# Development of Machining Feature Recognition Software Module

**Wisnu Aribowo †**
Department of Industrial Engineering
Bandung Institute of Technology, Indonesia
Tel: (+62) 22-2506449, Email: wisnu@ti.itb.ac.id

**Mohammad Mi'radj Isnaini**
Department of Industrial Engineering
Bandung Institute of Technology, Indonesia
Tel: (+62) 22-2506449, Email: iis@ti.itb.ac.id

**Anas Ma'ruf**
Department of Industrial Engineering
Bandung Institute of Technology, Indonesia
Tel: (+62) 22-2506449, Email: maruf@ti.itb.ac.id

**Abstract.** In feature based machining, automatic feature recognition is the part that extracts machining features from a product model produced by CAD systems. Numerous researches have come up with many feature recognition algorithms, yet practical implementations are still limited and relatively inaccessible to small machining companies in developing countries due to their high cost. This paper reports the progress of development of a new feature recognition software module that is based on open source libraries, software, and standards. It will be a part of a larger project to develop a cost effective toolchain encompassing the whole stages of CAD/CAM for metal machining. It takes standard IGES/STEP solid models as input and the output is the list of identified features. One common problem that is not fully solved in machining feature recognition is the difficulty in handling complex work pieces with non-standard topological relations and intersections of features. A fully automatic feature recognition solution is difficult to realize. Instead, the developed module alleviates those practical problems by combining the geometrical and topological data retrieved from solid model with hints or feedback provided by human operators. This interactive approach proves to be practical as demonstrated by evaluations using solid models of real machining workpieces with varying complexities.

**Keywords:** feature recognition, CAD/CAM, open source

## 1. INTRODUCTION

In numerical control based metal machining production system, machining processes are carried out by various CNC machines based on digital instructions in the form of G&M codes. A typical workflow of the automated production system starts when a part design is created in a CAD system. Based on the model, the sequence of machining processes is determined, including the specification of machines as well as all the necessary fixtures and tools. For each process, the machining parameters, e.g. feed rate, cutting speed, machining pattern, etc. are selected. Those activities are within the scope of Computer Aided Process Planning (CAPP). Finally, based on all those data and parameters, the Computer Aided Manufacturing (CAM) part generates the machining instructions in form of NC programs automatically. Before actually executing the program in the real machines, a series of simulation may be performed to verify the technical feasibility of the machining plan.

In such framework, machining features of the part are important entities in determining the machining processes, all the way until the NC program generation. Features data may be incorporated right from the start by the use of feature based CAD systems. However, in case where the production systems opt not to use such 'feature by design' approach, the features data have to be identified from CAD models or drawings. The recognition of machining features lies right in the interface between CAD and CAPP-CAM subsystems, and thus is an important step in the integration of CAD/CAM automation.

A feature recognition (FR) system extracts raw data, such as geometrical and topological elements, from the 3D model of part. By analyzing the data, the goal of the FR system is to identify all features that should be processed or machined to realize the part. The features can range from standard parametric features, for instance tap holes and rectangular pockets, until non-standard features having complex boundaries.

Researches on approaches and methods to do feature recognition have been abundant in the literature. A popular method is the graph-based method (Di Stefano, Bianconi, & Di Angelo, 2004), where relationships among faces in a solid model is represented as a graph. Machining features can be identified by recognizing the patterns in the graph. In the volumetric decomposition approach (Sundararajan & Wright, 2004), a solid model is represented as series of Boolean operations where the final workpiece is the subtraction result of negative volumes from the stock material. The shapes of those negative volumes identify the machining features of the workpiece. In the hint-based approaches (Vandenbrande & Requicha, 1993), recognition of features from a model is performed by identifying traces of elements which can function as hints of the existence of features. For example, two faces that face each other in a cavity may be a hint of the existence of a slot feature. Besides those relatively popular methods, other feature recognition approaches include artificial intelligence, ontology (Wang & Yu, 2014), and hybrid combination of existing methods (Rameshbabu & Shunmugam, 2009).

Common problems in feature recognition that have not been fully addressed are difficulties in recognizing features in complex workpieces having many features intersections (Han, Pratt, & Regli, 2000; Babic, Nesic, & Miljkovic, 2008). The problems, beside other factors such as the lack of information in the model, have caused theoretical algorithms of feature recognition to be so complex that practical implementations become hard. A fully automatic feature recognition system without human intervention to this day may be still too difficult to realize. Practical algorithms have to be able to combine the hints received from the workpiece model and hints received by human.

On the practical side, currently most FR softwares are relatively expensive and some are only offered as modules of commercial application suites. This often becomes problem for industries, especially for small industries in developing countries, because they do not have much financial capability and they also do not posses enough research and development capacity to be able to develop automation solutions by themselves.

This paper reports the current progress of the development of an effective feature recognition software that can be used as a tool to perform machining feature recognition. The input for the software is 3D solid model in standard IGES format, while the output is list of feature data, including the feature types, parameters, and locations. Two important characteristics of the software are that it is interactive and uses open standards. In some cases, for example where two features or more intersect, a simple automatic rule would not be able to identify features in a robust manner. To address the problem, our approach is to allow user to provide additional hints. In practice, this approach is welcome because design intents – in the sense of machining features – are not always possible to provide completely in the solid model. Meanwhile, with the use of open standards and libraries, our motivation is to provide the solution to as many potential users as possible, especially small metal machining industries in developing countries.

.

## 2. DEVELOPMENT OF ALGORITHM AND SOFTWARE

### 2.1 Feature Recognition Algorithm

Through analysis of various machining products, we categorize machining features into three: cylindrical hole features, non-cylindrical features, and non-cylindrical features with irregular sides. Table 1 lists the types of machining features. Each feature type has a set of unique parameters. For

Table 1. Types of machining features.

| GROUP | FEATURE TYPE |
|---|---|
| (A) Cylindrical hole features | 1. BLIND BORE |
| | 2. THROUGH BORE |
| | 3. COUNTERBORE |
| | 4. DRILL |
| | 5. TAP |
| | 6. REAMER |
| (B) Non-cylindrical features | 7. FACE |
| | 8. 2-SIDE POCKET |
| | 9. 3-SIDE POCKET |
| | 10. 4-SIDE POCKET |
| | 11. STEP |
| | 12. SLOT |
| | 13. SIDE |
| (C) Non-cylindrical features with irregular sides | 14. CUTTERPATH |
| | 15. CUTOFF |
| | 16. ARBITRARY HOLE |

example, any feature of type BLIND BORE has three parameters, i.e. diameter, depth, and chamfer size, while the parameters of SLOT features are width, length, and depth. For the feature types in group (C), in addition to those scalar valued parameters, they also have a connected polylines which define the irregular feature sides. Other than parameters, the feature data also include one or more points that define the locations of feature instances.

Features that belong to group (A) are relatively well-structured. The recognition domain is relatively small, and thus the recognition can be done automatically. In a typical workpiece that we observe most features are of this group. Therefore, the automatic recognition needs to be robust. The feature recognition process in general is performed in following steps:

(a) Preparation of coordinate systems. By observing the bounding box of the solid model, several coordinate systems are created. In addition to the global coordinate system, by default six more coordinate systems are created, which represent the front, rear, right, left, top, and bottom planes.

(b) Extraction of geometrical and topological data from the solid model. A boundary representation (B-rep) model contains inter-related geometrical and topological elements. A shell consists of faces, and a face contains loops, where each loop is defined by connected edges, which in turn an edge has two vertices. Each element has its respective geometrical representation, for example the shape of a face is defined by a surface model which is bounded by an outer loop.

(c) Detection of cylinders. A cylindrical feature, e.g. a COUNTERBORE, is composed of one or more connected coaxial cylinders. In turn, a cylinder is composed of one or more convex coaxial cylindrical faces. This step identifies all cylinders in the model, along with all parameters, i.e. diameters and locations of the two ends.

(d) Identification of cylindrical features. In this step, adjacent cylinders are grouped and arranged based on the locations. Rules, e.g. the composition of cylinders in the group, determine the feature type(s). For example, Figure 1 shows a group of three connected cylinders where two of them are straight. Rules are then used to determine the specific feature type. The snippet of the algorithm for that specific case is as follows:
   - If the combination of diameters and depths is found in the TAP data, then the feature type is TAP.
   - If not found, and if the combination of diameters and depths is found in the COUNTERBORE data, then the feature type is COUNTERBORE.
   - If not found, then it will be spit into two features of type THROUGH/BLIND BORE and BLIND BORE.

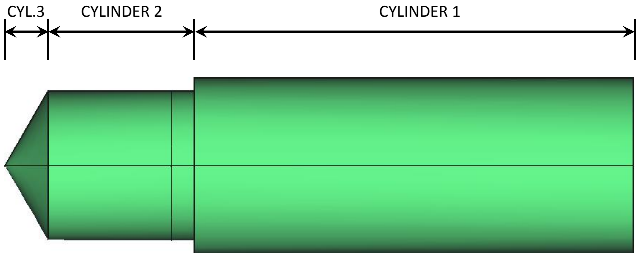


Figure 1: An example of a group of connected cylinders

(e) Determining the parameters. The parameters are derived from the dimensions of the composing cylinders. For example, a TAP feature has four main parameters: diameters of depths of large and small cylinders. The axis orientation determines which plane the feature belongs to. The plane can be any one of the six standard planes or an inclined plane. The feature origin is represented in the plane's coordinate system.

(f) Features grouping. Features that have same parameter values and same plane are grouped into one feature group with multiple feature instances. Feature instances only differ in their locations.

At times, features may intersect each other. In that case, a cylinder may be broken and appears in the model as two cylinders that share same axis vector and are separated by small distance. If the algorithm detects such case, it is handled in step (c) by fusing those two cylinders as one. Another special case is when features are stacked. For example, a TAP feature is made at the bottom of a BLIND BORE. In that case, a cylinder group may contain many cylinders. This case is handled by performing the analysis of step (d) in bottom-up manner.

Features in group (B) are actually possible to detect automatically in some cases. However, in the current state the recognition has to be triggered by user. One consideration is that the workpieces often are results from casting process. There is no distinction in the solid model whether negative volumes in the workpiece are intended to be machined or not. If the recognition is to be done automatically all of them would be processed by the algorithm, which would output incorrect results. Another consideration is that those features are often 'not perfect', for example because of intersecting features or because the model was not correctly made in the first place. In that case, inputs from the user are required to correctly identify the features.

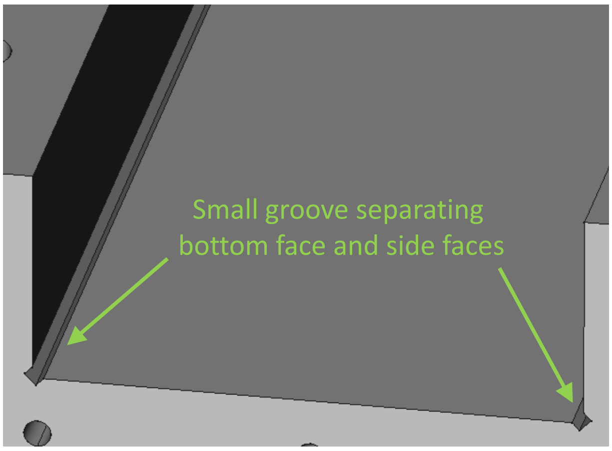


Figure 2: An example of improper slot in solid model

The user triggers feature recognition by adding one empty feature data. User then needs to determine the bottom face of the feature by selecting the face in the visualization panel. The rest is handled by the algorithm in similar manner as the algorithm for feature recognition of group (A), but this time the identification (step (c) and (d)) is performed on faces directly instead of cylinders. For some difficult cases, user may need to supply more than the bottom face. For example, Figure 2 shows an example of improper slot, where the bottom face is somehow not directly connected to the side faces. In that case, the user may supply more data by selecting the side faces as well. This kind of user interaction may go as far as the user specifying a specific feature type, in which case the algorithm is forced to use that specific type.

Feature types in group (C) are identified in a manner similar to recognition for group (C). However, because the feature shape is not regular, one of the parameters is a polyline which defines the shape of the irregular sides. While the algorithm for group (C) works for some cases, in the current state it is still considered experimental.

In all cases, parameter values, including the location of feature instances, are editable. It means the user can easily change or adjust the values after automatic recognition by the FR algorithm.

## 2.2 Software Architecture and User Interface

The feature recognition algorithm is implemented as a computer software. Input for the software is a solid model of a workpiece in common standard IGES format. This format is selected because of it is already supported widely by many CAD applications, so that models created by different CAD systems can be used as the input as long as they can be translated to IGES format.

The core feature recognition algorithm is written in C++ language as a dynamic library, utilizing the OpenCascade library for data extraction and operations in geometrical and topological levels. The user interface is developed as an individual module in open source FreeCAD software. It is written in Python language and Qt library. The database is implemented in open source Firebird database management system. Data exchange between the core algorithm library and the Python user interface is carried out in XML format. The use of open source (or at least free) libraries and software in the software development allows us to maintain flexibility of the licensing and developments in the future.

Figure 3 shows the main user interface of the software. The *main panel* displays the visualization of the workpiece model along with visualization of all recognized features. The recognized machining features in general are displayed in green, while red and purple colors denote the features that are currently selected by the user. The panel on the right side, namely *work panel*, contains detail data and information regarding features. It shows the list of all recognized features along with their respective type, parameters, and locations of instances. Changes of values by the user are reflected immediately in the visualization of the feature. In the work panel, user can also view and edit the coordinate systems data, mainly shifting or changing the origins. In addition, user can also add, edit, and delete records of standard data, e.g. standard data used to define TAP feature. The output of the FR software is list of feature instances and all their data, to be passed on to subsequent subsystem in CAD/CAM toolchain.

## 3. EVALUATION OF SOFTWARE PROTOTYPE

To demonstrate and evaluate the effectiveness of the developed solution, we test the software prototype against a few solid models of real workpieces used by industries. The first model, shown in Figure 4, contains many cylindrical features (group (A)) of various types and sizes. Some of them are intersecting features. The result shows that most cylindrical features can be identified correctly, including the intersecting features. This test also shows that features in inclined planes can be recognized as well. In that case, the inclined coordinate system is also added to the system.

The second test model contains, in addition to cylindrical features, three LONGHOLEs and two STEPs, as shown in Figure 5. The recognition actions of those non-cylindrical faces are triggered by the user, as explained in previous section. The test shows that the software is able to recognize those two features and their parameters. They include the angles of the LONGHOLE features. The LONGHOLE features are identified at the TOP plane, while the STEPs are at the BOTTOM plane.

The third test is to confirm the ability of the software in handling improper features, as shown in the example in Figure 2, where the user has to supply additional hints to the system.

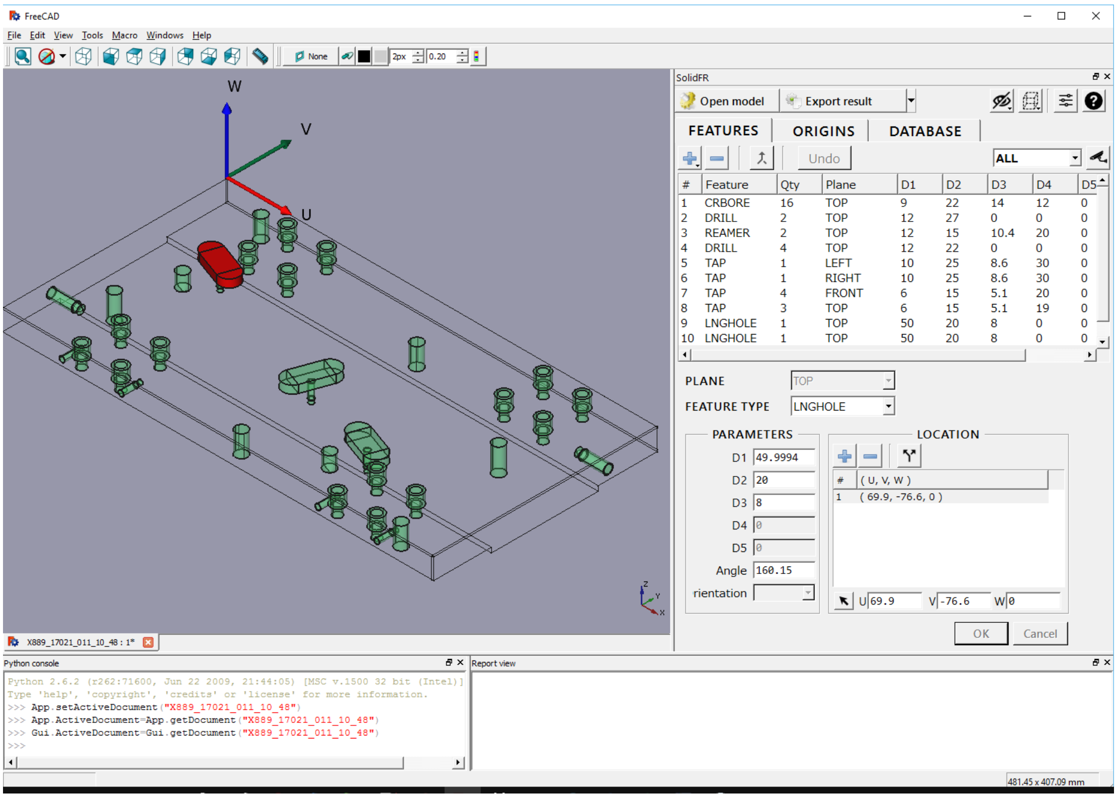


Figure 3: The main user interface

The recognition result, as shown in Figure 6, shows that the SLOT feature can be recognized by the software. In that case, additional data, in the form of bottom face and side faces, are supplied by the user.

## 4. CONCLUDING REMARKS

Feature recognition is an important step within the CAD/CAM integration framework. The goal is to define the machining features that will be processed in subsequent subsystems. This paper laid out the development of algorithms and software that have to work well in practice. For that reason, this research does not attempt to delve deeply into the theoretical domain of Feature Recognition. Instead, this research tries to address the real problems faced by industry in a simple yet effective way. To realize that, this research develops algorithm that use rules with combined hints extracted from the solid model and hints given by the user. The tests, although in this stage is still not comprehensive, have shown that the software prototype shows promising results.

Future work will be directed toward the improvement of the algorithms and software to be more robust. More comprehensive experiments are needed to demonstrate the effectiveness of the solution. Finally, the solution will be integrated with solutions in downstream stages of CAD/CAM toolchain.

## ACKNOWLEDGMENTS

This research was funded in part by the PUPT grant from Ministry of Research, Technology, and Higher Education of the Republic of Indonesia.

## REFERENCES

Babic, B., Nesic, N., & Miljkovic, Z. (2008). A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry, 59*, 321-337.

Di Stefano, P., Bianconi, F., & Di Angelo, L. (2004). An approach for feature semantic recognition in geometric models. *Computer-Aided Design, 36*, 993-1009.

Han, J., Pratt, M., & Regli, W. (2000). Manufacturing Feature Recognition from Solid Models: A Status Report. *IEEE Transactions on Robotics and Automation, 16*(6), 782-796.

Isnaini, M. M., & Shirase, K. (2014). Review of Computer-Aided Process Planning Systems for Machining Operation - Future Development of a Computer-Aided Process Planning System -. *International Journal of Automation*

*Technology, 8*(3), 317-332.

Rameshbabu, V., & Shunmugam, M. (2009). Hybrid feature recognition method for setup planning from STEP AP-203. *Robotics and Computer Integrated Manufacturing, 25*, 393-408.

Sundararajan, V., & Wright, P. (2004). Volumetric feature recognition for machining components with freeform surfaces. *Computer-Aided Design, 36*, 11-25.

Vandenbrande, J., & Requicha, A. (1993). Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(12), 1269-1285.

Wang, Q., & Yu, X. (2014). Ontology based automatic feature recognition framework. *Computers in Industry, 65*, 1041-1052.

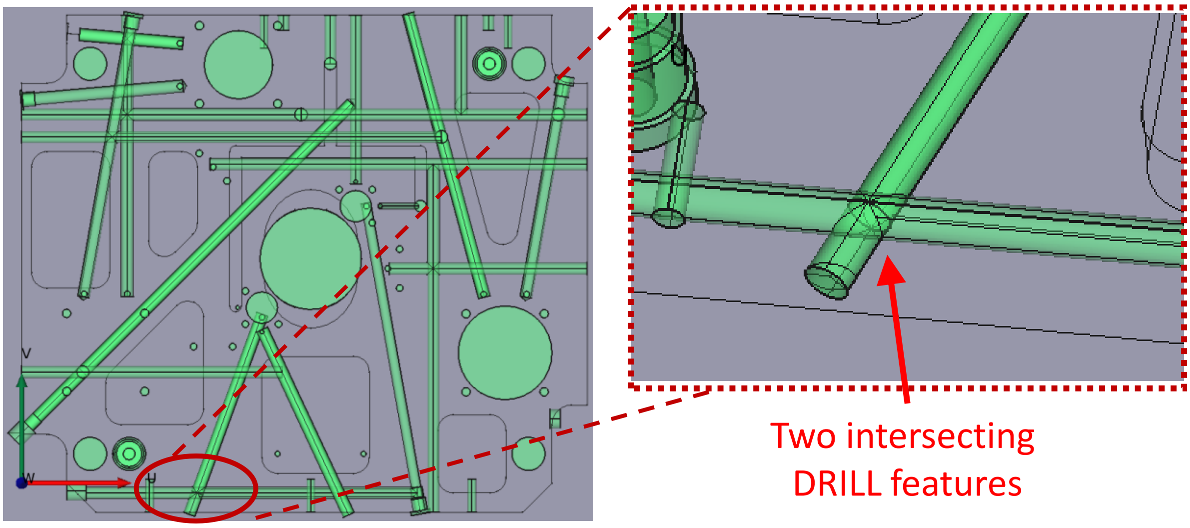


Figure 4: Test model 1

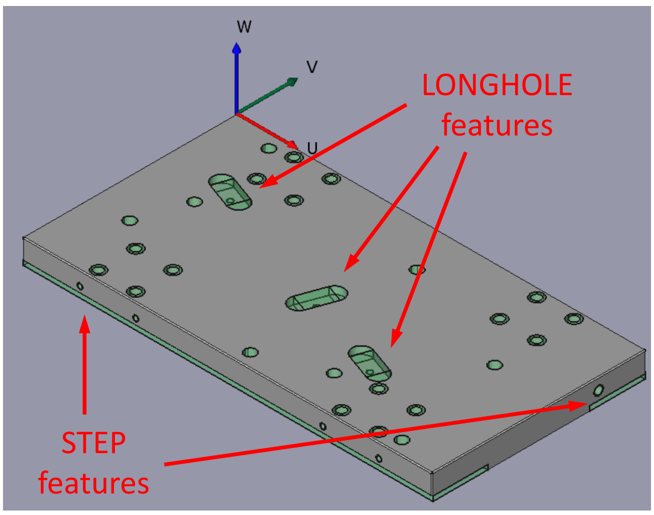


Figure 5: Test model 2

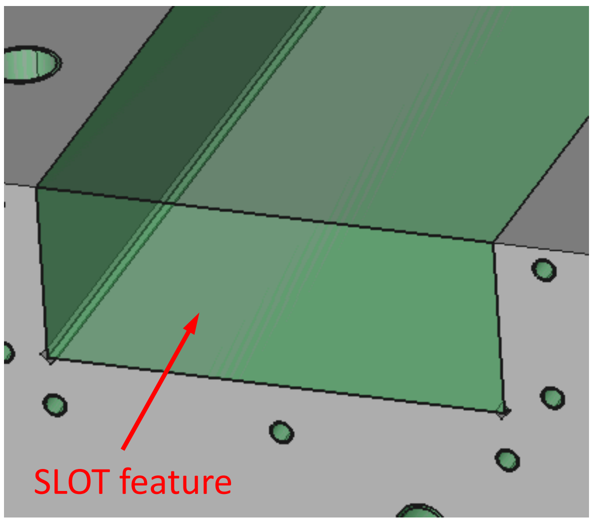


Figure 6: Test model 3