 patch partitions. For simplicity, the 3 patch partition is selected for machining. The optimal partition and tool path are shown in Figure 7-7(a). The feed direction for all the patches is along the positive  $X$ - axis. This partition resulted in straight boundaries and regular shapes, which are appropriate for machining. The projected normals plane for this partition shows well separated and tight clusters that can be machined properly using a single tool orientation (Figure 7-7(b)).

**Table 7-3 Numerical simulation tests for Surface 2**

	$R_1$ (mm)	$R_2$ (mm)	Tool path length	Mach. Time 3+2-axis (automatic)	Mach. time 3+2-axis (manual)
1 patch	13.05	6	16,687 mm	8.78 min	<b>8.78 min</b>
2 patches	13.05	6	15,834 mm	8.48 min	9.33 min
3 patches	13.05	6	15,358 mm	<b>8.38 min</b>	10.08 min
4 patches	13.05	6	15,034 mm	<b>8.36 min</b>	10.91 min
5 patches	13.05	6	15,070 mm	8.53 min	11.93 min
6 patches	13.05	6	14,871 mm	8.58 min	12.83 min



**Figure 7-7 Machining parameters for surface 2**



**Figure 7-8 Cutting test surface 2**

Figure 7-8 shows the machined surface photo. The actual machining times for this surface were a slightly longer but still close to the estimated times. The machining time savings for the 3+2-axis compared with the 3-axis and 5-axis were 14 % and 22 %, respectively.

**Table 7-4 Machining time comparison for Surface 2**

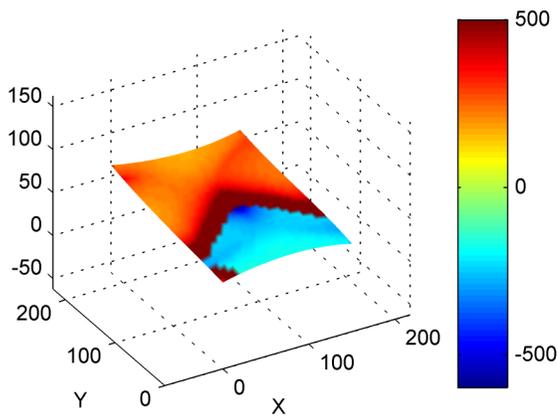
	<b>Number of patches</b>	<b>R<sub>1</sub> (mm)</b>	<b>R<sub>2</sub> (mm)</b>	<b>θ (degrees)</b>	<b>Tool path length</b>	<b>Machining Time</b>
3+2-axis (manual)	1	13.05	6	5	16,687 mm	8.85 min
3+2-axis (automatic)	3	13.05	6	5	15,358 mm	<b>8.51 min</b>
3-axis	1	19.05	-	-	18,734 mm	9.86 min
5-axis	1	5.525	4	6	11,016 mm	11.05* min

\* Estimated time

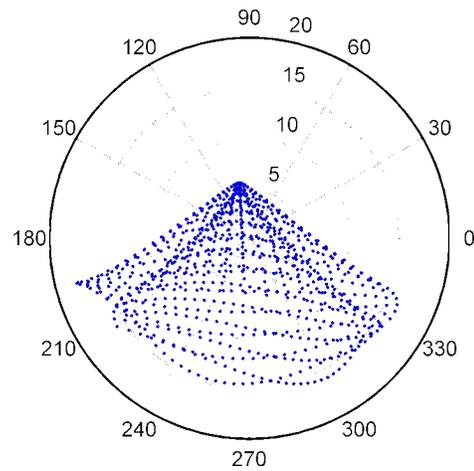
In the 5-axis test the machining time was estimated because the tool orientations required to machine it were off the limits of the machine. The estimated time was determined by extrapolating the time obtained to machine 80% of the surface. This problem was not encountered in either 3+2-axis or 3-axis machining. This surface is difficult to machine with 5-axis because the required tilting angles have a range close to 80 degrees. To machine this surface it may be required to calculate sub-optimal tool orientations that are on the limits of the machine or to machine half of the piece using one workpiece orientation and then re-orient the piece to machine the other portion of the surface.

### 7.3 Surface 3

The third machining test was conducted on a low curvature surface. This surface has a concave and convex section. This surface has large Rolling Ball radius which allows using the largest tool available for all the machining configurations. The projected normals plane show that the majority of the tilting angles are below 15 degrees (with respect to the z- axis), as shown in Figure 7-9(b).



(a) Rolling Ball Radius



(b) Projected Normals Plane (degrees)

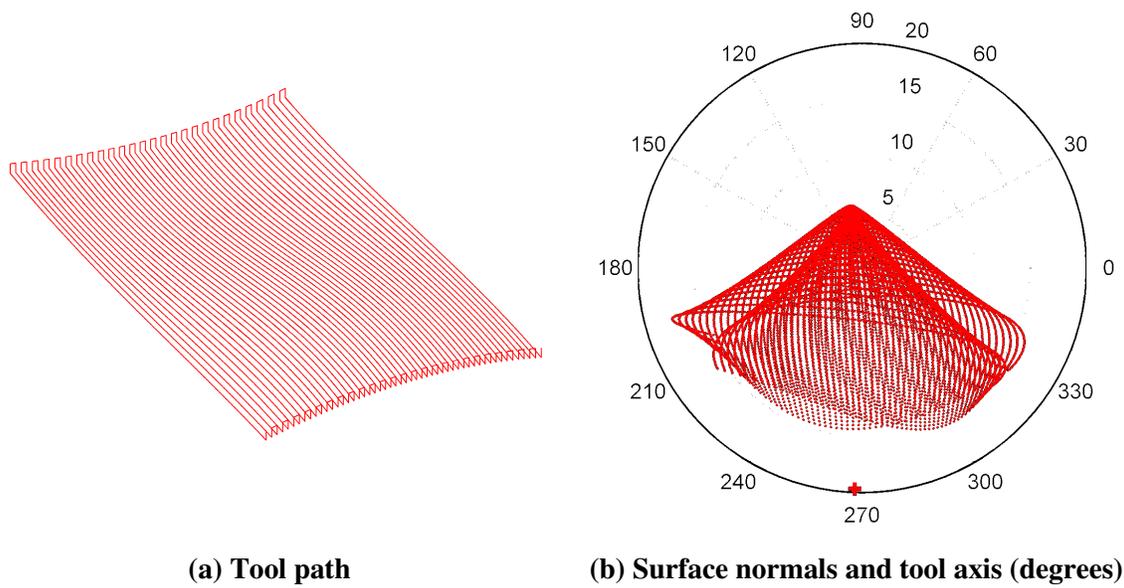
Figure 7-9 Surface properties of Surface 3

Table 7-5 Numerical simulation tests for Surface 3

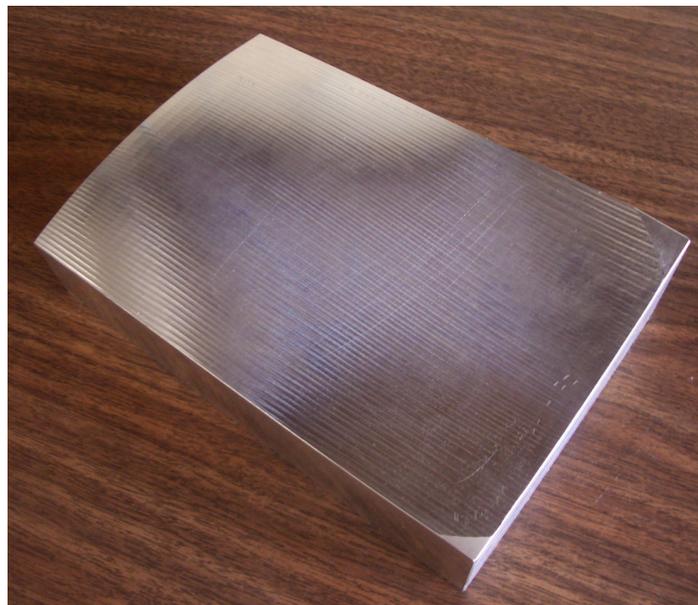
	$R_1$ (mm)	$R_2$ (mm)	Tool path length	Mach. Time 3+2-axis (automatic)	Mach. time 3+2-axis (manual)
1 patch	13.05	6	12,834 mm	<b>6.75 min</b>	<b>6.75 min</b>
2 patches	13.05	6	13,734 mm	7.38 min	8.23 min
3 patches	13.05	6	13,275 mm	7.29 min	8.99 min
4 patches	13.05	6	13,068 mm	7.33 min	9.88 min
5 patches	13.05	6	12,997 mm	7.44 min	10.84 min

Table 7-5 presents the machining times for surface 3. Based on the machining times obtained, the 3+2-machining test should be machined as a single patch. In comparison with the other surfaces, in this surface there are no reductions in the tool path length as the number of patches is increased. In this case, having more tool orientations does not reflect in larger side steps due to the small variations in the surface normals, yet, the partitions result in additional travel in between patches and shorter tool passes.

The tool path that provides the minimum machining time is shown in Figure 7-10(a). The feed direction for this surface is along the negative  $Y$ - direction. Figure 7-10(b) illustrates the distribution of the surface normals and the tool axis vector for this surface.



**Figure 7-10 Machining parameters for surface 3**



**Figure 7-11 Cutting test surface 3**

The machined surface photo is shown in Figure 7-11. The time comparison for this surface is shown in Table 7-6. The results obtained with the 3+2-axis show savings of 30% and 47% approximately, the largest for all the machining tests. Even when there were only small changes in curvature, the 5-axis machining test suffered from the slower rotations of the axes that resulted in smaller actual feed rates. Although the 3-axis tool path length is almost three times longer than the 5-axis machining test, the machining time is shorter by approximately 20%.

**Table 7-6 Machining time comparison for Surface 3**

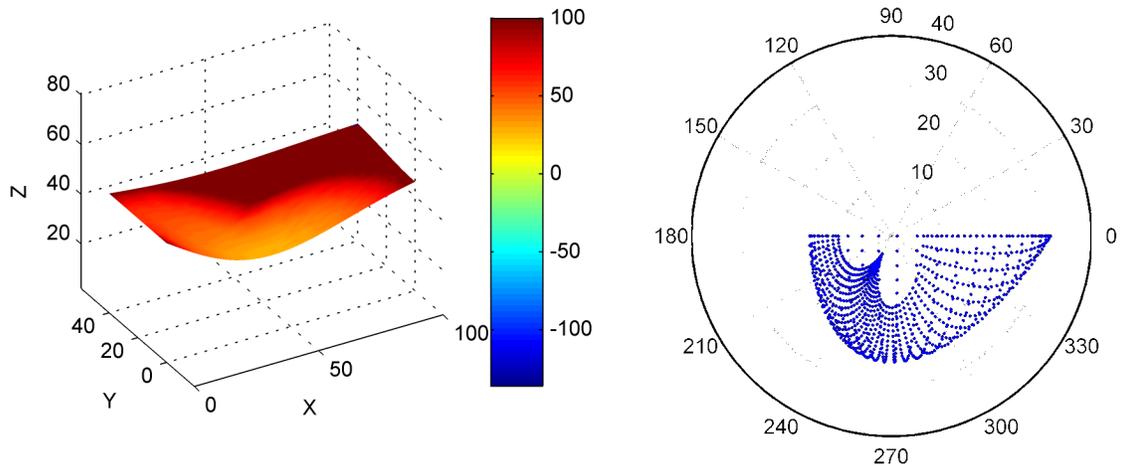
	Number of patches	R <sub>1</sub> (mm)	R <sub>2</sub> (mm)	θ (degrees)	Tool path length	Machining Time
3+2-axis (manual)	1	13.05	6	5	12,834 mm	<b>6.67 min</b>
3+2-axis (automatic)	1	13.05	6	5	12,834 mm	<b>6.67 min</b>
3-axis	1	19.05	-	-	18,448 mm	9.61 min
5-axis	1	13.05	6	5	6,763 mm	12.51 min

## 7.4 Surface 4

The last cutting test was conducted on a smaller surface (approximately 10% of the area of the other surfaces). This surface, in comparison to the other surfaces that were described using Bézier equations, is described using an algebraic equation (given in equation (3.3)). This required making changes in the original program used to generate the tool paths, as the program was based on the parametric equations and Bézier parameters. The new program developed was created to be able to produce tool paths by only requiring the surface coordinates and surface normals at specific points from the surface. These changes were developed for a future implementation of the program for industrial parts.

Figure 7-12 shows the Rolling Ball Radius and projected normals plane for this surface. The first plot shows that there is a portion of the surface where the Rolling Ball Radius is small, around 20-30 cm.

A small radius may limit the use of larger effective radius that can cause gouging. The projected normals plane shows that the normals are grouped in a compact space. In this case the tips of the normals are in the lower quadrants, which indicates that it may be convenient to machine all the points using a single tool orientation.



(a) Rolling Ball Radius

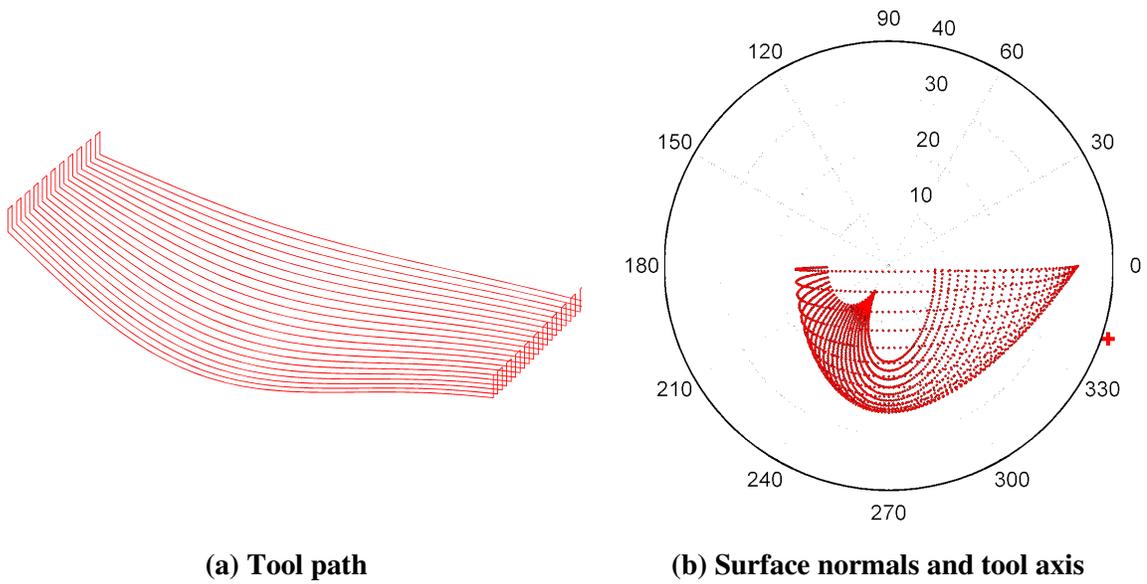
(b) Projected Normals Plane (degrees)

**Figure 7-12 Surface properties of Surface 4**

The machining times and tool path lengths for this surface are listed in Table 7-7. Considering that this surface was smaller and that the tool path lengths were not diminishing as the number of partitions was increased, the simulation tests were conducted up to 4 patches only. The single patch resulted in the smallest tool path length and shortest machining time for this surface. The single patch used a larger tool in comparison to the other partitions. As the number of patches is increased, the effective radius of the tool gets bigger because the tool is closer to the surface normals. The larger effective radius, however, restricts the partitions to use larger tools due to the small radius of some points on the surface.

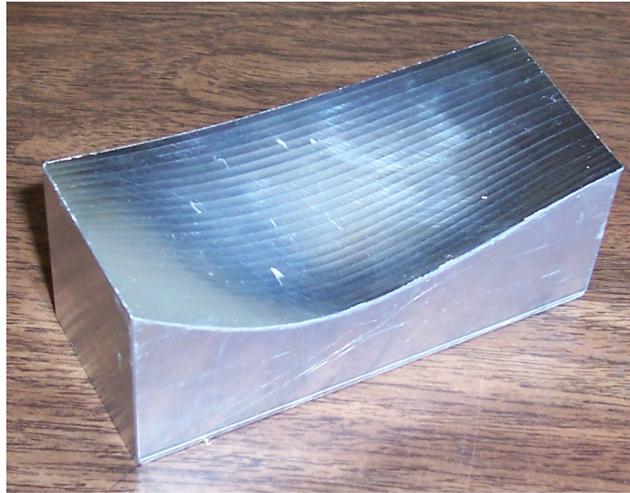
**Table 7-7 Numerical simulation tests for Surface 4**

	<b>R<sub>1</sub></b> <b>(mm)</b>	<b>R<sub>2</sub></b> <b>(mm)</b>	<b>Tool path</b> <b>length</b>	<b>Mach. Time</b> <b>3+2-axis (automatic)</b>	<b>Mach. time</b> <b>3+2-axis (manual)</b>
1 patch	6.7	6	2,813 mm	<b>1.48 min</b>	<b>1.48 min</b>
2 patches	2.35	4	3,373 mm	2.02 min	2.87 min
3 patches	2.35	4	3,862 mm	2.45 min	4.15 min
4 patches	2.35	4	3,340 mm	2.31 min	4.86 min



**Figure 7-13 Machining parameters for surface 4**

Figure 7-13 shows the tool path and the tool axis for surface 4. The feed direction for this surface is along the positive X- direction. The machined surface photo is presented in Figure 7-14.



**Figure 7-14 Cutting test surface 4**

Table 7-8 shows the machining time comparison for surface 4. The minimum machining time for this surface was obtained in the cutting tests conducted using the 3-axis machine. This test used the largest tool available and resulted in a shorter tool path length compared to the 3+2-axis machining test. The 5-axis machining test used the smallest tool available which resulted in longer machining time. The angle of inclination for this test was 6 degrees, as the 5 degrees inclination resulted in gouging.

**Table 7-8 Machining time comparison for Surface 4**

	Number of patches	R <sub>1</sub> (mm)	R <sub>2</sub> (mm)	θ (degrees)	Tool path length	Machining Time
3+2-axis (manual)	1	6.7	6	5	2,813 mm	1.56 min
3+2-axis (automatic)	1	6.7	6	5	2,813 mm	1.56 min
3-axis	1	19.05	-	-	2,448 mm	<b>1.36 min</b>
5-axis	1	2.35	4	6	2,086 mm	2.75 min

## 7.5 Summary

Initially, the objective of this research was to develop a methodology that could improve the machining efficiency of 3-axis. The 3+2-axis machining strategy was intended to provide more flexibility and reductions in machining time. However, as the investigation went on it was noted that this methodology had also the potential to become an alternative for 5-axis machining. 3+2-axis could be used as learning tool for an eventual migration for simultaneous 5-axis, but also, as noted in the results obtained in the machining tests, can be considered an efficient and cost-effective alternative for surface machining.

The tests conducted have verified that the developed strategy can identify the number of patches that provides the lowest machining time while satisfying the surface requirements. These tests provide a good perspective of the advantages of 3+2-axis machining. For completeness, the proposed 3+2-axis machining was compared with other multi-axis machining strategies.

The tests presented in this chapter showed some of the applications of the projected normals plane for multi-axis machining. This visualization tool can be used as a guide to determine some machining parameters.

Studies in this work focused on improving the quality of the surface finish. An improved method for tool path was developed in this work. This tool path strategy that lifts the tool between passes and overlaps passes helped to reduce the boundary marks. Also, the constant feed rate obtained in the 3+2-axis machining tests helps to obtain a consistent surface finish.

In some of the machining tests the difference between the minimum machining times obtained with the 3+2-axis manual and with the 3+2-axis automatic were not significant. In some cases, for simplicity it is more convenient to machine the surface using as a single patch. Further tests should

include machining of larger pieces and more complicated shapes. It should be convenient to evaluate the partitioning scheme in surfaces that require several tool orientations.

The 3+2-axis machining tests resulted in machining time savings and were easier to program over 5-axis machining. Once the tool orientation is locked, the tool moves only in three linear axes, which allows predicting the trajectory of the tool more easily. The 3+2-axis machining tests were simpler to run. In contrast, the 5-axis tool paths require special attention to prevent any type of collisions, as the workpiece orientation is changed continuously during cutting. Also noted in these experiments was that the tool seem to be more rigid in 3+2-axis machining, due to the locked position of the tool during cutting. Further studies should be conducted to evaluate the use of higher feed rates.

## Chapter 8

### Conclusion

This work showed that 3+2-axis machining can be an efficient and practical alternative for surface machining. Although this method showed improvements in machining time, the biggest advantage is the reduced investment in machine cost and operator training.

3+2-axis machining is cost-effective alternative for surface machining. This technique facilitates the transition from 3-axis machining and can be used as a learning tool to develop the knowledge for a migration to simultaneous 5-axis machining.

The objective of the development of the 3+2-axis machining method is to provide a guide to the CNC operator through the process of making programs for machining complex surfaces. This work described the procedures required to machine a surface.

The proposed 3+2-axis machining method is based on the partitioning of surfaces. 3+2-axis machining normally requires identifying regions that can be machined using a particular tool orientation. However, the partitioning of surfaces is frequently conducted subjectively by the operator. The lack of guidelines to conduct this procedure and insufficient research in this field were the main motives to conduct this study.

The uncertainty of what surface properties should be included in the partitioning process can be disconcerting. To address this issue, graphical and numerical tests were conducted to evaluate the influence of the parameters. Tests were conducted to evaluate different combination of parameters and identify groups that result in good partitions.

Partitioning depends on the geometric properties used for clustering that form the feature vector. The effect of various geometric properties was studied on sample surfaces and a list of properties

belonging to three categories namely proximity, orientation and shape were identified. It was shown that although these properties can be grouped in various combinations and with varying weights, the combination of the surface coordinates and the normal vectors consistently results in good partitions.

The number of partitions depends on the user and the surface at hand. The number of partitions is difficult to determine because of two opposing effects. A large number of patches leads to a better match between the tool and the workpiece, but it also leads to many tool re-orientations between patches. On the other hand, if the number of patches is small, the benefit of the method is not fully realized since the shape of the tool may vary greatly from that of the surface. Accordingly, a technique for selecting the optimum number of partitions was presented in this work. This technique is conducted by estimating the machining time for a range of patches selected by the user. The partitioning that results in the smallest machining time is selected for machining.

An examination of different approaches for the classification of cutter contact points was also presented in this work. Based on numerical and graphical studies, the Minimum Intra-Class Distance (MICD) method was implemented in the proposed 3+2-axis machining for the identification of boundaries of the patches. This technique was developed to guarantee that the cutter contact points are machined with the appropriate tool parameters.

This work presented new and improved methods for tool path generation. A new graphical tool, the projected normals plane, was developed to assist in the selection of machining parameters. This projected normals plane offers a visual representation of the surface normals, and is used in this work in the process of selecting feed directions and tool orientations.

The results obtained in the cutting tests verified the numerical estimations. The experiments confirmed that the proposed 3+2-axis machining strategy is less difficult to program, and resulted in shorter machining times than 5-axis machining. This strategy requires only the surface coordinates

and surface normals, which can facilitate its further implementation for different types of surfaces and industrial parts.

This work also compared the 3+2-axis method with existing 3-axis and 5-axis machining techniques. This comparison with other multi-axis machining methods is not conducted in related work reported in literature. The tests confirmed that 3+2-axis machining is an effective alternative for machining of complex surfaces.

### **8.1 Future work**

The results obtained in this work show the potential that 3+2-axis machining strategy can have in the manufacturing industry. This work lays out the foundation for further research and for the implementation of the method for the machining of industrial parts.

Although the machining conditions for the 3+2-axis machining and 5-axis were the same, 3+2-axis machining operations can be optimized with the use of higher machining feed rates that can result in further improvements.

The proposed methodology developed in this work provides a platform to implement future machining strategies. The surface partitioning method has only been applied to 3+2-axis machining. It would be useful to adapt this methodology for 5-axis machining. As well, some other methods developed in this work such as the projected normals plane and the methods to determine an appropriate feed direction, tool orientation and tool size could also be used for 5-axis machining.

## Bibliography

- [1] G. W. Vickers and K. Quan (1989) Ball-mills versus end-mills for curved surface machining. *ASME Journal of Engineering for Industry* 111:22-26.
- [2] P. J. Gray (2004) Tool positioning for 3+2-axis and 5-axis surface machining. *Ph. D. Thesis*, University of Waterloo.
- [3] M. Munlin, S. S. Makhanov and E. L. J. Bohez (2004) Optimization of rotations of a 5-axis milling machine near stationary points. *Computer-Aided Design* 36:1117-1128.
- [4] K. Sheltami, S. Bedi and F. Ismail (1998) Swept Volumes of Toroidal Cutters Using Generating Curves. *Int. Journal of Machine Tools and Manufacture* 38:885-870.
- [5] R. Wein, G. Elber, O. Ilushin and D. Halperin (2004) Continuous path verification in multi-axis NC-Machining. *Proceedings of the Twentieth Annual Symposium on Computational Geometry*: 86-95.
- [6] H. D. Cho, Y. T. Jun and M. Y. Yang (1993) 5-axis CNC milling for effective machining of sculptured surfaces. *Int. J. Prod. Res.* 31(11):2559-2573.
- [7] P. Gray, S. Bedi, F. Ismail, N. Rao and G. Morphy (2001) Comparison of 5-Axis and 3-Axis Finish Machining of Hydroforming Die Inserts. *Int. J. of Advanced Manufacturing Technology* 17:562-569.
- [8] P. Schmoll (2001) HSC-Technology in Die & Mould – and Series Production. *DMG Canada*.
- [9] A. Kaldos, I. F. Dagiloke and A. Boyle (1996) Computer Aided Cutting Process Parameter Selection for High Speed Milling. *Journal of Materials Processing Technology* 61:219-224.
- [10] R. I. King and R. L. Vaughn (1984) Synoptic Review of High-Speed Machining from Salomon to the Present. *American Society of Mechanical Engineers* 12:1-13.
- [11] A. Warkentin, F. Ismail and S. Bedi (2000) Multi-point tool positioning strategy for 5 axis machining of sculptured surfaces. *Computer Aided Geometric Design* 17: 83-100.
- [12] N. Rao, F. Ismail and S. Bedi (1997) Tool Path Planning for 5-axis Machining Using the Principal Axis Method. *Int. Journal of Machine Tools and Manufacture* 37(7):1025-1040.
- [13] N. Rao, S. Bedi and R. Buchal (1996) Implementation of the Principal-Axis Method for Machining of Complex Surfaces. *International Journal Advanced Manufacturing Technologies* 11:249-257.

- [14] P. Gray, F. Ismail and S. Bedi (2003) Rolling Ball Method for 5-axis surface machining. *Computer-Aided Design* 35:347-357.
- [15] P. Gray, F. Ismail and S. Bedi (2004) Graphics-assisted Rolling Ball Method for 5-axis surface machining. *Computer-Aided Design* 36(7):653-663.
- [16] P. Broomhead and M. Edkins (1986) Generating NC data of the machine tool for the manufacture of free-form surfaces. *International Journal of Production Research* 24: 1-14.
- [17] S. X. Li and R. B. Jerard (1994) Five-axis machining of sculptured surfaces with flat-end cutter. *Computer-Aided Design* 26:165-178.
- [18] Y. Huang and J. H. Oliver (1994) Non-constant NC tool path generation on sculptured surfaces. *International Journal Advanced Manufacturing Technologies* 9:281-290.
- [19] C. G. Jensens and D. C. Anderson (1993) Accurate tool placement and orientation for surface machining. *Journal of Design and Manufacturing* 3: 251-261.
- [20] J. Pi, E. Red and G. Jensen (1998) Grind-free tool path generation for five-axis surface machining. *Computer Integrated Manufacturing Systems* 11:337-350.
- [21] Y-S Lee (1998) Non-isoparametric tool path planning by machining strip evaluation for 5-axis sculptured surface machining *Computer-Aided Design* 30(7):559-570.
- [22] B. Lauwers, J. P. Kruth and P. Dejonghe (2001) An operation Planning System for Multi-Axis Milling of sculptured Surfaces. *International Journal Advanced Manufacturing Technologies* 17:799-804.
- [23] T. Kim and S. E. Sarma (2001) Toolpath generation along directions of maximum kinematic performance; a first cut at machine-optimal tool paths. *Computer-Aided Design* 34: 453-468.
- [24] C. J. Chiou and Y-S Lee (2002) A machining potential field approach to tool path generation for multi-axis sculptured surface machining. *Computer-Aided Design* 34:357-371.
- [25] J-M Redonnet, W. Rubio, F. Monies and G. Dessin (2000) Optimising tool positioning for end-mill machining of free form surfaces on 5-axis machines for both semi-finishing and finishing. *International Journal Advanced Manufacturing Technologies* 16(6):383-391.
- [26] B. Lauwers, P. Dejonghe and J. Kruth (2003) Optimal and collision free tool posture in 5-axis machining through the tight integration of tool path generation and machine simulation. *Computer-Aided Design* 35(5):421-432.

- [27] Y-S Lee (1997) Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining. *Computer-Aided Design* 29(7):507-521.
- [28] C-S Jun, K. Cha and Y-S Lee (2003) Optimizing tool orientations for 5-axis machining by configuration-space search method. *Computer-Aided Design* 35:549-566.
- [29] P. Gray, F. Ismail and S. Bedi (2003) Cusp Modeling for 5-Axis Surface Machining. *Symposium on Solid Modeling and Applications* 340-345.
- [30] C. Yun, J. W. Park, H. Shing and B. K. Choi (1998) Modeling the surface swept by a generalized cutter for NC verification. *Computer-Aided Design* 30(8):587-594.
- [31] W. L. Ralph and M. Loftus (1993) The application of an inclined end mill machining strategy on 3-axis machining centres. *Int. Journal of Machine Tools and Manufacture* 33(2):115-133.
- [32] S-H Suh and J-K. Kang (1995) Process planning for multi-axis NC machining of free surfaces. *International Journal Production Research* 33(10):2723-2738.
- [33] S-H Suh, J-J. Lee and S. Kim (1998) Multiaxis machining with additional-axis NC system: Theory and development. *International Journal Advanced Manufacturing Technologies* 14:865-875.
- [34] Z. C. Chen, Z. Dong and G. W. Vickers (2003) Automated surface subdivision and tool path generation for 3+2-axis CNC machining of sculptured parts. *Computers in Industry* 50: 319–331.
- [35] A. Roman (2003) 3+2-axis patch-by-patch NC machining of sculptured surfaces. *MASc Thesis*, University of Waterloo.
- [36] A. Roman, S. Bedi, F. Ismail (2006) Three-half and half-axis patch-by-patch NC machining of sculptured surfaces. *Int. Journal of Advanced Manufacturing Technology*, 29:524-531.
- [37] R. O. Duda, P. E. Hart and D. G. Stork (2001) *Pattern Classification*, Wiley Interscience.
- [38] M. Friedman, A. Kandel (1999) *Introduction to pattern recognition*, Statistical, structural, neural and fuzzy logic approaches, World Scientific, Signapore.
- [39] A Tutorial on Clustering Algorithms.  
[http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial\\_html/kmeans.html](http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/kmeans.html)
- [40] K. Nezis and G. Vosniakos (1997) Recognizing 2D shape features using a neural network and heuristics. *Computer Aided Design* 7:523-539.
- [41] J. C. Bezdek (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press.

- [42] Matlab : Statistical Toolbox User's Guide. (1998) The Mathworks, Inc.3.
- [43] G. Berks, D.G. Keyserlingk, J. Jantzen, M. Dotoli, H. Axer (2000) Fuzzy Clustering- A Versatile Mean to Explore Medical Database, ESIT, Aachen, Germany.
- [44] Matlab : Fuzzy Logic Toolbox User's Guide. (1998) The Mathworks, Inc.3.
- [45] S. Theodoris and K. Koutroumbas (2003) Pattern Recognition, Academic Press, Elsevier.
- [46] S. J. Leon (1998) Linear Algebra with Applications 5<sup>th</sup> edition Prentice Hall.
- [47] B. Kolman (1997) Introductory Linear Algebra with Applications. Prentice Hall.
- [48] M. Wölfel, H. Kemal Ekenel (2005) Feature Weighted Mahalanobis Distance: Improved Robustness for Gaussian Classifiers 13th European Signal Processing Conference, EUSIPCO 2005, Antalya, Turkey, 2005-09.
- [49] R. Donovan (2001) A new distance measure for costing spectral discontinuities in concatenative speech synthesizers, Proc. 4th ISCA Tutorial and Research Workshop on Speech Synthesis.
- [50] L. Klein (2004) Sensor and Data Fusion: A Tool for Information Assessment and Decision Making. Press Monograph.
- [51] [http://en.wikipedia.org/wiki/Euclidean\\_distance](http://en.wikipedia.org/wiki/Euclidean_distance)
- [52] X. Huang, A. Acero, and H.W. Hon (2001) Spoken language processing. Prentice Hall.
- [53] <http://www.ics.uci.edu/~eppstein/gina/voronoi.html>
- [54] <http://www.cs.sunysb.edu/~algorithm/files/voronoi-diagrams.shtml>
- [55] Matlab : Functions (1998) The Mathworks, Inc.3.
- [56] O. Joseph (1998) Computational Geometry in C, Cambridge University Press, Cambridge
- [57] Steve S. Skiena (1998) The Algorithm Design Manual, Springer.
- [58] Joseph O'Rourke (2000) Computational Geometry in C, Cambridge University Press.
- [59] Mark De Berg, Marc Van Kreveld, Mark Overmars, and O. Schwartzkopf. (1997) Computational Geometry : Algorithms and Applications, Springer.
- [60] E. Fix and J.L. Hodges (1951) Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties, Technical Report TR4, US Air Force School of Aviation Medicine, Randolph Field, Tex.
- [61] H. Wang (2006) Nearest neighbors by neighborhood counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v 28, n 6, p 942-95.
- [62] C. Elkan (1999) Results of the KDD '99 Classifier Learning Contest, <http://www.cs.ucsd.edu/users/elkan/clresults.html>.

- [63] H. Hayashi, J. Sese, and S. Morishita (2001) Optimization of Nearest Neighborhood Parameters for KDD-2001 Cup ‘the Genomics Challenge’,” technical report, Univ. of Tokyo, <http://wwwtsujii.is.s.utokyo.ac.jp/GENIA/WS/PDFfiles/Morishita.pdf>.
- [64] [http://en.wikipedia.org/wiki/Nearest\\_neighbor\\_\(pattern\\_recognition\)](http://en.wikipedia.org/wiki/Nearest_neighbor_(pattern_recognition))
- [65] S.A. Dudani (1976) The Distance-Weighted k-Nearest-Neighbor Rule,” *IEEE Trans. Systems, Man, and Cybernetics*, vol. 6, pp. 325-327.
- [66] <http://www.mentorsoftwareinc.com/cc/gistips/TIPS0199.HTM>
- [67] Robert D. Cook. (1995) Finite element modeling for stress analysis. John Wiley and Sons Inc.
- [68] Edward R. Champion Jr. (1992) Finite element analysis manufacturing engineering, McGraw Hill Inc.
- [69] Y. K. Cheung and M. F. Yeo (1979) A practical introduction to finite element analysis. Pitman publishing limited.
- [70] J. H. Vandenbrande, and A. A. G. Requicha (1993) Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(12):1269–1285.
- [71] A.K.W. Chan and K. Case (1994) Process planning by recognizing and learning machining features. *International Journal of Computer Integrated Manufacturing* 2: 77-99.
- [72] S. Joshi and T. C. Chang (1988) Graph-based heuristics for recognition of machined from a 3D solid model. *Computer-Aided Design* 20(2):58–66
- [73] L. De Floriani (1989) Feature extraction from boundary models of 3D objects. *IEEE Pattern Analysis and Machine Intelligence* 11(8):785–798
- [74] Y. S. Kim (1994) Volumetric feature recognition using convex decomposition. *In Advances in Feature Based Manufacturing*, J. J. Shah, M. Mantyla and D. S. Nau (eds) 39-63
- [75] H. Sakurai (1995) Volume decomposition and feature recognition Part 1: Polyhedral objects. *Computer Aided Design* 11:833-843
- [76] M. R. Henderson and S. Prabhakar (1992) Automatic form-feature recognition using neural-network-based technique on boundary representations of solid models. *Computer Aided Design* 7:381-393
- [77] W. D. Li, S. K. Ong and A. Y. C. Nee (2003) A hybrid method for recognizing interacting machining features. *International Journal of Production Research* 41(9):1887-1908

- [78] Z. C. Chen (2002) Optimal and Intelligent Multi-Axis CNC Tool Path Generation for Sculptured Part Machining, *Ph. D. Thesis*, University of Victoria
- [79] M. Negnevitsky (2002) Artificial Intelligence. *Addison Wesley*