A New Methodology to Extract Machining Features From Prismatic Parts

Sreeramulu Dowluru, Suman P, and Satish Kumar Adapa

Abstract—Feature recognition systems are now widely identified as a cornerstone for conceiving an automated process planning system. Various techniques have been reported in the literature, but a few of them acquired a status of generic methodology. This research aims to exploit the concept of automatic feature recognition system. The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program that is created specifically to extract the features from the geometrical information based on a geometric reasoning approach by using object oriented design software. A feature recognition algorithm is used to recognize different features of the part such as step, holes, etc. Finally, a sample application description for a workpiece is presented for demonstration purposes.

Keywords—Feature Recognition, B-rep, Concavity/convexity.

I. INTRODUCTION

The integration of Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) has received significant attention in the recent years according to the development of faster computing power tools. However, the actual integration between CAD and CAM, for the downstream applications such as process planning, can be achieved only when the manufacturing information can be obtained directly from 3D solid model and hence automate the process planning functions [1]. This automatic extraction of manufacturing information from CAD systems play an important role to facilitate the concurrent engineering concept in order to achieve the link between the design and manufacturing activities. This successful link can be considered as fundamental step to automate the product development from the design stage all the way to manufacturing and shipping stages. Hence, the total life cycle of the product can be reduces dramatically. [2] The proposed methodology is developed for 3D prismatic parts that are modeled in any CAD software. The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program.

Sreeramulu Dowluru is with Mechanical Engineering Department of Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, INDIA (corresponding author’s phone: +919490457054; e-mail:dowlurusreeram@gmail.com).

Suman P, was with Centurion University of Technology and Management. He is now with the Department of Mechanical Engineering, Paralakhemundi, Odisha, INDIA (e-mail:suman.inertia09@gmail.com).

Satish Kumar Adapa is with Mechanical Engineering Department of Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, INDIA (e-mail:adapa.satish@gmail.com).

II. RELATED WORK

There have been considerable researches on the feature recognition systems. Automated feature recognition has been an active research area in solid modeling for many years and is considered to be a critical component for integration of CAD and computer-aided manufacturing. Mike Pratt [3] gave an overview on the three major algorithmic approaches for feature recognition and mentioned several drawbacks of them also proposed several open research areas. JungHyun Han [4] made a survey on feature recognition and merits of several algorithms of feature recognition: graph pattern matching, cell based decomposition, convex hull decomposition and Hint based reasoning. In graph-based approach, boundary representation of the part is converted into a graph which involves a set of nodes and their attributes. Joshi and Chang [5] developed a graph named the Attribute Adjacency Graph (AAG) to represent features in which each face of the part is represented as a node, and each edge or face adjacency is represented as an arc. Sashikumar Venkataraman [6] presented a graph based frame work for feature recognition. The feature recognition step involved finding similar sub graphs in the part graph. The novelty of this framework lied in the usage of a rich set of attributes to recognize a wide range of features efficiently. W.F. Lu [7] gave an approach to recognize features from a data exchanged part model. A litany of algorithms for the identification of design and machining features are proposed. A.F.M. Anwarul Haque [8] explained manufacturing feature recognition of a rotational component using DXF file. In this work geometric information of a rotational part is translated into manufacturing information through a Data Interchange Format (DXF). Emad S. Abouel Nasr [9] discussed a methodology for extracting manufacturing features from CAD system.

The system takes a neutral file in Initial Graphics Exchange Specification (IGES) format as input and translates the information in the file to manufacturing information. The boundary (B-rep) geometrical information of the part design is then analyzed by a feature recognition program that is created specifically to extract the features from the geometrical information based on a geometric reasoning approach.
III. PROPOSED METHODOLOGY OF RECOGNIZING FEATURES IN PRISMATIC PARTS

The proposed methodology for recognizing different milling features constitutes two major modules.

Module-1: Finding of Concavity/Convexity relation between adjacent faces
Module-2: Feature extraction from concavity database

A. Module-1: Methodology to Find out Relation between Faces

Face surface of a simple milling features like pocket, slot, step, blind slot, blind step, etc. are represented as planes that contain certain number of edges depending upon the solid model. These features are further classified into concave or convex as attributes in the geometric feature class. Concave features consists of two or more concave faces and convex features are decomposed of either one or more convex faces or the interaction between other features in object as shown in Figure 1.

To determine concave features is basically to identify concave faces which are defined by a concave edge that connects two adjacent faces. Similarly, convex features can be identified by the convex edge which connects two adjacent faces. The concave and convex edges are determined by the concavity test that adapted and modified from Shu-Chu Liu et.al, 1996. The concave edge test used in work is based on cross product of the normal vectors of the two faces joined by a given edge. Figure 2 shows the symbols used in this test where the ith face is designated as Fi, its corresponding normal direction vector is defined as Ni in the upward direction with respect to the given face, and the kth edge is designated as Ek. For each edge of the designed part, the edge (Ek) shared by two faces (Fi and Fj) where the order is right to left from the left side of the edge view perspective.

The direction vectors of the faces as described above (Ni and Nj). Finally the edge’s directional vector is given with respect to face Fi using loop that contains the edge (Ek).

![Fig. 1 Convex and Concave features and edges](image)

![Fig. 2 Example of Concave edge](image)

1. The cross (vector) product \((V_{CP})\) of the directional vectors of the faces is determined as follows
\[
(V_{CP}) = N_i \times N_j
\]
2. The direction of the edge \(E_k\) with respect to the face \(F_i\) is determined. The normal vector \(N_i\) of face \(F_i\) must be the first component in the cross product of step 1.
3. If the direction vector of edge \(E_k\) from step 2 is in the same direction of cross product \((V_{CP})\), then the edge \(E_k\) is convex edge that concludes \(F_i\) and \(F_j\) are convex faces, otherwise, it will be concave edge and \(F_i\) and \(F_j\) are concave faces.

The normal vector for any plane will be read from B-Rep database [10] and direction of an edge can be found by the following methodology.

**Methodology used to find the edge direction:** Edge directions in the object models can be defined such that, when one walks along an edge, its face is all the time on the left-hand side and the edge is located in the external loop of a face, then its direction will be in anti-clockwise direction relative to the surrounding face [9]. On the other hand, when an edge is located in the internal loop of a face, its direction will be clockwise as shown in Figure 3.

![Fig. 3 Direction of edge w.r.t. face](image)
A simple algorithm has been developed to find out edge direction using the information (i.e. vertex points of the edge and normal of the plane containing that edge) from B-Rep data of the part. Edge directions of all the edges of each plane \( P_i \) can be obtained w.r.t. plane \( P_i \) using the proposed algorithm. This algorithm proposed only for the planes that are parallel to global XY, YZ and ZX planes. The edge direction can be determined for each planes based on its normal direction. Different steps for finding out edge direction for the plane surfaces having normals in \(+Z, -Z, +Y, -Y +X\) or \(-X\) direction are given below.

**Step-1:** If the normal of the plane = \(+Z\)

(i) Select edge having least \((X, Y, Z)\) coordinates with constant \(Y\) (i.e. \(Y_1= Y_2\)) and name the edge as \(E_1\) and its direction is \(+X\).

(ii) Give the numbering to remaining edges as \(E_2, E_3, \ldots E_n\) immediately after \(E_1\).

(iii) If \(Y_2 > Y_1\) of \(E_2\), the direction of \(E_2\) is \(+Y\) otherwise \(-Y\)

(iv) If \(X_2 > X_1\) of \(E_3\), the direction of \(E_1\) is \(+X\) otherwise \(-X\)

(v) Similarly repeat steps (iii) and (iv) till last edge i.e. \(E_n\).

**Step-2:** If the normal of the plane = \(-Z\)

(i) Select edge having least \((X, Y, Z)\) coordinates with constant \(X\) (i.e. \(X_1= X_2\)) and name the edge as \(E_1\) and its direction is \(+Y\).

(ii) Give the numbering to remaining edges as \(E_2, E_3, \ldots E_n\) immediately after \(E_1\).

(iii) If \(X_2 > X_1\) of \(E_2\), the direction of \(E_2\) is \(+X\) otherwise \(-X\)

(iv) If \(Y_2 > Y_1\) of \(E_3\), the direction of \(E_3\) is \(+Y\) otherwise \(-Y\)

(v) Similarly repeat steps (iii) and (iv) till last edge i.e. \(E_n\).

**Step-3:** If the normal of the plane = \(-Y\)

(i) Select edge having least \((X, Y, Z)\) coordinates with constant \(Z\) (i.e. \(Z_1= Z_2\)) and name the edge as \(E_1\) and its direction is \(+X\).

(ii) Give the numbering to remaining edges as \(E_2, E_3, \ldots E_n\) immediately after \(E_1\).

(iii) If \(Z_2 > Z_1\) of \(E_2\), the direction of \(E_2\) is \(+Z\) otherwise \(-Z\)

(iv) If \(X_2 > X_1\) of \(E_3\), the direction of \(E_3\) is \(+X\) otherwise \(-X\)

(v) Similarly repeat steps (iii) and (iv) till last edge i.e. \(E_n\).

**Step-4:** If the normal of the plane = \(+Y\)

(i) Select edge having least \((X, Y, Z)\) coordinates with constant \(X\) (i.e. \(X_1= X_2\)) and name the edge as \(E_1\) and its direction is \(+Z\).

(ii) Give the numbering to remaining edges as \(E_2, E_3, \ldots E_n\) immediately after \(E_1\).

(iii) If \(Z_2 > Z_1\) of \(E_2\), the direction of \(E_2\) is \(+Z\) otherwise \(-Z\)

(iv) If \(Y_2 > Y_1\) of \(E_3\), the direction of \(E_3\) is \(+Y\) otherwise \(-Y\)

(v) Similarly repeat steps (iii) and (iv) till last edge i.e. \(E_n\).

**Step-5:** If the normal of the plane = \(-X\)

(i) Select edge having least \((X, Y, Z)\) coordinates with constant \(Y\) (i.e. \(Y_1= Y_2\)) and name the edge as \(E_1\) and its direction is \(+Z\).

(ii) Give the numbering to remaining edges as \(E_2, E_3, \ldots E_n\) immediately after \(E_1\).

(iii) If \(Y_2 > Y_1\) of \(E_2\), the direction of \(E_2\) is \(+Y\) otherwise \(-Y\)

(iv) If \(Z_2 > Z_1\) of \(E_3\), the direction of \(E_3\) is \(+Z\) otherwise \(-Z\)

(v) Similarly repeat steps (iii) and (iv) till last edge i.e. \(E_n\).

**Step-6:** If the normal of the plane = \(-X\)

(i) Select edge having least \((X, Y, Z)\) coordinates with constant \(Y\) (i.e. \(Y_1= Y_2\)) and name the edge as \(E_1\) and its direction is \(+Z\).

(ii) Give the numbering to remaining edges as \(E_2, E_3, \ldots E_n\) immediately after \(E_1\).

(iii) If \(Y_2 > Y_1\) of \(E_2\), the direction of \(E_2\) is \(+Y\) otherwise \(-Y\)

(iv) If \(Z_2 > Z_1\) of \(E_3\), the direction of \(E_3\) is \(+Z\) otherwise \(-Z\)

(v) Similarly repeat steps (iii) and (iv) till last edge i.e. \(E_n\).

For any plane having inner loop i.e. face bound and the edge curves are lines, the above representation for the edges (in inner loop) are in reverse direction as explained above.

**Concavity/Convexity Relation Algorithm:** Algorithm for finding out concave/convex relationship between faces from boundary representation database is shown in Figure 4. A generalized java program has been written for concavity test algorithm from B-Rep database file of any part which has been modeled on any platform. Following are the various steps involved in proposed algorithm for convexity/concavity relationship

**Step 1:** Read the B-Rep database obtained from GDE algorithm and store all required coordinate data including vertex points of edges, edge directions and plane directions for all plane type faces.

**Step 2:** Check for common edge between any two faces \(F_i\) and \(F_j\).

**Step 3:** If there is a match found, then find out the cross product of Normals between matched faces i.e. \((V_CP) = N_i \times N_j\)

**Step 4:** Extract edge direction with respect to face \(F_i\).

**Step 5:** If the cross product of the normals is equals to common edge direction then the above faces \(F_i\) and \(F_j\) are convex to each other, then print \(+1\), else they are concave, then print \(-1\).

**Step 6:** Stop

**Terminology used in the algorithm shown in Figure 4.0**

\(F = \) Face
\(P = \) Plane
\(VP = \) Vertex point
\((E_a)_i = \) edge ‘a’ of plane \(- i\)
\((E_b)_j = \) edge ‘b’ of plane \(- j\)
\(E_k = \) Edge direction
\(N_i = \) Normal of plane \(- i\)
\(N_j = \) Normal of plane \(- j\)
\((V_CP) = \) cross product of two normals

The output of this algorithm includes the common edge between faces and convex/concave relationship between faces which are having this common edge. This algorithm will determine the relational topology which is basically converted.
with adjacency relationships between faces and edges. The attributes used for face and edge are shown in Table I.

![Algorithm diagram](image)

**Fig. 4 Algorithm to find out the convex/concave faces**

**TABLE I**

<table>
<thead>
<tr>
<th>FACE AND EDGE ATTRIBUTES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Edge/Face</strong></td>
</tr>
<tr>
<td>Convex</td>
</tr>
<tr>
<td>Concave</td>
</tr>
</tbody>
</table>

**B. Module-2: Feature Extraction Algorithm**

Extraction of features from concavity relationship database between faces involves various steps. Input for this algorithm is the database obtained after successful execution of java program for concavity/convexity test algorithm, which consist concavity/convexity relationship between all faces. The feature extraction algorithm proposed in this work is based on production rules listed in [9] Steps involved in feature extraction algorithm are

**Step 1:** Read the concavity relationship database and count the number of concave relations each face is having and store the count in a variable concavity count for respective face.

**Step 2:** Take all the faces with maximum concavity count (MCC) and next maximum concavity count (MCC-1).

**Step 3:** Check for the concavity relationship between the faces with MCC, with all other faces with MCC-1 in concavity relationship database.

**Step 4:** Count the number of faces, those are having concavity relationship from step 3.

**Step 5:** Classify the features depending upon number of faces from step 4.

**Step 6:** Repeat the step 2 to step 5 for all faces in concavity relationship database excluding the faces those belongs to classified feature in step 5.

**Step 7:** Repeat the step 6 until all face numbers in concavity relationship database are involved in a feature.

**IV. IMPLEMENTATION OF PROPOSED METHODOLOGY VIA EXAMPLE**

An example part shown in Figure 5, having different milling features and it has been modeled in CATIA. This example consists of a total 25 faces all are planes.

**Fig. 5 Example part**

The B-Rep data [10] is given as input the algorithm given in the above section-3 and it will determine the relational topology and the adjacency relationships between the faces and edges are then determined. The output of the successful execution of the algorithm is given in Figure 6.

**Fig. 6 Output window from JAVA for the example part**

Edge - 3 of face – 4 and edge - 4 of face – 12 are equal and these faces are having convex (+1) relation between them.
Like this Figure 6 shows relationship between any two faces having common edge. Since the concave faces will form a feature, the faces which are having concave relations are shown in Table II.

### Table II

**Concavity Relation Database for the Example Part**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Face numbers</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Face-1</td>
<td>Face-2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>17</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>19</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>20</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>21</td>
<td>23</td>
<td>24</td>
</tr>
</tbody>
</table>

Now the features of the part can be extracted from concavity relation database using feature extraction algorithm discussed in section-3. The step by step execution of feature extraction algorithm is explained below.

**Feature Extraction from Concavity Relationship Database for Faces**

**Step 1:** Count Number of concave edges for each face (from Figure 6 or Table II) and the details are shown below.

<table>
<thead>
<tr>
<th>Face number</th>
<th>Concavity count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 5 6 8 9</td>
<td>3 3 3 3 3</td>
</tr>
</tbody>
</table>

Face 24 is having maximum concavity count (4) i.e. face 24 is having concavity relation with four faces and faces 20, 21, 22, 23, 5 and 8 are having second maximum concavity count (3).

**Step 2:** Take all face numbers with maximum concavity count and next maximum concavity count from step-1 and the details are shown below.

<table>
<thead>
<tr>
<th>Face number</th>
<th>Concavity count</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 5 8 20</td>
<td>4 3 3 3</td>
</tr>
<tr>
<td>21 22 23</td>
<td>3 3 3 3</td>
</tr>
</tbody>
</table>

The relationship among these faces w.r.t. the data given in Figure 6 is as follows.

5-24 No relation
21-24 -1
8-24 No relation
22-24 -1
20-24 -1
23-24 -1

The faces that are having concavity relationship with their concavity count are given below.

**Face number** | 24 | 23 | 22 | 21 | 20
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Concavity count</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

According to the production rules 24, 23, 22, 21 and 20 will form a five faced **Rectangular pocket**

**Step 3:** Now repeat step 2 for remaining features excluding the faces that already formed the rectangular pocket and the details are given below.

<table>
<thead>
<tr>
<th>Face Number</th>
<th>5 8 2 6 9 11 12 15 16 17 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concavity count</td>
<td>3 3 2 2 2 2 2 2 2 2 2</td>
</tr>
</tbody>
</table>

The faces that are having the concavity relationship among the above faces with their concavity count are given below.

5-8 -1
5-9 -1
5-6 -1
8-6 -1
8-9 -1

**Step 4:** Repeat step 2 for remaining features and the details are given below.

<table>
<thead>
<tr>
<th>Face Number</th>
<th>15 16 17 19 13 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concavity count</td>
<td>2 2 2 1 1</td>
</tr>
</tbody>
</table>

The faces that are having the concavity relationship among the above faces with their concavity count are given below.

2-11 -1
2-12 -1
11-12 -1

The faces 2, 11 and 12 will form a **blind step**

**Step 5:** Repeat step 2 for the rest of the faces and the concavity relation among these faces with their concavity count are given below.

<table>
<thead>
<tr>
<th>Face Number</th>
<th>15 16 17 19 13 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concavity count</td>
<td>2 2 2 1 1</td>
</tr>
</tbody>
</table>

The faces that are having the concavity relationship among the above faces with their concavity count are given below.

15-16 -1
16-17 -1
15-17 -1

The faces 15, 16 and 17 will form a **blind step**

**Step 6:** Repeat step 2 for the rest of the faces and the concavity relation among these faces with their concavity count are given below.
Faces 19, 13 and 18 will form a through slot.

There is no other concave faces are remaining for extraction, stop the extraction cycle and store the features in a data base. Table III shows the list of features obtained from feature extraction algorithm for the example part (Figure 5). Column – 2 of the table shows the type of feature, column – 3 represents the faces will together to form corresponding feature (These face can be shown on example part in Figure 5) and the column - 4 shows the attributes of the corresponding feature.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Feature Name</th>
<th>Face numbers</th>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rectangular Pocket</td>
<td>20, 21, 22, 23</td>
<td>(20, 80, 80) and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and 24</td>
<td>(60, 120, 100)</td>
</tr>
<tr>
<td>2</td>
<td>Blind slot</td>
<td>5, 6, 8 and 9</td>
<td>(130, 80, 60) and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(200, 120, 100)</td>
</tr>
<tr>
<td>3</td>
<td>Blind step</td>
<td>2, 11 and 12</td>
<td>(0, 160, 50) and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(23, 200, 100)</td>
</tr>
<tr>
<td>4</td>
<td>Blind step</td>
<td>15, 16 and 17</td>
<td>(0, 0, 50) and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(23, 40, 100)</td>
</tr>
<tr>
<td>5</td>
<td>Through slot</td>
<td>13, 18 and 19</td>
<td>(80, 0, 60) and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(120, 200, 100)</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

Recognition of features in prismatic parts is a tedious task when compared to rotational parts. The methodology proposed in this work is a practical approach to the total integration of CAD and CAPP a prismatic parts by feature recognition from STEP files. Two algorithms (algorithm to find out relation between adjacent faces and algorithm to extract features) for identifying milling features like slot, pocket, step etc. are developed and implemented. The proposed milling feature recognition algorithms are unable to identify slant features and the work may be extended to identify the slant features.

REFERENCES


http://dx.doi.org/10.1109/70.897789