FEATURE RECOGNITION-BASED PROCESSING OF STEP FILES

Viktor TANCSA [1)], Patrik Dominik PÖRDI [1)], Dániel BERENCSI [1)], Lajos KUTROVÁCZ [1)]

[1)] *Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Hungary*

Abstract

The purpose of this paper is to present software that provides machining time prediction by the means of feature recognition processing STEP files in an object-oriented Python environment. In the theoretical part, an introduction is given to the structure of STEP file format, the most common file format of the known CAD systems. The way of recognizing simple and more complex features using regular expressions will be shown. In the practical- based section, the developed algorithms for the exploration of the geometric properties and features that deliver databases that are essential to recognize machining features will be detailed. Using the prepared data, the material that needs to be removed and the type of machining will be given and using an experimental model the estimated machining time can be presented.

Keywords: Feature recognition; STEP file processing; Feature similarity analysis; Machining time prediction

1. Introduction

Today, the biggest industry challenges include maximizing speed and efficiency, including minimizing downtime and manufacturing errors and automating processes. This is a multi- factor issue as it is influenced by many people, disciplines, and environmental conditions. The entire life cycle of a product from the onset of demand to the start of production is a long and resource-intensive iteration process along with this issue. In the first step, the product designer creates a 3D model, and then the manufacturing engineer checks the manufacturability, designs the required operating elements, and lists the equipment required for the specific machining. It can be seen that even between these two areas, conditions, contentious situations and recurring problems arise that require constant consultation. It is often the case that a given product cannot be manufactured or, with existing infrastructure, production is too costly, inefficient, or unfeasible. As a result, the product may have to be redesigned several times, or the expansion or modification of the equipment may be considered. This is no longer an easy issue to deal with, but if we look more closely at things, we must not forget the management side, including managers, economic analysts, and decision-makers, who can override any decision given the cost implications of change. Their goal is to maximize the expected profit, which means minimizing production time and costs incurred without compromising the quality of the end product. The common goal of the participants is to find the cheapest, and usually the most automated alternative that best meets their needs. Achieving this goal requires compromises on all sides, in the development of which it is essential to overcome the communication difficulties arising from the professional qualifications of the various participants.

In 21st centuries manufacturing, it is unavoidable to speak about Industry 4.0 and 5.0. Former means the implementations of highly developed, fully automated manufacturing systems, that are capable to track and simulate the whole journey of goods, furthermore, the machine-learning algorithms and the big data analysis can be integrated as well. Latter put emphasis on merging human creativity and craftmanship with the provided developed manufacturing systems. To sum up the main goal of manufacturing in the upcoming decades, to find all of the steps of the production cycle that are not automated yet but capable for that to minimize human intervention, therefore increase consistency and sufficiency. Moreover, traverse the human resources in directions that can lead to a more sustainable and ergonomic end product.

The software that will be presented is a possible example. Its aim is to provide a bridge between the fields that are involved in the planning of serial productions, therefore simplify the presented process. The idea is basically to find the machining features and count the amount of material to be removed. The former carries the type of machining and by the means of a database based on previous manufacturing information an experimental model can be defined and the machining time and other pieces of information, the expected cost, for example, can be predicted. Therefore, many unnecessary steps can be avoided. For instance, the different versions of a product can be compared. The manufacturing engineer gets a simplified list of operations, while the management side gets economic and logistical information, thus the optimal solution can be found, and each part can start to concentrate on that one.

2. The basics of step files

The CAD systems provide different file formats, depending on the purpose of use and the required details. Types as IGES or STL deliver mostly geometric information, whereas STEP contains additional manufacturing information, tolerances for instance.

STEP is known as "Standard for Exchange of Product model data", the ISO10303 is the ISO standard for mechanical STEP parts. For different applications, different protocols are provided. Jimmy N. [1] described some, such as AP214 which contains core data for automotive mechanical design processes, or AP242 that is used in managed model-based 3D engineering.

The STEP files have a boundary-based, so-called B-Rep structure, which means the boundary entities are stored in such topological order where the connection between them is represented as geometric definitions. In other words, if we aim to process a STEP part from its faces through its edges to its vertices, then surfaces, curves, and points need to be dealt with. According to Bhandarkar and Nagi [2] and Malleswaria [3], this topological order and the relations can be represented in a tree structure, see in (Figure 1). The files are stored in highly structured text format, therefore nested loops of regular expression need to be used during processing.

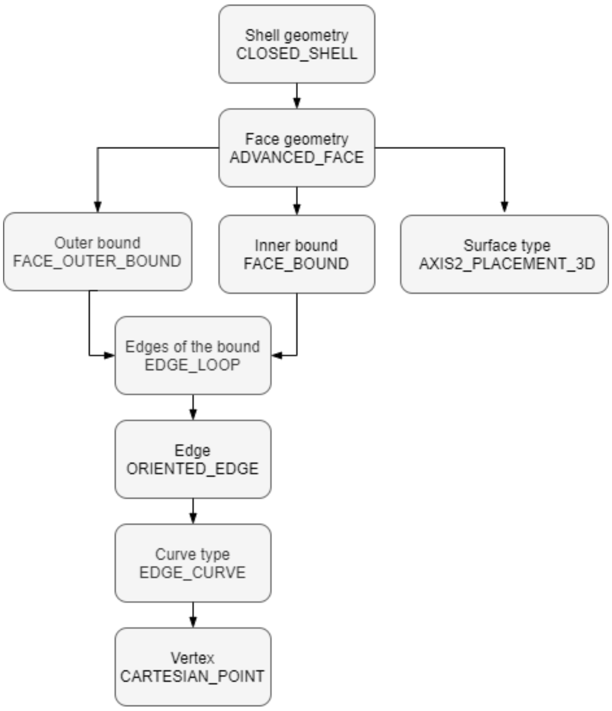


Fig. 1.: The topological structure of STEP

Every solid geometry stored in STEP can be processed using its detailed boundaries, as you can see above. The algorithm needs to follow the presented hierarchical path of strings, it is shown in (Figure 2), to get the necessary information of the solid, from its shells to its vertices, to be able to use the file for any purposes. Most of the entities have additional attributes besides the connecting elements, usually, Boolean flags to orient them, as a result, closed chains are given, it was also stated by Bhandarkar, Nagi.[2]

```
#15=FACE_BOUND('',#35,.T.);
#16=FACE_BOUND('',#41,.T.);
...
#24=FACE_OUTER_BOUND('',#32,.T.);
...
#32=EDGE_LOOP('',(#101,#102,#103,#104));
#33=EDGE_LOOP('',(#105,#106,#107,#108));
...
#42=LINE('',#218,#56);
...
#70=CIRCLE('',#165,3.5);
#72=CIRCLE('',#168,2.5);
...
#83=VERTEX_POINT('',#244);
...
#98=EDGE_CURVE('',#82,#83,#53,.T.);
#99=EDGE_CURVE('',#83,#77,#54,.T.);
...
#132=ORIENTED_EDGE('',*,*,#96,.T.);
...
#140=ADVANCED_FACE('',(#28),#19,.T.);
#141=ADVANCED_FACE('',(#29),#20,.T.);
...
#144=CLOSED_SHELL('',(#136,#137,#138,#139,#140,#141,#142,#143));
```

Fig.28. - Part of a STEP file

3. Features

If we are talking about 3D geometries the features can be described in several ways. Nasr and Kamrani [4] emphasized their significance to a specific application. According to Venu and Komma [5] in the process of planning, we need to look for higher-level features such as holes, pockets, chamfers, etc. In STEP they are represented by lower-level features, those are 2D and 3D entities, the hierarchy can be seen in the previous section.

Srivastava and Komma [6] presented the product data model, which contains information, that is needed by the CAM and CAPP system, including geometry, tolerances, processes, and numerical control information, etc. The higher-level features are named manufacturing features.

Bhandarkar and Nagi [2] summarized the work of Shah and Rogers [7]. The features are classified related to product engineering applications. The manufacturing features are called form features, a set of surfaces, a combination of geometric and topological entities, made during design, by knowing them the type of manufacturing operation like drilling, turning or milling can be concluded.

Venu, Komma, and Srivastava [8] also stated, that lower-level features and their geometric and topological connections interpret the higher-level features. Therefore, we need to go down into details to be able to use the given information and be able to reconstruct the solid. The classification of Bhandarkar and Nagi [2] is based on the effect of the form features in changing the fundamental shape of the solid. There are four main groups

- Void features: Voids are enclosed within the outer shell.
- Internal form features: They alter the internal shape of a surface, depending on the connection to other faces they are blind or through internal form features.
- External form features: The external shape of the solid is modified, using edge, face, or vertex modification.
- Connectivity modifying form features: They alter the link between faces, chamfers, and rounded edges for instance.

Feature recognition-based processing of 3D solids is a highly popular topic. Malleswaria [3] defined the main areas including Syntactic pattern recognition, Rule-based recognition, Hint based recognition, Graph-based recognition, and Artificial neural network systems-based recognition. Rameshbabu and Shunmugam [9] defined a hybrid feature recognition method that uses a face adjacency graph and subtracts the volume. The authors of [10] investigated methods of recognizing freeform machining features by the means of manufacturability analysis. Bhandarkar and Nagi [2] used a STEP-based feature extraction algorithm for prismatic solids which simplify the B- representations.

Feature recognition

Bhandarkar and Nagi [2] investigated the problem and explained the solution deeply. To recognize features in the STEP we have to look for the previously mentioned cornerstones and the presented tree structure.

First, we need to find a closed shell and start to investigate its faces, each has face normal, outer boundaries, and optionally inner boundaries, which would mean the existence of an internal form feature. Those are closed edge loops, which are divided into oriented edges. Those edges lay between their start and end vertices, and the edge curve parameter describes their type, for instance, line or circle.

Most of the levels of the tree contain Boolean operators to orient the geometries. The figure below presents a key rule of this B- representation. The bounding edges of a face are oriented so that the material will be found on its left side. Therefore, the edges that are part of the outer boundary are oriented counterclockwise, whereas the edges of the inner boundary are oriented clockwise, see (Figure 3). Due to this structure, the holes and pockets can be found, to decide wheatear they go through the material or not the relationship to other faces of the geometries, that they are determined by, needs to be looked at.

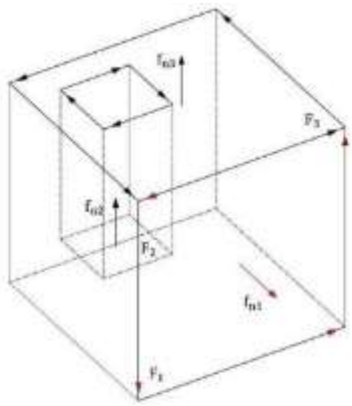


Fig. 3: Orientation of the edges and the face normal

To be able to find the external form features such as steps and slots, the convexity and concavity of their edges need to be looked at, which can be deduced from the relationship of the face normal by the means of mathematical equations. Depending on the complexity of the feature the exact calculation can be difficult. Malleswari [3] stated the key rules of detecting cylindrical surfaces see in (Figure 4):

- Rule 1: EDGE_CURVE construction must be circle, line, circle, line (or) line, circle, line, circle
- Rule 2: Radius of circles in EDGE_CURVE is equal to cylinder radius.
- Rule 3: Two coordinates of circle centers are the same and the third coordinate is different.
- Rule 4: Surface must be CYLINDRICAL_SURFACE.

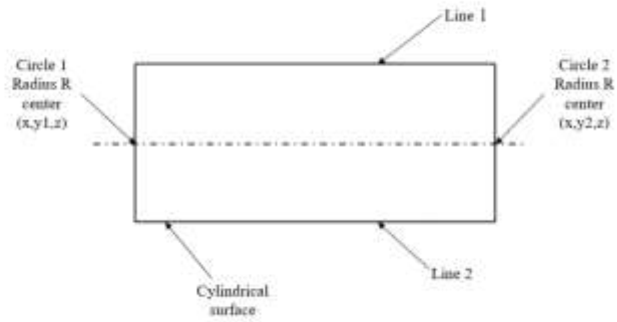


Fig. 4.: Cylindrical surfaces

Using the face normal we are able to decide whether it is a hole and if it is the faces sharing this edge cure to categorize it into blind or through holes.

4. The created software

The aim was to collect all of the information of a STEP file, that is needed to calculate estimated manufacturing values, such as manufacturing time, manufacturing cost, etc. The program is written in Python, which is an interpreted, object-oriented, high-level programming language. We decided to use this language for several reasons. It has several existing libraries, that were beneficial to use in our case. Moreover, it has a quite simple syntax and not a complicated structure, therefore a user- friendly environment is provided to create, debug, and edit the more complicated algorithms.

To be able to extract the necessary data from STEP files the text-based information needs to be processed using regular expressions. According to Jeffrey [11], to deal with regular expressions a search pattern, a sequence of characters basically, needs to be introduced. The combination of the regular expressions and the given map presents a huge amount of data, for instance, faces, edges, vertices, and their pose. Using the above-described Boolean-based topology the faces are defined and by the means of their concave or convex edges the form features can be detected. It has been done using a matrix that presents the connection between the faces of the part. There are three occasions, they can be linked through a concave or convex edge in other cases they are not linked. In the matrix, they are marked with 1, -1, and 0 respectively. Using this matrix, the part can be reconstructed, therefore the form features such as pockets and steps can be detected. The holes can also be found using the method described above. We tested our algorithm on several parts, a good example is the one called “Cheese”, see in (Figure 5). The processed and organized data are presented in databases on the user interface.

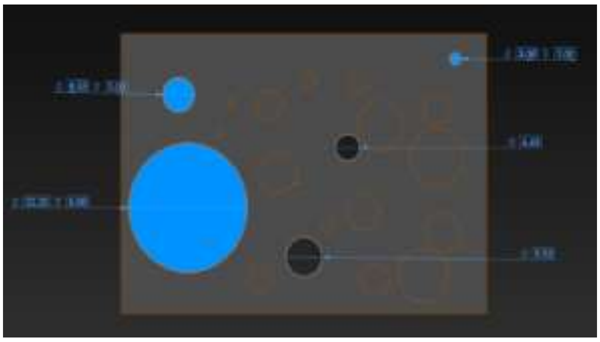
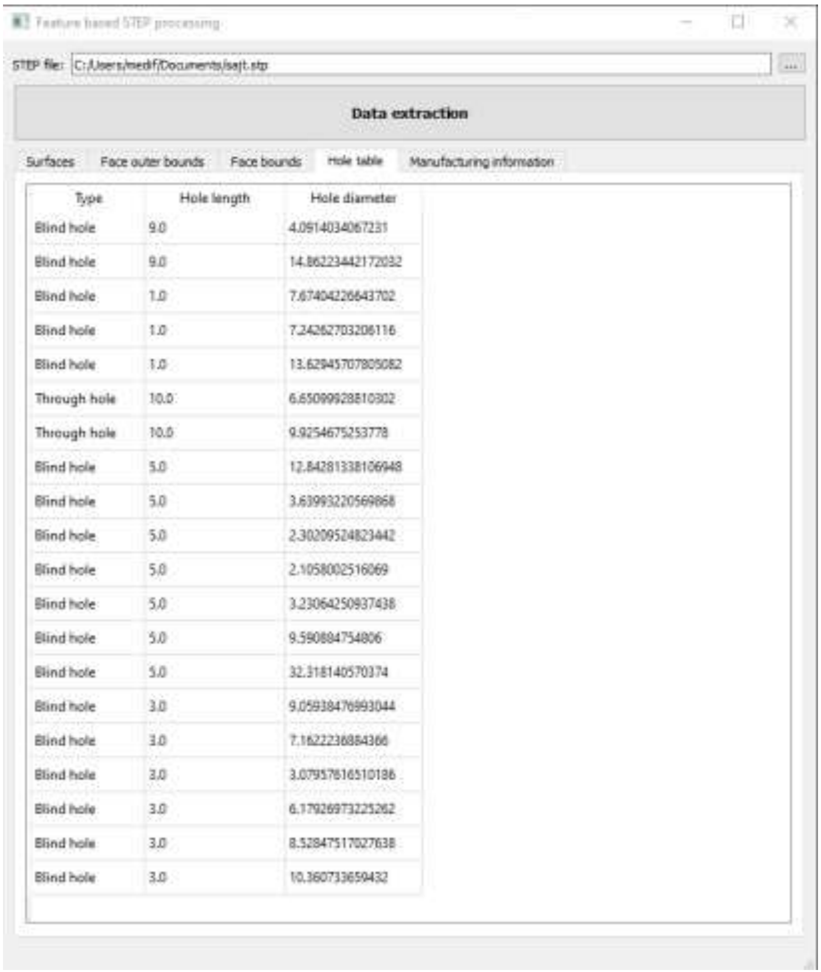


Fig. 5.: "Cheese"

The processed and organized data are presented in databases on the user interface seen in (Figure 6). The users can look at the geometrical information of the parts.



Type	Hole length	Hole diameter
Blind hole	9.0	4.0914034067231
Blind hole	9.0	14.86223442172032
Blind hole	1.0	7.67404226643702
Blind hole	1.0	7.24262703206116
Blind hole	1.0	13.62945707805082
Through hole	10.0	6.65099928810302
Through hole	10.0	9.9254675253778
Blind hole	5.0	12.84281338106948
Blind hole	5.0	3.63993220569868
Blind hole	5.0	2.30209524823442
Blind hole	5.0	2.1058002516069
Blind hole	5.0	3.23064250937438
Blind hole	5.0	9.590884754806
Blind hole	5.0	32.318140570374
Blind hole	3.0	9.05938476993044
Blind hole	3.0	7.1622236884366
Blind hole	3.0	3.07957816510186
Blind hole	3.0	6.17926973225262
Blind hole	3.0	8.52847517027638
Blind hole	3.0	10.360733659432

Fig. 6.: Database on the user interface

5. Calculation of the estimated values

According to JungHyun, Kang, and Choi [12], to estimate production time and cost, we need information on how much material and how to remove it. The three most common estimates of machining time can be made on the basis of machine tool performance and specific cutting forces, or by taking into account the cutting parameters assigned to the tool and empirical values. Of these, we dealt with the latter two areas, where we used two basic procedures. Turning and milling. We distinguish two types of this, smoothing and roughing. Roughing is volume-proportional machining while finishing is surface-proportional. For our calculations, we used the correlations characteristic of the machining technology.

We also distinguish two types of input parameters. There is something we extract and use from the software we create, but there is something we can't extract from the STEP file. Examples of such input parameters are cutting speed and feed rate. These and other non-extractable data are aggregated in a subroutine in the form of technology and tools. The tools are assigned the cutting parameter limits that are characteristic of them. These limits are used as missing input parameters and the machining time and material deposition of a surface element can be calculated. The user gets a default setting stored in a database that can be customized for the selected tool.

So first investigating the shape and the enclosing dimensions of the part the type of manufacturing, in this case, milling or turning and the additional drilling, can be decided. The raw material can be inserted into the program if not, it is calculated from the enclosing dimension. The recognized form features can be converted into volumes that need to be removed. Using the database and formulas of the type of machining an estimated machining time can be presented. Each machine has its own cost proportional to the machining time. To sum it up, the calculations are algorithmized to obtain the surface elements to be machined gives the total amount of material removed and the total machining time. From these, a rudimentary estimated production cost can be calculated. The user interface is shown in (Figure 7).

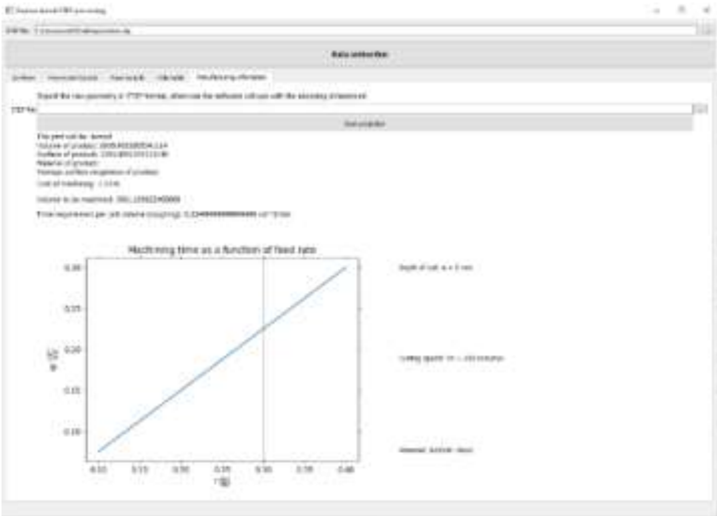


Fig. 7.: The estimated values

Conclusion

Overall, our method works, we can recognize form features, however, there are development opportunities and simplifications that need to be introduced to be able to recognize complex form features. Recognition of planar and cylindrical features is already accomplished with high accuracy, however, the combination of the two is currently still a challenge, it could be done using machine learning. Estimation of production time and the cost is currently rudimentary, here basically big data analysis is what should be implemented in the future.

Anknowledgment

The authors would like to acknowledge the support provided by the CEEPUS III HR 0108 project.

References

[1] Nguyen, Jimmy. “Best STEP File to Use: AP203, AP214, and AP242.” Accessed September 15, 2021.https://www.capvidia.com/blog/best-step-file-to-use-ap203-vs-ap214-vs-ap242.

[2] Bhandarkar, Mangesh P., and Rakesh Nagi. “STEP-Based Feature Extraction from STEP Geometry for Agile Manufacturing.” Computers inIndustry 41, no. 1 (January 2000): 3–24.

[3] Malleswari, V N. “Automatic Recognition of Machining Features Using STEP Files.” International Journal of Engineering Research 2, no. 3 (2013): 11.

[4] Nasr, Emad Abouel, and Ali K. Kamrani. Computer-Based Design and Manufacturing: An Information-Based Approach. New York: Springer, 2007.

[5] Venu, Bitla, and Venkateswara Rao Komma. “STEP-Based Feature Recognition from Solid Models Having Non-Planar Surfaces.” International Journal of Computer Integrated Manufacturing 30, no.10 (October 3, 2017): 1011–28.

[6] Srivastava, Deepanshu, and Venkateswara Rao Komma. “Development of STEP AP224 Extractor for Interfacing Feature Based CAPP to STEP-NC (AP238).” International Journal of Automation and Computing 16, no. 5 (October 2019): 655–70.

[7] J. Shah, M. Rogers, Functional requirements and conceptual design of the feature-based modeling system, Computer Aided Engineering Journal 5

[8] Venu, Bitla, Venkateswara Rao Komma, and Deepanshu Srivastava. “STEP-Based Feature Recognition System for B-Spline Surface Features.” International Journal of Automation and Computing 15, no.4 (August 2018): 500–512.

[9] Rameshbabu, V., and M.S. Shunmugam. “Hybrid Feature Recognition Method for Setup Planning from STEP AP-203.” Robotics and Computer-Integrated Manufacturing 25, no. 2 (April 2009): 393–408.

[10] Cai, Na, Soumiya Bendjebla, Sylvain Lavernhe, Charyar Mehdi-Souzani, and Nabil Anwer. “Freeform Machining Feature Recognition with Manufacturability Analysis.” Procedia CIRP 72 (2018): 1475–80.

[11] Friedl, Jeffrey E. F. Mastering Regular Expressions. 3rd ed. Sebastapol, CA: O’Reilly, 2006.

[12] Han JungHyun, Mujin Kang, and Hoogon Choi. “STEP-Based Feature Recognition for Manufacturing Cost Optimization.” Computer-Aided Design 33, no. 9 (August 2001): 671–86.