STEP-Based Feature Recognition of Orthogonal Primitives of Prismatic Parts

Deepanshu Srivastava¹, Venkateswara Rao Komma²*, Bitla Venu³

¹Research Scholar, Department of Mechanical Engineering, M. N. National Institute of Technology, Allahabad, 211004, Email: deepanshu.sripb@gmail.com
²Assistant Professor, Department of Mechanical Engineering, M. N. National Institute of Technology, Allahabad, 211004, Email: kvrao@mnnit.ac.in
³Research Scholar, Department of Mechanical Engineering, M. N. National Institute of Technology, Allahabad, 211004, Email: b.venu1973@gmail.com

Abstract

Standard for Exchange of Product Model Data (STEP) is being used as one of the effective neutral formats for exchange and sharing among different CAD systems. STEP also helps in integrating CAD/CAM system, so that the total manufacturing time and cost is reduced with increased product quality. Feature is the term used in manufacturing that defines geometric, topological and surface information of the product model. Feature recognition is the identification of features so that CAD can be integrated with CAM. In this work, product model information is extracted from STEP text file (AP203) with the program developed in Java programming language. Further, a hybrid approach is used for recognition of features in the product model. In the hybrid approach, Attributed Adjacency Matrix (AAM) and a modified syntactic pattern approach are combined in feature recognition. In the present work, the scope is limited to identify orthogonal primitives of prismatic parts only. In this work, syntactic pattern approach is modified by replacing the conventional alphabets with numbers in the pattern string. The sum of the numbers used in pattern string and number of concave edges (extracted from AAM) are used for recognizing the features. The working of the developed Java program is demonstrated by applying on the STEP AP203 files of sample prismatic products which contain different orthogonal primitives such as through slot, through step, blind step etc.

Keywords: STEP, AAM, Syntactic pattern recognition.

1 Introduction

Computer and Information Technology have been introduced into industry to relieve particular bottleneck in individual processes. However, since the introduction of computer, there has been increased understanding that different processes use the same information (e.g. geometry). In industry, we have to send product data through different medium. Product data is all the data that represents a product. Usually engineering drawing contains all data. Integration and transfer of product data is very important. Product data exchange is the enabling technology that has evolved to support these issues, particularly engineering data about product.

STEP (Standard for Exchange of Product model data) provides a mechanism that describes a complete and unambiguous product definition throughout life cycle of product (i.e. design, manufacturing, utilization and disposal). It permits different implementation methods can be used for storing, accessing, transferring and archiving product data. It is used not only for neutral file exchange, but also a basis for implementing and sharing product data bases.

Important Application Protocols (APs) of STEP in reference to CAD/CAM integration are AP203, AP224, AP240 and AP238. AP203 is “Configuration controlled 3D design of mechanical parts and assemblies”. It represents the model in terms of their Boundary representation (B-Rep) format. AP224 is “Mechanical product definition for process planning using machining feature”. It bridge gap between design and manufacturing by providing machine part information that ensures design information is 100% complete, accurate, computer interpretable and reusable. AP240 is “Numerical Control Process Plan for Machined parts”. It defines macro (i.e., high level) process plans for machining parts. AP238 (STEP-NC) is “Application interpreted model for computerized numerical controller”. It is the next step in manufacturing after AP240, this AP defines micro process planning function.

2 Literature review

Recently, research in the product design and manufacturing using STEP is gaining importance. Researchers are using STEP neutral format for feature recognition. One of the important articles is by Bhandarkar and Nagi (2000). Bhandarkar and Nagi (2000) reported feature recognition, from low level geometric entities of product design representation within a CAD model to facilitate
process planning and manufacturing activities, that has significant importance in computer integrated manufacturing system (CIMS). Sharma and Gao (2002) developed a prototype called STEP-enabled Manufacturing Planning System (SMPS) which can generate process plans and associated documents from AP224 files automatically, without any user interaction. McCormack and Ibrahim (2002) represented attributed adjacency (AA) feature recognition and represent it as a matrix or an arc node. This created a feature recognition technique that involves scanning a matrix or graph for a combination of zeros (concave relationships/edges) and ones (convex), or smaller arc node graph that are predefined to features. Verma and Rajotia (2007) reported that recognition of machining feature is a vital link for the effective integration of the various modules of CIMS. A unique method of representing a feature, called feature vector, was developed. This helped in building incrementally a feature library as per the requirement of the specific domain. Arivazhagan et al. (2009) presented a STEP AP203-214 based Machinable Volume Identifier (MVI) to identify the finished cut machinable volume in prismatic parts by deducting rough-machined part from final parts. MVI uses the predefined syntactic pattern string stored in the string database checked with the generated string of feature to determine the shape of machinable volume stored in the volume database. Rameshbabu and Shunmugam (2009) described a hybrid approach that effectively uses volume subtraction and face adjacency graph to recognize manufacturing feature from 3-D model data in STEP AP203 format. Nagarajan and Reddy (2010) developed a STEP-based platform independent system for design and manufacturing feature recognition.

From the literature review, it is observed that there is no unique method which is applicable to recognize all features. Therefore, there is a need of hybrid method which is applicable for more number of machining features. From the literature review it is also observed that Syntactic Pattern Approach is not used with Attributed Adjacency matrix (AAM) approach. Therefore, in this paper authors have made an attempt to combine these two approaches for effective recognition of machining features.

3 Methodology for Feature Recognition

With the help of Pro-E solid model has been made and is stored in STEP format (AP203). Different steps in Feature recognition methodology are listed in Table 1, followed by the details of the steps.

| Step-A | Reading STEP AP203 file with the help of Java program developed by the authors. |
| Step-B | Extraction of geometric information. |
| Step-C | Creation of Attributed Adjacency Matrix for faces. |
| Step-D | Describing Concavity and Convexity in Attributed Adjacency Matrix. |
| Step-E | Applying the modified Syntactic Pattern Approach for Feature Recognition. |

Step-A

In this step, STEP AP203 file in its text format is provided as an input to a Java program developed by authors. The java program reads the STEP file line by line.

Step-B

In this step, different geometric entities, such as lines, vertices, oriented edges, edge loops, advanced faces, are extracted. Java objects are created for all the geometrical entities available in the STEP file. Java objects corresponding to the geometrical entities are stored in such a way that they are linked to each other through their unique IDs. The unique IDs are line numbers of the geometric entities in the STEP file. An example for interlinking of entities is as follows: an edge loop of a face has references to the set of oriented edges of the face. In this paper, oriented edges are orthogonal lines.

Step-C

In this step, the geometric information is used for checking common edges between the advanced face entities. An advanced face has its corresponding lines, the lines can be compared with the lines of other advanced faces to form an attributed adjacency matrix (AAM). If any two faces are having common line the value of AAM for corresponding faces is assigned as ‘1’ otherwise ‘0’ is assigned. Pseudo code of the algorithm for generating the AAM is given below. Let N be the number of advanced faces and Li and Lj are the number of lines in face i and face j respectively.

```
for face i, i=0,1,2,...,N
  for face j, j=0,1,2,...,N
    if i ≠ j
      for line k, k=0,1,2,...,Li
      for line l, l=0,1,2,...,Lj
        if face(i).line(k) = face(j).line(l)
          Face(i)(j)=1;
        else
          Face(i)(j)=0;
      }]]]>}
```

Step-D
In this step, concavity and convexity of two connected faces are determined by the direction of the cross product between normal vectors of the two faces. If the direction of the cross product is same as the direction of the common oriented edge (line) with respect to the first face then the two faces are concave otherwise convex.

Method for calculating cross-product between two faces in contact is as follows: Let two connected Face1 and Face2 are perpendicular to each other as shown in Figure 1.

![Figure 1 Faces showing normal vector](image)

Normal vector to Face1

Normal vector to Face2

**Normal vector of face can be retrieved from the geometric entity 'plane' of the advanced face.** Let normal vector to face1 is \( A = a_1 \mathbf{i} + a_2 \mathbf{j} + a_3 \mathbf{k} \) and normal vector to face2 is \( B = b_1 \mathbf{i} + b_2 \mathbf{j} + b_3 \mathbf{k} \), then

\[
A \times B = \begin{vmatrix}
    \mathbf{i} & \mathbf{j} & \mathbf{k} \\
    a_1 & a_2 & a_3 \\
    b_1 & b_2 & b_3
\end{vmatrix}
\]

\[= (a_2 b_3 - a_3 b_2) \mathbf{i} - (a_1 b_3 - a_3 b_1) \mathbf{j} + (a_1 b_2 - a_2 b_1) \mathbf{k}\]

Where \( i, j, k \) are unit vectors in \( X, Y, \) and \( Z \) directions respectively.

In this work, the concavity or convexity is filled in AAM. The steps of the algorithm that modifies the AAM with concavity or convexity is given below.

i). **General format for advanced face entity in STEP AP203 is:**

`#advancedfaceID=ADVANCED_FACE(#face outer bound ID, #plane ID,True/False Flag).` Usually normal vector to face is outward from surface but if it is protruding inside of surface/solid then it is “F” (i.e. False), else for tangentially outward it is “T” (i.e. True). So, if advanced face flag is “F” then reverse the direction of normal vector of advanced face.

ii). **Calculate the cross product between two connected advanced faces.**

iii). If the resultant vector is in the same direction of the common line direction with respect to the first face then edges are concave and fill ‘1’ in the corresponding location of AAM.

Else edges are convex and fill ‘0’ in the corresponding location of AAM.

**Step-E**

In this step, feature recognition is performed. As features are formed by the connected faces, which are having concave edges and concavity is always important in feature recognition. In this work, features are identified by using some information of AAM and a modified syntactic pattern approach.

In conventional syntactic pattern approach, alphabetical strings are used for feature recognition. But in this work, a novel approach is used by replacing the alphabets with numbers. The main drawback with the conventional syntactic pattern approach is that the exact matching of alphabets in a string is not possible as the starting alphabet of string is governed by STEP file format. The sequence of line ID in STEP file format will get modified if the user, while making the geometric model in Pro-E, takes a different plane in sketcher and a different starting point. For a considered plane and direction (either clockwise or anticlockwise), the possible number of strings that will form are equal to the number of vertices in the face. Therefore, it is difficult to exactly match the string for feature recognition. To ease the pattern matching, in this work, alphabets are replaced with numbers and the sum of numbers in the string is used for pattern matching. Let \( X, Y \) and \( Z \) directions, which are mutually perpendicular, are assigned the numbers as shown in Figure 2. These values are assigned randomly to edges of the faces, which are in \( x, y, \) and \( z \) directions respectively.

![Figure 2 New methodology for syntactic pattern direction](image)

For example, to define a through slot in \( YZ \) plane, as shown in Figure 3, let us consider clockwise direction and the start point is right bottom corner of the drawing. Syntactic pattern string is formed by concatenating the numbers corresponding to the line directions of the feature. In this case, it is ‘42151215’ and the sum of the number string is 21, as shown in Figure 3(a). Irrespective of the starting point of the drawing, the sum of the number string is 21. If we use conventional alphabetical string, to recognize the same through slot we have to compare with eight strings (one string for each vertex as the starting point...
of the edge loop). However, if the number string is formed by considering anticlockwise direction, its
syntactic string pattern is ‘24542451’ and the number string is 27, as shown in Figure 3(b). Therefore, if
the edge loop lies in YZ plane and the sum of the number string is either 21 or 27 it may be a through slot.

The sums of pattern strings of considered features for different planes and for different features are stored in feature library. Sample feature library is given in Appendix-A.

If the sum of number string of a feature to be recognized matches with the sum of number string of a feature stored in the feature library, then the program recognizes the feature while simultaneously considering the information from AAM such as number of concave edges and the number of connected faces in the feature.

4 Results and Discussions

The above modified syntactic pattern approach for feature recognition is applied on different sample parts for testing its effectiveness.

4.1 Test Part 1

In this case, a prismatic part with through slot as shown in Figure 4 is considered. Concavity in AAM for the part shown in Figure 4 is given in Table 2. In the table, ‘Face-ID’ is the unique line number corresponding to the advanced face entity in STEP file.

The numbers of concave edges are the sum of the ‘1’s in either upper or lower half of the AAM. In this case it is two. The numbers of faces connecting the concave edges are the number of Face–IDs with concave edges. In this case it is three.

Now modified syntactic pattern approach is applied to faces (i.e. for its edge loop) which are connected to all the faces with concave edges. In this case, we get two such faces, one is the front face and the other is rear face. The edge loops of the front and rear faces have 8 numbers of oriented edges each. If the numbers of concave edges are two and the numbers of faces connecting the concave edges are three and if either of the following three conditions is true for front and rear faces then it is a through slot. The conditions are: i) if the face is on X-Y plane and the sum of syntactic number string is either 25 or 31. ii) if the face is on X-Z plane and the sum of syntactic number string is either 35 or 29. iii) if the face is on Z-Y plane and the sum of syntactic number string is either 21 or 27.

4.2 Test Part 2

In this case, a prismatic part with through protrusion as shown in Figure 5 is considered. Concavity in AAM for the part shown in Figure 5 is given in Table 3.

The numbers of concave edges are the sum of the ‘1’s in either upper or lower half of the AAM. In this case it is two. The numbers of faces connecting the concave edges are the number of Face–IDs with concave edges. In this case it is three.

Now modified syntactic pattern approach is applied to faces (i.e. for its edge loop) which are connected to all the faces with concave edges. In this case, we get two such faces, one is the front face and the other is rear face. The edge loops of the front and rear faces have 8 numbers of oriented edges each. If the numbers of concave edges are two and the numbers of faces connecting the concave edges are three and if either of the following three conditions is true for front and rear faces then it is a through slot. The conditions are: i) if the face is on X-Y plane and the sum of syntactic number string is either 25 or 31. ii) if the face is on X-Z plane and the sum of syntactic number string is either 35 or 29. iii) if the face is on Z-Y plane and the sum of syntactic number string is either 21 or 27.
From the Table 3, the numbers of concave edges are two. The numbers of faces connecting the concave edges are four.

As mentioned for the previous test part, modified syntactic pattern approach is applied to the front and rear faces. The edge loops of the front and rear faces have 8 numbers of oriented edges each. If the numbers of concave edges are two and the numbers of faces connecting the concave edges are four and if either of the following three conditions is true for front and rear faces then it is a through protrusion. The conditions are: i) if the face is on X-Y plane and the sum of syntactic number string is either 25 or 31. ii) if the face is on X-Z plane and the sum of syntactic number string is either 35 or 29. iii) if the face is on Z-Y plane and the sum of syntactic number string is either 21 or 27.

Incidentally, the conditions for modified syntactic pattern approach for through slot and through protrusion are same, however, the numbers of faces connecting the concave edges are different.

When the sides of the through slot are not orthogonal (i.e. either inclined inwards or outwards) then the number of concave edges and the number of faces connecting the concave edges are same as the orthogonal through slot. However, the sum of the syntactic number string will be different from the orthogonal through slot. Therefore, the proposed modified syntactic pattern approach will be more useful for features with nonorthogonal edges. However, in this paper, the scope is limited to orthogonal primitivies.

5 Conclusions

In this work, a hybrid approach for feature recognition is proposed. Attributed adjacency matrix and modified syntactic pattern approach are simultaneously used for feature recognition. Sum of the syntactic number string is used for making the feature recognition easier. The working of the methodology is demonstrated with two prismatic parts with orthogonal primitivies. In this work, single orthogonal primitive features such as through slot, through step, blind step, blind slot etc. were considered in the part. The present work can also be extended to recognize multiple orthogonal features in a product and/or non-orthogonal features in the product. Further scope is the feature recognition of parts with interaction and intersection features.

References


## APPENDIX-A

<table>
<thead>
<tr>
<th>Primitive Feature Type</th>
<th>No. of Internal Loops</th>
<th>No. of Concave Edges</th>
<th>Edge loop pattern where syntactic approach is used</th>
<th>No. of faces connecting concave edges</th>
<th>Syntactic Pattern sum on planes X-Y</th>
<th>Y-Z</th>
<th>X-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Through Step</td>
<td>0</td>
<td>1</td>
<td></td>
<td>2</td>
<td>21, 18, 24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Through Slot</td>
<td>0</td>
<td>2</td>
<td></td>
<td>3</td>
<td>25 or 31, 21 or 27, 29 or 35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blind Step</td>
<td>0</td>
<td>3</td>
<td></td>
<td>3</td>
<td>21, 18, 24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Through Pocket</td>
<td>2</td>
<td>4</td>
<td></td>
<td>4</td>
<td>14, 12, 16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blind Slot</td>
<td>0</td>
<td>5</td>
<td></td>
<td>4</td>
<td>25 or 31, 21 or 27, 29 or 35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blind Pocket</td>
<td>1</td>
<td>8</td>
<td></td>
<td>5</td>
<td>14, 12, 16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Through Protusion</td>
<td>0</td>
<td>2</td>
<td></td>
<td>4</td>
<td>25 or 31, 21 or 27, 29 or 35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Side Protusion</td>
<td>0</td>
<td>2</td>
<td></td>
<td>3</td>
<td>21, 18, 24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Protusion</td>
<td>0</td>
<td>3</td>
<td></td>
<td>4</td>
<td>25 or 31, 21 or 27, 29 or 35</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>