Proceedings of 10th International Conference on Precision, Meso, Micro and Nano Engineering (COPEN 10)
December 07 – 09, 2017, Indian Institute of Technology Madras, Chennai – 600 036 INDIA

Automatic Feature Recognition from STEP Files using JSDAI for Enhanced Feature Recognition, Locating Surface Selection and Tool Path Generation

Anirudh Mohan Vaniyam Valappil [1], **Manu R.** [2]
[1, 2] National Institute of Technology Calicut, Department of Mechanical Engineering
673601, India

Abstract

Automatic Feature Recognition (AFR), involves reading a CAD file to find information about the features represented in the CAD model Fixturing affects significantly the quality, the cycle time of production and the total cost of manufacturing a product. The design and verification of fixtures by digital means has become necessary in reducing lead time. Successful implementation of Computer Aided Fixture Design (CAFD) leads to the integration of the fixture design process to a Computer Integrated Manufacturing (CIM) environment.

Concurrent engineering (CE) has become the preferred business and engineering methodology thanks to its many advantages. An important aspect of modern CE systems is Computer Aided Process Planning (CAPP) wherein work of planning the part manufacture is taken over by computers. Most CAD systems do not lend themself well in CAPP environments as data used in these systems is design data which has to be translated to manufacturing data for CAPP. Creation of manufacturing data in part program form requires a lot of effort. Hence to facilitate CE, the design data may be read and interpreted by an AFR program and the recognized feature data may be translated to machine code by it.

The objective of the current project is to apply AFR to extract feature information from STEP based CAD models. The extraction of complex intersecting features will also attempted. The extracted feature information is then used for suggesting locator positions for fixture design and generating NC codes for the machining of parts by a Java program using JSDAI.

Key words: Automatic Feature Recognition/Extraction, AFR, CAE/CAx, STEP, JSDAI

1. INTRODUCTION

The term 'Feature' simply refers to a prominent attribute or aspect of a part. Every feature of every part has its own property and uses. It is a feature that converts a part into a useful product. Feature Recognition is a process of extracting relevant data to interpret component features by reading a Computer Aided Design (CAD) file which contains the digital model of the part under consideration. Automatic Feature Recognition (AFR) is a process in which the extraction and interpretation procedure is implemented through the use of a computer program. It provides an ideal solution to the problem of integrating and automating the various Computer Aided Engineering (CAE) activities. Feature recognition involves identification of higher level design and manufacturing features such as slots, holes etc., and lower level geometric features such as faces, planes, edges, vertices etc. The extraction and recognition of features from a CAD model is done from the lower geometric feature level. It facilitates Computer Aided Process Planning (CAPP) which links CAD and CAM as in addition to the common automated design and manufacturing activities, other activities can be enhanced such as preparation of cost estimates without much guesswork, expedited fulfillment of quote requests, quick generation of inspection routines for computer driven coordinate measuring machines, etc. Other analysis and planning applications will similarly benefit. Automatic Feature Recognition has grown to be a hot topic for manufacturing industries employing Computer Aided Manufacturing (CAM) approach. The process involves reading the data contained in a file containing the CAD model of the part using predefined algorithms executed by computer programs to identify, group and combine relevant parts of the data to generate information regarding all the important features of the part required for any Computer Aided Engineering application.

A fixture is simply a mechanism used in manufacturing perform the triple function of holding the workpiece, positioning it correctly with respect to a machine tool and acting as a support for holding it during machining operations. Fixturing, being one of the earliest operations in manufacturing, affects significantly the quality, the cycle time of production and the total cost of manufacturing any given product. According to Zhang et al [1], the cost of designing and manufacturing appropriate fixtures can account for as much as 20% of the total cost of a manufacturing system and according to Nixon et al [2] almost 40% of part rejections are due to dimensioning errors attributed to poor fixture design. Fixture design process is also tedious and time-consuming. It often relies heavily on fixture design engineers' experience and knowledge along with a lot of trial and error practice. Further, the designing and manufacturing fixtures can take several days or even longer to complete by traditional methods. In order to overcome these difficulties, research on the use of computers to design and manufacture fixtures has been carried out. The advances in this field has led to the automation of the process of fixture design and manufacture through the use of computers and this process is commonly referred to as Computer Aided Fixture Design (CAFD). With the advancement of CAE technologies, the design and verification of fixtures by digital means has become an absolute necessity in reducing the lead time involved in the process. Successful implementation of CAFD will lead to the integration of the fixture design process to a Computer Integrated Manufacturing (CIM) environment. As the fixtures needed for nearly geometrically similar parts vary greatly depending on even small feature details, the extracted feature information provided by AFR is of great use in CAFD. This is true even for single fixture setups where the locating and clamping surfaces are to be selected correctly to avoid movement and/or damage to the component being machined.

Concurrent engineering (CE) refers to an approach used in product development where the functions of design engineering, manufacturing engineering and other functions are integrated i.e. design, development and manufacturing stages of products are run simultaneously, rather than consecutively. CE, while being challenging in its initial implementation phase, has become the preferred business and engineering methodology thanks to its advantages such as reduced marketing time and development costs and better quality products that are produced at the highest

ISBN: 978-93-80689-28-9

levels of productivity and has been replacing traditional sequential engineering in more and more organizations. An important aspect of modern concurrent engineering systems is Computer Aided Process Planning (CAPP) wherein the work of planning the production of a part from the design to the manufacture stage is taken over by a computerized system. CAPP is the link between design and manufacturing in a CIM environment. However most CAD systems do not lend themself too well in a CAPP environment as the data used in these systems is design data which has to be translated to manufacturing data for process planning. The creation of manufacturing data in the form of a part program requires a lot of time and complex decision making. Hence to facilitate concurrent engineering the design data may be read and interpreted by an Automatic Feature Recognition program and the recognized feature data may be translated to machine code by it. The AFR program may also facilitate CAPP by using the extracted feature information for CAFD, bringing with it, its own set of advantages in a CIM environment as explained earlier. This helps to completely automate the CAPP process, bringing with it all the advantages of CE.

Most of the earlier efforts in AFR were concerned with extracting data from IGES files. Liu et al. [3] extracted features of a prismatic part using the CAD data stored in IGES file format. The AFR system that was developed used an object-oriented data structure designed to allow the modeling and manipulation of complex parts using a C program for an IGES file. The system used a methodology composed of three algorithms that translated the data in a CAD file into manufacturing features that could be used in CAPP.

Sharvan Kumar and P. K. Jain [4] created an AFR system named PRIZCAPP which could extract holes, slots and pockets from a wireframe model made in AutoCAD. It could recognize features like slots, steps, holes and pockets of different profile geometries by matching it with a basic hole feature.

In the late 90s, ISO 10303 known as STEP (STandard for the Exchange of Product model data) came into prominence as the go to neutral file format for Computer Aided Engineering applications. STEP file was meant to be used for the representation and unambiguous exchange of computer-interpretable product information throughout the lifecycle of a Product. STEP was originally developed by the International Standards Organization ISO, under Technical Commitee-184, Sub Commitee-4 with the official title: “Automation Systems and Integration, Product Data Representation and Exchange” and informal Title: “Standard for Exchange of Product model data” and acts as a Neutral file format for all CAE systems. A STEP file consists of several parts which can be, at the simplest level, grouped into three categories. The content or description method of the file is represented using EXPRESS, the grammar and semantics of which are both described and standardized in STEP.

Bhandarkar and Nagi [5] developed a feature extraction system that takes as input, a STEP file defining the geometry and topology of a part and generated as output, a STEP file with form-feature information in AP224 format for form feature-based process planning using object-oriented modeling principles and C++ programming language. The algorithm was developed for prismatic solids produced by milling operation.

Rameshbabu and Shunmugam [6] proposed a hybrid approach that effectively uses volume subtraction and face adjacency graphs to recognize manufacturing features from 3-D model data in STEP AP-203 format.

However, these papers suggest that there is still a need for more improved and advanced AFR methods and its application to CAFD and CAM. There is tremendous scope for improvements which could result in increasing the efficiency of feature extraction for application in design of fixtures and manufacture of the component, thereby reducing the overall manufacturing cost and time.

2. OBJECTIVES AND METHODOLOGY

The primary objective of the current project is to apply the AFR technique to extract feature information from STEP based CAD models. The extraction of complex intersecting features was also attempted. A program developed in Java using the Java Standard Data Access Interface (JSDAI) toolkit for the Java Eclipse platform was used to achieve the research objectives. Unlike most AFR programs based on neutral file formats which use simple string searching algorithms to perform feature recognition, JSDAI converts the STEP file entities into Java classes and objects which enables easier access to the geometrical entities represented in a STEP file and enables more complex manipulations and operations to be done with the files. The scope of the present work is limited to prismatic components manufactured in a single fixture setup with features consisting mostly of holes and pockets in order to keep the code as simple and streamlined as possible while being capable enough to recognize maximum possible combinations of features. The final version of the program should be capable of achieving the project objectives without being overly complex in its logic or structure. The tighter focus on the type of the features recognized would allow for the recognition of holes of any radius and depth, cavities of any number of dimensions and depth and most importantly, for the recognition of nearly all possible combinations (intersections) of holes and cavities of any number and level (complexity) of intersection. The accurately recognized feature data could act as groundwork for achieving a full-fledged CIM environment.

As secondary objectives, the use of the extracted feature information for suggesting locator positions for fixture design using minimum clamping force criterion was attempted. Using the accurate data made available by achieving the primary objective, locating surfaces could be easily recognized regardless of the complexity of the features present on the component. The generation of the NC codes required to manufacture the component was also attempted. The final version of the program must be capable of generating a .NC file containing the G codes needed to accurately manufacture the component with minimum idle movements while taking all practical machining considerations into account.

3. FEATURE RECOGNTION

After the component was created in a CAD software, it was saved as a STEP AP 214 file in a specified directory. The path to this file was specified in the program. When the program is run, it starts finding all Advanced Faces with a plane geometry i.e. component surfaces in each direction, whose surface geometry is a plane and stores them in three separate arrays named Facesps depending on the directions of their plane normals. After this each of the arrays are sorted separately based on the distance of each plane from their respective parallel planes at the origin. This is necessary as the planes appear at random entity numbers in the STEP file and there in no arranged sequence to their placement in the file. Then the first and final member of each of the three Facesps arrays are stored in a separate array designated as Facesmin and Facesmax for each direction that indicates closest and farthest planes of the component from origin in that direction. The program then calculates and stores the distance between Facesmin and Facesmax in each direction as the dimension of the stock in that direction. This algorithm in addition to finding component dimensions, also helps to establish permanent references in each direction in the form of Facesmin and Facesmax which will be of use in the recognition of every feature that exists on thee component. Then all the Advanced faces with a cylindrical geometry are identified and their position, depth and diameter are recorded. Intersecting and non-intersecting holes are identified during this part of the program. Next, the Facesps array is iterated through to find those Advanced faces with bounds that are not completely circular to find cavities. Those without any circular bounds are directly recognized and their position, depth and dimensions are noted. Intersecting cavities are also distinctly recognized in this step as they belong in different Facesps arrays. If the bound is a combination of both circular and linear bounds then it is recognized as a separate element and classified in to one of the three cases of hole-cavity intersections and their feature information is recorded separately.

4. FIXTURE DESIGN

Degrees of freedom of an object refer to the number of distinct independent motions that the object can exhibit in its current configuration. An object in space has a total of 12 degrees of freedom; 6 rotational and 6 translational. For proper machining, the work piece must be held in a pre-defined manner against the tool so that only the required relative motion occurs between the tool and the workpiece. This is done by using appropriate fixtures in which all the required degrees of freedom of the workpiece are restricted as per the requirements of the machining operation. The devices that restrict a work piece's movement in a fixture are called locators. Locators act as a positive stop for the workpiece against which motion does not occur. For a particular machining operation, using minimum number of locators, the required degrees of freedom of the workpiece must be restricted. This is ensured by following the 3-2-1 principle. Six locators are used which restrict nine degrees of freedom of the workpiece. The remaining three degrees of freedom will be restricted by the clamps. Clamps do not act as positive stops, instead hold the work in place by virtue of friction acting between the clamp surface and the clamped surface. It must be always ensured that the largest of the forces that occur during machining in any direction are absorbed by the locators and not by the clamps as it leads to the requirement of greater clamping forces which is not recommended. In the program, the Primary locating surface is chosen as one of the faces perpendicular to axis of the features on which there are lesser number of bounds. The pairs of surfaces are chosen based on the relative areas of the surfaces. The program then determines the relative torque produced during machining to create each feature. This torque is then used to obtain the magnitude of forces that must be applied at each surface to arrest motion. The surfaces in each direction requiring maximum force are chosen as the locating surfaces in order to minimize the clamping force needed. The program then designates these locating surfaces by using distinct colours (Green for Primary, Violet for Secondary and Yellow for Tertiary) on three output STEP files for the understanding of the human operator or in case of a fully automated system, simply passes the information regarding the identified surfaces to the CAPP system.

5. NC CODE GENERATION

For the generation of NC codes, during program execution, every important coordinate of every recognized feature such a hole centres, cavity bounding coordinates etc., were uniquely stored during the recognition process itself. The code then generates a file with the extension NC and then writes appropriate G codes for the creation of the recognized features. No canned cycles were used directly in the code to maintain the universal nature of the code as all canned cycles are not supported by every CNC controller.

For the generation of holes, a peck drilling method was adopted. In this method, the entire hole depth was split into ten smaller incremental depths. Once the drill is positioned over the hole centre, it drills into the stock removing material up to a depth equal to one-tenth of the hole depth. Then, it retracts to a plane at a height 1 unit above upper stock surface to remove the material from the drill and also to allow coolant entry. Then it goes down once again, and drills to a total depth of two-tenth of the total hole depth, removing material from an additional one tenth. This procedure is repeated until the complete hole is made. For the generation of cavities, initially a zig zag tool path was followed in which the tool moved up by a distance equal to 75% of the tool diameter for every horizontal linear motion. The diameter of thee tool was selected as one-fourth of the width of the cavity. The tool motions were further staggered by a distance equal to half the tool diameter in either direction on each side to simulate cutter compensations. Any intersecting features would also be generated properly as the code has distinctly recognized these features into a combination of holes and cavities rather than an unknown shape.

6. CASE STUDY

A case study was conducted using a relatively complex component consisting of multiple intersecting features. It requires a stock with minimum dimensions of 30, 20 and 10 units. It has three holes and three cavities of unique dimensions. One of these holes is a through one and intersects a blind hole. The other blind hole intersects a blind cavity and a through cavity. The through cavity also intersects the second blind cavity. The component along with its detailed dimensions are shown in Fig. 1.

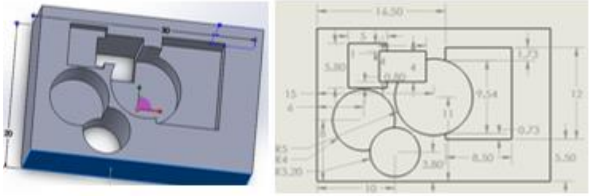

Fig. 1. Component for case study

Recognition of such features would be nearly impossible for a common string searching algorithm alone due to the complex intersecting features present. It is very difficult to select the locating surfaces based on observation alone as the number, position and size of each feature is different relative to every other feature. The complexity of features also poses a problem for NC code generation in case of simpler algorithms due to the fact that the STEP files never directly indicate if an intersection has occurred or not. This often leads to cases where such programs generate erroneous G code or redundant code that results in unnecessary idle motions. The simple and streamlined yet capable program developed using JSDAI for the current work does not suffer from any of these issues and recognizes all the features accurately. During execution it collects and stores a large amount of data for each unique feature which is then used to achieve the secondary objectives of determining the locating surfaces and generating the NC code needed to make the component. Once the program has begun execution, it reads and interprets the STEP file and then computes and stores feature information. Using this accurate feature information, it determines the locating surfaces and indicates these locating surfaces on three STEP files having dimensions equal to stock dimensions with feature boundaries marked on them and locating surfaces colored distinctly as shown in Fig. 3. Also .NC files are generated based on the extracted feature data that contains G codes needed to manufacture the component. Finally all the required feature information is displayed as the program output based on the data collected as shown in Fig. 2.

The validity of the NC codes generated is confirmed by simulating the tool path in tkCNC Editor 3.0 which generates a visual representation of the tool path motions indicated by the G code created by the program and allows this tool path to be overlaid on the upper surface of the original CAD file for complete confirmation as shown in Fig. 4.

```
M150350ME
Required stock dimensions :30.0*10.0*20.0
LOCATING SURFACES IDENTIFIED

Feature:1

Blind Hole of radius=4.000000000000001and Depth=6.0
At a distance = 6.001153130530721 from Tertiary Locating surface
At a distance = 8.0 from Secondary Locating surface

Feature:2

Blind Hole of radius=5.000000000000001and Depth=8.0
At a distance = 15.0 from Tertiary Locating surface
At a distance = 11.0 from Secondary Locating surface

Feature:3

Through hole of radius=3.1999999999999984
At a distance = 10.0 from Tertiary Locating surface
At a distance = 3.8 from Secondary Locating surface

Feature:4
Blind cavity of dimensions=
0.7303039929152728 * 8.5 * 12.0 * 8.500000000000004 * 1.730303992915271 * 9.539392014169458
With Depth= 4.0

At a distance = 20.75 from Tertiary Locating surface
At a distance = 11.5 from Secondary Locating surface

Feature:5
Blind cavity of dimensions=
1.0 * 0.9999999999999964 * 5.0 * 5.800000000000001 * 4.999999999999999 * 0.8000000000000007 * 1.0 * 4.
With Depth= 2.0

At a distance = 6.5 from Tertiary Locating surface
At a distance = 15.1 from Secondary Locating surface

Feature:6

Rectangular Through cavity of dimensions=
4.0 * 6.0
At a distance = 11.0 from Tertiary Locating surface
At a distance = 15.0 from Secondary Locating surface
COLOUR CODED STEP FILES GENERATED
NC CODES GENERATED
```

Fig. 2. Program output

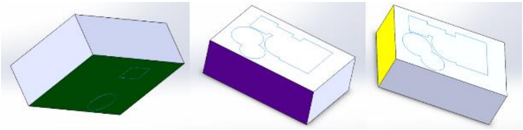

Fig. 3. Locating surfaces

7. CONCLUSION

In the current work, Automatic Feature Recognition from STEP files has been attempted using a Java program written with the JSDAI API in order to determine stock dimensions, recommend locating surfaces and generate NC codes for manufacturing the component for a single fixture setup in which the component has features limited to mostly holes and cavities. The main contributions of this research have been

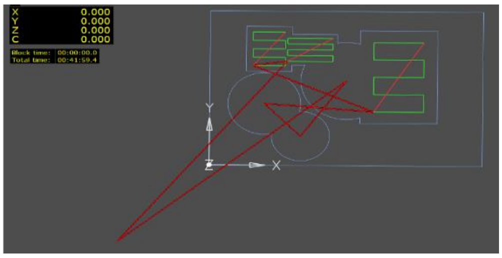

Fig. 4. Tool path simulation

- Providing a simple but capable code for determining stock dimensions and reference surfaces
- Developing an efficient and compact code for recognizing simple as well as intersecting holes simultaneously without additional work
- Developing a cavity recognition code that is capable of detecting any feature formed by any combination of line segments and also recognizing complex intersecting features to a certain extent.
- Developing an algorithm and code to easily identify locating surfaces and generate NC codes to manufacture components using single fixture setups

The AFR system developed in the current work is not entirely perfect and a few modifications have to be incorporated to expand the program for greater practical application. This includes enabling recognition of non-circular curved features. Also a procedure for perfect and exact recognition of circular curved intersecting features must be added to the code. A method should be developed if possible, to color individual faces of a STEP file component uniquely

4. REFERENCES

[1] Zhang, W,J., and Bi, Z,M., “Flexible fixture design and automation: review, issues, and future directions”, International Journal of Production Research, Vol. **39**, 2001, pp. 2867–94.

[2] Nixon, F., "Managing to achieve quality", McGraw Hill: Maidenhead,1971.

[3] Liu,S., Gonzalez, K., Chen, J., 1996, “Development of an automatic part feature extraction and classification system taking CAD data as input,” Computer in Industry, **29**, pp. 127-150.

[4] Kumar, S., Jain, P,K., 1998, “Automatic feature extraction in PRIZCAPP,” International Journal of Computer Integrated Manufacturing, **11**, pp. 500-512.

[5] Bhandarkar, M , P., Nagi, R., 2000, “STEP- based feature extraction from STEP geometry for agile Manufacturing,” Computer in Industry, **41**, pp. 3-24.

[6] Shunmugam, M, S., and Rameshbabu, V., “Hybrid Feature Recognition Method for Setup Planning from STEP AP-203”, Robotics and Computer Integrated Manufacturing, Vol. **25**, 2009, pp 393-408.