Jurnal Mekanikal
December 2011 No 33, 01-14

IMPLEMENTATION OF HEURISTIC REASONING TO RECOGNIZE ORTHOGONAL AND NON-ORTHOGONAL INNER LOOP FEATURES FROM BOUNDARY REPRESENTATION (B-REPS) PARTS

Zheng Y, J M. Taib* and M M Tap

Faculty of Mechanical Engineering
Universiti Teknologi Malaysia,
UTM Skudai, 81310, Johor, Malaysia

ABSTRACT

B-Reps models are represented as complicated network of topology and geometry that will pose difficulties in the recognition of features. To simplify the search process, heuristic reasoning is adopted. Reasoning on topology will search and define the features, and geometry data are then used to extract the geometrical properties. This search process will identify features emanated from inner loops that lead to the recognition of island, pocket, and through pocket. Since the search is mainly based on the topology, the orthogonal and non-orthogonal features can be identified using the same procedure. Then, heuristics reasoning is again applied to features especially nested of pocket and island as well as isolated island to convert the design feature into machining features. Features such as pocket and through pocket and a new feature which is called prismatic ring are identified. The implementation of heuristic reasoning to recognize isolated design features with the aim to recognize machining will simplify the search process on the complicated of B-Reps data. The focus of this paper is to demonstrate the application of heuristics reasoning by showing the programming algorithm based on Visual Basic Language (VBA) using Application Programming Interface (API) of Solidworks CAD software.

Keywords : *Feature recognition, design feature, geometric modeler, heuristic identifier*

1.0 INTRODUCTION

Heuristic reasoning is one of the artificial intelligent concepts to reach the human level decision made by computer [1]. Heuristic allows intuitive judgment or even common sense to make decision based on the experience, knowledge etc, replaces the numerical method or mathematic to seek the solution. It involves the study of human way of thinking and translating it into meaningful algorithm for computing purposes. It is one of the ways to implement the heuristic reasoning for engineering related work.

In the area of feature recognition, perception of human and computer on the CAD model diverges. Human perceives the part as a model with specific volume or shape which has been removed or added, whilst computer sees the part as a network of the topology and geometry. This low-level definition by computer needs to be translated into human level of understanding. Therefore, heuristic reasoning on human perception based on the low-level computer part definition is the method adopted here to identify the machining features of the inner loops.

* Corresponding author : jamalt@fkm.utm.my

Currently, island has been defined differently by designers and process planners. Designers regard island as isolated feature to be added to the base model. Process planners will treat the island as the volume to be removed around the island and this causes the basic feature definition by the process planners to become more complicated to designers. In view of this, the system developed here will identify the features as isolated design features using the heuristic reasoning based on the topology and geometry data; island, pocket, and through pocket are successfully identified at this stage.

The searching method is mostly based on the topology data that causes the orthogonal and non-orthogonal features to be identified based on similar method. Orthogonality of the features is based on the profiles that emanate from the features. The inner profiles are regarded as orthogonal when all the edges are parallel to X, Y, and Z axis. When even one of the edges on the profiles is not parallel to the main axis, the profiles are regarded as non-orthogonal profiles that emanate from the features.

The system will then define the machining features using the heuristic reasoning on the features without having to revisit the complicated network of the topology and geometry data of the B-Reps part. Island, pocket, and through pockets are recognized as design feature. Since island is the only feature that has different definition from the view point of process planner, isolated island and nested features of island and pocket will be converted to machining features. A new feature, called prismatic ring, is introduced to show the volume to be removed during machining.

The focus of this paper is to elaborate the implementation of heuristics reasoning based on the topology and geometry data of the B-Reps into useful machining features. This paper also shows the programming algorithm to demonstrate the implementation of heuristics reasoning to identify the feature.

This paper starts with a preview of previous work in feature recognition in Section 2. The section will also explain on the data structure and the purpose of the paper. Section 3 will present the feature recognition algorithm and implementation of the algorithm which is discussed in detail in Section 4. Section 5 illustrates the algorithm with three parts as the illustrative examples. Finally, Section 6 will discuss the advantages of the approach and conclusion.

2.0 REVIEW OF PREVIOUS RESEARCHES

There are two purposes of the feature recognition, one is to transfer the information from CAD system to the CAPP system and the other is to transfer the features from one CAD system to another. This contributes to the automation connection which researchers are keen to achieve.

In graph based approach, the B-Reps of the part is translated into a graph where, for example, its nodes represent faces and its arcs represent edges. Additional information such as edge-convexity is incorporated into the graph. Gavankar [2] focused on the identifying morphological features which are known as the characteristic attributes of an object shape such as chamfers, protrusions and depressions. However, the faceted boundary models must be converted into their exact form before this feature extraction technique can be implemented and it is applicable only to the model representation allowing multiple edge loops such as those employing the winged-edge data structure. Verma and Rajotia [3] illustrated a new edge classification scheme to extend the graph-based algorithms for curved faces implemented in Visual C++ (VC++) using an Alan, Charles, Ian's System (ACIS) 3D solid modeling toolkit. The drawback of this system is its inability to automatically break the compound features into appropriate primitive features.

Rule based method identifies a feature based on certain pre-specified rules that are characteristic to the feature. The idea of the rule-based method for feature extraction is that rules are used to capture the knowledge about geometric and topological properties of form features. Features are recognized on the basis of certain pre-specified rules that are characteristic to the features. Babic, Nesic and Miljkovic [4] have reviewed the feature recognition work with rule-based pattern recognition. In 1998, Jain and Kumar [5] presented a system which takes a wire frame part presentation model, imported from AutoCAD dxf file. It is developed for prismatic parts (hole, step, slot and protrusions with orthogonal boundary faces), a case of syntactic pattern recognition implementation not so often met in practice. The frame model (3D) is translated in a vertices-edges graph (3D), for each of the six boundary planes of a parallelepiped. For this method, its application is limited to 2D prismatic parts, rotational parts with turning features and any axis symmetric volumes. Success for non-axis symmetric 3D part or rotational parts with non-turning features has been very limited.

Convex hull decomposition was originally evaluated in the seminal feature recognition work of Kyprianou [6] in 1980, formalized by Woo [7] in 1982 by producing a decreasing convex hull algorithm which decomposes the work piece either as a series of additive or subtractive solids. A polyhedron convex hull is determined, circumscribed around a part. The difference in volume between the part and its convex hull is defined as an alternating sum of volumes (ASV). In 1990, Wang and Chang [8] developed "backward-growing approach", for a more effective treatment of intersecting form feature problem, which can be recognized as particular complex forms or sets of trivial forms. This approach was implemented in the system developed by Nagaraj and Gurumoorthy [9], which is also based on predefined manufacturing features. The cavity volumes, regarding the most distant outer surface of the part, are defined and, in an iterant process, filled with predefined manufacturing primitives (cuboid, wedge, cylinder, etc.) .

Cell based decomposition method is firstly subtracted from its convex hull, and the process is repeated until each volume is equal to its own convex hull, decomposing the input model into a set of intermediate volumes and manipulating the volumes in order to produce features. Cell decomposition methods are simpler and easier to visualize, because they use a spatially enumerated model of the part [10].

Neural network recognition approach is one of the most promising approaches that can be used to overcome this deficiency for its two major characters: learning and recall [11]. Sunil and Pande [12] researched on an intelligent system for recognizing prismatic part machining features from CAD models using an artificial neural network. Machining feature families having variations in topology and geometry are represented by a unique 12-node vector scheme. The B-Reps CAD model in ACIS format is preprocessed to generate the feature representation vector, which are then fed to the neural network for classification. The Artificial Neural Network (ANN)-based feature recognition system was trained with a large set of feature patterns and optimized for its performance. A wide range of complex machining features allowing variations in feature topology and geometry can be recognized efficiently by the system.

Efforts have been made to advance the recognition researches by combining some characteristics of existing approaches. Rahmani and Arezoo [13] worked on the recognition of interacting milling feature with a hybrid hint-based and graph-based framework. Rameshbabu and Shunmugam [14] used both volume subtraction and face adjacency graph to realize the recognition process based on STandard for the Exchange of Product data (STEP) AP-203 of 3D CAD model as the input file for the system and the identified machining features are clustered for the setup of process planning.

The topology and geometry data structure depends on the solid kernel or translator. Due to variation of data, Guan et. al [15] uses ANN to identify machining features from STEP data. Due to the nature of data in STEP file, the lexical method is adopted to extract the data and Attribute Adjacency Graph (AAG) feature template of the features are used to train AAN prior to the feature recognition task. Similarly, Li et al [16] used lexical method to extract the B-Reps data of the part from STEP AP203 file, and the rule matching and feature template were applied to identify the features.

2.2 Data Structure

B-Reps model is represented using topology and geometry data. Within B-Reps model, different data structure exists depending on the type of solid kernel or standard used in the data collaboration. Parasolid and ACIS are the famous solid kernels that are widely used in addition to the data structures that have been developed by standard institution such as Initial Graphics Exchange Specification (IGES) and STEP.

Solidworks uses Parasolid as its solid kernel and the topology and geometry data is shown in Figure 1. Topology relationship is vertical while the geometrical relationship is horizontal. Therefore, access to the geometry must be via the respective topology. Body and loop are the topology data that do not have any geometry.

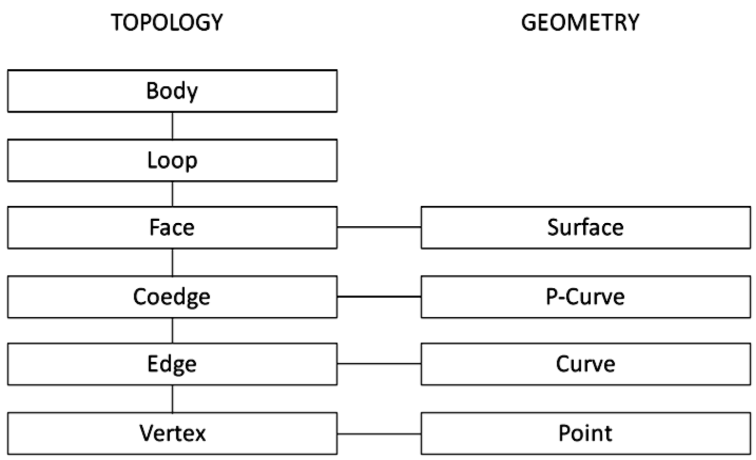


Figure 1 : Topology and geometry data structure

This vertical relationship allows specific topology data to be accessed at every level. Therefore, any level such as Face can access all the topology data using the following API, such as Face.GetBody, Face.GetLoops, and Face.GetEdges. However, in order to access the respective geometry, topology must be interfaced first. Face.GetSurface function can be assessed when Face is currently being visited. Similarly Edge.GetCurve function to get the geometry of the edges can be assessed during assessing the edges.

Each part comprises of a number of face, loop, edge, and vertex. Therefore, to access each geometry, the system has to visit all the faces. The following program structure can be used

```
ThisFace = Body.GetFirstFace
For j = 1 To NoFace Step 1

        surface = face.GetSurface()
        … accessing the geometry

        Face = Body.GetNextFace()
Next j
```

3.0 FEATURE RECOGNITION APPROACH

Figure 2 illustrates the overall approach of the system from input to output. A four-stage feature recognition is implemented here; input, recognition of isolated feature, followed with recognition of alternative feature definition when the features are nested, and finally output.

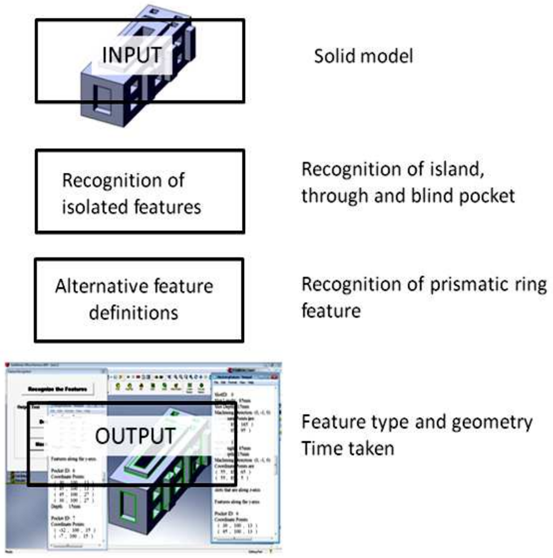


Figure 2 : Overall feature recognition approach

3.1 Input Model

The system is executed within the Solidworks platform on the current active model. To interface the solid part and its B-Reps data structure, Application Programming Interface (API) is utilized. The API interfacing is implemented using Visual Basic for Application (VBA) programming language.

The system focuses on the recognition of features that are emanated from inner loop. Features such as pocket, through pocket, and island (Figure 3(a)) are the features identified based the presence of the inner loop. These features can be isolated, arrayed and nested as shown in Figure 3(b).

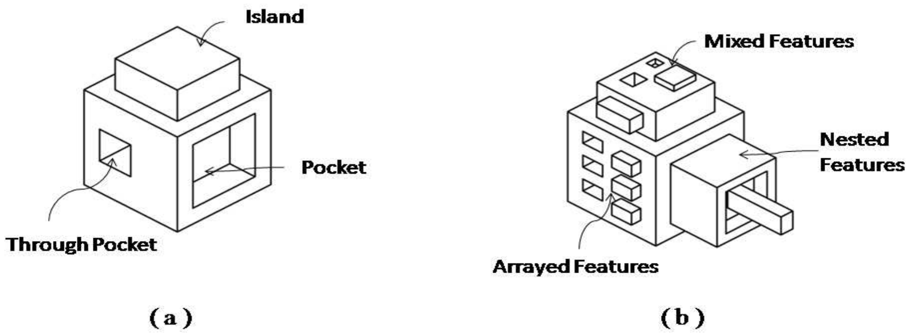


Figure 3 : Isolated, arrayed, and nested features

3.2 Recognition of Isolated Features

Recognition of isolated features is a three-level search process (as shown in Figure 4), they are;

i. *Identification of faces with more than one loop*: The faces that have more than one loop indicate the presence of inner loop/s and therefore, the first level will identify these faces.

ii. *Primary Differentiation*: Primary Differentiation examines the inner loop and classifies the inner loop either as convex or concave inner loop. Island is then identified when convex inner loop is detected, whilst concave inner loop will have to go through the next level that is the search process of Secondary Differentiation.

iii. *Secondary Differentiation*: When a profile is extruded to create a cavity or cut feature, another loop is formed in addition to the profile. To recognize the feature, this loop will be the key factor in determining whether the inner loop will emanate a through or blind pocket. Therefore, Secondary Differentiation level will carry this task that leads to the recognition of pocket or through pocket.

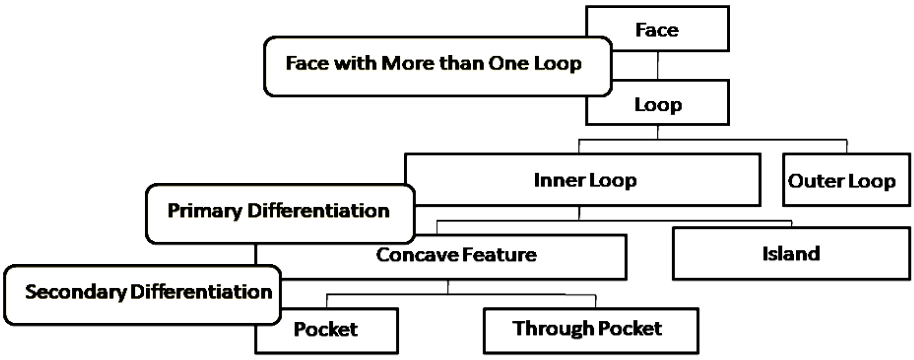


Figure 4 : Procedure to recognise isolated features

3.3 Alternative Feature Definition

Previous stage focuses on the recognition of the isolated features. This algorithm can handle isolated, arrayed, and nested features. In the case of nested features, alternative feature can be defined and therefore, the aim of this stage is to define the alternative feature. If the alternative feature exists on the part, ring prismatic features are identified from the nesting of blind pocket and prismatic 'ring' as shown in Figure 5.

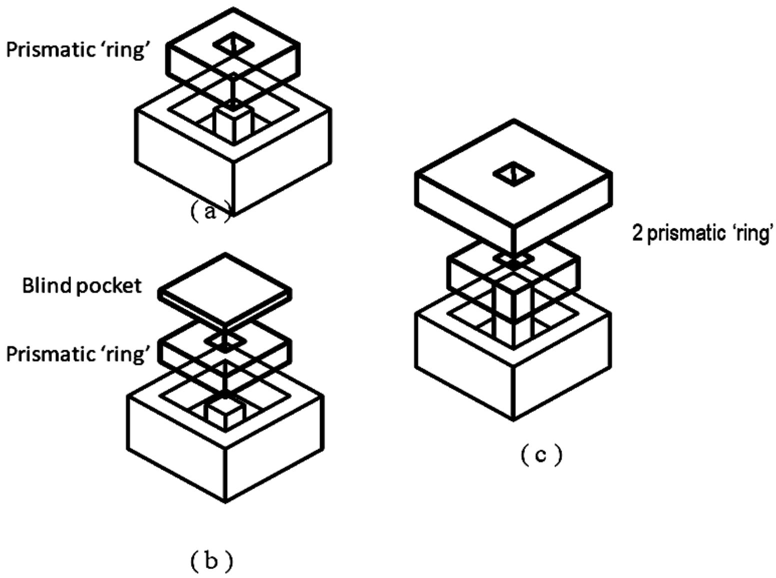
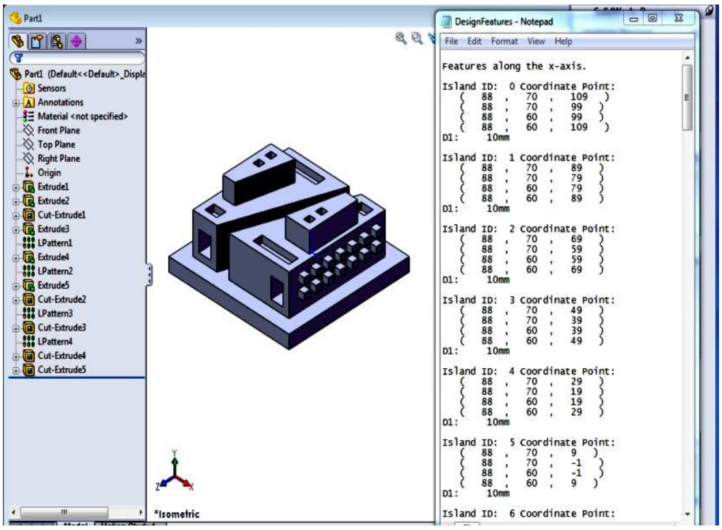


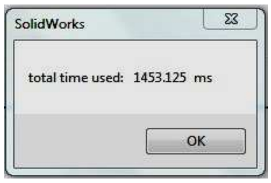
Figure 5 : Alternative feature definition

3.4 Output

The output of the system is the features written in the note pad file displayed automatically after the recognition. The features recognized are island, pocket and through pocket for isolated features, whilst ring feature is the additional feature in the case of the alternative feature definition. Figure 6(a) shows how the note pad displays the recognized feature. The system also displays the time taken to recognize the features as shown Figure 6(b).



(a) List of features



(b) Execution time

Figure 6 : Output displayed

4.0 SYSTEM IMPLEMENTATION

This section will discuss the procedure to recognize the features including the programming techniques to interface the topology and geometry of the part as well as heuristics reasoning to identify the feature.

4.1 Identify The Inner Loop

The system will start by counting the total number of faces on the part and save it as an accumulator. The system will examine the first face and continued to the next until all the faces in the model have been checked. On each face, the system will retrieve the number of loops. If the number of loops is equal to 2 or larger, the loops will be used for feature recognition in the primary differentiation stage. The following is the programming for this process;

```
LoopCnt = ThisFace.GetLoopCount

If LoopCnt >= 2 Then
     Set ThisLoop = ThisFace.GetFirstLoop
     …
     Call PrimaryDifferentiation(ThisLoop)
     Set swThisLoop = swThisLoop.GetNext

End If
```

4.2 Primary Differentiation Stage

The aim of Primary Differentiation Stage is to determine whether the inner loop is convex or concave. The concavity of the loop is determined based on the cross product of the normal of two adjacent faces as well as detecting the coedge on the loop. Each coedge is formed by two adjacent faces. Hence, the first step taken by the system is to identify two adjacent faces using the coedge.

The cross product of the normal of two adjacent faces is then compared with the tangent of the coedge. Convex inner loop is detected when the direction of the cross product is in the opposite direction to the tangent of the coedge and this will lead to recognition of the island, but vice versa for concave inner loop. Figure 7 illustrates this procedure.

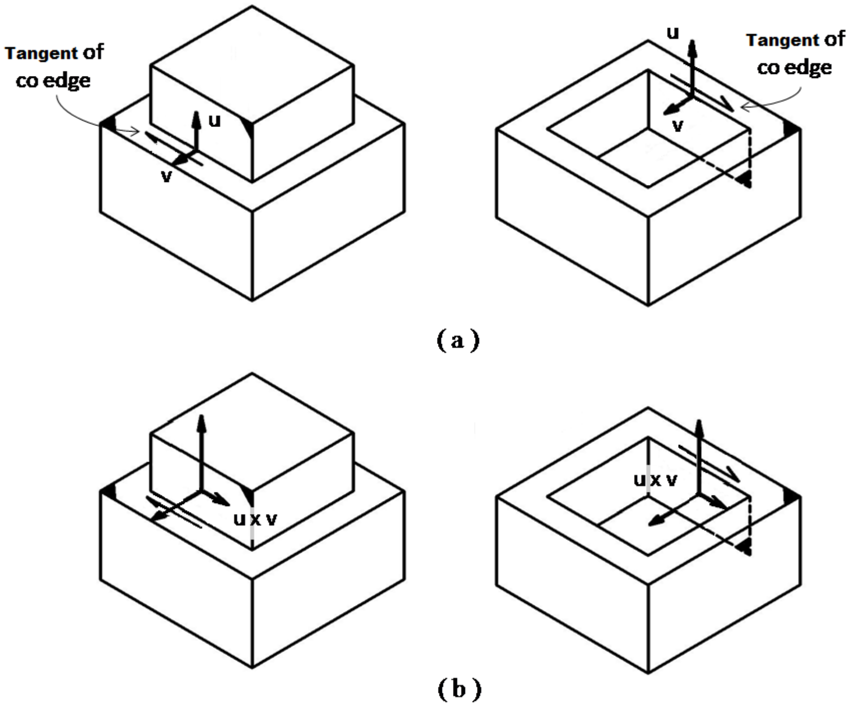


Figure 7 : Determination of convex and concave loop

Here, the network of topology is used in such a way that from loop, the system will visit the first coedge and this coedge will lead to the identification of its partner coedge. From the coedge, the system will firstly identify the face which the coedge belongs to and as a result, the normal of two adjacent faces can be determined from the geometrical properties. Then the system calculates the cross product and compares it with the direction of the coedge.

For inner loop, the direction of the coedge will be in the clockwise direction to ensure the loop is an inner loop. Apart from checking the concavity of the inner loop, the system also stores the vertices of the loop when examining the loops. The core algorithm for programming purposes is as follows:

```
Function PrimaryDifferentiation(swThisLoop As SldWorks.Loop2) As Variant
       If ThisLoop.IsOuter = False Then
          NoVert = ThisLoop.GetVertexCount
          Set ThisCoedge = ThisLoop.GetFirstCoEdge
          Set PartnerCoEdge = ThisCoedge.GetPartner
          varThisNormal = GetFaceNormalAtMidCoEdge(ThisCoedge)
          varPartnerNormal = GetFaceNormalAtMidCoEdge(PartnerCoEdge)
```

```
            varCrossProduct        =        GetCrossProduct(varThisNormal,
            varPartnerNormal)
            varTangent = GetTangentAtMidCoEdge(swThisCoedge)

           If VectorsAreEqual(varCrossProduct, varTangent) = False Then
                 Call GetBossParameter(ThisLoop)
                     Else
                 Call SecondaryDifferentiation(ThisLoop)

             End If
           End If
        …
End Function
```

4.3 Secondary Differentiation

Secondary Differentiation will determine whether the concave inner loop emanates from a through or a blind pocket. Each pocket is constructed using the extrusion of the loop at a given distance. Therefore, examining the loops will identify the feature. Through pocket is constructed when both end loops are inner loops, whilst the pocket is constructed when inner loop is connected to outer loop. Therefore, this unique characteristic is used to differentiate between them.

Primary Differentiation function will pass the inner loop to Secondary Differentiation function. When the system examines the inner loops, there is a possibility that the inner loop belongs to another through hole. Therefore the inner loops will be stored to an array variable called *ThruPocLoop*. When a through pocket is found, it will check against the inner loop of the through pockets that have been previously recognized. When the system recognized the through pocket, both inner loops will be stored to an array variable called *ThruPocLoop*. When the inner loop belongs to one of the through pocket’s inner loop, the system will exit from the subroutine Secondary Differentiation. The programming is as follows;

```
If ThruPocID > 0 Then

       For j = 0 To ThruPoc Step 1
       If ThisLoop Is ThruPocLoop(j) Then Exit Function
       Next j
End If
```

With the known inner loop, the system proceeds with the identification of feature that emanated from the inner loop. To do this, the system will visit the coedges followed with the edges. For each edge on the loop, the start and end points of the edges are retrieved.

For each vertex, there will be three incident edges. On each vertex, the system will examine the start vertex of the first edge and retrieve the end vertex of the three incident edges. Based on the coordinates of all the vertices on the incident edges, only one vertex will not be in the same plane as the innerloop. This vertex must be belonged to other loop therefore the height or the depth can be found by calculating the distance between the vertex and the plane of the innerloop.

When the heuristics reasoning identified that the vertex is not on the same plane, the vertex is set to variable called *Vert*. The vertices of the incident edges are then examined using the procedure shown below;

```
Set ThisCoedge = ThisLoop.GetFirstCoEdge
For j = 0 To 1 Step 1
       Set ThisEdge = ThisCoedge.GetEdge
       Set Vert = ThisEdge.GetStartVertex
       vEdgeArr = Vert.GetEdges
```

```
…
For Each vEdge In vEdgeArr
        Set ThisEdge = vEdge
        Set SVert = ThisEdge.GetStartVertex
        Set EVert = ThisEdge.GetEndVertex
        SPt = SVert.GetPoint
        EPt = EVert.GetPoint

                …
(heuristics reasoning to find the vertex that is belonged to other loop)

        Set Vert = EPt
    Next vEdge
Next j
```

When the vertex is determined, the system will proceed to visit the incident edges. On each incident edge, the system will visit the coedge and loop. When inner loop is detected on one of the incident edges, and shows the presence of a through pocket, through pocket detector variable (*ThruPoc*) is set to TRUE. Therefore, when *ThruPoc* is TRUE, through hole feature is identified, whilst if it is FALSE, pocket is detected and all the related parameters are saved to the features as illustrated in the following programming;

```
vEdgeArr0 = Vert.GetEdges
Set ThruPoc = FALSE
For Each vEdge0 In vEdgeArr0
        Set ThisEdge = vEdge0
        vCoEdgeArr = vEdge0.GetCoEdges

        For Each vCoEdge In vCoEdgeArr
                Set ThisLoop = VCoEdge.GetLoop
                If ThisLoop.IsOuter = False Then
                        ThruPoc = TRUE
                End If
        Next vCoEdge
 Next vEdge0
```

4.4 Alternative Feature Definition

From the previous stages, through pocket, pocket, and island are recognized as isolated features However, these features may be nested to each other since island can be inside of a pocket. The nested pocket and island can provide alternative feature definition.

The alternative feature definition depends on the height of island and depth of the pocket. When the height and depth is equal, it can generate another feature that is regarded as prismatic ‘ring’ feature as shown in Figure 5(a). When the depth of the pocket is larger than the height of the island, it generates a prismatic ring from the floor of the pocket to the top of the island and a pocket from the top of the island to the top face of the pocket. This is shown in Figure 5(b). Finally, when the top face of the island is higher than the top face of the pocket, it generates two prismatic ring features as shown in Figure 5(c).

The presence of the inner loops can lead to identification of the nested features. Therefore, the search of nested features can be incorporated into the search of the isolated features. As mention previously, the pocket is defined when the other loop is an outer loop

When the system examines the other loop, the system can visit the adjacent faces of the loop. These adjacent faces comprise of wall and bottom faces of the pocket. The face that contributes to the identification of nested island is the bottom face. Therefore, the normal of the faces will be cross examined to identify the bottom face.

When the bottom face is identified, the system examines all the edges on the faces, then the search proceeds with coedges of edges. For every coedge, the system will identify the loop. If the loop is categorized as inner loop, *PocketFace* variable, which is tag to the face, is set to TRUE. The programming for the algorithm is;

```
For Each vEdge0 In vEdgeArr0
      Set ThisEdge = vEdge0
      vCoEdgeArr = vEdge0.GetCoEdges
      For Each vCoEdge In vCoEdgeArr
            Set swCoEdge = vCoEdge
            Set PocketLoop = swCoEdge.GetLoop
            If PocketLoop.IsOuter = False Then
                  PocketFace = True
            End If
      Next vCoEdge
Next vEdge0
```

When the bottom face is detected and tagged with a TRUE *PocketFace* variable, the recognition of the features will be carried out using Primary and Secondary Differentiation Stage.

5.0 ILLUSTRATIVE EXAMPLES

Figure 8 shows the first illustrative example, namely Part A. The part comprises nested features of islands, pockets, and through pocket. The search of the faces with inner loops is based on their normal vectors. Six faces are detected to have inner loops. Four of them have one inner loop each and the other two faces have two inner loops. Therefore a total of eight inner loops (*lp1 to lp8*) are detected and these inner loops are shown in Figure 9(a).

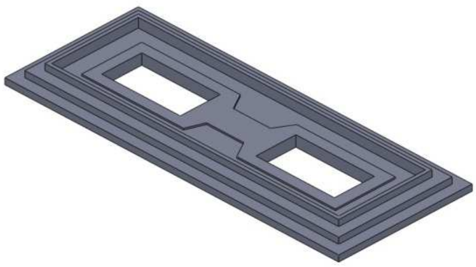


Figure 8 : Illustrative Example Part A

Firstly, Primary Differentiation Stage will catergorize whether the inner loops will emanate cavity or island. This is carried out using the first co-edge of the loop and its partner. Based on normal of the faces which both co-edges belong to, catergorization of the inner loops is carried out. At this stage, three islands are detected as shown in Figure 9(b), whilst the rest of the inner loops are defined as cavity feature which will be further defined in Secondary Differentiation Stage. The vertices of the loops (*lp1, lp2 and lp4*) will define the vertices of the features and one of the incidence edges of the inner loop will define the height of the feature.

Cavity inner loops will have to go the Second Differentiation Stage to identify whether the inner loop emanates through or blind pocket. To do this, the vertices of the incident edges are examined. The system proceeds with the search process using the vertex that is not on the same plane. Based on this vertex, similar process is adopted to indicent edges. If the edges are outer loop, blind pocket is identified and if the edges are inner loop, through pocket is identified. *ThruPoc* variable is set to TRUE to eliminate of the possibility of defining the same through pocket when the system visits this face again. Figure 9(c) shows the loop with *ThruPoc* is set to TRUE.

The next stage is to define the alternative features if the features are nested. This is done when blind pocket is identified by examining the bottom face. In the case of Part A, three faces with *PocketFace* variable is set to TRUE as shown in Figure 9(d). Based on the inner loops of the face, there are two isolated islands and one nested of pocket and island are detected. Based on the reasoning on the geometrical properties of the features, three primatic rings, one pocket, and two through pockets are detected as shown in Figure 9(e).

A total of six features were recognised, three islands, one blind pocket, and two through pockets as the first set feature definition. Since there are isolated pocket and a nested feature, alternative feature definition can be defined, which are three prismatic rings, one pocket, and two through pockets. These identified features and their geometries are saved into *DesignFeat.txt* file. The time taken to identify the features is 421.9 ms.

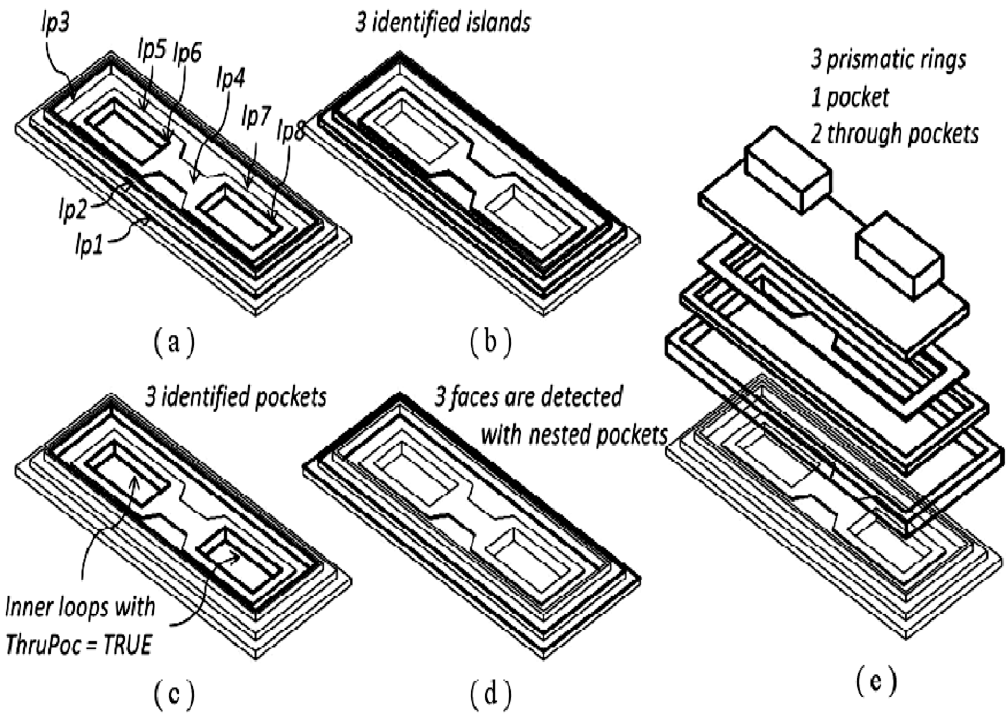
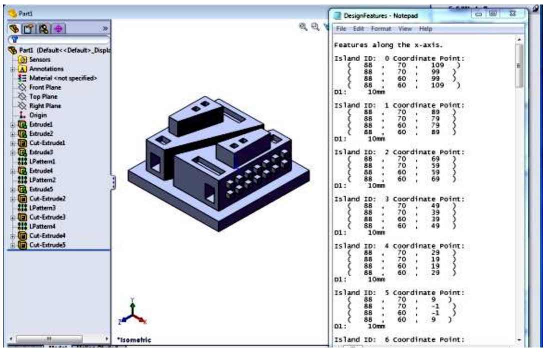
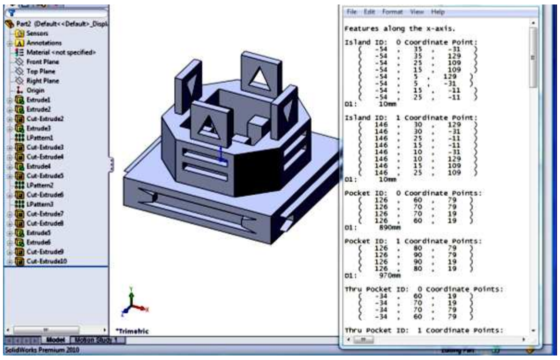


Figure 9 : Feature recognition algorithm of Part A

The system is also tested on other models, to show that the system is able to identify features from a complex part. Since the purpose of the test is to show the capability of the system, it is tested on parts, namely Part B and C shown in Figure 10(a) and 10(b) respectively. Both parts are 3D parts and comprises of orthogonal and non-orthogonal features. The system is able to recognise 42 features from Part B, which can be broken up into 32 islands, two through pockets, and eight blind pockets. In the case of the Part C, 30 features are identified which comprise of 11 islands, 10 through pockets, and nine through pockets. The times taken to identify the features are 1433.6 ms and 1296.9 ms for respective Part B and Part C.



(a) Part B



(b) Part C

Figure 10 : Illustrative Examples

6.0 DISCUSSION AND CONCLUSION

The main purpose of the paper is to show heuristics reasoning used in the recognition of orthogonal and non-orthogonal inner loop features from B-Reps model. The system has successfully adopted a similar search to recognize the features regardless whether they are orthogonal or not. The main reason for this is because the core of the search process are based on the topology data such as faces, loops, coedges, edges, and vertices. Geometrical properties will then be searched to allow the reasoning in defining the features.

Even though the system poses limitation such as it is limited to prismatic feature only, it has shown that the heuristics reasoning, based on the topology data, followed by the affirmation stage using the geometry data, has the following advantages;

i. The generic approach is able to recognize orthogonal and non-orthogonal features

ii. Minimal entities are visited to define the features

iii. Alternative features definition is capable of identifying the other possible feature definition on nested features.

REFERENCES

1. McCarthy J, 2007, *From here to human-level AI*, Journal Artificial Intelligence, Vol 171, pp 1174-1182
2. Gavankar P, 1993, *Graph-based recognition of morphological features*, Journal of Intelligent Manufacturing, Vol. 4, pp 209-218.
3. Verma A K, and Rajotia, S., 2004, *Feature vector: a graph-based feature recognition methodology*, International Journal of Production Research, Vol. 42, pp. 3219 -3234.
4. Babic B, Nesic N, and Miljkovic. Z, A review of automated feature recognition with rule-based pattern recognition, Computers in Industry, Vol. 59, pp. 321-337.
5. Jain P K and S Kumar, 1998 *Automatic feature extraction in PRIZCAPP*, International Journal of Computer Integrated Manufacturing, Vol. 11, pp 500-512
6. Kyprianou L K, 1980, *Shape Classification in Computer Aided Design*, Ph.D. dissertation, Christ College, Univ. Cambridge, Cambridge, U.K.
7. Woo T C, 1982, *Feature extraction by volume decomposition*, in Proc. Conf. CAD/CAM Technology in Mechanical Engineering, 1982.
8. Wang M T, and Chang T C, 1990, *Feature recognition for automated process planning*, In: Proc. Manuf. Int. 90, Part-2: Adv. Manuf. Syst., ASME, New York, pp. 49–54
9. Nagaraj H S and B. Gurumoorthy, 2001, *Automatic extraction of machining primitives with respect to preformed stock for process planning*, Journal of Manufacturing Systems, Vol. 20, pp. 210–222.
10. Subrahmanyam S, and Wozny M, 1995, *An Overview of Automatic Feature Recognition Techniques for Computer-aided Process Planning*, Computers in Industry, Vol. 26, pp. 1-21.
11. Ozturk N, and Ozurk, F, 2001, *Neural network based non-standard feature recognition to integrate CAD and CAM*, Computers in Industry, Vol. 45, pp. 123-135.
12. Sunil V B and Pande S S, 2009, *Automatic Recognition of Machining Features Using Artificial Neural Networks*. The International Journal of Advanced Manufacturing Technology, Vol. 41, pp. 9-10.
13. Rahmani K and B Arezoo, 2007, *Hybrid Hint-based and Graph-based Framework for Recognition of Interacting Milling Features*, Computers in Industry, Vol. 58, pp. 304-312.
14. Rameshbabu V, and Shunmugam M S, 2009, *Hybrid feature recognition method for setup planning from STEP AP-203*, Robotic and Computer-Integrated Manufacturing, Vol. 25, pp. 393-408.
15. Guan X, Meng G, and Yuan X, 2010, *Machining feature recognition of part from STEP file based on ANN*, 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE), , 26 – 28 August, Changcun, China
16. Li T, Fu C, Guan P, Yu T, and Wang W, 2010, *Milling Feature Recognition and Construction for Structural Parts Based on STEP*, International Conference on Digital Manufacturing & Automation, 18-20 December, Chansa, China