International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:5, 2016

An Automatic Feature Extraction Technique for 2D Punch Shapes

Awais Ahmad Khan, Emad Abouel Nasr, H. M. A. Hussein, Abdulrahman Al-Ahmari

***Abstract*—**Sheet-metal parts have been widely applied in electronics, communication and mechanical industries in recent decades; but the advancement in sheet-metal part design and manufacturing is still behind in comparison with the increasing importance of sheet-metal parts in modern industry. This paper presents a methodology for automatic extraction of some common 2D internal sheet metal features. The features used in this study are taken from Unipunch ™ catalogue. The extraction process starts with the data extraction from STEP file using an object oriented approach and with the application of suitable algorithms and rules, all features contained in the catalogue are automatically extracted. Since the extracted features include geometry and engineering information, they will be effective for downstream application such as feature rebuilding and process planning.

***Keywords*—**Feature Extraction, Internal Features, Punch Shapes, Sheet metal, STEP.

I. INTRODUCTION

A Feature is a given shape or characteristic in a CAD model that carries some engineering significance such as a specific function in terms of design; or useful, high-level information for some downstream application [1]. Feature has some intrinsic properties like feature type, feature shape, feature volume etc. Sheet metal features have widely used in mechanical, electronics and telecommunication industries. The automatic extractions of these features are important for downstream manufacturing functions.

Two most popular approaches in feature extraction are feature based design and feature recognition. In feature based design method, designer is limited to a specific library of features while creating the solid model. Each of these features can be mapped to one or more manufacturing processes in downstream applications [2]. In feature recognition approach, efficient algorithms are applied to extract specific information automatically from the CAD model. Various approaches have been used in feature recognition for prismatic, polyhedral etc. parts like [3] used a hint based approach patterns of convex and concave relationships between the faces of a part to identify features. References [4] and [5] developed the hint-based methodology that looks for “characteristic traces” of features in the surface geometry of a part, and generates feature hints. Reference [6] presented a methodology to recognize intersecting and non-intersecting features respectively from the 2D orthographic projections of a part. Reference [7] proposed a process orientated feature recognition system in which the pattern recognition is used to transform the surface boundary model of a component into an Attributed Adjacency Graph (AAG). Reference [8] proposed an algorithm for the identification of design and machining features from a data exchanged part model.

Sheet metal feature recognition systems have been proposed that take 3D models as input, build pattern or feature graphs for feature recognition [9], [10]. Reference [11] used vertex-edge graphs and face-edge graphs for feature recognition from wire-frame models [11]. Reference [12] developed integrated sheet metal CAD/CAM systems, all of which rely on feature definition during the design stage itself and are system dependent. The automatic feature extraction using neutral file format e.g., STEP has greater applicability than the other systems. STEP provides a computer interpretable representation and exchange of product data. The overall objective of STEP is to provide a mechanism that is capable of describing product data throughout the life cycle of product, independent from any particular system [13]. Reference [14] proposed an approach for shape representation and interoperability of product models from STEP over the internet in a distributed virtual prototyping environment. Reference [15] proposed a methodology based on attribute adjacency matrix for feature recognition. The system works well for determination of implicit feature type like internal, external, flange etc. The limitation of the system is the explicit definition of these features.

In this article, STEP AP 203 is used to extract the geometric and topological information of CAD Model. The real advantage of using STEP is its extensibility and exchange of data in neutral file format. The geometric output file is created by reading and defining entities like face, edge etc. from STEP AP 203 file using object oriented approach. The next step is to apply algorithms for identification of feature loop, faces and edges. The rule based conditions are developed to recognize each feature separately.

Awais A. Khan is an Assistant Professor in Advance Manufacturing Institute, College of Engineering, King Saud University, Saudi Arabia and Mechanical Engineering Department, University of Engineering & Technology, Lahore, Pakistan (e-mail: kawais@ ksu.edu.sa).

Emad Abouel Nasr is an Associate Professor in Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia and Mechanical Engineering Department, Faculty of Engineering, Helwan, University, Egypt.

H.M.A Hussein is an Assistant Professor in Advance Manufacturing Institute, College of Engineering, King Saud University, Saudi Arabia and Mechanical Engineering Department, Faculty of Engineering, Helwan, University, Egypt.

Abdulrahman Al-Ahmari is the Dean of Advance Manufacturing Institute, College of Engineering, King Saud University, Saudi Arabia and Professor in Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia.

II. METHODOLOGY

The proposed methodology presented in this paper consists of three main phases: (1) a geometric output file, (2) feature extractor (3) a features recognizer.

A. Geometric Output File

The part design is introduced through computer aided design (CAD) software and it is represented as a solid model by using CSG technique as a design tool. The CAD software generates and provides the geometrical information of the part design in neutral file format (STEP AP 203). The format allows the ability to communicate with the various CAD/CAM systems. A partial list of STEP entities is shown in Fig. 1.

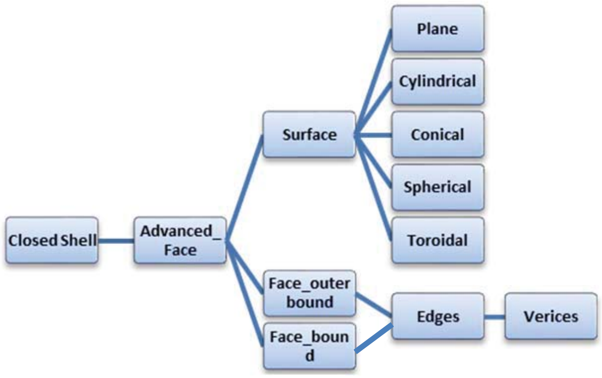
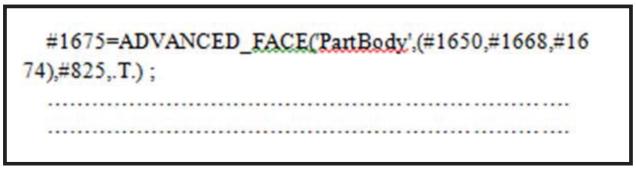


Fig. 1 STEP Data Structure

The relevant STEP entities are mapped into C++ classes using object oriented technique. Each face, edge and vertex point has been given a distinctive id.

B. Feature Extraction process

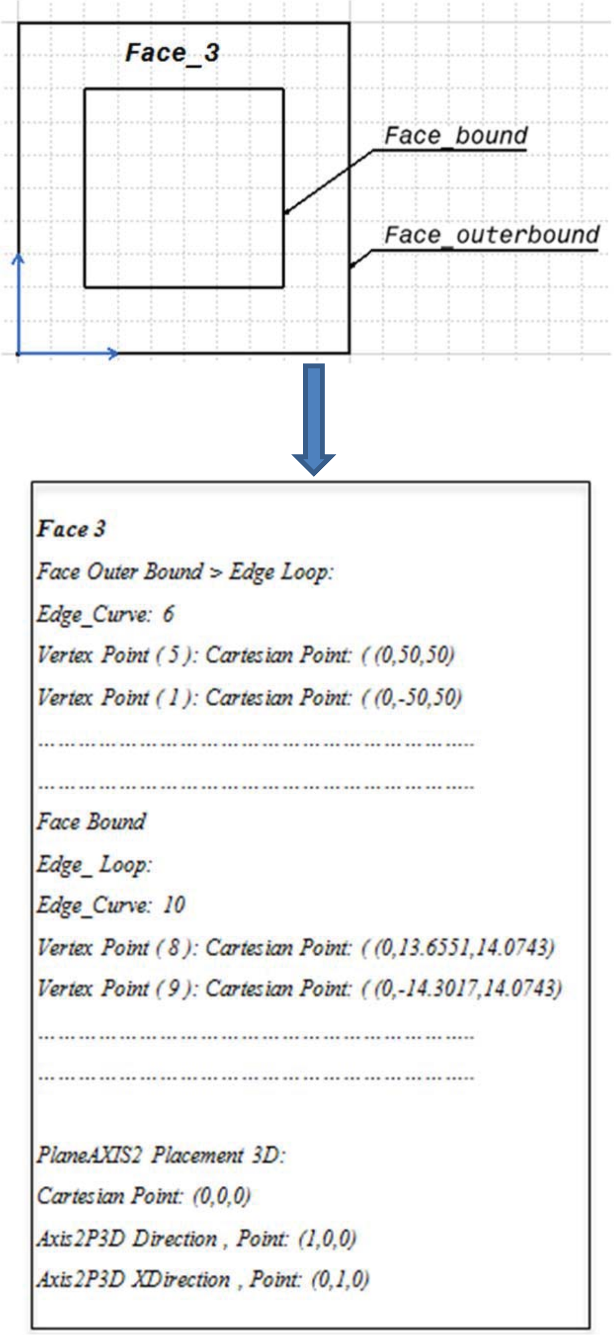
The data extraction file is now analyzed for further classification in the output file for feature extraction. In the STEP file, the Face_outerbound and Face_bound are the parts of Advanced_face entity. For example, the following line describes Advanced_face entity in STEP file (Fig. 2).



```
  #1675=ADVANCED_FACE('PartBody',(#1650,#1668,#16
74),#825,.T.) ;
  ...................................................................
  ...................................................................
```

Fig. 2 Advanced Face Entity

In this line, the numbers #1650 in the parenthesis represents face_outerbound loop and other two #1668 and #1674 indicates the presence of face_bound inside face_outerbound. The line number (#825) outside the parenthesis represents surface type. Therefore, in the data extraction output file, Face 1675 contains 1 external edge loop (#1650) and 2 internal edge loops (#1668 and #1674). The information is critical for the existence and identification of internal features. The internal feature edge loops are extracted separately for each face_bound entity along with its face_outerbound. Therefore, all extracted face_bound loops are placed inside respective face_outerbound loop with the same face identification number as shown Fig. 3. Now, these face_bound loops have been extracted and re-generated as 2-dimensional loops. The extraction and regeneration process of these face_bound loops are presented in the following steps:



Face 3

Face Outer Bound > Edge Loop:

Edge_Curve: 6

Vertex Point (5): Cartesian Point: ((0,50,50)

Vertex Point (1): Cartesian Point: ((0,-50,50)

....

....

Face Bound

Edge_ Loop:

Edge_Curve: 10

Vertex Point (8): Cartesian Point: ((0,13.6551,14.0743)

Vertex Point (9): Cartesian Point: ((0,-14.3017,14.0743)

....

....

PlaneAXIS2 Placement 3D:

Cartesian Point: (0,0,0)

Axis2P3D Direction , Point: (1,0,0)

Axis2P3D XDirection , Point: (0,1,0)

Fig. 3 Face_outerbound and Face_bound loops

Step 1:Start
Step 2:Read all the faces of data extraction file
Step 3:Extract the faces that have face_bound entries

Step 4: Extract face_bound loops of the faces found in step 3

Step 5: Mark constant coordinate entry throughout the edge_loop (the coordinate just represents sheet thickness)

Step 6: Remove this entry and re-generate the edge loops as 2-D

The next step is to define the edges inside these loops. In the STEP file, edge curves are classified as line and circle edge curves. The line edge curves can further be classified as straight and inclined line edges on the basis of vector direction of these curves. The circle edge curves are classified as concave or convex circle edge curves as shown in Fig. 4.

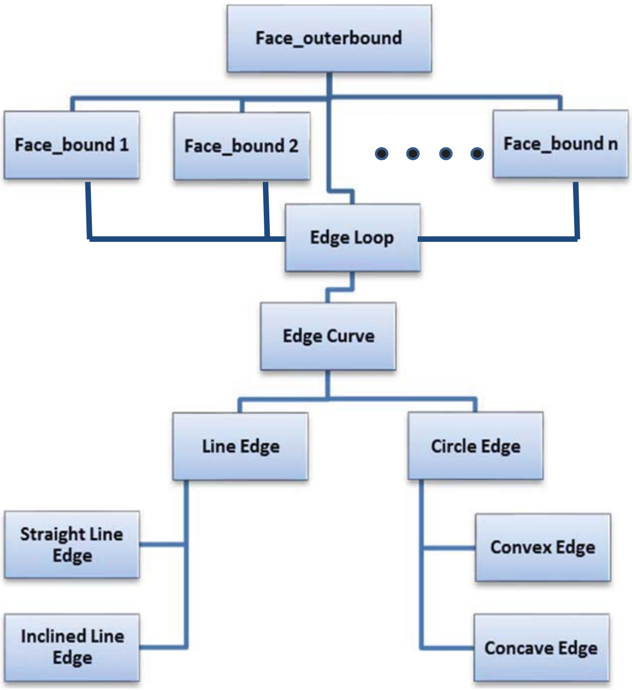


Fig. 4 Face-Loop-Edge Hierarchy

The inside or indented side of the curve is called concave side. The concave circular curve is like the inside of a circle or a bowl. Conversely, the convex side of a curve is the outside or the side of the curve away from the center of the circle. It is like the outer surface of circle. It is vital to determine the concave or convex curvature of the part in order to clarify the actual shape / feature of the part in sheet metal feature recognition technique. The concave or convex edge test used in this research is based on vector direction point of each edge. If the vector direction point is inside the edge vertex points, the curve is outward (convex curve) whereas if vector direction point is outside the edge vertex points, the edge is inward (concave curve) as shown in Fig. 5.

The following algorithm is established to precisely ascertain the concavity or convexity of the curve.

[1]. Start

[2]. Extract 2-D edge_loops of face_bound entries

[3]. Extract circular edges and their vector direction points from these edge_loops

[4]. Extract Cartesian point coordinates from vertices of these circular edges

[5]. Calculate the mid-point between the two vertices and find their absolute values

[6]. Compare values found in step 5 with absolute value of vector direction point

[7]. If the mid-point values are greater than vector direction point values, the arc is outward (convex) otherwise inward (concave)

[8]. Stop

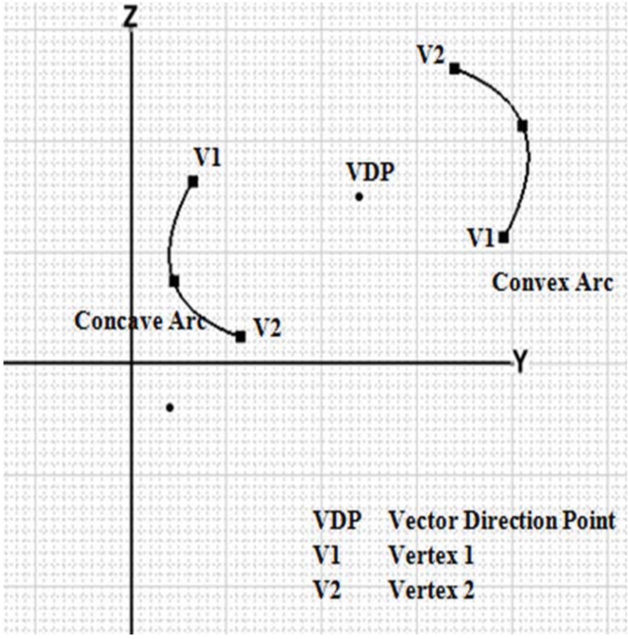


Fig. 5 Concave and Convex Arcs

The semi-circular edge curves are identified by comparing their vector direction points with the mid-point between the vertices. If points are same, the arc is categorized as semi-circular edge curve.

The parallelism and perpendicularity among line edges are established by finding the angle between their vector direction points. If the angle found is 0 or 180 degrees, the edges are parallel. If the angle between the vectors of two edges is 90 or 270, the edges are perpendicular.

The next level in the hierarchy is vertices. The start and end of the vertex point is used to determine the edge curve dimension. The distance formula is used for the calculation as

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

C. Feature Recognizer

The 2D punch features considered in this article are taken from UnipunchTM. The feature library contains standard sheet metal features from their catalogue as shown in Fig. 6.

The feature extractor file contains refined data of these internal features entities as straight line edge curve, inclined line edge curve, concave/convex curve etc. along with their dimensions. In the recognition steps, rules are now formulated based on feature extraction procedure to recognize all types of punch shapes present in feature library. Every feature has been given a unique feature id. Table I represents rules to identify selected features from feature library for illustration purpose.

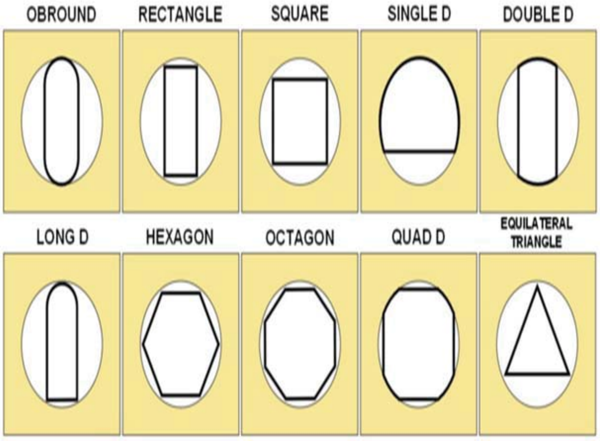
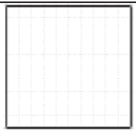
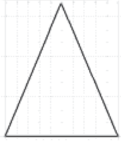
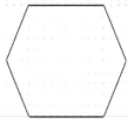
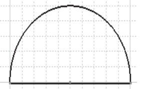
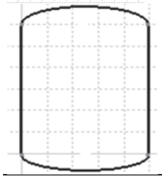


Fig. 6 Common 2-D Punch Shapes (Courtesy UnipunchTM)

TABLE I
FEATURE RECOGNITION RULES

2-D Shape	Rule	Feature Name
	*IF all edges are of type straight line edges* *IF all edges are connected by common vertex numbers* *IF non-connecting edges are parallel And connecting edges are perpendicular* *IF all Edges have same Lengths*	Feature is Square
	*IF all edges are of type straight line edges* *IF all edges are connected by common vertex numbers* *IF non-connecting edges are parallel And connecting edges are perpendicular* *IF parallel Edges have same Lengths*	Feature is Rectangle
	*IF all edges are of type inclined line edges* *IF all edges shared 2 common vertex numbers* *IF sum of angles between the edges is* 180^0	Feature is Triangle
	*IF two edges are of type straight line edges* *IF four edges are of type inclined line edges* *IF inclined edges are connected with each other and straight edges* *IF straight edges are parallel* *IF sum of angles between the edges is* 720^0	Feature is Hexagon
	*IF 1 edge is of type semicircular convex edge And 1 edge of type straight line edge* *IF both edges share common vertices*	Feature is Single-D
	*IF 2 edges are of type convex arcs And 2 edges are straight line edges* *IF straight line edges are parallel to each other* *IF each convex arc is connected to both straight edges by common vertices*	Feature is Double-D

As described in the feature recognition Table I, 2-D features are identified automatically by applying algorithms and rules from both feature extractor and recognition phases respectively. Several geometric factors like edge type, edge connections edge orientations, dimensions etc. are considered during recognition process. Each feature has been given a specific name and unique feature id.

III. CONCLUSION

The methodology presented in the article successfully validated by extraction and recognition of all features included in the feature library along with their dimensions. The feature extractor phase is pivotal for automatic recognition of these diversified features. The STEP AP 203 contains object oriented entities that can easily map in to C++ classes to create geometric output file. The feature extractor file lays the foundation for automatic recognition by extracting and classifying geometric output data using suitable algorithms. The rules are then developed and applied to each group of extracted data for recognition of these features.

The automatic recognition is useful for die/punch manufacturers. In the future work, detailed feature library with more complex features will be added.

ACKNOWLEDGMENT

This work was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (12-INF2816-02).

REFERENCES

[1] J. J. Shah and M. Mantyla. Parametric and feature-based CAD. John Wiley, 1995.
[2] Tennety, Chandu. Machining Feature Recognition Using 2D Data of Extruded Operations in Solid Models. Diss. Ohio University, 2007.
[3] S. Joshi and T. S. Chang. Graph-based heuristics for recognition of machined features from a 3D solid model. Computer-Aided Design, 20(2):58–66, March 1988.
[4] J. H. Vandenbrande. Automatic recognition of machinable features in solid models.
[5] J. H. Vandenbrande and A. A. G. Requicha. Spatial reasoning for automatic recogni- tion of machinable features in solid models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(12):1269–1285, December 1993.
[6] R. Ganesan and V. Devarajan. Intersecting features extraction from 2D orthographic projections. Computer-Aided Design, 30(11):863–873, 1998. PhD thesis, University of Rochester, USA, 1990.
[7] P. Holland, P. Standring, H. Long and D. Mynors, “Feature extraction from STEP (ISO 10303) CAD drawing files for metal forming process selection in an integrated design system” J. Material Process Technology, PP. 446–455, 2002.
[8] W. F. Lu., "An approach to identify design and manufacturing features from data exchanged part model. Computer-Aided Design 35, 979–993, 2003.
[9] B.O Nnaji, T.S Kang, S. Yeh and J.P Chen, Feature reasoning for sheet metal components. Int. J. Prod. Res., 1991, 20, 1867–1896.
[10] T.S Kang and B.O Nnaji, Feature representation and classification for automatic process planning systems. J. Manufact. Syst., 1993, 12, 133–145.
[11] R. Jagirdar, V.K Jain and J.L Batra, Charecterization and identification of forming features for 3-D sheet metal components. Int. J. Mach. Tools Manufact., 2001, 41, 1295–1322.
[12] J.C Choi and C. Kim, A compact and practical CAD/CAM system for the blanking or piercing of irregular shaped-sheet metal products for progressive working. J. Mater. Proc. Technol., 2001, 110, 36–46.

[13] L. David. "Fundamentals of STEP implementation." STEP Tools, Inc (1999).

[14] Lee, J.Y., Shape representation and interoperability for virtual prototyping in a distributed design environment. Int. J. Adv. Manufact. Technol., 2001, 17, 425–434.

[15] T.R. Kannan and M. S. Shunmugam. "Processing of 3D sheet metal components in STEP AP-203 format. Part I: feature recognition system." International Journal of Production Research 47.4 (2009): 941-964.