AUTOMATED VOLUMETRIC FEATURE EXTRACTION FROM THE MACHINING PERSPECTIVE

BY

HESAMODDIN AHMADI

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgements

First and foremost, I would like to take this opportunity to express my most sincere gratitude and appreciation to my supervisor, Dr. Zhang Yun Feng, for his invaluable guidance, advice, and discussions throughout the entire duration of the project. He has been greatly helpful not only for his expertise and knowledge, but also for his continuous support.

I would also like to thank Dr. Lingling for her guidance and taking the time to help me. I am especially grateful for her friendship. She has always been there to listen and support me over the past few years.

Thanks are also given to my family for their never failing prayers, love and support. I could not have made it this far in life without them.

I would also like to thank all those people I met in the Internet who gave me much useful information of my research.

Last, but not the lease, I would like to thank A*STAR for providing me the research scholarship to support my studies.

Contents

List of Tables

List of Figures

Summary

It is well known that computer-aided process planning (CAPP) is the bridge between computer-aided design (CAD) and computer-aided manufacturing (CAM). Especially, with the competition in the market place, more and more companies want to improve their product efficiency and reduce cycle time. Under this condition, CAPP is developed integrating with other manufacturing functions.

The role of CAPP is to obtain CAD data of a part and then generate a sequenced set of instructions to manufacture the part. In order to do that, CAPP has to interpret the part in terms of features. Therefore, feature recognition could be considered as a front end to the CAPP function.

The focus of this thesis is to present a new feature recognition method aiming at recognizing volumetric features from the delta volume (DV), which is the material difference between the part and the stock. The volumetric feature can then be used for feature-based tool path generation directly. To this end, the DV is firstly decomposed into accessible delta volumes (ADVs) along all possible tool approach directions (TADs). The ADVs along each TAD are then decomposed into individual volumetric features (drilling, $2\frac{1}{2}$ D milling, and 3D milling) in which feature interaction problems are resolved and a feasible removal sequence is also established. The proposed algorithm allows multiple feature interpretations with valid manufacturability.

The developed method has been implemented and case studies show that it is able to handle complicated realistic parts that can be produced using a 3-axis machining centre and there is no limitation to the shapes of final part and stock.

CHAPTER 1

INTRODUCTION

1.1 Background

By decreasing the cost of computing and increasing its capability, nowadays, computers are widely used in design and manufacturing industries. Competition in a modern market demands production of high quality products in the shortest possible time. In response to fulfill this requirement, companies devote much effort to develop technologies which can improve productivity and Computer Integrated Manufacturing (CIM) is as an effective tool to increase manufacturing competitiveness [1].

Computer-Aided Process Planning (CAPP) is a key to CIM and is the application of computer to assist process planners in the planning functions [3]. This chapter presents a brief review of related concepts involved in the development of a CAPP system.

1.2 Computer-Aided Process Planning (CAPP)

A process is a method to manufacture parts from raw materials into the desired form. There are various manufacturing processes used for converting raw material into finished parts. These processes include casting, forging, punching, forming, machining, heat treatment, plating and so on. Among them, the machining process plays an important role in the manufacture of parts. The commonly used machining processes include various operations, such as turning, milling, drilling, grinding, broaching, etc., depending on the required shape, dimension, accuracy and surface quality of the part.

A process plan is, then, a sequence of instructions which determines exactly how a product can be made in the most efficient and effective way. Process planning provides a link between the design and manufacturing functions. After a product is designed, planning the processes of its components is the first step of preparatory work for manufacturing. The quality of a process plan should be evaluated from both technological and economical standpoints [2].

At present, computers are widely used in design and manufacturing. Computer aided-process planning (CAPP) is the application of computers to aid the process planner to offload some of the manual woks by using information and computerized algorithms to select proper manufacturing conditions [2].

1.3 CAD /CAM Integration and CAPP

CAPP serves as a bridge between CAD and CAM. It determines how a design will be made in a manufacturing system. Without successful CAPP, it is impossible to transform the design information into manufacturing. It is for this reason that CAPP is often referred to as a critical step in achieving CIM.

CAD systems generate graphically oriented information and may go as far as geometrically identifying material to be removed during machining. In order to produce NC instructions for CAM equipment, basic decisions regarding equipment to be used, tooling and operation sequence need to be made. This is the function of CAPP. Hence, without elements of CAPP, there would not be such a thing as CAD/CAM integration.

Although many technical problems arising in CAD and CAM are complicated and are difficult to solve, most of them are deterministic and involve a limited number of factors. CAPP, however, involves substantial technological decision making and the relationships among these CAPP decisions are complicated. This indicates the level of difficulty associated with CAPP [1].

1.4 Input to CAPP

In the conventional manufacturing system, two sets of information are presented to a process planner in form of engineering drawing [3]:

1) The geometrical and technological constraints in the part.
2) The manufacturing resources available on the shop floor.

Thus, engineering drawing can be considered as a bridge between design and manual process planning functions. Analogously, the development of CAPP system requires computer modeling for the following items:

1) Part modeling: It means computerized representation of part to be manufactured.
2) Manufacturing resources: This information should be made available to the CAPP system during its decision making procedure.
3) Process plan: It involves representation of the resultant process instructions in a structured form.

CAPP can be viewed as a modeling of the above elements and the interaction between them. The remained of this chapter is focused on the part modeling methods in CAPP systems.

As it is discussed in the previous section, one of the mandatory steps towards automation of process planning is to describe the part in a computer interpretable format. However, since human expertise and knowledge plays a major role in a manufacturing system, realization of the part model in a CAPP system seems to be a complex task.

Part modeling has become a key research issue since the introduction of CAPP. Generally, there exists three basic sets of data which completely describe the design content of the part [3]:

- **Geometrical data:** the geometric data give the basic description of the shape. For example diameter of a hole, depth of groove, width of a keyway, etc. constitute this type of data.
- **Technological data:** The information pertaining to tolerance and surface finish can be referred to as technological data, e.g., circularity, diametrical tolerance, etc.
- **General data:** Certain global characteristic that are applicable to the part as whole are often added to the to the design specifications. These global attributes include quantity to be produced, work material, design number, part name, functional specifications of the part and other task dependent details.

In the following current approaches on the generation of geometrical information of the part from the physical shape of the product are introduced.

1.5 Generation of Geometrical Details

There are two major methods for part modeling in the CAPP system development [4]; *CAD Models* and *Feature Based Models.*

1.5.1 CAD Models

Geometric shape of the part plays a major role in design and manufacturing functions. Generation of CAD/CAM systems can be seen as the logical outcome of this observation. Unfortunately, due to the following reasons geometric information stored in CAD data base is not structured to facilitate CAPP.

1) Low Level Data [4]:

   CAD-generated objects exist in terms of low level points, lines, arc and solids which are irrelevant to the manufacturing planning task. Therefore, the CAD data base needs a re-interpretation to extract manufacturing related knowledge from the part. This knowledge can be used by the process planning system and other downstream applications to proceed without the human intervention.

2) Non-Manufacturability [3]:

   It may happen that a part represented in a CAD system is not manufacturable. Hence, it is essential in to have a modeling system that supports model manufacturability check and geometric validation.

2) Lack of Design Intent [6]:

   Design intent is the intellectual arrangement of features and dimension of design. Design intent governs the relationship of the features in the part. Something that CAD cannot do is incorporate design into a model. They could display a design but the geometry does not hold design information beyond the actual lines and circles required for the construction of the object.

Hence, CAD models cannot be used directly without further processing for manufacturing applications like CAPP and this gap needs to be bridged to obtain coupling of CAD and CAM.

1.5.2 Feature-Based Models

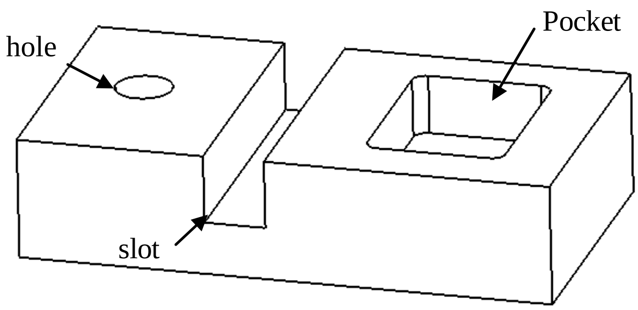
The mentioned limitations of CAD-generated model have led to the interest in using the concept of *form feature* (shape elements) for part modeling in CAPP.

Informally features are generic shapes or other characteristics with which engineers can associate knowledge useful for reasoning about the part [5]. Features represent a collection of low level entities which are packed in a meaningful form (like hole, slot, thread, groove, etc) and hence provide information at a higher conceptual level. In features, groups of geometrical entities are coupled with technological information needed for process planning functions to link between design and manufacturing.

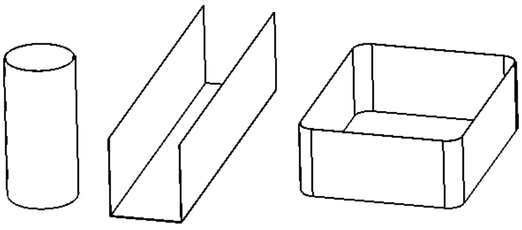
Features can be defined from different viewpoints, such as design, analysis, assembly, and function. Hence, there may be several co-existing feature models of the same product design [4]. In our research, the main viewpoint is manufacturing in which features represent shapes and technological attributes associated with manufacturing operations and tools.

A feature model is a data structure that represents a part in terms of its constituent features [34]. Figure 1-1a shows a feature model example. The part is represented in terms of a hole, slot, and pocket. These features can be used by CAPP to generate manufacturing instructions to fabricate the part. For example, CAPP typically generates a drilling operation for the hole feature.

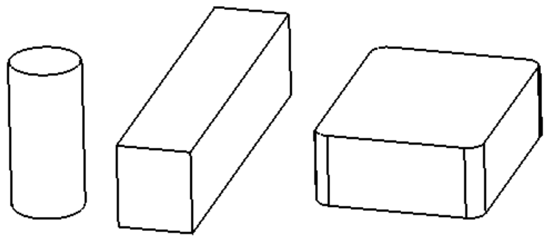
Manufacturing features may be represented both as surfaces and as volumes. Surface feature is a collection of faces of the model while volumetric feature represents the material to be removed by the rotation of cutting tool. Figure 1-1b and 1-1c shows both the surface and volumetric features of the part.



(a) part and features



(b) surface features



(c) volumetric features

Figure 1-1: Feature examples [34]

Volumetric features are necessary in automated process planning for relating a feature to the extent of material to be removed from a part, and for capturing the global characteristics of a part, such as tool accessibility [7]. It has become evident that volumetric features are more desirable not only for supporting feature creation and manipulation, but also for the reasoning activities in generative process planning.

1.6 Methods to Create Feature-Based Model

Methods to create a feature based model can be classified into two main categories [34]: *feature recognition* and *feature-based design*, as depicted in Figure 1-2.

1.6.1 Feature-Based Design Approach

In this approach, the part geometry is defined directly in terms of design features and geometric models are created from the features. This method is schematically shown in Figure 1-2.

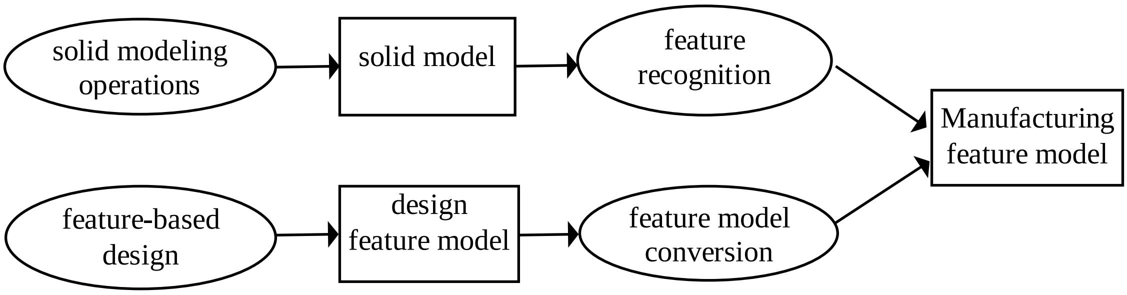
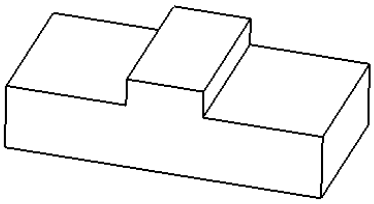


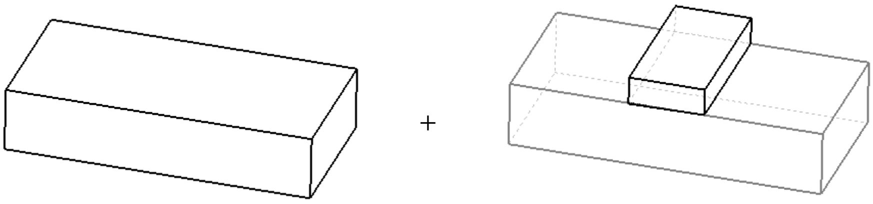
Figure 1-2: Feature model generation [34]

Unfortunately, design by feature method has several drawbacks. Firstly, there is a discrepancy between design feature model and machining feature model [4]. An example of this discrepancy is shown in Figure 1-3. In this example, the part is designed by adding one rib to the base block. However, from machining perspective, this part should be fabricated by removing the two steps from the enclosing block. Hence, feature based design systems need an additional step to convert the design features into machining features which is called *feature model conversion* as shown in Figure 1-2.

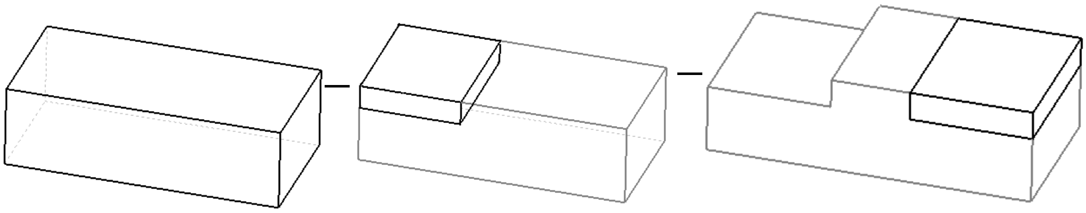
Another problem of design by feature approach is related to the existence of multiple feature models. One part can be interpreted in many number machining feature models especially when feature interaction occurs in the part. However, in the design by machining feature approach, the designer only describes the part in one set of features which may not be best for machining practice [34].



(a) part



(b) design feature model



(c) machining feature model

Figure 1-3: Difference between design features and manufacturing features [34]

1.6.2 Automated Feature Recognition Approach (AFR)

In this approach a geometric model is created first and then, a computer program processes the geometric information to discover and extract the features automatically [9]. Once the features are recognized, application oriented information can be added to the features for the completeness of the model. Compared to the previous approach in which the designer is limited to choosing the features from a predefined form feature library, in AFR the designer is allowed to use whatever geometric operations to create the CAD model and hence would be able to model complex parts.

Another advantage of AFR is that it assumes that all the features can be removed by milling and drilling operations and so it is not needed to recognize the specific type of the feature, other than its boundary corresponding to the final machining surfaces [8]. For example it does not matter whether a removal volume is a pocket or L shape slot since tool paths can be generated without knowing this distinction.

To sum up, compared to feature based design, the advantages of automated feature recognition are significant savings in time and human resource, as well as ensuring the desired part functionality without being limited in design creativity by the possibilities of the predefined form feature library [9].

Based on the discussion in the previous sections, we can draw a conclusion that AFR technique is an important tool for achieving a true integration of CAD/CAPP/CAM. Figure 1-4 schematically demonstrates the role of AFR in CAD/CAPP/CAM integration. As can be seen in the diagram, AFR could be considered as the primary but critical step in the transmission of CAD data into downstream applications. Without having a high performance AFR system success in the consequent steps are difficult to be achieved.

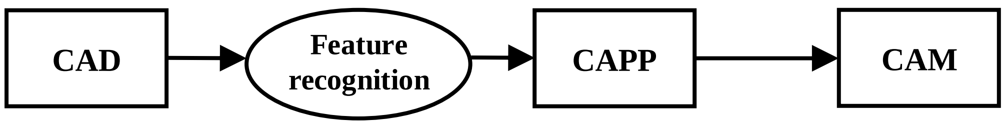


Figure 1-4: Diagram of AFR and CAD/CAPP/CAM Integration

1.7 Objectives

The main objective of this thesis is to develop a feasible feature recognition system for the integration of CAD and CAM. The input to the system is CAD models of the stock and the part and output would be a set of sequenced manufacturable volumetric features that could directly be used by CAM functions for NC part programming.

Generating a direct link between CAD and CAM does not mean that the role of process planning is eliminated. However, in the developed framework, tasks of feature recognition and CAPP are merged together to some extent.

1.8 Overview of the Thesis

This thesis contains 5 chapters. Chapter 1 gives the background of the problem studied in this thesis, as well as the motivation and objective of the research work.

Chapter 2 is a review of related work in feature recognition and its integration with CAPP system. Conclusions drawn from the review, which simulate the work of this thesis, are also given. Chapter 3 describes the main stages of developed system in detail. Various figures are used to visualize the steps for better understanding of the concepts. Chapter 4 presents system interface. Moreover, 3 case studies are used to validate the developed algorithm. Chapter 5 presents the conclusion on the results and contributions of the research work. The comments on future work are also given.

CHAPTER2

LITERATURE REVIEW

This chapter presents a summary of the previous research works related to the issues studied in this thesis. There is a large amount of literature on feature extraction. However, some of the previously developed methods have been replaced by newer techniques that have overcome their limitations. In this chapter, we will only focus on relatively successful techniques which are still being actively pursued.

2.1 AFR Technique Review

Generally, methods for automated feature extraction with rule-based pattern recognition consist of three phases: identification of structure in a part representation, formation of the feature, matching the feature with some predefined pattern using if-then rules. The main shortcoming of rule-based systems is a lack of a unique form feature library, which becomes a serious problem when an extracted feature cannot be matched with any form feature pattern that exists in the library and hence cannot be recognized.

There are various methods of rule-based pattern recognition. However, in the following only the most active approaches are reviewed and discussed. It is also necessary to mention that this survey is restricted to feature recognition techniques that can recognize features removable by three-axis milling machines.

2.2 Graph-based Approach

The graph based approach was firstly introduced by Joshi and Chang [12]. In this approach, the boundary model of the part is used to create an attributed face adjacency graph (AAG). Nodes of AAG represent faces and arcs of AAG represent edges of the model. Moreover, additional attributes such as edge-convexity are assigned to the corresponding arcs of the graph [11, 12].

To recognize the features of interest, firstly each form feature template is modeled using AAG to generate a graph pattern. Secondly, the AAG of the model is searched to match with the form features' AAG to recognize the features. In order to facilitate the searching, the following heuristic is used to simplify the AAG of model: *Face whose all boundary edges are convex does not form part of a feature and, therefore can be deleted from AAG.*

This approach is quite successful for non-intersecting depression type features where the feature AAG is found as a complete sub-graph in the part AAG [34]. However, this approach faces many difficulties when only portion of a feature AAG is present in the model due to feature intersection. Feature intersection is a crucial problem in AFR, and considerable effort has focused to address this issue.

Marefat and Kashyap [13] proposed a novel solution to deal with interactions. They define features by cavity graphs that extend a feature's AAG to include some geometric constraints on the orientations of the feature faces. To recognize interacting features, they firstly restore the missing arcs and add them into the part graph. Then, they generate all hypothesized features by sub-graph matching and non-valid hypotheses features are eliminated using rule-based reasoning. However, in this approach, it is not guaranteed to identify the exact set of missing links and if few unnecessary links are added to the graph, the features may not be recognized or some bogus features may be recognized. Trika and Kashyap [14] extended this approach by proposing an algorithm that can compute the exact set of missing arcs. However, in their algorithm both the part domain and feature classes are limited to polyhedral parts and seven basic machining feature classes. Moreover, single interpretation is extracted in their approach. The searching algorithm for restoring missing links is also very exhaustive.

Another problem concerning graph-based method is that the manufacturability of recognized features is not ensured. In graph based method , the extraction method is only based on the geometric shape of the model and manufacturing information that accounts for features accessibility, selection of cutting tools, etc., have not been taken into consideration.

Graph pattern analysis has also been criticized for computational complexity. The procedure of graph matching involves using sub-graph isomorphism algorithm which is a well known NP-hard problem. However, this criticism may be incorrect. Fast algorithm for recognizing cavity features were developed by Field and Anderson [15] for arbitrary shaped cavities that are common in machining applications and occurs often when features intersect. In their algorithm, edges are not only attributed by convex/concave but also exterior/interior classification. This classification facilitates the searching operator and reduces the computation complexity of the search to linear in the number of edges.

2.3 Hint-based Approach

Vandenbrande and Requicha [31] observed that looking for exact patterns of faces, edges and/or vertices is unsuitable for most of practical problems due to the existence of immense variety of feature interactions in the model. They proposed to use topological, geometrical and heuristic information about the part as the hints of presence of a certain features. Then the largest possible volume consistent with the hint is generated and tested for validity. Regli and Nau [32] proposed a similar methodology and named it trace-based approach. Later, Han and Requicha [33] improved the method by using different sources such as user input, tolerance attributes and design features for the generation of hints. In their developed system, instead of generating all the feature interpretations which is very exhaustive, a heuristic is used to generate one interpretations and the user can interact to generate alternative interpretations. The latest version of hint-based approach [35] aims to facilitate sequencing process in an overall CAPP system,a tool database is linked to the recognizer in order to generate only manufacturing features.

Many other researchers have contributed to enhance the method with completeness of class of features to be recognized, efficiency of algorithms, use of additional information as hints, and independence from a modeler applied for the part's design [36, 37, 38].

There are several limitations concerning the hint-based technique. Hints are unique to each feature class, so the recognition algorithm is dependent on the feature type or we can say that this approach is feature library dependent [9]. The other problems is that in hint-based approach the number of traces which imply the location of features is more than the number of good features to recognize and as a result large number of hints my not lead to the creation of valid machining features [34]. In addition, it may be inefficient to check all the traces for the existence of valid features. Finally, hint-based technique involves conducting considerable number of

Boolean operations which is costly for practical cases with large numbers of machining features.

2.4 Volume Decomposition Approach

Volume decomposition approach is based on decomposing the delta volume into a set of intermediate volumes and then combining the volumes in order to produce features. This approach can be divided into two classes: convex hull decomposition and cell-based decomposition.

2.4.1 Convex Hull Decomposition

Convex Hull approach was first implemented by Woo [16] after the seminal work of Kyprianou [17] and later was extended by Kim [18]. An envelope (convex hull) around a part is firstly determined. The difference in volume between the part and it convex hull is defined as an alternating sum of volumes (ASV). Kim [18] proposed a remedy for non-convergence, initiating remedial partitioning procedure –ASV with partitioning (ASVP) and, since then, his research group worked to successfully implement the method. More details on convex hull approach can be found in [19, 20].

Although convex hull decomposition approach is interesting from the computational geometry viewpoint, this technique has limited success in handling realistic parts. Current convex hull decomposition methods can only deal with polyhedral features and cylindrical features which interact with them along the principal directions, with constant-radius blending. However, most practical domains include curved parts with complex feature interactions. There are some other drawbacks too. One of them is that the convex hull decomposition is completely separated from the feature recognition and methods proposed in [18,19,20] are often incapable of producing recognizable features.

Dong and Vijayan [58] developed a similar technique in which features are extracted using an approach called “blank surface – concave edge”. In their system, first an overall volume (total volume that should be removed from blank stock) is produced and then concave edges of the part are used to partition the overall volume into intermediate volumes which will finally be matched to machining form features. The pattern matching process is based on if-then rules. Their technique is simple but not applicable for complex parts. However, the idea is interesting because their partitioning procedure is done based on machining perspective.

2.4.2 Cell-based Decomposition

In all cell-based decomposition approaches, the methodology includes four steps: (1) the overall removable volume (delta volume) is obtained by Boolean subtraction of the finished part from stock; (2) delta volume is decomposed into cells by using extended boundary faces as cutting surfaces; (3) cells are concatenated to get macro volumes that can be removed in a single tool path; (4) macro volumes are classified into machining features. Methods used for decomposing the delta volumes are: extension and intersecting all faces of the body to construct “minimal”, convex, solid volumes [21-25] or extension of those faces sharing concave edge using half spaces [26]. In all of these approaches, the faces of model should be analytical faces otherwise they cannot be extended. Another problem specific to the first approach is that generation of cells by extending all the faces of part is computationally expensive and may lead to the generation of void, redundant or invalid cells.

Two methods have been used for re-composition of cells: (a) a time consuming procedure to combine all adjacent cells until a convex volume is generated [21, 22, 24, 25]. This method is costly and may produce many identical feature sets. (b) selective combination using cell adjacency [26]. Compared to the previous one, this method is more efficient and it never produces redundant combinations.

For volume classification, some researchers have reverted to methods used in conventional boundary based methods, such as feature specific attributed graphs based on topology and geometry [25, 27]. Others have used volume classification based on tool approach directions/accessibility. A generalization of this is classification based on rotational and translational degrees of freedom that can be related to machining operations [26, 28].

The main problem specific to this approach is the global effect of local geometry [34]. Machining operation usually leave its traces on the localized area of the part. However, globally extending the faces associated with the localized feature trace may result in the generation of huge number of cells which is difficult to deal with. Woo [29] addressed this problem by enabling the faces to be extended only over the concave edges, reducing the computational complexity more than 10 times.

Although a large number of re-composition alternatives could be considered as an advantage for this method because it generates all possible process plans, the resulting combinatorial explosion is a major drawback. In the most recent research, Woo and Sakurai [30] present the development of an algorithm for scalability of complex parts in order to reduce computational exhaustion and improve applicability of cell-based approach.

2.5 Hybrid Approach

In the hybrid technique, researchers attempted to develop a feature recognition algorithm by combining some fundamental concepts of several basic techniques mentioned in previous sections.

Gao and Shah [39] proposed an approach that combines graph–based method with hint-based method. They have effectively addressed the problem of feature intersections for parts with planar and cylindrical faces. Moreover, Alternative feature interpretations can be generated by their hybrid approach. Nonetheless, its limitation to features with planar and cylindrical faces is a major shortcoming.

An example of combination of convex hull approach and graph-based approach is presented in [40]. The system can handle prismatic parts and recognize features from six basic tool access directions. Moreover, a limited class of free form features can be dealt with their algorithm. The major drawback of their system is the limitation regarding machining directions.

Subrahmanyam [41] made an attempt to combine hint technique with cell-based technique. He reduced the complexity of combinatorial problem of cell-based approach by removing all isolated features and using some heuristic–based method. Both problem of feature interactions and multiple feature interpretation are effectively addressed in his approach. In addition, manufacturability of recognized features is a major advantage of the system. However, this approach is limited to parts which can be machined with single set-up only.

Another hybrid method based on the combination of hint method and graph method is recently presented in [42]. To reduce the complexity while recognizing features, they proposed a method to remove fillets. Their system can recognize 2.5D, floorless or 3D features. The authors used several test parts from NIST design repository to prove the validity of their algorithm. However, like other hint-based technique their approach requires human intervention in the recognition stage.

2.6 AFR/CAPP Integration and Feature Sequencing

In order to effectively integrate feature recognition with process planning, firstly the manufacturability of recognized features should be guaranteed. Secondly, it is required to incorporate manufacturing resource knowledge into feature recognition. Moreover, if feature sequencing is done in early feature recognition stage, computational load of subsequent process planning system may be decreased significantly. However, in most of the reported approaches the reasoning is only based on the geometry of the part to be manufactured. In the following, few feature recognition approaches that made some attempts for the integration with CAPP/CAM are reviewed.

Corney, Clark and their associates [44, 45, 46] developed a feature recognition system known as FeatureFinder. The algorithm produces a set of manufacturing volumes, each of which represents the material to be removed by a manufacturing operation. In the first step, a tool approach direction is manually selected. Only one tool approach direction is considered at a time. Then a graph-based algorithm is employed to recognize the 2D profile of $2\frac{1}{2}$D feature volumes. Again user interaction is needed to select the suitable profile for feature volume generation. Once a valid profile is selected, the profile is swept along the access direction to generate the volume. The main advantage of their system is that the way they extract the features is useful in subsequent stages of process planning, such as sequencing the manufacturing operations. Their system has two major drawbacks. It requires human intervention for tool approach direction selection and validity check of 2D profiles. In addition, their system is limited to $2\frac{1}{2}$D features.

In [35] a hint-based approach is proposed which incorporates setup, machining and tool change costs into feature recognition procedure. The output of their algorithm is an optimal sequence of machining features. However, the proposed system is subject to combinatorial explosion.

Sakurai *et al.* [22] proposed some heuristics based on practical process planning to sequence the extracted maximal volumes for the machining operation. However, his sequencing method is only applicable to the simple parts and can not cover complex practical problems.

Kim *et al.* [47] proposed to use face dependency information for the generation of feature precedence relationships in the ASVP decomposition algorithm. Khoshnevis *et al.* [48] also presented a similar process planning system.

Manufacturability of features based on tool accessibly is investigated in series of research work conducted by Roberts and Henderson [49, 50, 51]. Along with this direction, Jurrens *et al.* [52] proposed a feature recognition system which can communicate with manufacturing resource library in order to select the available tools for the features. A feature recognition system that does process planning task is developed by Gaines and Hayes [53-55]. Their system is based on manufacturability and made adaptive to resources.

2.7 Summary

AFR is an important stage in transformation of CAD information into downstream applications. To eliminate the role of human in CAD/CAM integration, a fully automated CAPP system is required to be developed. However, despite of huge amount of efforts made in past 25 years, limited success is acieved in the area of feature recognition and the complete problem is far from being solved [9]. The main shortcoming of contemporary AFR systems are [10]: (1) complexity of the recognition algorithm, especially when feature interaction occur; (2) the domain of recognized features are limited-most of the current AFR systems mainly deal with orthogonal features; (3) the manufacturing information attached to the features is not rich enough to facilitate the subsequent process plan.

Our system attempts to overcome some of the limitations mentioned above. We developed a feature recognition framework with CAPP functionality in which manufacturable features are generated. In our system, problems of feature intersections and multiple feature interpretation is addressed from machining prospective.

CHAPTER 3

DESCRIPTION OF THE RECOGNITION METHODOLOGY

3.1 Introduction

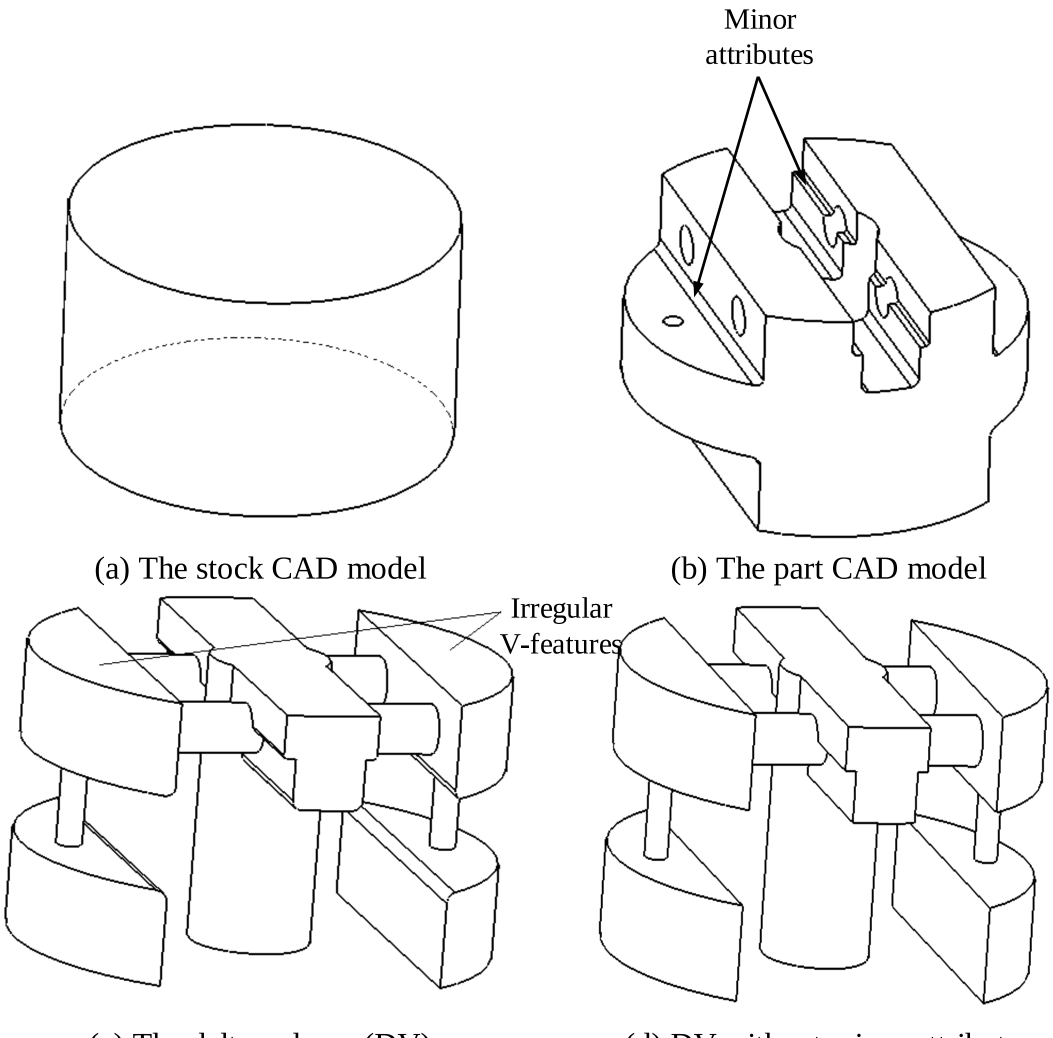
Currently, CAD and CAM systems are being widely adopted in parts manufacturing industry. Generally, CAD systems provide powerful means to design complex parts in three-dimension (3D) mode and the CAM systems take the 3D CAD model of a part and help to generate numerical control (NC) tool-paths and codes to produce it. However, the task of generating the tool-paths for a given CAD model of a part by using a commercially available CAM system is not trivial. Instead, the user may have to make the following decisions in this process:

(1) Identify the overall material removal volume, i.e., the delta volume (DV), which is the difference between the stock model and the part model (e.g., see Figure 3-1).

(2) Based on the available machines and cutters, decompose the DV into sub-DVs such that each sub-DV can be removed by a single machining process (e.g., end milling or drilling) along a feasible tool approach direction (TAD).

(3) Group the sub-DVs into different set-ups based on the same TAD and arrange the sub-DVs in the same set-up into a feasible machining sequence. Arrange the set-ups into a feasible sequence.

(4) For each sub-DV, select a machine and a cutter, and the CAM system can then be used to generate the corresponding tool-paths for removing the sub-DV.



(a) The stock CAD model (b) The part CAD model

(c) The delta volume (DV) (d) DV without minor attributes

Figure 3-1: An example of the stock, part, and the delta volume

The procedure described above is generally called the process planning process, which demands a substantial amount of expertise and experience. Over the last two decades, there has been much research effort, in the name of *computer-aided process planning* (CAPP), towards automating this procedure. However, in terms of real industrial application, limited success has been achieved. Apart from CAPP, there has been some specific effort towards automating steps (1) to (2) in the above procedure, namely *machining feature extraction*.

In the research literature, a number of definitions for the term *"feature"* exist depending upon the application domain. In the domain of CAPP, there are mainly two kinds of feature definitions. The first one is based on the part only, in which a feature is defined as a group of geometric entities that is meaningful to a particular machining process, e.g., a slot (vs. end-milling) and a hole (vs. drilling). The second one is based on the volumetric difference between the part and the stock (materials to be removed), in which a feature is defined as a volume that can be removed by a single machining process, e.g., a rectangular block (vs. end-milling) and a cylinder (vs. drilling). In the first definition, the materials to be removed are constructed from the final state of the feature, i.e., the stock is predetermined. While in the second definition, the stock can take any shape, from bulk materials to near-net shape materials such as casting and forging parts. Obviously, the second feature definition is more realistic in resolving the machining feature extraction problem. Therefore, in this paper, the second definition is adopted and the feature is named as *volumetric features* (V-features).

There are several challenges in extracting V-features from the DV. Firstly, the V-features in the DV are often intersected (see Figure 3-1c). Partitioning the DV into individual V-features must be based on machining practice such that the V-features can be removed one by one along the specified TADs and following the specified sequence. Moreover, there are often multiple choices when partitioning a DV. Optimization factors, e.g., high machining efficiency and/or low machining cost, also need to be taken into consideration. Secondly, some of the V-features may not be of regular shape. For example, the two blocks in Figure 3-1c can be treated as two rectangular blocks when generating tool-paths for an end-milling process. However, the boundaries of the two corresponding rectangular blocks must be specified. Therefore, in order to input the final V-features into the CAM system directly, those irregular shaped V-features must be converted to regular shaped V-features first. Thirdly, chamfers and round blended corners (so-called minor attributes) are often present in the parts (see Figure 3-1b). These minor attributes can be generated as when their parents V-features are removed. However, the dimensions of the minor attributes must be taken into consideration when selecting a cutter to remove the corresponding V-features.

Over the last two decades, there has been much research on feature extraction/recognition, but still complete problem is far from being resolved. While the approaches differ in their specific recognition processes, most employ general geometry-based operations to recognize diverse features. In specific, those approaches based on volume decomposition have shown that V-features can help achieve automated process planning for direct NC tool-path generation. However, an important issue, i.e., how to ensure the manufacturability of the V-features, is still not fully addressed.

In this research, a new feature extraction method based on delta volume decomposition is proposed, which focuses on extracting V-features with valid machining feasibility. The above mentioned challenges in feature extraction are effectively addressed. The resultant V-features can be directly used by the various

CAM functions available in most commercially available CAM system to generate tool-paths and NC codes. The V-features covered correspond to all the geometric features that can be created using the machining processes on a 3-axis machining centre.

3.2 Overview of the Proposed Approach

3.2.1 The volumetric features

Based on the geometric shape of the machined faces and the corresponding cutter, all the V-features can be categorized into two general types: the *drilling* V-feature and the *milling* V-feature. A drilling V-feature refers to a V-feature having a convex cylindrical machined face that can be created by drilling, profile-milling, reaming, and cylindrical grinding processes; and a milling V-feature refers to a V-feature having planar machined faces that can be created by end-milling, side-milling, and planar grinding processes. As a result, the *cylinder* type shown in Figure 3-2a is a drilling V-feature, the rest are milling V-features.

In terms of dimensionality, the milling V-features can be of $2\frac{1}{2}$D or 3D. A $2\frac{1}{2}$D milling V-feature is a volume that can be removed by continuous motion of the cutter along 1 or 2 axes only. A 3D milling V-feature, however, requires the cutter to move along x-, y-, and z-axes simultaneously. In this study, six regular shaped milling V-features are defined first (see the top images in Figure 3-2b-g, which are commonly encountered in 3-axis machining).

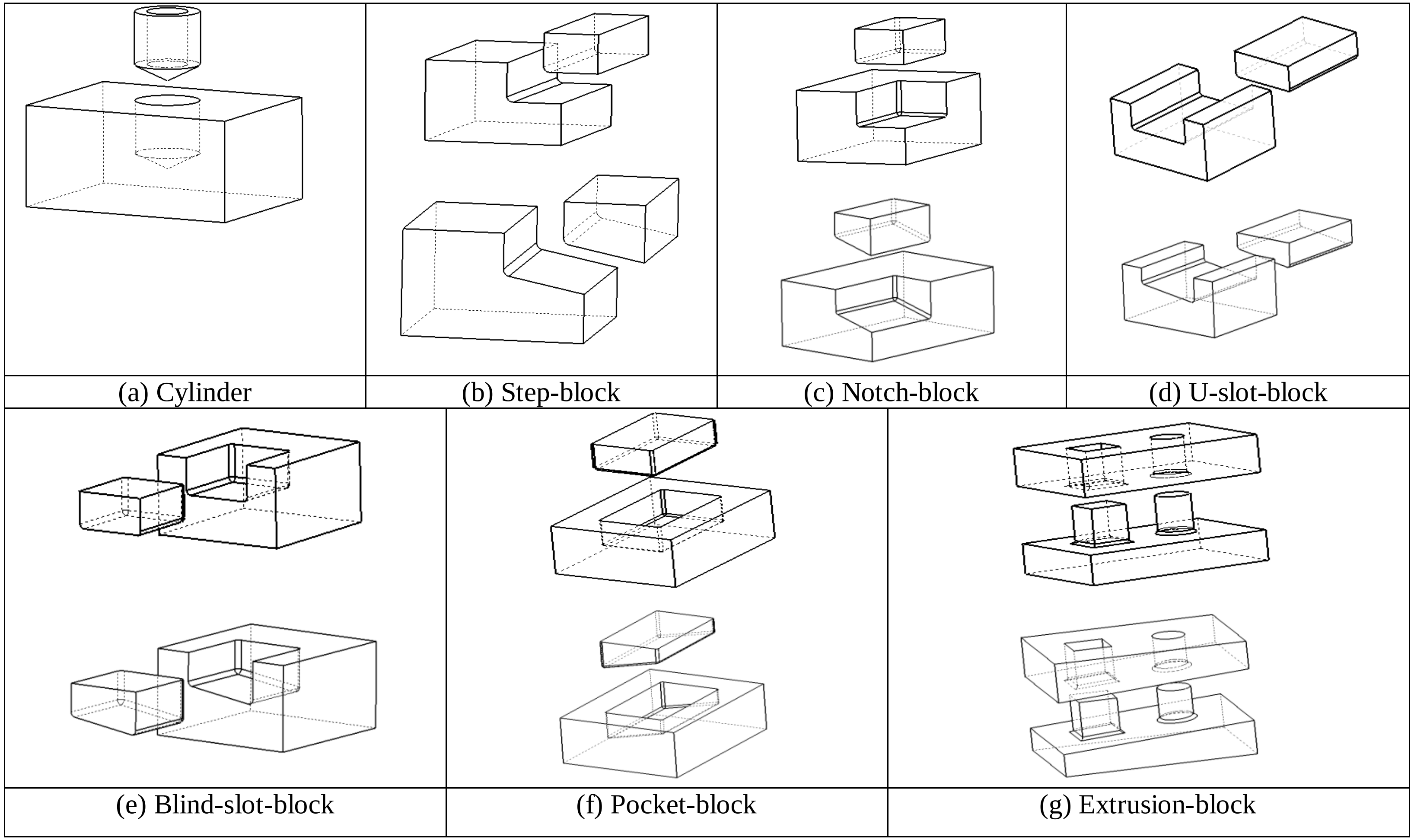


Figure 3-2: The V-features ($2\frac{1}{2}$ D and 3D) and their corresponding geometric features on the part

The images show both the V-features and their corresponding geometric features on the part. Each type of V-feature is defined by a specific data structure covering all the parameters. It is worth noting that the extrusion-bock shown in Figure 3-2g may have multiple holes of bosses or pads. In process planning, the type of a V-feature is the *major* attribute that determines the machining process to be used. On each V-feature, the *minor* attributes, such as blended corners, are also well defined. These minor attributes may not play any role in major process selection, but are critical factors for cutter selection. These $2\frac{1}{2}$D milling V-features will become 3D when some of the machined faces are of 3D (not planar or the planar machined faces are not orthogonal to each other) as shown by the bottom images in Figure 3-2b-g, which are also covered in this study.

3.2.2 The V-feature Extraction Procedure

The first step of our approach is to obtain the DV by Boolean subtraction of the part CAD model from the stock CAD model. The machined faces (MFs) on the DV are identified during which the minor attributes such as blended corners are also extracted. The pseudo codes for MF identification are illustrated in Algorithm 3-1. The minor attributes are then removed and replaced by a virtual edge such that the blended corners become sharp (see Figure 3-1d). The information of the minor attributes is linked to their virtual edges, which will be copied to their respective V-features later.

Algorithm 3-1: MF identification

Input: Volume (V), Part model (P);

Output: MFs list;

Steps:

a. Find faces of V and put them ino Vf_list;

b. Find faces of P and put them into Pf_list;

c. **For each**, face in Vf_list, **do**

  c.1. Get surface of the face, V_surface;

  c.2. **For each**, face in Pf_list , **do**

    c.2.1. Get surface of the face, P_surface;

    c.2.2. **If**, V_surface and P_surface are same, **then**

      c.2.2.1**.** **If**, edges of Vf_list face are same as edges of Pf_list face, **then**

        c.2.2.1.1. Put the Pf_list face ino MF_list;

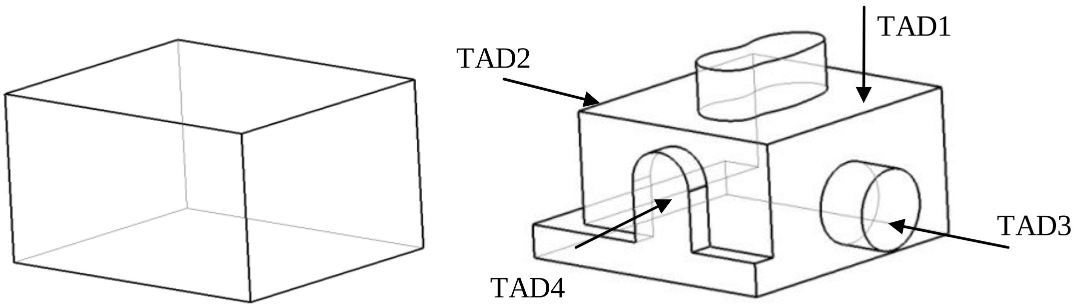
      c.2.2.2**.** **End if**

    c.2.3. **End if**

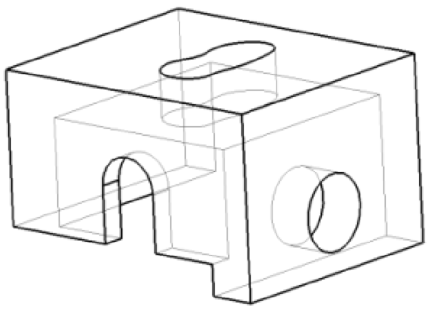
  c.3. **End for**

d. **End for**

In the second step, all the possible tool approach directions (TADs) for removing the DV are extracted. A TAD is an unobstructed direction along which a cutter can access and remove at least a portion of the DV. Apparently, the possible TADs are closely related to the MFs on the part model such that the MFs are in touch with the cutter's faces during the machining process. It was found that two kinds of MFs provide the clues for possible TADs: (1) a planar MF indicates a possible TAD along its normal vector (pointing to the material); (2) an internal cylindrical MF indicates two possible TADs along two directions of its axis (in case the cylindrical MF ends at a MF, the possible TAD that points away from the material is discarded). Following these two rules, the four possible TADs for the example shown in Figure 3-3 can be identified (see Figure 3-3b). It is worth noting that the possible TADs identified at this stage may be redundant or even invalid. They will be finally confirmed or rejected in the process of partitioning the DV into V-features. Algorithm 3-2 shows the detailed procedure for TAD list generation.



(a) The stock CAD model (b) The part with possible TADs



(c) The delta volume (DV)

Figure 3-3: An example for V-feature extraction

Algorithm 3-2: TAD list generation

Input: Part Model (P);

Output: TAD list;

Steps:

a. **For each,** face of P, **do**

  a.1. **If** , face is planar, **then**

    a.1.1. Create TAD, new_TAD opposite to the face normal;

    a.1.2. **If**, new_TAD is not in TAD_list , **then**

      a.1.2.1. Add new_TAD into TAD_list

    a.1.3. **End if**

  a.2. **End if**

  a.3. **If** , face is cylindrical , **then**

    a.3.1. Create TAD, new_TAD parallel to the axis of cylinder;

    a.3.2. **If** , new_TAD is not in TAD_list , **then**

      a.3.2.1. Add new_TAD into TAD_list;

    a.3.3. **End if**

    a.3.4. Create TAD, new_TAD opposite to the axis of cylinder;

    a.3.5. **If** , new_TAD is not in TAD_list, **then**

      a.3.5.1. Add new_TAD into TAD_list;

    a.3.6. **End if**

  a.4. **End if**

b. **End for**

In the third step, the DV is partitioned along the possible TADs, one at a time, into *accessible delta volumes* (ADVs). The ADVs along each TAD are then re-organized to form the final V-features. The procedure is as follows:

(1) Select a possible TAD.

(2) Applying partition operations to the DV along the TAD to obtain the ADVs, which is part of the DV that can be accessed in the selected TAD.

(3) Construct V-features by making use of the ADVs along the TAD.

(4) Update the DV by discarding the used ADV from the current DV.

The above procedure is repeated until the DV becomes empty. In step (3), there can be more than one way to construct V-features from the ADVs. To maximize the machining efficiency, we introduce the concept of *maximal V-feature*, which is, to a certain extent, similar to the one proposed in [23]. A *maximal V-feature* (maxV-feature) is a maximum portion of ADVs that can be removed by a single machining process with a 3-axis machining centre.

3.3 Generating ADVs from the DV

Given a possible TAD, there are 3 steps involved in the generation of the ADVs from the DV. Firstly, the MFs on the part model that are wholly or partially visible along the TAD are identified, which are called visible MFs. Secondly, the outline curve-segments of visible MFs are generated and used to decompose the DV into cells. Finally, the accessibility of each cell is checked and the accessible cells are the ADVs along this TAD. Pseudo codes for the identification of visible MFs are illustrated in Algorithm 3-3.

Algorithm 3-3: Visible MF identification

Input: Part model (P), TAD;

Output: Visible MFs list;

Steps:

a. Visible_MF_list=empty;

b. Use Algorithm 3-1 to get MF_list;

c. **For each,** MF in MF_list ,**do**

    c.1. Find all edges of MF;

    c.2. **For each**, edge, **do**

        c.2.1. Extrude the edge along -TAD to generate *Semi-infinite Surface*;

        c.2.2**.** **If,** *Semi-infinite Surface* is not wholly blocked by part model, **then**

            c.2.2.1. Add MF to the Visible_MF_list;

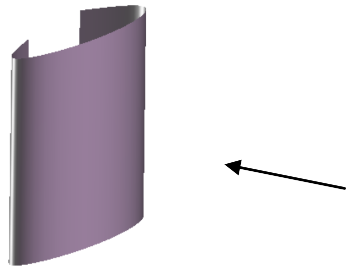
            c.2.2.2. Go to step b;

        c.2.3. **End if**

    c.3. **End for**

d. **End for**

The outline of a face is an important visibility feature of the face with respect to a viewing direction. It is the collection of curve-segments on the face that separate the front portion of the face from the back one. For a wholly visible face, the boundary curve-segments are effectively the outline curve-segments. For a partially visible face, however, the silhouette curve-segments need to be generated along the giving viewing direction.



(a) A face and a view direction

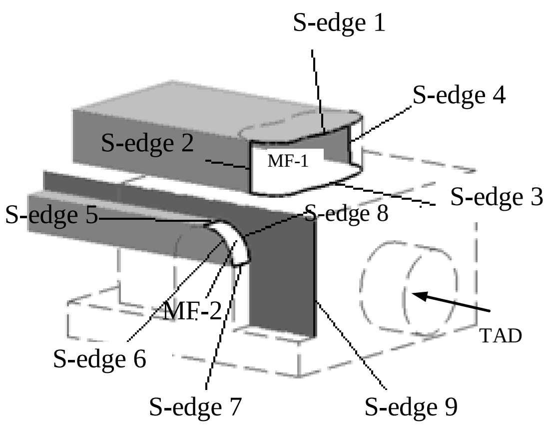
(b) Boundary of the face (c) Outline curve-segments

Figure 3-4: Outline curve-segments of a face along a viewing direction

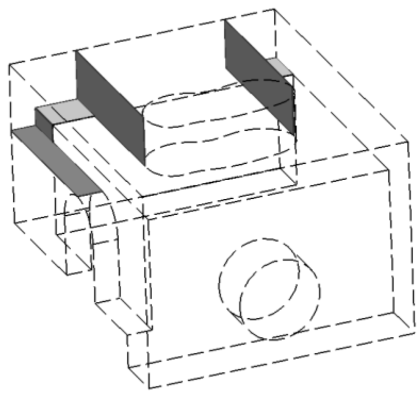
Figure 3-4a shows a partially visible face along a viewing direction and Figure 3-4b the boundary curve-segments. Figure 3-4c shows the 4 outline curve-segments in solid lines. From now onwards, the outline curve-segments of a MF along a given TAD are called silhouette edges (S-edges). In the following sections, detailed discussions are focused on how to decompose the DV into disjoint cells by using the S-edges along a possible TAD and how to check the accessibility of these disjoint cells.

3.3.1 Delta volume decomposition

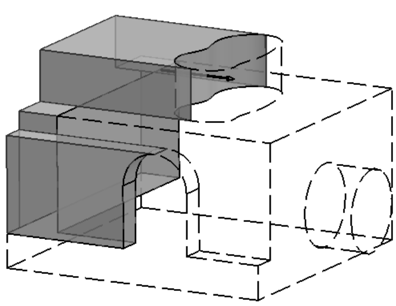
The example shown in Figure 3-3 is used here for illustration. For better clarity, only three visible MFs along the specified TAD, i.e., MF-1, MF-2, and MF-3, are used here as shown in Figure 3-5a. Firstly, each S-edge is swept along the TAD until the swept surface is obstructed by the part model or totally out of the part model. It can be seen that the swept surfaces of S-edges 1, 2, 4, 5, and 6 are created along the TAD without any obstruction, while the swept surfaces of S-edges 3 and 7 are obstructed by the part model from the beginning and fail to create. For S-edge 9, some portion of the swept surface is obstructed by the model whereas the remaining portion is created.



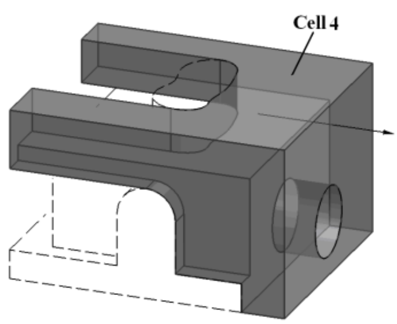
(a) The S-edges and with their swept surfaces



(b) The DV with intersection faces



(c) Inaccessible cells



(d) Accessible cells

Figure 3-5: An example of DV decomposition to obtain ADVs

Next, the swept surfaces obtained from the above procedure are checked to find their relationship with the DV. This is conducted by obtaining the *intersection faces* between the swept surfaces and the DV. If an intersection face lies on the MFs of the DV, it is discarded. The remaining intersection faces are added to the DV to create a new non-manifold body with internal faces (the intersection faces). For the example shown in Figure 3-5a, along the specified TAD, only the intersection surfaces related to S-edges 2, 4, 5 and 9 are located inside the DV. Figure 3-5b shows the final resultant non-manifold body, i.e., the DV and the intersection faces. This non-manifold body is called the non-manifold DV. Algorithm 3-4 illustrates the pseudo codes for the above procedure.

Algorithm 3-4: Generation of non-manifold DV

Input: Part model (P), Delta Volume (DV), TAD;

Output: Non-manifold DV;

Steps:

a. Use Algorithm 3-3 to get visible MFs of P;

b. **For each,** visible MFs of P ,**do**

  b.1. Find all S_edgs; (Parasolid Kernel provides a function which can be directly used in this step)

  b.2. **For each**, S-edge, **do**

    b.2.1. Sweep S-edge along TAD to generate a *Semi-infinite Surface*;

    b.2.2. Intersect *Semi-infinite Surface* with DV to get *Intersection-surface*.

    b.2.3. **If,** *Intersection-surface* lies on MFs of P, **then**

      b.2.3.1. Go to step b.2;

    b.2.4. **End if**

b.2.5. **Else**

b.2.5.1. Add *Intersection_surface* into DV as an internal face;

b.2.6. **End else**

b.3. **End for**

c. **End for**

In the final step, the non-manifold DV is decomposed into disjoint cells by extracting the manifold portions of DV. By the definition, the manifold portion of the DV is a volume for which the boundary faces are of the faces of the non-manifold DV and has no intersection faces inside. For the example shown in Figure 3-5, the DV is decomposed into 2 disjoint cells (see Figure 3-5c and d).

3.3.2 Identification of Accessible Cells

After the DV is partitioned into a set of cells, the accessibility of every cell along the specified TAD needs to be checked. A cell is accessible if there is a clear path for a cutter to approach the cell without any interference with the part model. A simple accessibility checking algorithm is developed based on ray casting analysis. For a given cell, a ray is firstly fired from any point inside the cell in the direction opposite to the specified TAD. If the ray hits the part model, the cell is inaccessible. Otherwise, the cell is accessible and called an *accessible delta volume* (ADV). For the example shown in Figure 3-5, Cell 1 is found inaccessible and Cell 2 is accessible and therefore the resultant ADV along the specified TAD. The detailed procedure of ADV generation is illustrated in Algorithm 3-5.

So far in this section, the procedure to generate the ADV from the DV along a single possible TAD is described. This procedure can be applied to the DV along every possible TADs. The result is a collection of ADVs along all the possible TADs, i.e., $(\mathrm{ADV}_i, \mathrm{TAD}_i), i = 1, 2, ..., n$, where n is the total number of possible TADs. The ADVs along different possible TADs may be overlapping, the following relationship should hold: $\bigcup_{i=1}^{n} \mathrm{ADV}_i = \mathrm{DV}$.

For all the ADVs along their respective TADs, a checking algorithm is applied to eliminate the redundant ones. This can be conducted by comparing a pair of ADVs: ADV_i and ADV_j. If ADV_i is totally contained inside ADV_j, ADV_i is removed as well as its respective TAD.

Algorithm 3-5: ADVs generation

Input: Part model (P) , non_manifold(DV);

Output: ADV list;

Steps:

a. Cell_list=empty; ADV_list=empty;

b. Extract manifold portions of non_manifold DV and put them into Cell_List;

   (Parasolid Kernel provides a function which can be directly used in this step)

c. **For each**, cell in Cell_list, **do**

   c.1. Find an arbitrary point inside the Cell’s volume;

   c.2. Sweep the point along –TAD to generate a volume called *wire_body*;

   c.3. Check if *wire_body* intersects with P(clash check);

   c.4. **If** ,clash does not exist, **then**

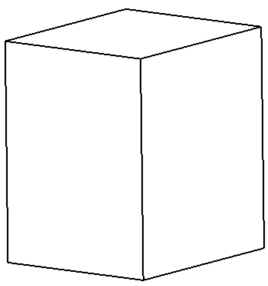
      c.4.1. Add the cell into the ADV_list;

   c.5. **End if**

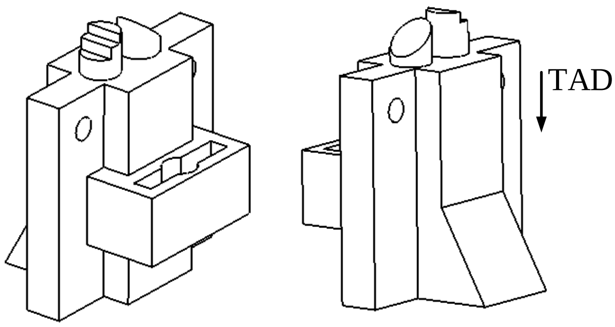
d. **End for**

3.4 Extraction of V-features from ADVs

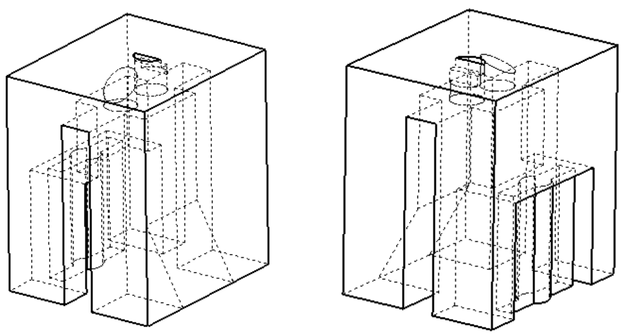
Given an ADV with its associated TAD, we have developed a feature extraction procedure that follows the natural machining practice, i.e., removing the materials from shallow to deep along the TAD. The idea is to section the ADV starting from the top by using a set of planes generated from the machined edges on the ADV perpendicular to the TAD, called *horizontal splitting planes* (HS-planes). By slicing the ADV using the HS-planes, a set of sub-ADVs are obtained, which are further partitioned into drilling and milling V-features, including $2\frac{1}{2}$D and 3D V-features. In the following sections, the details of this method are illustrated by following the example shown in Figure 3-6. It can be seen that there are both $2\frac{1}{2}$D and 3D V-features on the DV. The V-features are also heavily interacted presenting a good challenge to feature extraction.



(a) The stock model



(b) The part model from 2 views



(c) The ADV from the TAD

Figure 3-6: An example for V-feature extraction

3.4.1 Partitioning ADV into Sub-ADVs

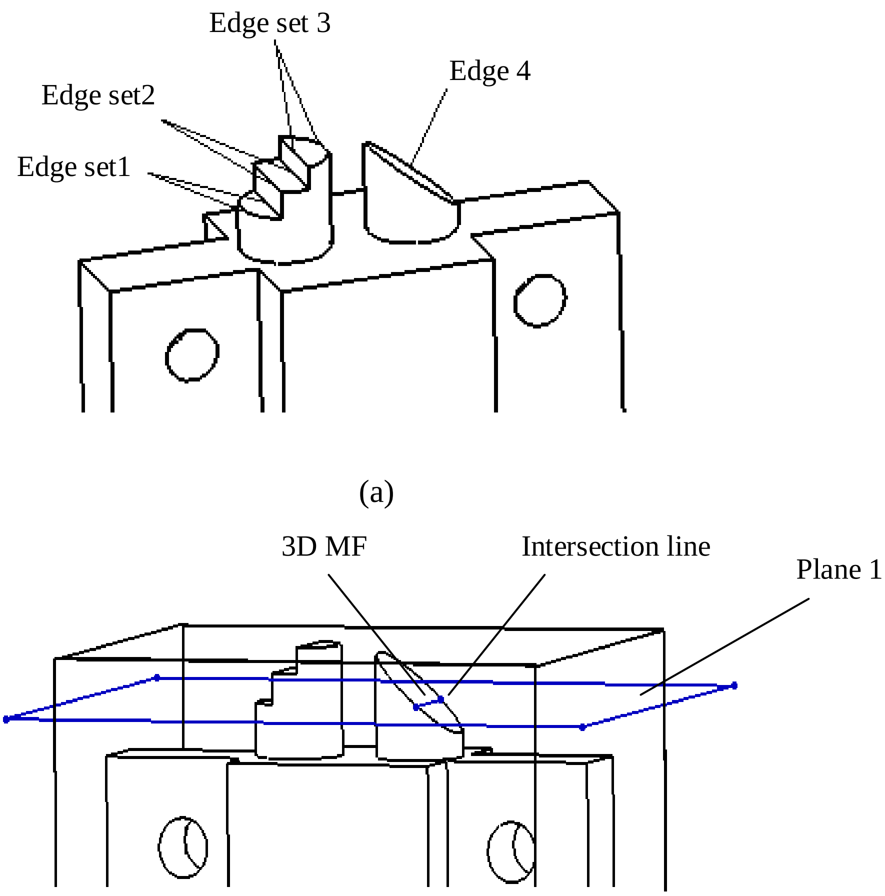
On an ADV, an edge is a *horizontal splitting edge* (HS-edge) for constructing a HS-plane if it satisfies the following conditions:

(1) It is a machined edge and planar.

(2) The plane containing the edge is perpendicular to the TAD. For example, Edge-4 in Figure 3-7a is not a HS-edge.

(3) It is not on the stock model. For example, Edge set-3 (see Figure 3-7a) is flush with the top plane of the stock model. Therefore, it is not considered a HS-edge.

(4) The HS-plane, generated by extruding the edge horizontally (perpendicular to the TAD), must not intersect with any 3D MFs on the ADV



(b)

Figure 3-7: Identification of HS-edges

Conditions (1)-(3) are geometric constraints. Condition (4) is more related to machining quality concerns. A 3D MF indicates the existence of a 3D V-feature. However, the creation of HS-plane follows a $2\frac{1}{2}$D milling approach. If the HS-plane intersects with a 3D MF, the end milling cutter may leave traces on the MF which is not acceptable if good surface quality is required.

In our approach, 3D V-features are to be extracted separately that can be removed by using 3D milling means. For example, Plane-1 (see Figure 3-7b), generated by extruding Edge set-1 (see Figure 3-7a) horizontally, intersects with the 3D MF (see Figure 3-7b). Therefore, Edge set-1 is not considered as a HS-edge. The same scenario also happens to Edge set-2.

Once all the HS-edges are identified, the shallowest HS-edge along the TAD is selected (see Algorithm 3-6) and the corresponding HS-plane is generated, which is then used to section the ADV. This results in several disjoint volumes, each being placed either above or below the HS-plane. The one that is above the HS-plane is named a sub-ADV. The volumes underneath the HS-plane are further partitioned by the deeper HS-planes, one at a time. The final result is a set of sub-ADVs. Figure 3- 8 provides an illustration of this partition process as follows:

(1) HS-plane 1 (the shallowest) splits the ADV into 3 disjoint volumes with sub-ADV 1 on top of HS-plane 1 (see Figure 3-8a).

(2) For the remaining 2 sub-volumes underneath HS-plane 1, HS-planes 2 and 3 are generated respectively (see Figure 3-8b). These two planes section the 2 volumes into 6 disjoint volumes with sub-ADV 2 on top of HS-plane 2 and sub-ADV 3 on top of HS-plane 3, respectively.

(3) Finally, since there are no more HS-planes in the deeper level, the remaining sub-volumes form the final set of sub-ADVs (see Figure 3-8c).

Algorithm 3-6: Identification of shallowest HS-edge

Input: List of HS-edges, TAD;

Output: Shallowest HS-edge;

Steps:

a. Choose an arbitrary point P in space;

   Minimum=1000;

b. **For each**, HS-edge, **do**

   b.1. Generate HS-plane;(HS-plane's normal vector should be opposite to TAD)

   b.2. Get origin point (O) and normal vector (N) of the HS-plane;

   b.3. Generate a vector directing from point O to Point P, OP;

   b.4. Dotproduct N and OP, dot= OP.N;

   b.5. **If**, dot<Minimum, **then**

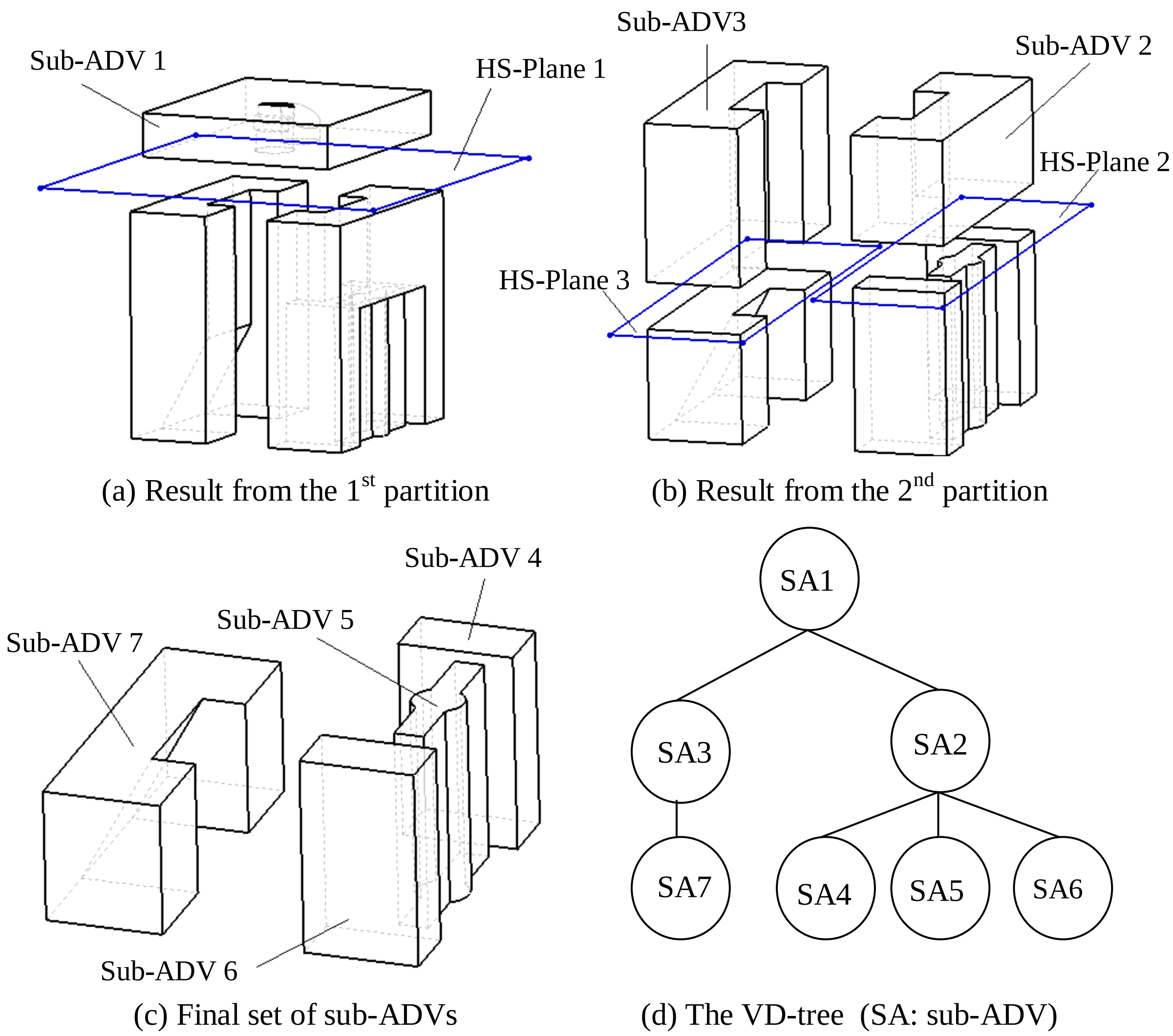
      b.5.1. HS-edge is shallowest edge;

      b.5.2. Minimum=dot;

   b.6**. End if**

c. End for

The resulted sub-ADVs and their spatial location are stored in a graph called the Volume Dependency Tree (VD-tree). The sub-ADVs from the first partition are stored as the top nodes. The sub-volumes resulted from the subsequent partitions form the remaining nodes in the corresponding levels (see Figure 3-8d). Algorithm for ADV partitioning s illustrated in Algorithm 3-7.



(a) Result from the 1st partition

(b) Result from the 2nd partition

(c) Final set of sub-ADVs

(d) The VD-tree (SA: sub-ADV)

Figure 3-8: The ADV partitioning process

Algorithm 3-7: ADV partitioning

Input: ADV, TAD;

Output: Sub_ADVs lisy

Steps:

a. Sub_ADVs_list = empty;

   Current_volumes = ADV;

   Temp_volumes = empty;

   Level_number=1;

b. **While**, Current_Volumes is non-empty, **do**

   b.1. **For each**, volume in Current_volumes, **do**

      b.1.1. Use Algorithm 3-6 to identify shallowest HS-edge;

      b.1.2. Cut the volume by shallowest HS-edge to generate top sub_volume and buttom sub_volumes;

      b.1.3. Add top sub-volume into Sub_ADV_list at Level_number;

      b.1.4. Add buttom sub-volumes into Temp_volumes;

   b.2. **End for**

   b.3. Current_Volumes=Temp_volumes;

      Temp_Volumes=empty;

   b.4. Level_number=Level_number+1;

c. End while

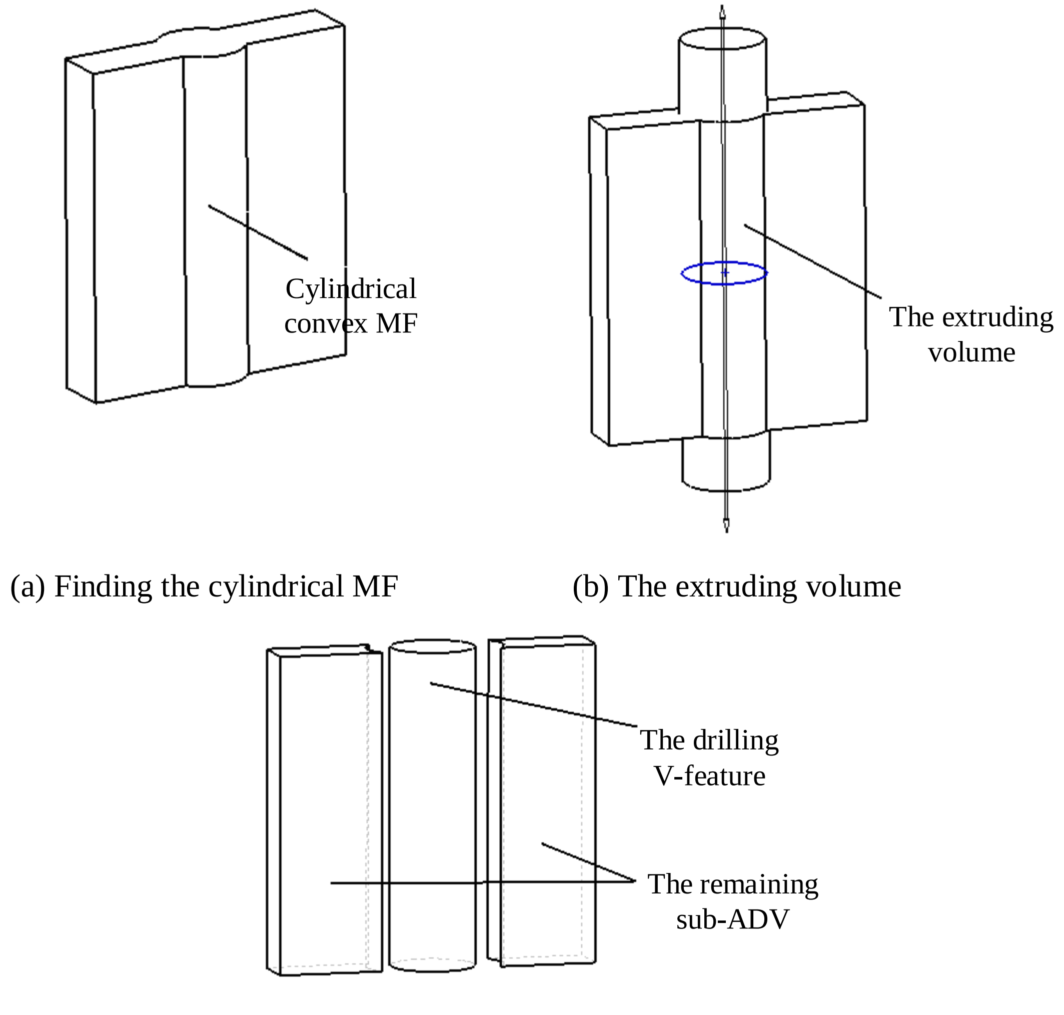
3.4.2 Extracting V-features from Sub-ADVs

Compared with the parent ADV, the sub-ADVs are much simpler. However, a sub-ADV may still not be removable by a single process due to two feature intersection scenarios: (1) intersection between milling and drilling V-features (e.g., sub-ADV5 in Figure 3-8c) and (2) intersection between $2\frac{1}{2}$D and 3D V-features (e.g., sub-ADV1 in Figure 3-8a). A 3-phase algorithm has therefore been developed that resolves the two intersection problems, respectively, before converting the simple sub-ADVs into the corresponding V-features.

(A) Further partition of sub-ADVs containing drilling V-features

From the sub-ADVs, the drilling V-features can be found by identifying the convex cylindrical MFs with their axes parallel to the TAD. By finding such a MF, one can determine both the location and radius of its circular profile. Subsequently, the circular profile is swept along both axis directions to generate a cylindrical extruding volume. The drilling V-feature to this drilling volume can be constructed by intersecting this extruding volume with the sub-ADV. The remaining of this sub-ADV is to be further checked for 3D V-feature interaction. As for the machining sequence, the drilling V-feature is to be removed before the remaining V-features. The detailed procedure is illustrated in Algorithm 3-8.

For the sub-ADVs shown in Figure 3-8, only sub-ADV5 is found to have a convex cylindrical MF as shown in Figure 3-9a. The circular base profile and its extruding volume are shown in Figure 3-9b. Figure 3-9c shows the drilling V-feature and the updated sub-ADVs after subtracting the drilling V-feature. In this case, the remaining sub-ADV forms a single V-feature.



(a) Finding the cylindrical MF (b) The extruding volume

(c) The drilling V-feature and sub-ADVs

Figure 3-9: An example for resolving $2\frac{1}{2}$ D and 3D V-feature intersections

Algorithm 3-8: Extraction of drilling V-features

Input: sub_ADV, TAD

Output: Drilling V_features;

Steps:

a. Use Algorithm 3-1 to identify all MFs of sub_ADV.

b. **For each**, MF of sub_ADV , **do**

b.1. **If** , MF is cylindrical face and convex, **then**

b.1.1. **If**, axis of MF is parallel to TAD, **then** b.1.1.1. Determine center point and radius of MF.

b.1.1.2. Create a circle with the data obtained in b.1.1.2, *circular profile*;

b.1.1.3. Extrude the *circular profile* along TAD and –TAD to generate a volume called *Extrusion_volume*;

b.1.1.4. Intersect *Extrusion_Volume* with sub_ADV to generate drilling V-feature;

b.1.1.5. Update the sub_ADV by subtracting the drilling V-features;

b.1.2. **End if**

b.2 **End if**

c. **End For**

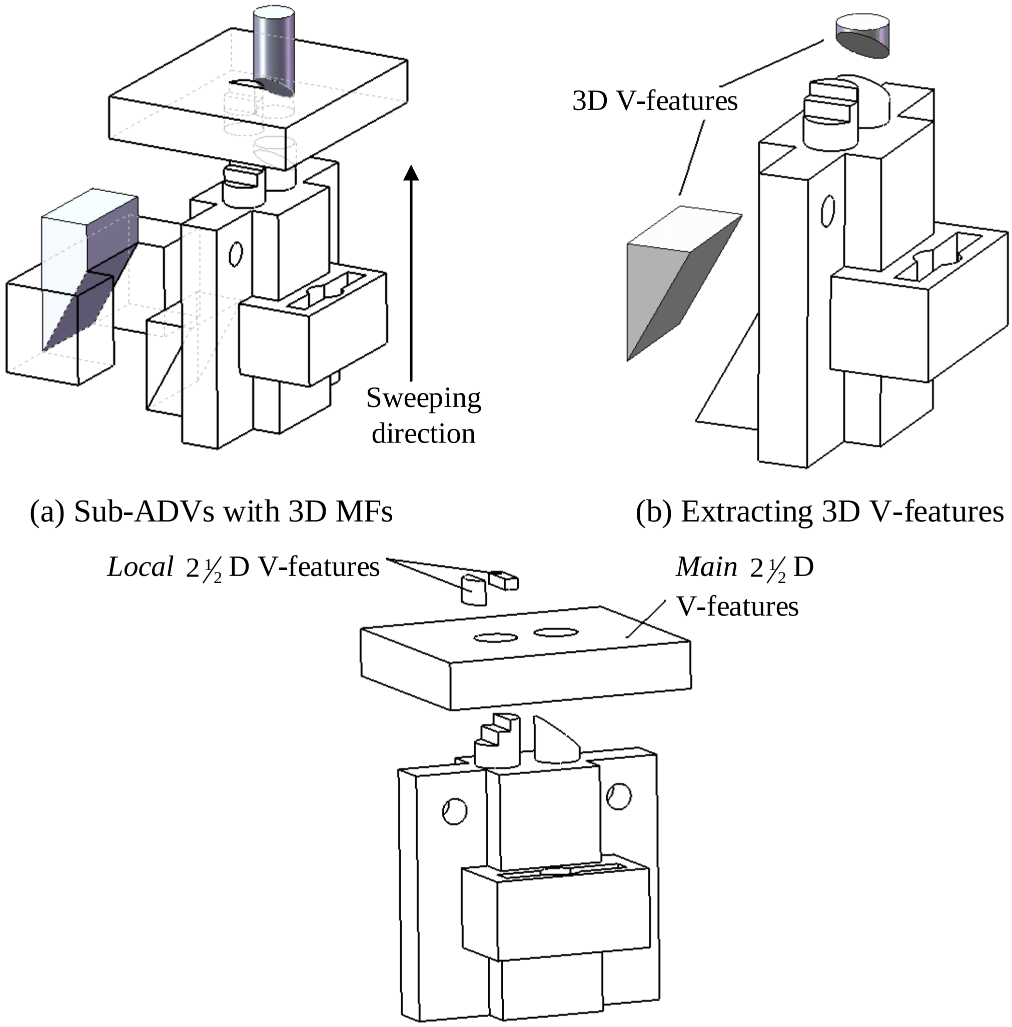
(B) Further partition of sub-ADVs containing 3D V-features

A 3D V-feature has a 3D MF with respect to a specified TAD. This property can be used to identify the sub-ADV having $2\frac{1}{2}$ D and 3D V-feature intersections. To resolve this kind of interaction, the following algorithm is applied:

(1) In this step, the 3D V-features are extracted. Firstly, the 3D MF is extruded along the −TAD to generate a corresponding swept volume. Subsequently, the 3D V-feature is created by intersection this swept volume with the sub-ADV. This process is illustrated by the example shown in Figure 3-10. For the sub-ADVs shown in Figure 3-8, sub-ADV1 and sub-ADV7 have 3D MFs as shown in Figure 3-10a. As a result, two 3D V-features are created and extracted from these two sub-ADVs (see Figure 3-10b). The sub-ADVs are

then updated by subtracting the created 3D V-features from the original sub-ADVs.

(2) For the updated sub-ADV, $2\frac{1}{2}$D milling V-feature interaction may exist since some machined edges in the sub-ADV may not be used to form HS-planes due to the presence of the 3D V-features (discussed in section 4.1). Therefore, we firstly find the MFs in the updated sub-ADV that satisfy the following two conditions: (1) perpendicular to the TAD and (2) not used to generate any HS-plane. Such MFs are then swept along the −TAD to generate the corresponding swept volumes. Subsequently, these swept volumes are intersected with the sub-ADV to generate the simple sub-ADVs, called the *local* $2\frac{1}{2}$D milling V-features. The remaining sub-ADV forms a simple sub-ADV, called the *main* $2\frac{1}{2}$D milling V-feature. In case, there is no $2\frac{1}{2}$D milling V-feature interaction, the remaining sub-ADV forms the *main* $2\frac{1}{2}$D milling V-feature. At the end of this step, the sub-ADV containing 3D-features is partitioned into a set of 3D V-features, a *main* $2\frac{1}{2}$D milling V-feature, and maybe a set of *local* $2\frac{1}{2}$D milling V-features. As for the machining sequence, the *main* $2\frac{1}{2}$D milling V-feature is to be removed first. There is no rigid precedence constraint among the removal of the 3D V-features and the *local* $2\frac{1}{2}$D V-features. For the example in Figure 3-10, the remaining of the top sub-ADV after 3D feature extraction is further partitioned into 1 *main* $2\frac{1}{2}$D milling V-feature and 2 *local* $2\frac{1}{2}$D V-features as shown in Figure 3-10c. Pseudo codes for the above procedure are shown in Algorithm 3-9.



(a) Sub-ADVs with 3D MFs

(b) Extracting 3D V-features

(c) Extracting 2½ D V-features

Figure 3-10: An example for resolving 2½ D and 3D V-feature intersection

Algorithm 3-9: Extraction of milling V-features

Input: sub_ADV, TAD

Output: Main 2½ D V-feature, Local 2½ D V-features, 3D V-features;

Steps:

a. 3D_V_feature_list= empty; Local_V_feature_list=empty;

   Main_V_feature_list=empty;

b. Use Algorithm 3-1 to identify all MFs of sub_ADV;

c. 3D_MF_ list=empty;

d. **For each**, MF of sub_ADV, **do**

c.1. **If** , MF is planar with normal non-parallel to TAD, **then**

c.1.1. Put the MF into 3D_MF_list;

c.2. **End if**

e. **End for**

f. **For each**, face in 3D_MF_list, **do**

f.1. Extrude the face along –TAD to generate *Extrusion_volume*;

f.2. Intersect *Extrusion_volume* with sub_ADV and add it to 3D_V_feature_list;

f.3. Update sub_ADV by subtracting the 3D_V_feature;

g. **End for**

h. CheckValue=true;

i. **While**, CheckValue is true, **do**

i.1. Use algorithm 3-10 to identify shallowest planar MF.

i.2. **If** , shallowest planar MF is not found, **then**

i.2.1. CheckValue=false;

i.2.2. Jump to step k;

i.3. **End if**.

i.4. Extrude the MF along –TAD to generate *Extrusion_volume*;

i.5. Intersect *Extrusion_volume* with sub_ADV and add it to Local_V_feature_list;

i.6. Update the sub_ADV by subtracting the Local V_feature;

j. **End While**

k. Put the Updated sub-ADV into Main_V_feature_list;

Algorithm 3-10: Shallowest planar face identification

Input: Part model(P), Volume(V), TAD;

Output: Shallowest planar face of V;

Steps:

a. Candidate_MF_list=empty;

b. Use Algorithm 3-1 to get MFs of V;

c. **For each**, MF of V, **do**

  c.1. **If**, MF is planar with normal opposite to TAD, **then**

    c.1.1. Create an infinite plane containing planar MF, *Infinite_plane*;

    c.1.2. Cut the V by using *Infinite_plane*;

    c.1.3. **If** , cutting process separates V into at least two components, **then**

      c.1.4.1. Add the MF into Candidate_MF_List;

    c.1.5. **End if**

  c.2. **End if**

d. **End for**

e. Select an arbitrary point in space, P;

Minimum=1000;

f. **For each** , face in Candidate_MF_List , **do**

  f.1. Get origin point(O) and normal vector(N) of face;

  f.2. Generate a vector directing from point (O) to point (P), OP;

  f.3. Dotproduct N and OP, dot=OP.N;

  f.4. **If** , dot < Minimum, **then**

f.4.1. Face is shallowest planar face;

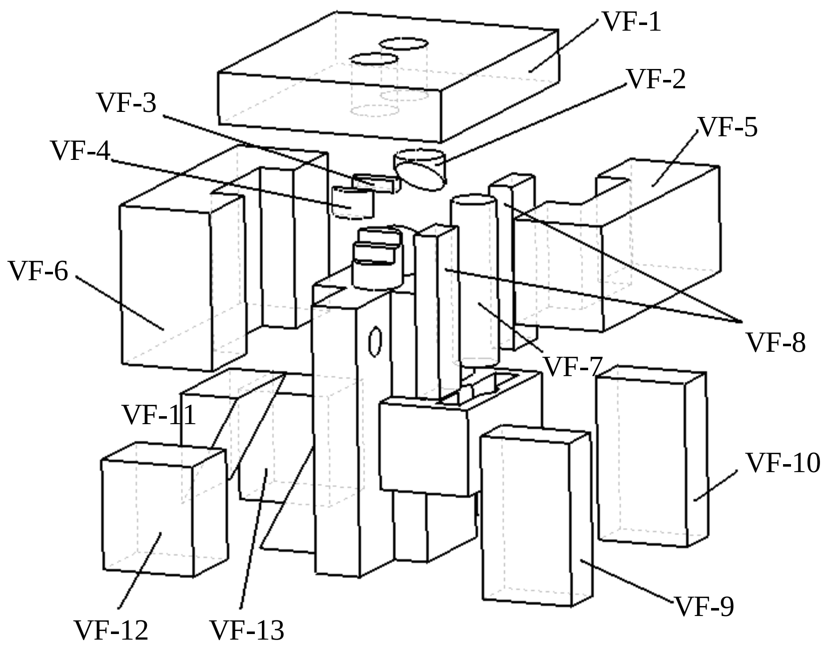
f.4.2. Minimum=dot;

f.5. **End if**

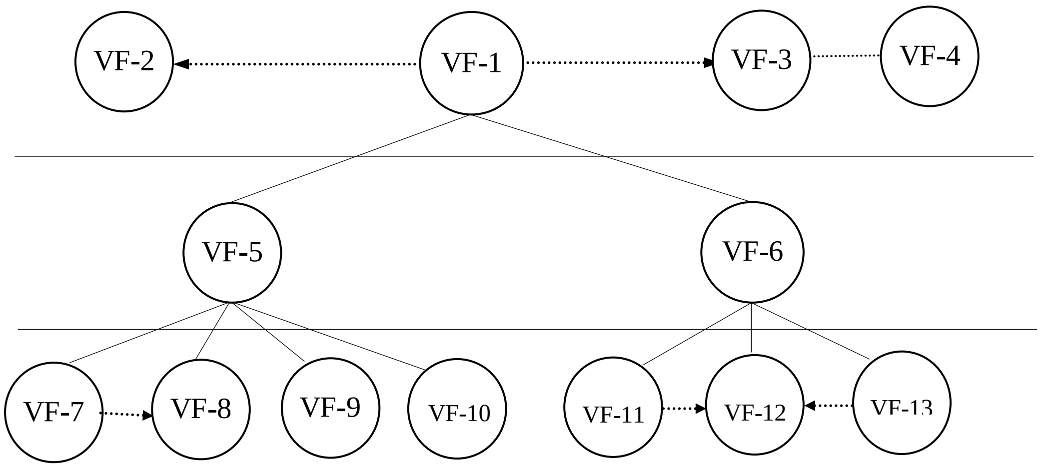
g. **End for**

(C) Converting simple sub-ADVs into V-features

After extracting the drilling and 3D V-features from the sub-ADVs through steps (a) and (b), the remaining sub-ADVs are all $2\frac{1}{2}$D milling V-features. The type of a $2\frac{1}{2}$D milling V-feature can be found by mapping the pattern of MFs to that of those shown in Figure 3-2. The dimensional information of the $2\frac{1}{2}$D milling V-feature can also be extracted. As a result, For the ADV shown in Figure 3-6c, a total of 13 V-features have been extracted as shown in Figure 3-11a. VF-2 and VF-11 are 3D V-features, VF-7 is a drilling V-feature. The rest are $2\frac{1}{2}$D milling V-features. Furthermore, the minor attributes, removed earlier, are attached back to their respective V-features by using the virtual edges as the links. At the same time, the VD-tree is expended to a V-feature dependency tree (VFD-tree) in which every node represents a V-feature. Furthermore, the nodes are placed in different hierarchical levels in which the nodes at the higher level are to be removed before those in the lower level. For nodes of the same level, a dashed arrow indicates the precedence of removal. As an example, the VFD-tree for the V-features in Figure 3-11a is shown in Figure 3-11b. There are totally 3 levels of nodes. The dashed arrows indicate that VF-1 needs to be removed before VF-2, VF-3, and VF-4; VF-7 before VF-8; VF12 and VF-13 before VF-11.



(a) The Exploded view of V-feature



(b) The final VFD-tree

Figure 3-11: The final set of V-features and VFD-tree

3.5 Multiple Feature Interpretation (Machining Sequence Generation)

The algorithm described in section 4 can be applied to all the ADVs along their respective TADs. As a result, a set of $2\frac{1}{2}$ D V-features (milling and drilling) and 3D V-features are extracted along each possible TAD. To determine the final machining sequence among the different TADs and the V-features along each TAD, a multiple interpretation problem arises. Moreover, some V-features along different TADs may be overlapping. To resolve these problems, two factors, i.e., accessibility and manufacturability are considered in the generation of the final TADs and their respective V-features as well as the sequence among them.

The algorithm starts from a set of possible TADs (TAD pool) and their corresponding ADVs (ADV pool). Firstly, a TAD is arbitrarily selected from the TAD pool and the partitioning algorithm described in section 4 is applied to its ADV. For each extracted V-feature, the information on available machines and cutters is checked to see whether the V-feature is removable. If the V-feature is not removable, it is eliminated. By doing so, the manufacturability of the confirmed V-features is guaranteed. Next, the confirmed V-features are removed one by one from the top node to the leave nodes in accordance to the VFD-tree structure. It should be noted that based on the VFD-tree's structure, a child node cannot be removed before removing the parent node. Up to this stage the machining sequence of all the V-features along this TAD is determined. This procedure is then applied to the remaining TADs, one by one, until the ADV pool becomes empty. The above procedure is illustrated in Algorithm 3-11 in form of pseudo codes.

Algorithm 3-11: Overall feature extraction algorithm

Input: Part model (P), Stock model (S);

Output: Drilling V-features, milling V-features;

Steps:

a. Subtract P from S to obtain the delta volume (DV).

b. CheckValue=true;

c. **While**, CheckValue is true, **do**

  c.1. Use Algorithm 3-2 to generate TAD list;

  c.2. Select a TAD;

  c.3. Use Algorithm 3-4 and Algorithm 3-5 to Convert DV into ADV.

  c.4. Use Algorithm 3-7 to partition ADV;

  c.5. Use Algorithm 3-8 to extract drilling V-feature;

  c.6. Use Algorithm 3-9 to extract milling V-features;

  c.7. Update DV by subtracting the drilling and milling V-features;

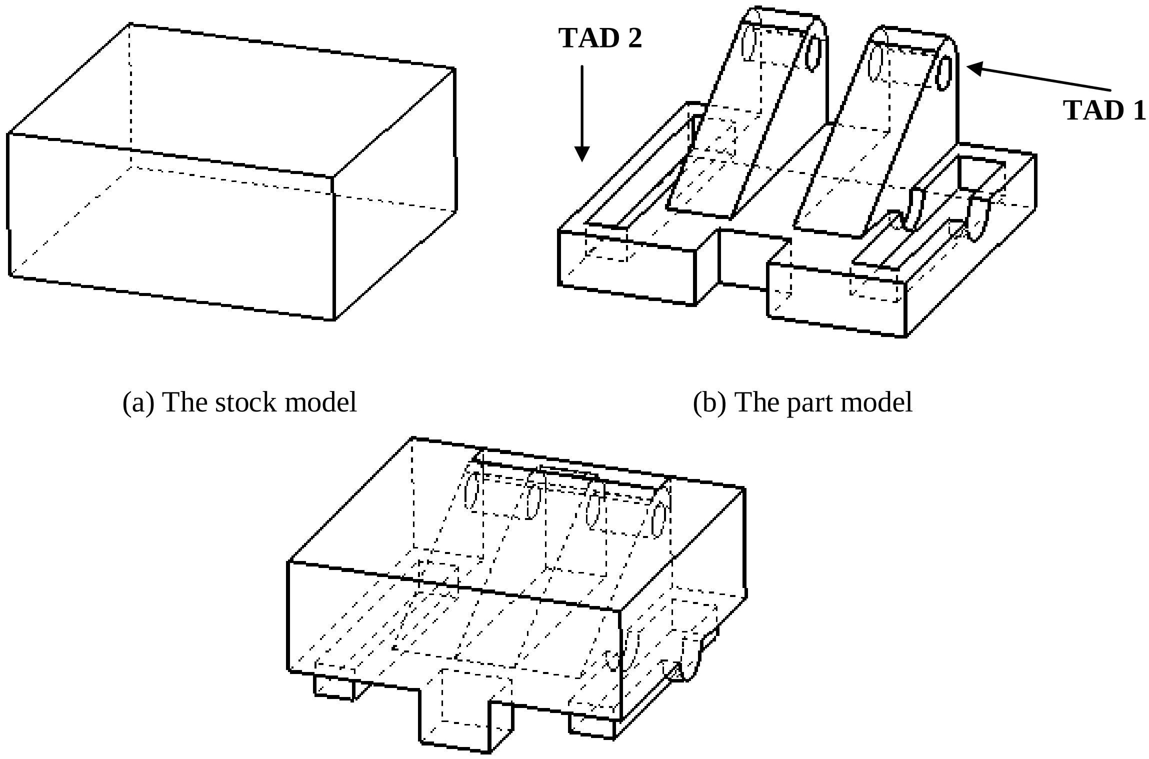
  c.8. **If** , DV is empty, **then**

    c.8.1 CheckValue =false;

  c.9. **End if**

d. **End While**

It is obvious that the above algorithm may still generate multiple interpretations in terms of the final set of TADs with their corresponding V-features and their sequences. However, the resulting feature interpretations are valid since the manufacturability and accessibility of the V-features are guaranteed. Therefore, this multiple interpretation property is actually preferred since one can select a satisfactory solution by comparing different results based on various criteria. Moreover, in our algorithm, V-feature extraction and sequencing are done simultaneously. By doing this, the computational load of the CAPP module will be reduced significantly.

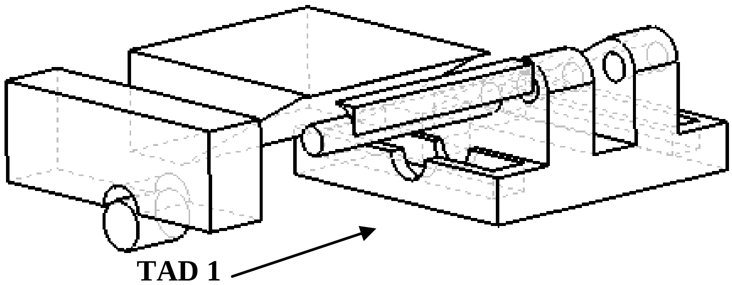


(a) The stock model (b) The part model

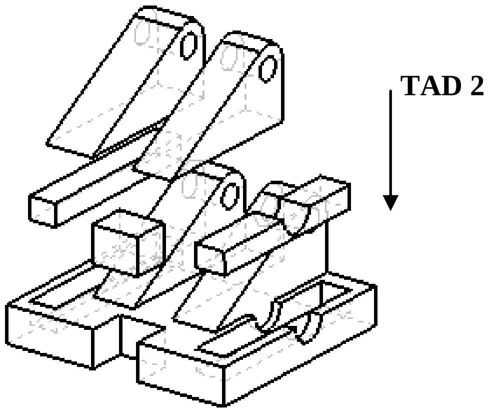
(c) The DV

Figure 3-12: An example of generating multiple feature interpretations

For illustration, Figure 3-12 shows the models of a stock and a part (Figure 3-12a and b) with 2 different TADs (TAD1, TAD2). The DV is shown in Figure 3-12c. In the first machining strategy, TAD1 is selected first. The extracted V-features along TAD1 are shown in Figure 3-13a and the extracted V-features for TAD2 are shown in Figure 3-13b. In the second machining strategy, TAD2 is selected first. The resulting V-features for TAD2 are shown in Figure 3-14a and the extracted V-features for TAD1 are shown in Figure 3-14b.

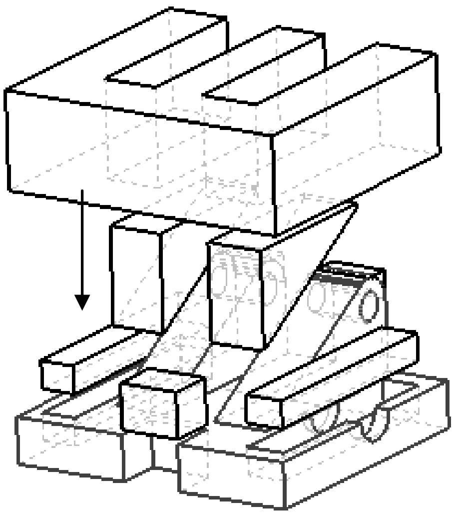


(a) V-features along TAD1

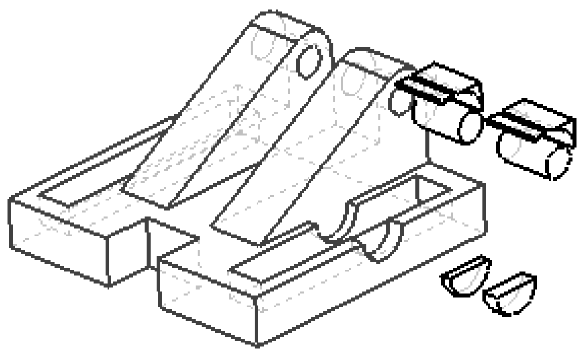


(b) V-features along TAD2

Figure 3-13: V-Feature extraction results for machining strategy 1



(a) V-features along TAD2



(b) V-features along TAD1

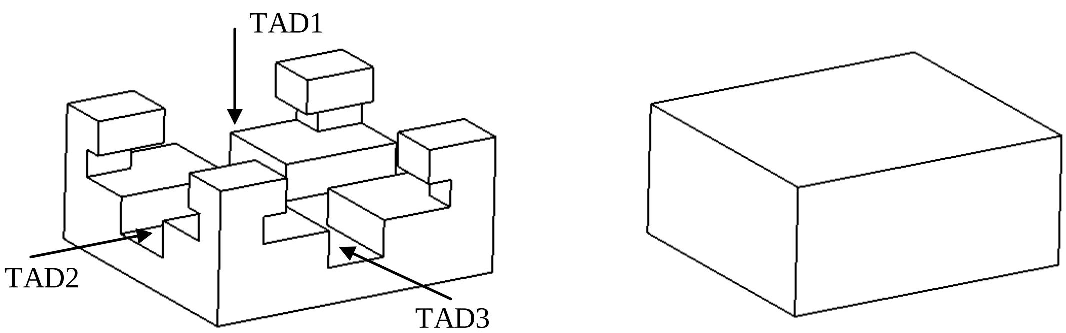
Figure 3-14: V-Feature extraction results for machining strategy 2

3.6 Discussion of the Developed Feature Recognition Approach

In this section, the developed feature recognition method is compared to the hint-based feature recognition (FR) technique which is recognized so far as the most powerful reported approach. However, the below discussion is general and could be valid for other FR methods which are described in chapter 2.

The basic logic behind all the developed hint-based FR techniques is discussed briefly in chapter 2. Although hint-based techniques may be successful in recognizing interactive features, but due to the following reason, all the hint-based techniques suffer from a major deficiency from CAD/CAPP/CAM integration point of view. In hint-based FR approaches, machining features are recognized merely based on the geometry of the part model (i.e. feature traces left on the geometry of the model) while other machining considerations such as required machining set-up and sequence, manufacturability of the extracted features, required tools and machine tools are left to be processed in CAPP/CAM modules. On the other hand, in complex real-world parts, there may exist thousands of feature hints in the part geometry. Let's assume that the existing hint-based FR algorithms are currently matured enough to deal with all the complex feature hints and can construct all the possible machining features from the features' traces left in the part geometry. Therefore, for complex parts, the output of FR module would be a large number of $2\frac{1}{2}$D and 3D machining features which are required to be further processed by CAPP module and therefore cannot directly be used by CAM applications.

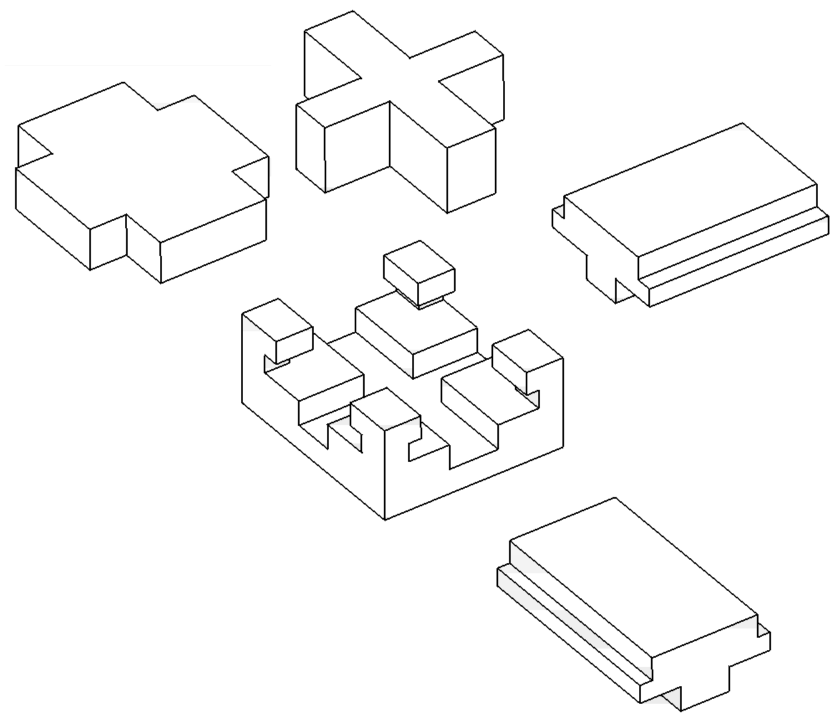
On the other hand, in our approach, by using the concept of TAD-based feature extraction, not only the volumetric features are extracted with the guaranteed manufacturability, but also a practical and near optimal machining sequence is established among the extracted features. In our approach, instead of attempting to create all the feature interpretations of the part, we aimed to create one valid set of machining features which is near to the actual machining practice. Meanwhile, as it is discussed in section 3.5, the developed FR approach is capable of creating multiple feature interpretations which can be used by CAPP to generate an optimal machining plan based on the evaluation criterion.



a) The part model b) The stock model

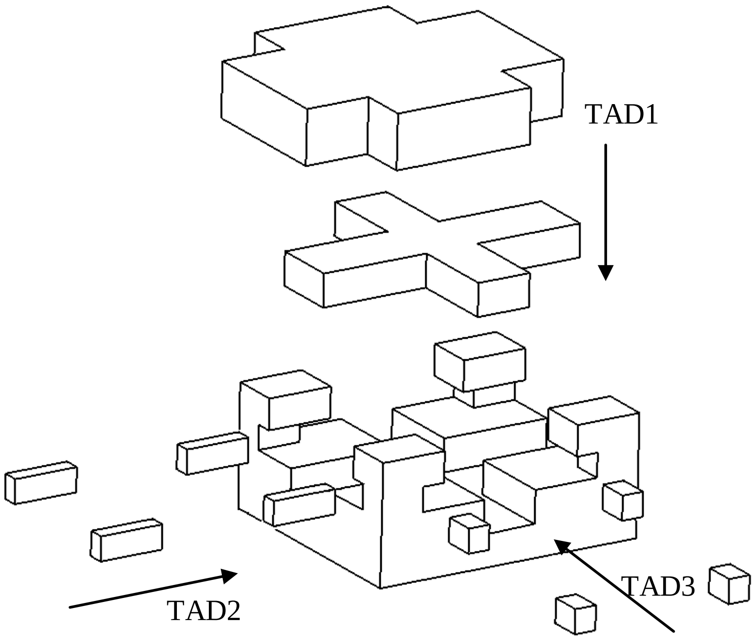
Figure 3-15: An example of a part and a stock

To better understand the above discussion, the two approaches are compared visually by using an example part which is depicted in Figure 3-15. It should be noted that in this example, the features are heavily interacted. Figure 3-16a shows the recognized features by using the hint-based FR technique which is described in [42] while Figure 3-16b shows the extracted V-features using the method presented in this chapter. As can be seen in Figure 3-16a, 4 maximal features are recognized in hint-based method and also the issue of feature interaction is effectively handled by the approach. However, as a consequence of heavy interaction among the features, 1) all the 4 features are not required to be removed from the part, 2) by removing any of the 4 features from the part, the shape of other three features would be affected significantly. Thus, an extra planning stage is required to further process all the 4 features and to determine which of the features are required to be machined first or to determine which of extracted features are excessive and are not necessary to be removed. On the other hand, in our approach, the V-features are extracted with respect to a selective sequence of available TADs. In this example, the sequence of TAD1→ TAD2 →TAD3 is selected. As can be seen from the result in Figure 3-16b, not only the presented approach effectively dealt with the problem of feature interaction, but also the extracted V-features are sequenced for the machining purpose in a practical manner. In addition, the extracted V-features are in their final shape of removal from the part. Therefore, the need for an extra planning stage is eliminated and the extracted V-features can be directly fed into CAM applications.



a) Extracted features by using hint-based technique

Figure 3-16: Comparison between hint-based technique and our approach



b) Extracted V-features and generated machining sequence using our approach

Figure 3-16: Comparison between hint-based technique and our approach-continued

CHAPTER 4

IMPLEMENTATION AND CASE STUDIES

In this chapter, the developed feature recognition system is tested to validate its capability. In the following sections, the system is introduced along with some examples.

4.1 System Interface-The Input

The current implementation of the system uses Parasolid library (Unigraphics Soloution Inc., 1999) as the geometry kernel, runs on Windows XP platform and in the Visual C++ environment.

Figure 4-1 shows the graphic interface of the developed system. The system includes 4 toolbars: *1-View Toolbar 2-Model Input Toolbar 3-Manual Feature Extraction Toolbar 4- Semi-Automated Feature Extraction Toolbar.*

The View Toolbar allows the user to select different viewing formats. Shaded display and wireframe display are two main viewing formats. Moreover, in wireframe display, the user can choose to show the hidden lines or hide them. Different items of this Toolbar are depicted in Figure 4-1.

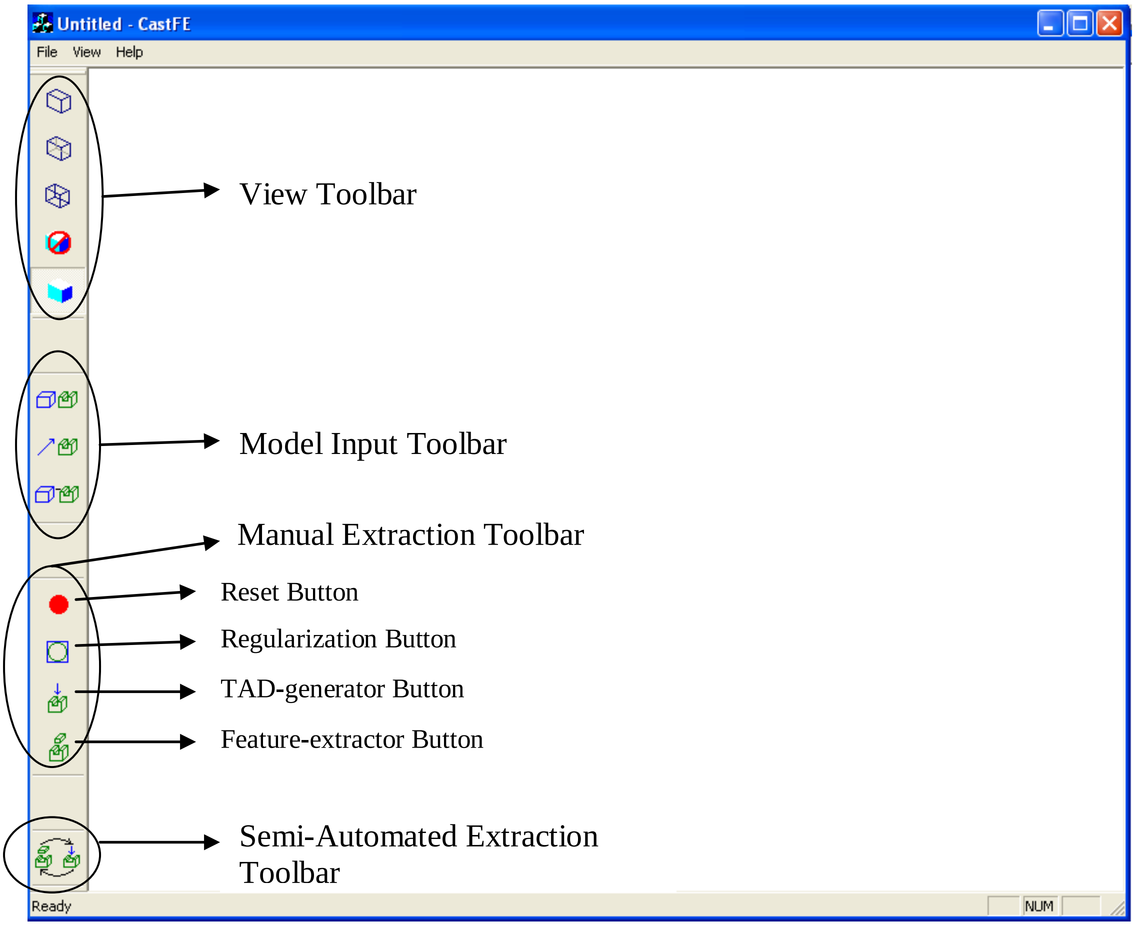


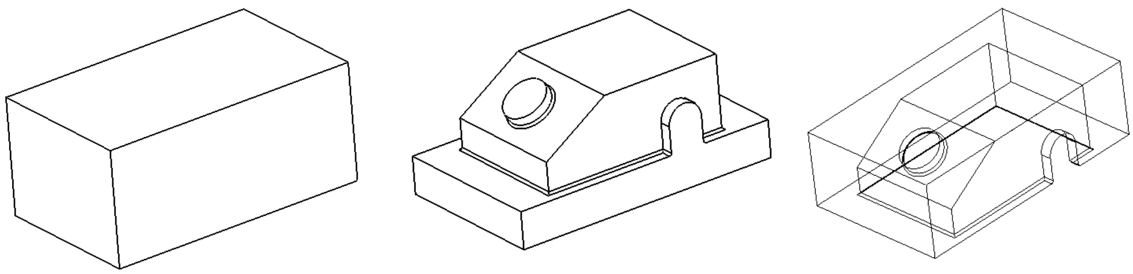
Figure 4-1: Interface of the system

The model could be generated by any CAD software (SolidWorks, UG, etc) and then should be converted and saved into Parasolid file format .xmt-txt which can be recognized by Parasolid Kernel. The items of the Model Input Toolbar are shown in Figure 4-1. With this toolbar, the user can read the part and the stock files from a certain path in the memory and bring them into the graphic area. Then, both the models can be transferred into the same origin and finally Boolean difference is applied between them to obtain the delta volume.

4.2 System Interface-Feature Extraction

Two toolbars are created for the feature extraction stage: *Manual Feature Extraction Toolbar and Semi-Automated Feature Extraction Toolbar.*

In the manual approach, all the stages of the algorithm are done interactively by the user step by step. First, the part and the delta volume are regularized. Then, by running the *TAD-generator* function, all the possible TADs are presented to the user to select an appropriate one. Next, by running the *Manual Feature-extractor* function, machining features are extracted one by one from top to down along the TAD. The main advantage of manual approach is that it enables the user to stop the feature extraction process in any level of VFD-tree (discussed in the previous chapter) in the selected TAD and choose another TAD. Therefore, it is not compulsory to extract all the features in the selected TAD. In the following, various steps of Manual Toolbar are visualized by the use of an example which is shown in Figure 4-2.



(a) The stock (b) The part (c) The DV

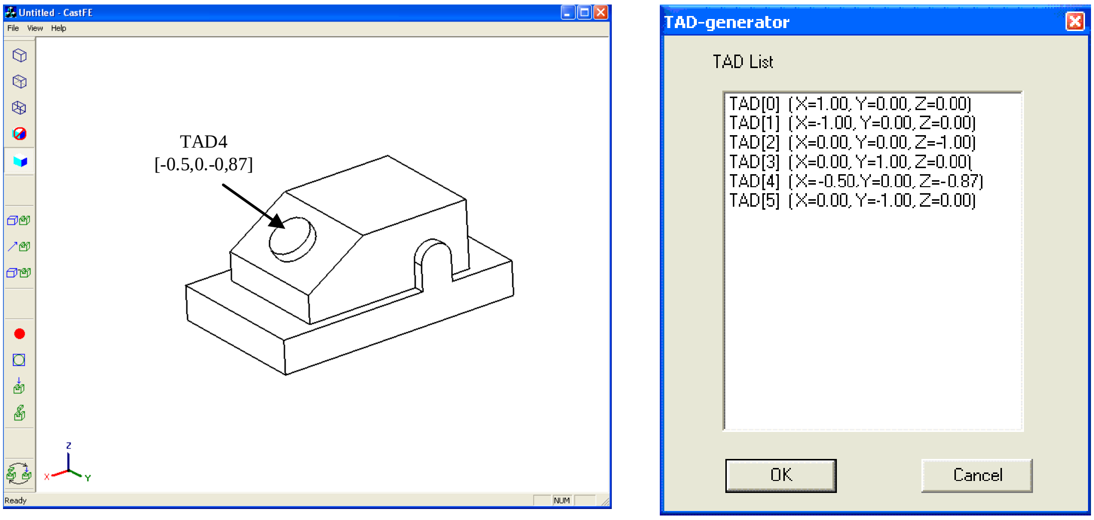
Figure 4-2: Example 1

The regularization icon is shown in Figure 4-1. By running this function, the input part is simplified by removing the minor attributes such as fillets and chamfers. Figure 4-3a shows the regularized model of the original model shown in Figure 4-2b.

The TAD-generator icon is shown in Figure 4-1 under manual extraction toolbar. As discussed in the previous chapter, TADs are generated by finding the planar and cylindrical faces of the model. Then, TADs will be normal of planar faces or axis of cylindrical faces. The resultant TADs for the example part are shown in Figure 4-3b. To start the extraction process, TAD4 [-0.5, 0, -0.87] is selected as the first TAD. (In the graphic area, red line represents X direction, green line represents Y direction and blue line represents Z direction).

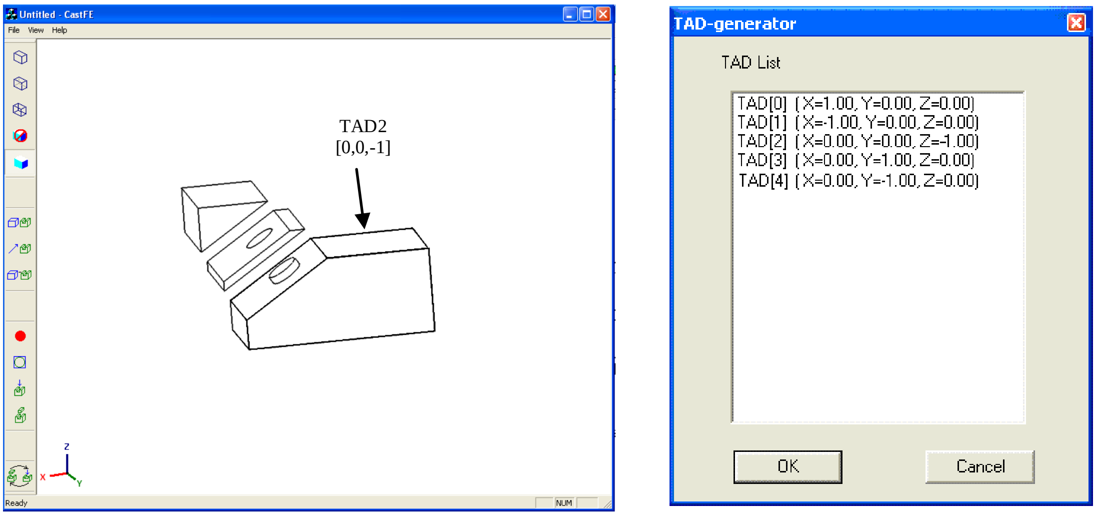
The Feature-extractor icon is shown in Figure 4-1. By every clicking of this button, one V-feature is extracted and removed from the stock from top to down along the TAD in accordance with VFD-tree. The user can decide to stop the extraction process at any point and run the TAD-generator function again to continue the extraction from a new TAD. As can be seen in Figure 4-4a, two top V-features are extracted. Figure 4-4b shows a new set of TADs after running the TAD-generator.

The process of TAD selection and feature extraction is repeated for the example part until all the delta volume is removed from the stock. In this example, TAD2 [0, 0, -1] in Figure 4-4b and TAD0 [0, -1, 0] in Figure 4-5b are selected for the second and third iteration respectively. The extracted V-features are shown in Figures 4-5a and 4-6.



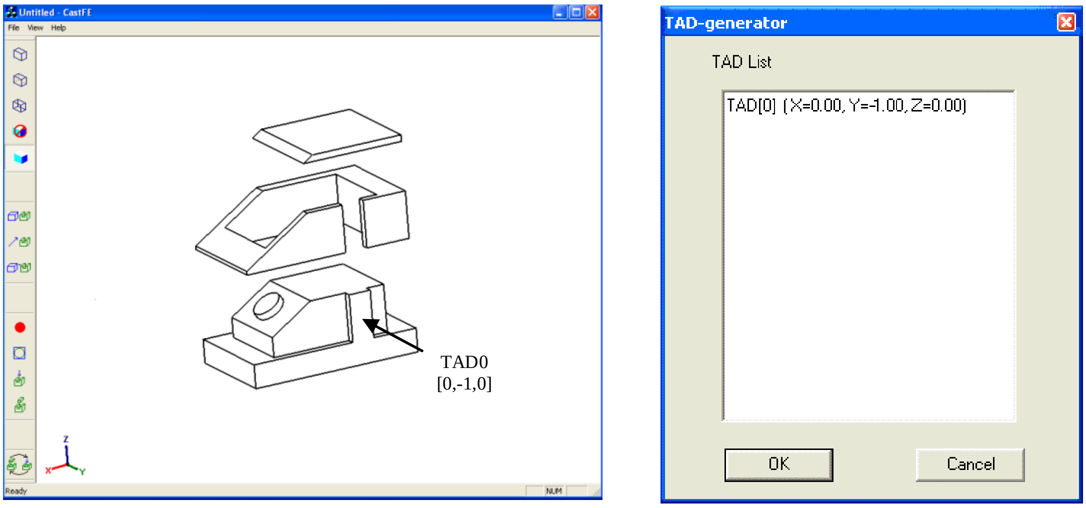
(a) Regularized model (b) first TAD List

Figure 4-3: Model Simplification and TAD list.



(a) Extracted V-features along first TAD (b) TAD list after first extraction

Figure 4-4: Extraction result after the first iteration.



(a) Extracted V-features along second TAD (b) TAD list after second extraction

Figure 4-5: Extraction results after the second iteration.

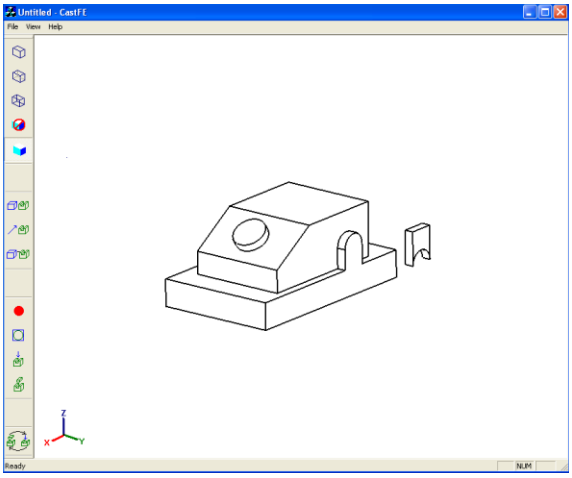


Figure 4-6: Extracted V-feature in the final iteration.

As can be seen from the figures, some of the extracted V-features are in irregular shape and apparently needs further partitioning to be used by CAM. However, by creating the enclosing block of the V-features, we can map the V-features to their corresponding standard V-features introduced in Figure 3-2. Table 4-1 shows the V-features and their regularization blocks. Moreover, for every V-feature, the corresponding standard V-feature type is given.

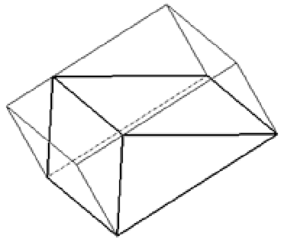
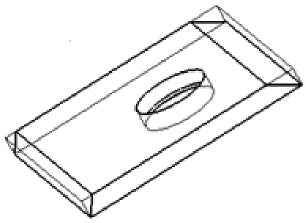
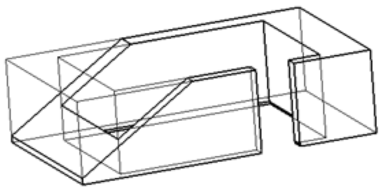
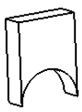
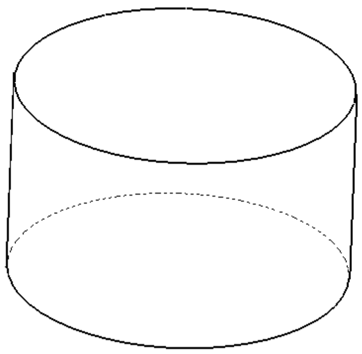
Tool Approach Direction	V-Features and Regularization Block	V-Feature Type
(0.5, 0, .87)		Pocket-Block
(0.5, 0, .87)		Extrusion-Block
(0, 0, 1)		Pocket-Block
(0, 0, 1)		Extrusion-Block
(0, 1, 0)		Step-Block

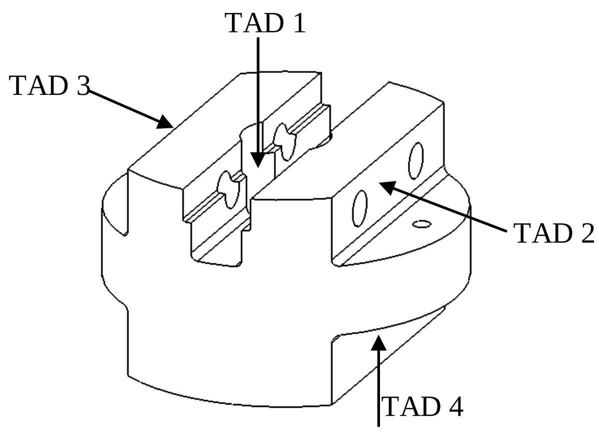
Table 4-1: V-feature mapping

The Semi-Automated Toolbar is shown in Figure 4-1. In this approach, steps of the regularization, TAD generation and feature extraction are packed and put into one loop. After the user imports the stock and the part, by clicking this icon, the model will be simplified and possible TADs are presented to the user. After selection of a TAD, all the features of VFD-tree will be extracted and removed from the stock. Then the program automatically checks whether further material needs to be removed from the stock. If so, it will present a new TAD list to the user again. This process of TAD selection and feature extraction will be continued until the delta volume becomes empty. To test the Semi-Automated approach, the part and stock shown in Figures 4-7a and 4-7b are used. For simplification, only the V-features are shown and graphic interface is eliminated.

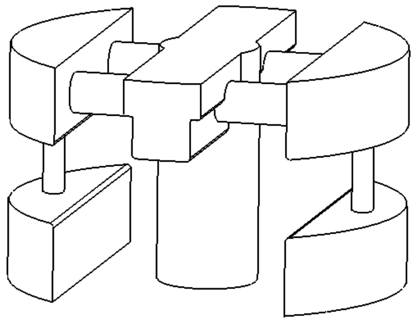
Figure 4-7b shows all the 4 possible TADs in respect to the part model. After DV partition and V-feature extraction, TAD4 is eliminated due to its redundancy. Among the remaining 3 possible TADs, there are 6 possible sequenced routes. The sequence of TAD1→TAD2→TAD3 is selected here for illustration. The V-features extracted from these 3 TADs are shown in Figure 4-7d, 4-6e and 4-7f respectively. There are 7 drilling V-features and 8 $2\frac{1}{2}$D milling V-features. Furthermore, it can be seen that the 4 drilling V-features shown in Figure 4-7e and all the $2\frac{1}{2}$D milling V-features are of irregular shape compared with the standard ones in Figure 3-2. As a result, these irregular V-features are corrected to their corresponding ones in Figure 3-2 while the final shape of the part is not affected. These standardized V-features can be used to generate NC programs directly.



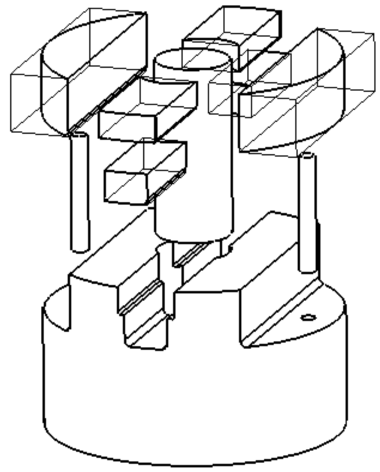
(a) The stock



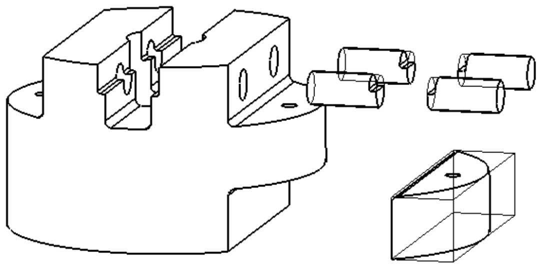
(b) The part



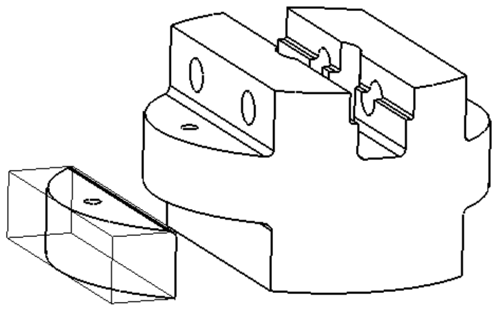
(c) The DV



(d) Extracted V-features along TAD1



(e) Extracted V-features along TAD2

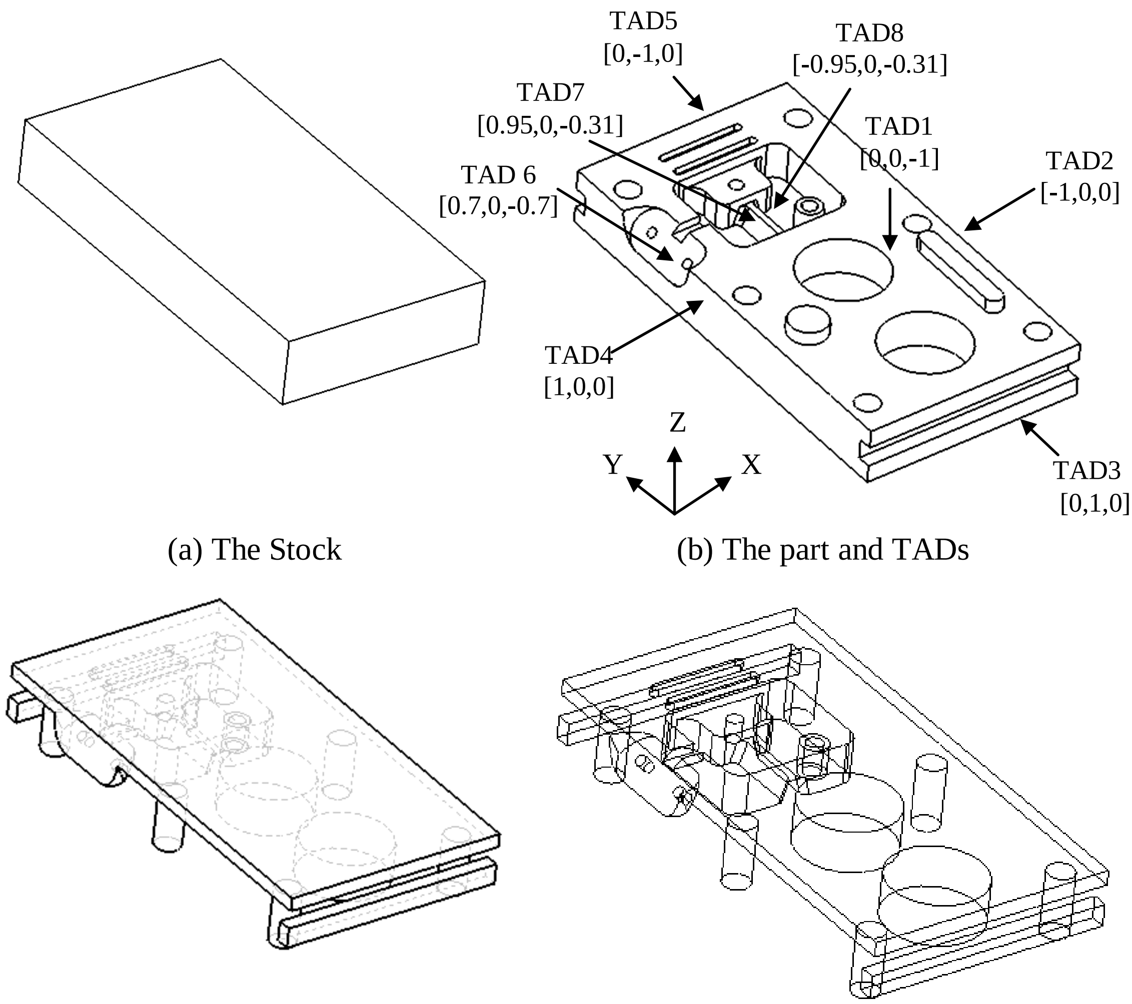


(f) Extracted V-features along TAD3

Figure 4-7: Example 2

4.3 A Case Study

Figure 4-8b shows a part which is modified version of TEAM part, a bench mark suited for testing feature recognition algorithm. Such parts are available at national design repository at NIST [60]. Unlike TEAM part which is completely $2\frac{1}{2}$D, in this case the rib inside the cavity has been made inclined, making the part to be non-$2\frac{1}{2}$D. The stock model and delta volume model are depicted in figure 4-8a, 4-8c and 4-8d. For better visualization, the delta volume is shown in both wireframe and solid formats. Based on the geometry of the part model, 8 possible TADs can be extracted which are shown in Figure 4-8b.

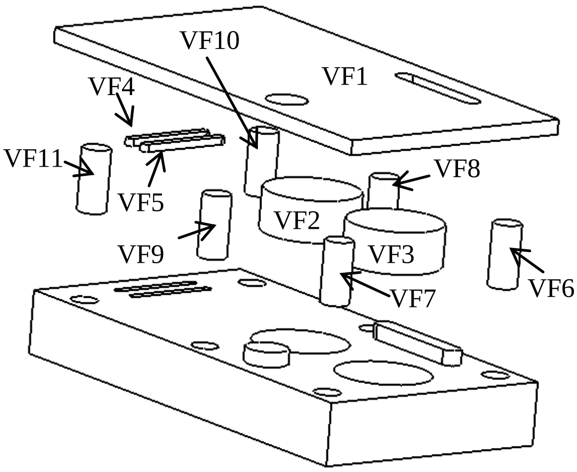


(a) The Stock (b) The part and TADs

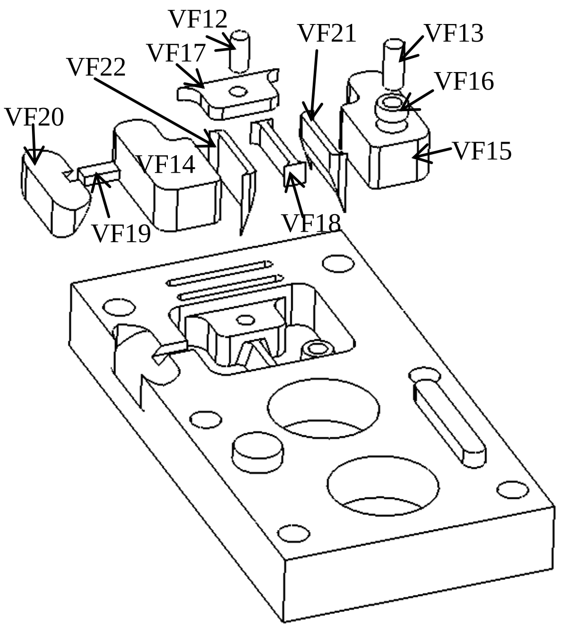
(c) The DV in solid format (d) The DV in wireframe format

Figure 4-8: Case Study

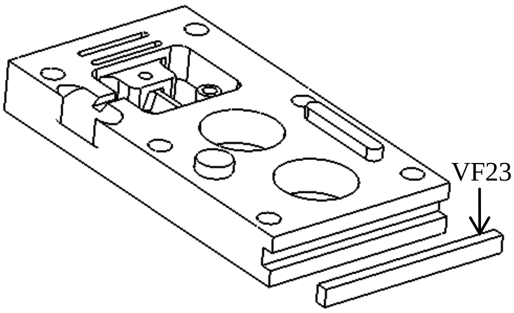
Here, the sequence of TAD1→TAD3→TAD5→TAD6 is selected for illustration. The extracted V-features are shown in Figures 4-9a to 4-9e. Remaining stock volume after extraction of V-features in each TAD is also shown.



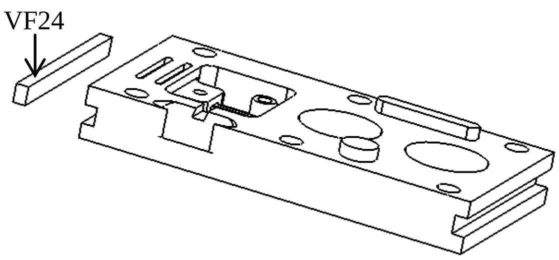
(a) Extracted V-features along TAD1-part1



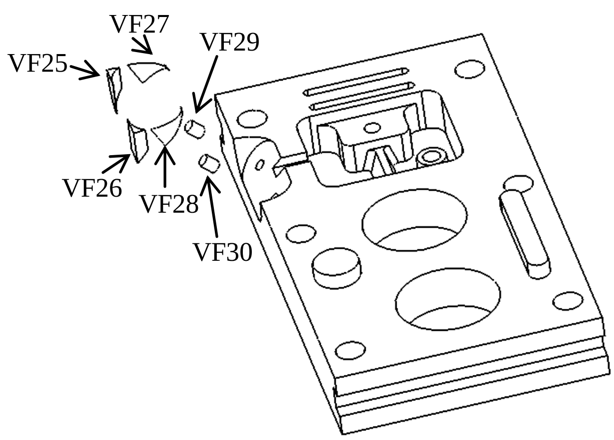
(b) Extracted V-features along TAD1-part 2



(c) Extracted V-features along TAD3



(d) Extracted V-features along TAD5

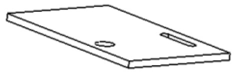
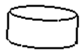
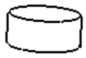
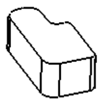
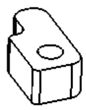


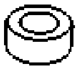
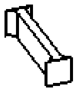
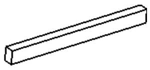
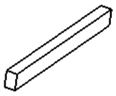
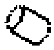
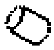
(e) Extracted V-features along TAD6

Figure 4-9: Extracted V-features of the case study

Table 4-2 shows one possible machining sequence of the extracted V-features which is consistent with structure of the resulted VFD-tree. Moreover, for each V-feature, the corresponding standard feature type in Figure 3-2 is given.

Table 4-2: Machining sequence of V-features for the case study part

Sequence Number	Tool Approach Direction	V-feature Name	Geometric Shape of V-feature	V-feature Type
1	TAD1	VF1		Extrusion-block
2	TAD1	VF2		Cylinder
3	TAD1	VF3		Cylinder
4	TAD1	VF4		Pocket-block
5	TAD1	VF5		Pocket-block
6	TAD1	VF6		Cylinder
7	TAD1	VF7		Cylinder
8	TAD1	VF8		Cylinder
9	TAD1	VF9		Cylinder
10	TAD1	VF10		Cylinder
11	TAD1	VF11		Cylinder
12	TAD1	VF12		Cylinder
13	TAD1	VF13		Cylinder
14	TAD1	VF14		Pocket-block
15	TAD1	VF15		Pocket-block

16	TAD1	VF16		Pocket-block
17	TAD1	VF17		Step-block
18	TAD1	VF18		Pocket-block
19	TAD1	VF19		U-slot-block
20	TAD1	VF20		Blind-slot-block
21	TAD1	VF21		Pocket-block
22	TAD1	VF22		Pocket-block
23	TAD3	VF23		U-slot-block
24	TAD5	VF24		U-slot-block
25	TAD6	VF25		Notch-block
26	TAD6	VF26		Notch-block
27	TAD6	VF27		Notch-block
28	TAD6	VF28		Notch-block
29	TAD6	VF29		Cylinder
30	TAD6	VF30		Cylinder

CHAPTER 5

CONCLUSION

5.1 Contributions

In this thesis, a new approach aiming at direct link between volumetric feature extraction and NC tool path generation for 3-axis machining is proposed and implemented. In the developed system, firstly the CAD model is simplified and minor attributes such as fillet and chamfer are removed from the model. Secondly, manufacturable V-features are extracted based on the feasible TADs. At the same time, in each TAD, extracted features are sequenced for the machining purpose in a near optimal and practical way. Finally, the removed minor features are attached again to the corresponding features.

The advantages of the proposed method are as follows:

(1) It is based on volumetric feature extraction from the delta volume, thus eliminating the restrictions on the shape of the stock. The minor attributes of the V-features are effectively retained. The irregular shaped $2\frac{1}{2}$D milling V-features are converted to their regular ones. Therefore, the resultant V-features can be directly fed into feature-based CAM functions.

(2) Feature interaction problems among drilling, $2\frac{1}{2}$D milling, and 3D milling V-features are effectively resolved by considering manufacturing practices.

(3) The proposed algorithm allows multiple feature interpretations with valid manufacturability. This provides much flexibility for process planning to pursue optimization.

(4) In the developed algorithm, feature extraction and feature sequencing are done in the same manner of practical process planning procedure and therefore the proposed framework has the potential to easily integrate with CAPP functions. Moreover, since manufacturability of features is guaranteed, it may significantly reduce the computational load of process planning function.

5.2 Future Work

In order to fully integrate the developed system with CAM function for real-life parts, several extensions are needed. Following are some of the recommendations for future work:

- The current method for resolving $2\frac{1}{2}$D milling feature interaction only allows a single V-feature interpretation, which needs further study.
- The task of sequencing the features in each TAD is done in a practical but not optimal way. To solve this problem, an evaluation criterion such as cost or time of machining should be considered in each stage.
- Currently in the developed system the process of TAD selection is done manually by the user. However, in order to fully automate the process, a TAD selection algorithm should be developed.
- Currently, the manufacturability of features is ensured from accessibility and geometric perspectives. However, in practical level, factors beyond the two above criteria should be taken into consideration such as availability of machines, tools, etc. Therefore, one of the future works could be integrating a

library of manufacturing resource with the current system to ensure the manufacturability of features from all the perspectives.

Bibliography

[1] H.P. Wang and J.K. Li. Computer-Aided Process Planning, *Elsevier Science*, 1991.

[2] Gideon Halevi and Roland D. Weill. Principles of Process Planning, A logical Approach, *Chapman & Hall*, 1995.

[3] Computer-Aided Design, Engineering, and Manufacturing Systems Techniques and Applications, Volume III, Operational Methods in Computer-Aided Design Edited by Cornelius T Leondes, 2000.

[4] Jami J. Shah and Martti Mäntylä. Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications, *Wiley, Chichester*, 1995.

[5] Jami J. Shah, M. Mantyla and DS Nau, Advances in Feature Based Manufacturing, *Elsevier*, 1994.

[6] http://www.ivcc.edu/perez/Design%20Intent.htm

[7] Xin Dong and Michael J. Wozny. A method for generating volumetric features from surface features. *Int. J. Comput. Geometry Appl*. Vol.1, No.3, pp.281-297, 1991.

[8] Shah, J.J. Assessment of features technology. *Computer Aided Design*, Vol. 23, No.5, pp.331-43, 1991.

[9] J.J. Shah, D. Anderson, Y.S. Kim and S. Joshi. A Discourse on Geometric Feature Recognition from CAD Models. *Journal of Computing and Information Science in Engineering*, Vol.1, No.1, 2001.

[10] Babic b., Nesic, n. and Miljkovic Z. A review of automated feature recognition with rule-based pattern recognition. *Computers in Industries,* Vol.59, No.4, pp.321-337, 2008.

[11] Ansaldi S., De Floriani L. and Falcidieno B. Geometric Modeling of Solid Objects by Using a Face Adjacency Graph Representation, *Proc. Siggraph '85, Computer Graphics*, Vol. 19, No. 3, pp. 131–139,1985.

[12] Joshi, S., and Chang, T. Graph-Based Heuristics for Recognition of Machined Featured Features from a 3D Solid Model, *Comput.-Aided Des*, Vol. 20, No. 2, pp. 58–66,1988.

[13] Marefat, M. and Kashyap R. Geometric Reasoning for Recognition of Three-Dimensional Object Features, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol.12, No. 10, 1990.

[14] Trika, S., and Kashyap, R. Geometric Reasoning for Extraction of Manufacturing Features in Iso-Oriented Polyhedrons. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol.16, No.11, pp.1087–1100, 1993.

[15] Fields M.C. and Anderson D.C. Fast Feature Extraction for Machining Features. *Comput.-Aided Des*. Vol. 26, No. 11, pp. 803–813, 1994.

[16] Woo TC. Feature extraction by volume decomposition. In: Proceedings of the Conference on CAD/CAM, 1982.p. 39-45

[17] L. K. Kyprianou. Shape Classification in Computer Aided Design. Ph.D. dissertation, Christ College, Univ. Cambridge, Cambridge, U.K., July 1980.

[18] Y.S. Kim. Recognition of form features using convex decomposition. *Computer-Aided Design*, Vol.24, No.9, pp.461–476, 1992.

[19] F. Pariente and Y.S. Kim. Incremental and localized update of convex decomposition used for form feature recognition, *Computer-Aided Design*, Vol. 28, No.8, pp.589–602, 1996.

[20] Y.S. Kim, Y. Kim, F. Pariente and E.Wang. Geometric reasoning for mill-turn machining process planning, *Computers and Industrial Engineering*, Vol. 33 (3–4), pp.501–504, 1997.

[21] Sakurai H. and Chin C. Definition and Recognition of Volume Features for Process Planning, *Advances in Feature Based Manufacturing*, J. J. Shah, M. Ma¨ntyla¨, and D. Nau, eds., Elsevier Science Publishers, New York, pp. 65–80.

[22] Sakurai, H. Volume Decomposition and Feature Recognition: Part I Polyhedral Objects, *Comput.-Aided Des.*, Vol.27, No.11, pp. 833–843, 1995.

[23] Sakurai H. and Dave P. Volume Decomposition and Feature Recognition: Part 2—Curved Objects, *Comput.-Aided Des.*, Vol.28, No. 6, pp. 517–537, 1996.

[24] Tseng Y. and Joshi S. Recognizing Multiple Interpretations in 2-1/2D Machining of Pockets, *Int. J. Prod. Res.*, Vol. 32, No. 5, pp. 1063–1086, 1994.

[25] Bezdek E.J., Thompson D.C., Wood K.L. and Crawford R.H., Volumetric Feature Recognition for Direct Engineering, *Direct Engineering: Toward Intelligent Manufacturing*, A. Kamrani and P. Sferro, eds., Kluwer Academic, Dordrecht, the Netherlands, pp. 15–69,1999.

[26] Shen Y. and Shah J. Feature Recognition by Volume Decomposition Using Half-Space Partitioning, *Proc. 20th ASME Design Automation Conference*, pp. 575–583, 1994.

[27] Marefat M. and Kashyap R. Geometric Reasoning for Recognition of Three Dimensional Object Features, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol.12, No.10, 1990.

[28] Ferriera, J., and Hinduja, S. 1990, ‘‘Convex Hull Based Feature Recognition Method for 2.5 D Components,’’ Comput.-Aided Des., **22**, No. 1, pp. 41–49.

[29] Y.Woo, Fast cell-based decomposition and applications to solid modeling, Computer-Aided Design 35 (2003) 969–977.

[30] Y. Woo, H. Sakurai. Recognition of maximal features by volume decomposition, Computer-Aided Design 34 (2002) 195–207.

[31] J.H. Vandenbrande, A.A.G. Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (12) (1993) 1–17.

[32] W.C. Regli, S.K. Gupta, D.S. Nau. Towards multiprocessor feature recognition, Computer-Aided Design 29 (1) (1997) 37–51.

[33] J.H. Han, A.A.G. Requicha, Integration of feature based design and feature recognition, Computer-Aided Design 29 (5) (1997) 393–403.

[34] J.H. Han, M. Pratt and W.C. Regli. Manufacturing feature recognition from solid models: a status report, *IEEE Transactions on Robotics and Automation*, Vol.16, No.4, pp.782–796, 2000.

[35] J.H. Han, I. Han, E. Lee, J. Yi, Manufacturing feature recognition toward integration with process planning, IEEE Transactions Systems, Man and Cybernetics. Part B. Cybernetics 21 (3) (2001) 373–380.

[36] S. Meeran, J.M. Taib, M.T. Afzal. Recognizing features from engineering drawings without using hidden lines: a framework to link feature recognition and inspection systems, International Journal of Production Research 41 (3) (2003) 465–495.

[37] A.D. McCormack, R.N. Ibrahim, Process planning using adjacency-based features, International Journal of Advanced Manufacturing Technology 20 (2002) 817–823.

[38] M.G.L. Sommerville, D.E.R. Clark, J.R. Corney, Viewer-centered geometric feature recognition, Journal of Intelligent Manufacturing 12 (2001) 359–375.

[39] S. Gao, J.J. Shah, Automatic recognition of interacting machining features based on minimal condition subgraph, Computer-Aided Design 30 (9) (1998) 727–739.

[40] V. Sundararajan, P.K. Wright, Volumetric feature recognition for machining components with freeform surfaces, Computer-Aided Design 36 (2004) 11–25.

[41] S. Subrahmanyam, A method for generation of machining and fixturing features from design features, Computers in Industry 47 (2002) 269– 287.

[42] K. Rahmani, B. Arezoo, Boundary analysis and geometric completion for recognition of interacting machining features, Computer-Aided Design 38 (2006) 845–856.

[43] Sormaz N, Corney JR, Clark DER, Tuttle R. A feature recognition algorithm for NC-machining. Proceeding of the IFIP TC5/WG5.2 Workshop on Geometric Modeling in Computer Aided Design 1996;223-32.

[44] G. Little, R. Tuttle, D.E.R. Clark, J. Corney, The Heriot-Watt Feature- Finder: CIE97 results, Computer-Aided Design 30 (13) (1999) 991– 996.

[45] G. Little, D.E.R. Clark, J.R. Corney, R. Tuttle, Delta-volume decomposition for multi-sided components, Computer-Aided Design 30 (9) (1998) 695–705.

[46] R. Tuttle, G. Little, D.E.R. Clark, J. Corney, Feature recognition for NC part programming, Computers in Industry 35 (1998) 275–289.

[47] Y. Kim, E.Wang, C. Lee, and H. Rho, "Feature-based machining precedence reasoning and sequence planning," in *Proc. 1998 ASME Design Engineering Technical Conf. (DETC98/CIE-5707)*, 1998.

[48] B. Khoshnevis, D. Sormaz, and J. Park, "An integrated process planning system using feature reasoning and space search-based optimization," *IIE Trans.*, vol. 31, no. 7, pp. 597–616, 1999.

[49] C. Roberts, R. Stage, N. Hubele, M. Henderson, and E. Perez, "A new approach to manufacturing features for evaluation and operational planning," in *Proc. 5th IFIP Workshop, WG 5.2*, 1996.

[50] R. Stage, C. Roberts, and M. Henderson, "A framework for representing and computing tool accessibility," in *Proc. ASME Design for Manufacturing Conf., DETC97/DFM-4323*, 1997.

[51] , "Generating resource-based flexible form manufacturing features through object driven clustering," *Computer Aided Design*, vol. 31, no. 2, pp. 119–130, 1999.

[52] K. K. Jurrens, J. E. Fowler, M. Elizabeth, and A. Algeo, "Modeling of Manufacturing Resource Information: Requirements Specification," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NISTIR 5707, July 1995.

[53] D. M. Gaines and C. Hayes, "A constraint-based algorithm for reasoning about the shape producing capabilities of cutting tools in machined parts," presented at the ASME Design for Manufacturing Conference, DETC97/DFM-4322, 1997.

[54] , "CUSTOM-CUT: A customizable feature recognizer," *Comput. Aided Des.*, vol. 31, no. 2, pp. 85–100, 1999.

[55] D. M. Gaines, F. Castano, and C. Hayes, "MEDIATOR: Reconfigurable feature recognition for a maintainable, extendible CAD/CAPP integration," *ASME J. Mech. Des.*, vol. 121, no. 1, pp. 145–158, 1999.

[56] W. Faheem, C. Hayes, D. M. Gaines, and J. Castano, "COORDINATOR: A robust setup planner that does early detection of fixture-feature interactions," in *ASME Design for Manufacturing Conference, DETC98/DFM-5741*, 1998.

[57] S.M. Lam, T.N. Wong, Recognition of machining features—a hybrid approach, International Journal of Production Research 38 (17) (2000) 4301–4316.

[58] J. Dong, S. Vijayan, Features extraction with the consideration of manufacturing processes, International Journal of Production Research 35 (8) (1997) 2135–2155.

[59] Kailash SB, Zhang YF, Fuh JYH. A volume decomposition approach to machining features extraction of casting and forging components. Computer Aided Design 2001; 33(8):605-17.

[60] Regli WC, Gaines DM. National repository for design and process planning. Computer Aided Design 1997; 29(12):895–905.