Computer-Aided Design 134 (2021) 102991

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Recognition of features in sheet metal parts manufactured using progressive dies

Yang Yang [a], Srichand Hinduja [a,*], Oladele O. Owodunni [b], Robert Heinemann [a]

[a] Department of Mechanical, Aerospace and Civil Engineering, University of Manchester, Manchester M13 9PL, UK
[b] Department of Mechanical Engineering, University of Chester, Chester CH2 4NU, UK

ARTICLE INFO

Article history:
Received 30 June 2020
Received in revised form 26 November 2020
Accepted 2 January 2021

Keywords:
Sheet metal parts
Progressive dies
Feature recognition
Thickness faces
Feature taxonomy

ABSTRACT

Feature recognition is an important stage in digital manufacturing as it supports activities such as automated process planning and part re-design. This paper describes a two-stage algorithm for recognising features in sheet metal parts which are manufactured using progressive dies. Representation of the part in neutral format (STEP AP203) facilitates the first stage of this algorithm in which the thickness faces are extracted in the form of chains. In the second stage, an extracted chain of faces is classified by, first, determining its topological and geometric characteristics and then comparing its characteristics with pre-set attributes for each feature. The system was developed in C++ and has been successfully tested on representative parts containing features ranging from simple bends to composite features, such as a cut-out with lance.

1. Introduction and literature review

1.1. Introduction

The importance of features in design and manufacturing cannot be overemphasised. From a manufacturing viewpoint, features are vital in the downstream activities performed in computer-aided process planning. For example, if the shape of a feature is known, the tooling engineer can design a tool to match the feature shape. Designers, on the other hand, can use feature technology to determine the degree of similarity between components with the intention of re-using existing designs and shortening the design time [1,2]. Features can also be used to link design and manufacturing by developing feature-based support systems which provide advice to the designer if one or more of the process parameters become unsuitable for manufacture [3].

The automated extraction and recognition of features were first started in the 1980s in rotational and prismatic components by researchers [4–7]. These researchers considered only machined components probably because they were more commonly found in industry and also because the features present in these machined parts are more regular [8]. It was not until the early 1990s when sheet metal parts received some attention on extracting geometrical features in sheet metal parts [9]. Several other researchers [10–20] have taken the pioneering work reported in [9] further and the academic research on automatic recognition of sheet metal features has now found some adoption in commercial software. Most commercial CAD systems have sheet metal modules with some degree of automation of the feature recognition function.

While automated recognition of sheet metal features has made vast progress, there are several significant improvements that need to be addressed. These improvements include: (i) need for clarity on the purpose of an automated algorithm — whether as skill-eliminating or skill enhancing; (ii) clarity on the coverage of features; (iii) the need to eliminate unnecessary user interaction; (iv) achieving efficient processing through elimination of redundant operations, and (v) flexibility of the algorithms to make them easier to customise and extend to new contexts and experience of practitioners.

This paper describes a method that achieves the above-mentioned improvements. Some of the highlights of the algorithm described herein are: consideration of mainly the thickness-face groups to extract the features from an input file which is in STEP AP203 format, with the thickness of the part being automatically determined, and identification of the features using generic feature characteristics rather than pre-defined graphs and it suggests an enhanced taxonomy which takes into consideration the location of a feature. The work reported in this paper focuses on sheet metal components fabricated using progressive dies. This focus enables the research to have clarity in the coverage of features and ensures that the algorithm has correspondence to concrete physical reality.

* Corresponding author.
E-mail address: sri.hinduja@manchester.ac.uk (S. Hinduja).

https://doi.org/10.1016/j.cad.2021.102991
0010-4485/

1.2. Literature review

In the early 1990s, Nnaji et al. carried out pioneering work on automated extraction of geometrical features in sheet metal parts [9]. Using a solid model of a part in IGES format, they separated all the edges into three groups (top, base and thickness faces) and, to simplify the feature extraction, discarded two groups which contained all the edges belonging to the base and thickness faces. Extraction of features from the remaining group i.e. top surface, was based on a simplified connectivity relationship between the edges. In their seminal work, Lentz and Sowerby formalised these groups by labelling them as s- and t-faces but they utilised only the top (s-face) and thickness (t-face) face groups to construct a modified version of face-adjacency hypergraphs [10]. Their work was focused on extracting drawn and bend features from these hypergraphs, features that consisted principally of quadric surfaces and included the transition faces between two concave/convex faces. Zhang et al. also subdivided the faces into s- and t-face groups but they utilised only the base and top s-face groups to form pairs of matching faces which enabled them to extract features [11]. A unique feature of their work is that the recognition is done in four stages with the user able to interact and correct, if necessary, the output at the end of each stage, thus ensuring that the output from each stage is suitable for processing by the next stage.

Instead of subdividing into s- and t-face groups, Kannan and Shunmugam calculated the mid-plane of the component, assuming, of course, that the part is of constant thickness [12]. The mid-plane is akin to a thin foil. This simplifies the feature extraction because the feature boundary can be represented by edges, vertices and loops, a simplification which was also deployed by Jagirdar et al. [13]. Although it simplifies the extraction, additional pre-processing effort is required to determine the mid-plane. Other researchers that have adopted the mid-plane approach include Kulkarni et al. and Salem et al. [14,15]. Salem et al. focused their work on the extraction of bends and the subsequent unfolding of these bends, thus enabling them to process plan the sheet metal part. Kannan and Shunmugam's algorithm to determine the mid-plane involves determining s-face pairs but Woo points out the drawbacks in identifying these face pairs in thin-walled parts [16]. To make the extraction of these face pairs more robust, Woo suggests a divide-and-conquer approach which subdivides the solid model of the part into smaller parts and then synthesises their mid-surfaces.

A very different approach was adopted by Liu et al. who did not subdivide the input data into t- and s-groups; instead, they studied the stamping process with a view to defining the characteristics of the different features [17]. They subsequently developed extraction algorithms, one for each feature. Within each algorithm, they programmed pre-defined patterns of faces and edges and were able to extract common features such as holes, bends and cut-outs. However, they admitted that they were able to extract only 80% of the features, whilst the remaining 20% had to be manually extracted.

For the recognition of deformation features, it becomes unnecessary to subdivide into t-face groups. Instead, Gupta and Gurumoorthy divided the faces into primitive faces such as a planar face (which they referred to as a "wall"), thickness face ("shell face") and "bend end faces" [18]; they then built a library comprising face adjacency graphs ("basic deformation feature" graphs), which topologically defined the connectivity between primitive faces for specific deformation features. The same authors describe the extraction of free form features in [19]. Since such features are embedded within a free form surface, there is no clear edge which separates the free form feature from its parent free form surface. They determined the curve separating the free form feature from its parent surface by calculating the principal curvatures at various points on the surface. According to Zhang et al., it requires a considerable amount of computational effort to calculate these curvatures [8]; they suggested a more efficient method wherein the original B-spline representation of a surface is transformed into a Bezier surface; this enabled them, using symbolic computation, to more easily compute the Gaussian and mean curvatures. They were then able to classify a group of free form faces into different types of features (e.g. saddle, valley, ridge, etc.), depending on the pattern of the combined signs of the curvatures. Like other researchers, they used pre-defined rules to classify combinations of these faces as depression or protrusion features.

A similar approach was adopted by Sunil and Pande who also computed the Gaussian and mean curvatures to segment the part into regions [20]. But their segmentation algorithm is probably time-consuming since the input was in STL format and the curvatures had to be computed at each and every vertex of the triangles. Having segmented the triangles into regions, they were able to classify them into free form features such as darts, dimples and dents. Their reason for using an STL format is that it had become by then more universally supported by all CAD/CAM systems and continues to do so. The disadvantage of using STL format is that some of the geometric information is lost. For example, a cylindrical face is replaced by several faceted faces. There are several other neutral formats which are more popular such as IGES and STEP. The former, as Nagarajan and Reddy [21] state, is for geometry data exchange whereas STEP is intended for product data exchange, which unsurprisingly has made it the *de facto* standard for input files for recent research on feature recognition.

Most of the research reviewed above recognises sheet metal features using deterministic algorithms such as Attributed Adjacency Graphs with fixed predefined patterns. More recently, emerging research in the area of machine learning has been deployed for the automatic recognition of features. Qi et al. [22] were able to classify objects and parts from point cloud data based on maxpooling and T-net algorithms. Zhang et al. [23] were able to go one step further by recognising features in a machined part. Their research was based on Convolutional Neural Networks and the resulting system, FeatureNet, required 144,000 training datasets of 3D-voxel models, converted from B-Rep data, to recognise 24 selected independent features with an accuracy of 97.4%. Shi et al. [24] adopted a different strategy; they generated as many as 24 sectional images from the CAD model data which were then subsequently processed to obtain the machined features. In Cao et al.'s work [25], the input data consisted of Face Adjacency Graph patterns of B-rep models. Using Graph Neural Networks (GNN) they were able to detect not only features recognised by FeatureNet but also intersecting features.

These algorithms based on machine learning are promising but they possess significant drawbacks making them not as efficient and capable as deterministic algorithms for the recognition of sheet metal features. These drawbacks are:

(i) the need for a large amount of data for training the models (for example, GNN required 15,486 CAD models to train the neural network);
(ii) the need for retraining the models when new shapes are encountered, and
(iii) the difficulty of extracting feature parameters (for example, after recognising a feature, FeatureNet is unable to generate the geometric characteristics of a feature).

Commercial systems which are capable of recognising sheet metal features are discussed in detail by Gupta and Gurumoorthy, who concluded that these commercial systems have the disadvantage of requiring the user to input the thickness of the sheet metal part [18]. The user also has to specify a reference face to initialise the recognition of features. Furthermore, intermediate steps, such as folding and unfolding, are required during the recognition process. Although the 2020 versions of commercial CAD software such as SolidWorks [26] and CATIA [27] have overcome some of these limitations, since these systems are proprietary, it is difficult to determine the methodology used, and its efficacy, to extract the features from the model data.

1.3. Gaps in the literature

The literature discussed above indicates some gaps that need to be addressed in the current research on automated feature recognition. These include the following:

(i) The location of features is not determined during the extraction stage.
(ii) Very often sheet metal features such as a jog, lance and bridge, have a relief; researchers have tended to ignore the presence of a relief in these features.
(iii) Most of the feature extraction algorithms require a considerable amount of pre-processing effort such as calculating the mid-plane.
(iv) In some cases, the user has to identify the base surface and input the thickness.
(v) Whilst there are several methodologies (hint-based, concavity, extended concavity, convex volume decomposition, see [28] for a review) for extracting and identifying features in machined components, sheet metal researchers have relied mostly on the following graph-based methods for identification purposes.

  (a) In [13], an edge-adjacency (EA) graph, degree of vertices and degree of loops are used.
  (b) In [12,15,17], face adjacency in the form of graphs or matrices is used.
  (c) An enhanced version of AAG such as the face-adjacency hypergraph (FAH) method is described in [10].

These graph-based techniques, whilst being very efficient, rely on the existence of pre-defined features in the library.

2. Definitions and feature taxonomy

2.1. Terminology

The terms used in this research are defined as follows:

(i) *Thickness (T)*: It is assumed that the part is manufactured from a sheet of constant thickness (see Fig. 1(a)).
(ii) *Thickness face (TF)*: A thickness face is one which has a pair of straight edges that are parallel to each other and equal in length to the *thickness value*.
In rare cases, a surface face may also have a pair of parallel edges equal in length to the thickness, as, for example, in the open slot shown in Fig. 1(b). To avoid f_i being wrongly classified as a thickness face because it has edges e_i and e_k equal in length to the thickness, the solid angles θ_1 (included between edges e_i and e_j), and θ_2 (included between edges e_j and e_k) in a thickness face must be less than 180°.
(iii) *Fan-shaped thickness face (FsTF):* A planar thickness face is one which has a pair of concentric circular edges and a pair of thickness edges, as illustrated in Fig. 1(a).
(iv) *Surface face*: A face that belongs to either the top or base surface, as shown in Fig. 1(a).
(v) *Thickness face chain (TFC)*: A group of connected thickness faces which form a closed chain. There are two types of closed chains:

  (a) Internal thickness face chain (ITFC): A group of connected thickness faces which form a closed chain and which lie in the interior of the part. Chain *f* (Fig. 1(b)) consists of two semi-cylindrical and two planar faces and is an example of an *ITFC*.
  (b) Boundary thickness face chain (BTFC): A group of connected thickness faces which form a closed chain and represents the external boundary of the part. Chain *g* in Fig. 1(b) is a closed chain of planar faces describing the boundary of the part and is an example of a *BTFC*.

2.2. Feature taxonomy

The earliest feature taxonomies [9,29] were mostly developed using heuristic approaches. Gupta and Gurumoorthy [30], and particularly Sanfilippo [31] presented more formalised approaches through the use of applied computational ontologies. Their approaches resulted in an attempt to create one unified taxonomy for all contexts. Most recent research in applied computational ontology [32] indicates that instead of attempting a unified taxonomy, it is better to align the taxonomies of different application contexts with the involvement of the users and owners of the content. This state-of-the-art in ontology has influenced the taxonomy presented in this paper to be a template taxonomy developed by a procedure that is explicit and empirically grounded.

The procedure for creating the template taxonomy (shown in Fig. 2) includes the following steps.

Step 1: Determination of feature instances from representative sheet metal parts.

To ensure empirical grounding, the development of the taxonomy started from 52 representative sheet metal parts provided by the industrial partner. This set of parts was analysed to identify a list of 1293 instances of sheet metal features $\{F_1, ..., F_{1293}\}$ they contained.

Step 2: From feature instances to feature types.

In this step, the 1293 instances of features were clustered into types of features using the corresponding manufacturing operations and tools. This means that for circular holes in the feature instances, association with a punching operation and a circular punch tool, means these belong to a circular hole feature type.

Step 3: Identification of feature categories.

The feature types $\{F_{t1}, \ldots, F_{tn}\}$ were generalised into 4 feature categories $\{F_{C1}, \ldots, F_{C4}\}$, depending on the general operations required to make these feature types, i.e. cut, general bend, deform or combination.

Step 4: Identifying sub-categories of feature categories.

Within each feature category, there are sub-categories of features. For example, in feature category "Cut", there are sub-categories of feature categories such as "Clip Cut" feature and "Hole Cut" feature. The taxonomy needs to be flexible so that content owners can choose to have more sub-categories.

Step 5: Ordering sub-categories of feature categories.

To ensure the taxonomy is explicit about its completeness/incompleteness, a systematic enumeration was carried out within each feature category. For example, in the "cut" feature category, the enumeration according to feature complexity produced the

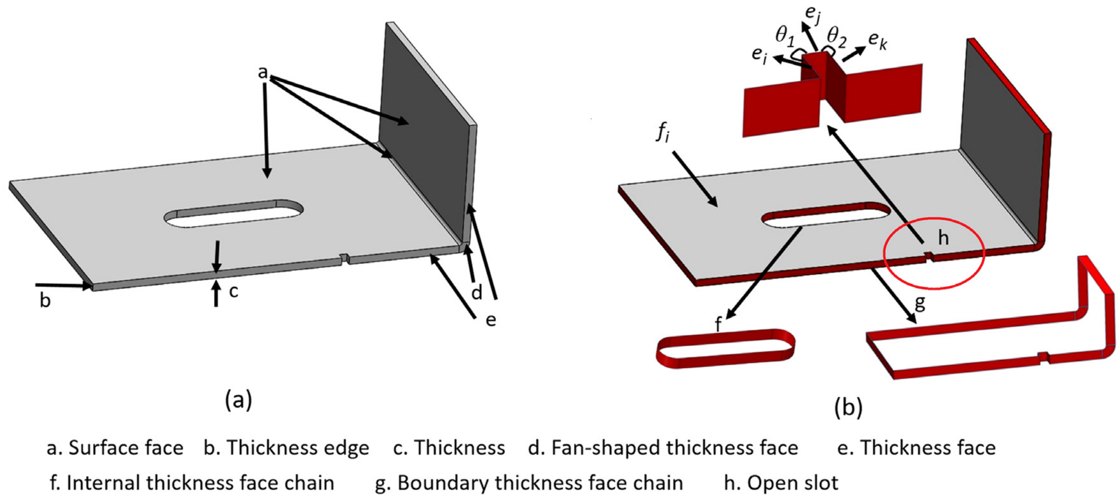

Fig. 1. Terminology.

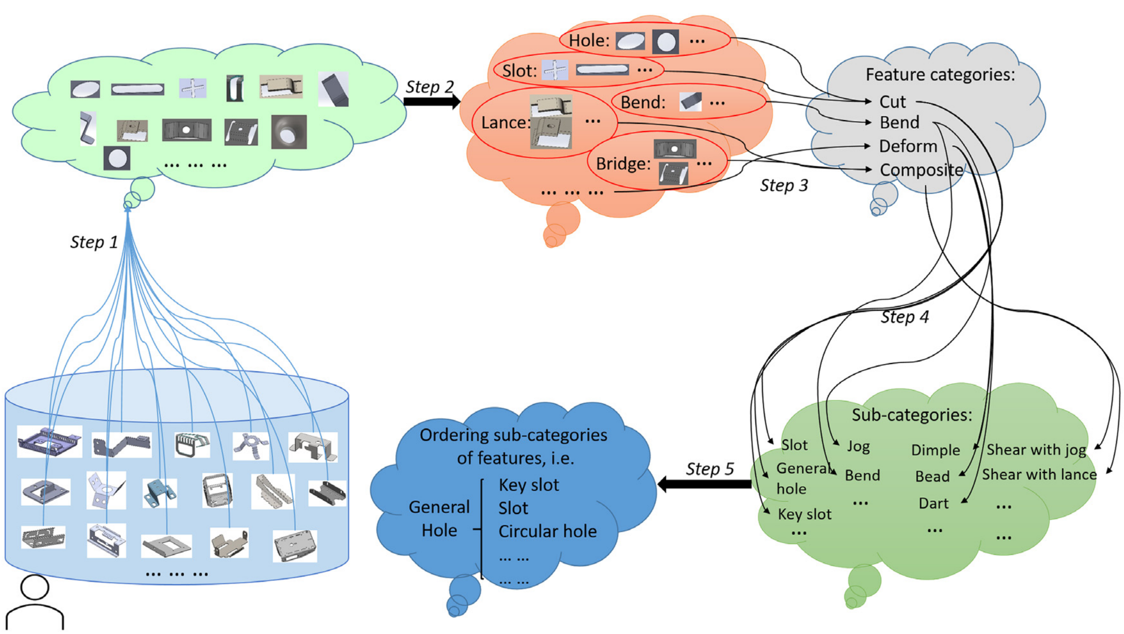

Fig. 2. Logic for producing a feature taxonomy for sheet metal parts.

order for the feature sub-categories: “Clip”, “Notch”, “Open Slot” and “General Hole”.

The resulting taxonomy is shown in Fig. 3 and some examples from this taxonomy are shown in Fig. 4. As mentioned above, rather than being a unified taxonomy for all application contexts, the proposed taxonomy acts as a template for facilitating a dialogue between domain content owners during taxonomy alignment. For example, the fact that the “clip cut feature” is not present in the taxonomy proposed by Gupta and Gurumoorthy [30] indicated in Fig. 3, could act as a dialogue starter between the content owners.

3. Feature extraction

Feature recognition is carried out in two stages: extraction and classification. The extraction is done in three phases, which are extraction of internal and boundary thickness face chains, extraction of sub-chains from the boundary chain and finally, extraction of deform features. The extraction is done from input data, which is in STEP AP203 format. Methods deployed for these extractions include:

(i) thickness face chain method to extract internal and boundary features ;
(ii) concavity and extended methods to extract slots, notches and reliefs;
(iii) convexity method to extract clips, and
(iv) inner loop method to extract deform features.

The STEP format was selected for the input data because not only is it available on all CAD systems, but the use of this format is also very helpful in the extraction as it is a boundary representation (B-rep) of the part, i.e. it contains geometrical and topological information of the part. The geometrical information consists of surfaces, lines and point information while the topological information comprises faces, loops, edges and vertices. The topological and geometrical information make the extraction of thickness faces relatively straightforward when compared to techniques such as projecting or unfolding a bridge into a planar surface or calculating the mid-plane [12,15].

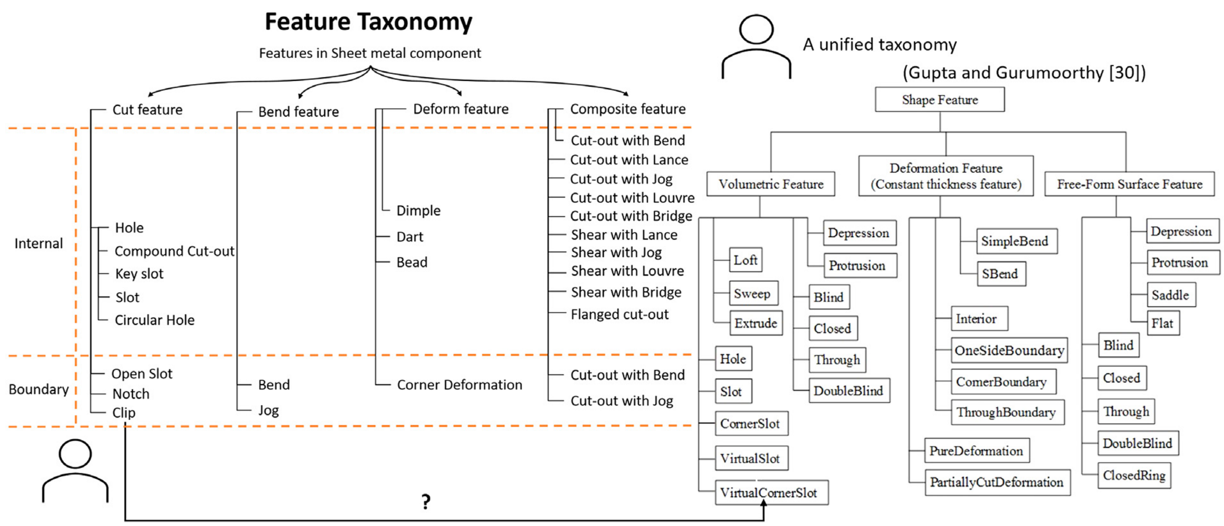

Fig. 3. Feature taxonomy.

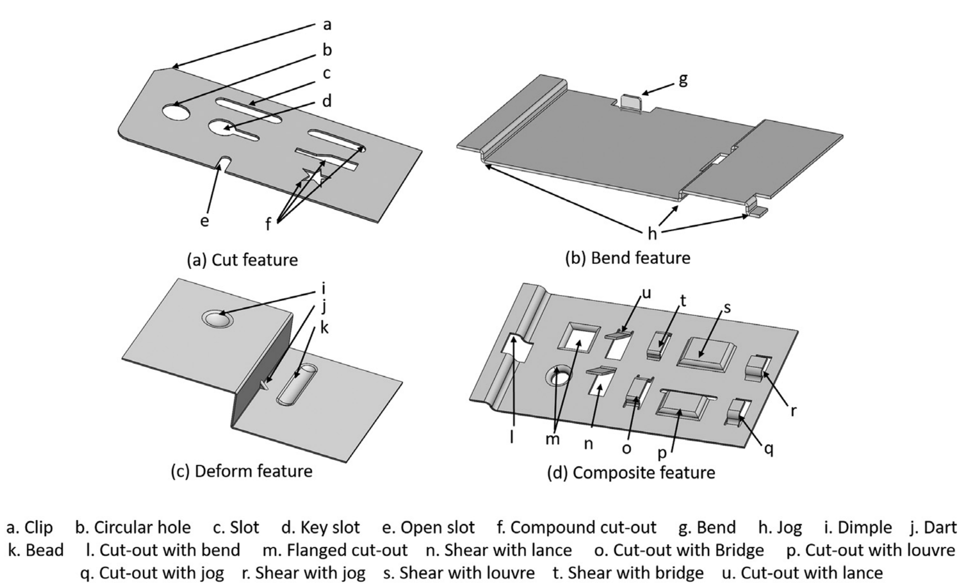

Fig. 4. Examples of cut, bend, deform and composite features.

3.1. Extraction of thickness face chains

The thickness face chain method, in some ways, bears some similarity to the concept of 3D edge loops suggested by Lai et al. [33], the similarity being that both are closed loops. However, the thickness face chain consists of faces instead of edges. Lai et al. used their loop concept to extract features in machined components whereas, in this paper, a thickness face chain is used to extract features in sheet metal parts. The advantage of using a thickness face chain is that extraction of such a chain from a B-rep data model does not require as much processing effort as that required for extracting a 3D loop.

The steps required to extract the thickness face chains from the input data are described below.

Step 1: Determination of part thickness.

The first step is to determine the thickness of the part. Rather than rely on the user having to input the value, the system automatically determines the thickness using an empirically grounded procedure.

This procedure calculates the lengths of all the straight edges in the part. (Circular arcs and B-splines curves are discarded because thickness edges are always straight.) All edges greater than a pre-set value are discarded. Next the frequency of edge lengths below this pre-set value is determined and the most frequent length is initially assumed as the thickness. The procedure then proceeds to build a chain of thickness faces with the most frequent length. If this length is not the actual thickness of the part, it will fail to build a closed chain. In such cases, the next most frequent length is assumed to be the thickness and the procedure to build chains of thickness faces is repeated. This frequency–length algorithm prevents the length of small edges which may be present in the part from being assumed as the thickness of the part.

The pre-set value represents the maximum value of thickness of a sheet metal part. According to [34], sheet metal parts have a maximum thickness of 6 mm thick. If the algorithm encounters a part with a thickness greater than this value, feedback is provided informing the user that the part may not be a sheet metal part.

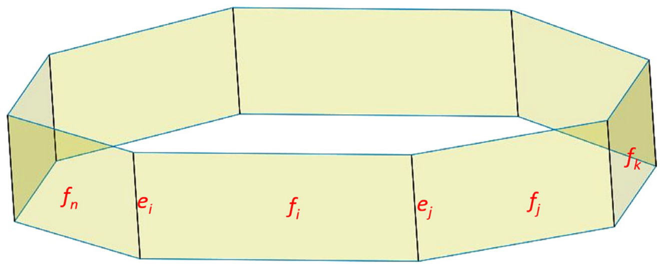

Fig. 5. Extraction of thickness face chain.

This pre-set value is not hard coded, instead it is stored in a file which can be edited by the user.

Step 2: Subdivision into t- and s-face groups.

In this step, all the faces are segregated into three groups, a thickness face group and two groups for the top and bottom surfaces. Only the thickness face group is required for extracting all the internal and boundary chains.

Step 3: Extraction of thickness face chains.

Once the group containing all the thickness faces has been determined, chains are constructed. To determine these chains, any one face f_i is selected as the seed face (see Fig. 5). Assume that the two thickness edges of this face are given by e_i and e_j. Next a search is made for an adjacent face which contains either edge e_i or e_j. If one assumes that the system searches for a face containing edge e_j, then the next face to be in the chain will be to the right of face f_i. This process of searching for the next face in the chain which shares an edge with the current face is continued until face f_n is reached; this face shares edge e_i with the seed face resulting in chain i (TFC_i). All the faces contained in chain TFC_i are removed from the group of thickness faces.

A new search is initiated for the next chain and this process is repeated until no further chains (TFC_i, $i \in 1, n$) can be formed.

Step 4: Classification of thickness face chains.

For each chain, three range values, i.e. $(X_{max}\text{-}X_{min})$, $(Y_{max}\text{-}Y_{min})$ and $(Z_{max}\text{-}Z_{min})$, are calculated and these range values are used to compare the chain sizes. For a chain to represent the boundary, two of its range values must be greater than the corresponding values for any other chain. If, for the sake of simplicity, one assumes that the thickness is in the Z-direction, then the X_{range} and Y_{range} for the boundary thickness face chain *(BTFC)* must be greater than the corresponding ranges of any other chain. The other chains are referred to as internal thickness face chains *(IFTC)* and they represent internal features.

3.2. Dividing the boundary chain into sub-chains

The boundary of the part may contain features which are represented by boundary sub-chains (BsC); each sub-chain, or a pair of sub-chains, represents a boundary feature. These boundary sub-chains are open whereas the parent boundary thickness face chain is closed.

Some of the features that can exist on the boundary are open-ended slots, clips, notches, jogs and bend. Several different methods are deployed to extract sub-chains relating to these features and these methods are described below.

3.2.1. Extraction of fan-shaped thickness face sub-chains

This method extracts sub-chains containing fan-shaped faces which are always present in features such as a bend or jog. These features contain at least *one* pair of fan-shaped faces. So, the first step is to search the boundary thickness face chain and identify all the fan-shaped faces. Fig. 6 shows the thickness face chain for the component in Fig. 4(b), and this boundary chain contains a total of fourteen fan-shaped faces, e.g. faces f_1 and f_5. As mentioned earlier, a fan-shaped face must be planar and must contain two concentric circular edges and two thickness edges. The next step is to pair these fan-shaped faces. For faces f_i and f_j to be partner faces, the following conditions must be satisfied.

(i) The face normals fn_i and fn_j must be in opposite directions (see Fig. 7(a)).
(ii) The line connecting the centres of the concentric circular arcs of f_i and f_j must be parallel to the normals fn_i and fn_j.
(iii) The two larger circular edges, e_i and e_j, belonging to the fan-shaped faces f_i and f_j respectively, should be identical in a geometric sense i.e. part of the same cylindrical surface (see Fig. 7); the same must be also true of the two smaller circular edges e_{ii} and e_{jj}, both of which should also be part of another but smaller cylinder. In most cases, e_i and e_j will be edges of the same cylindrical face but in some cases, as in Fig. 7(b), they are not part of the same face because of a cut-out in the cylindrical face.

Analysis of the boundary thickness face chain results in seven fan-shaped pairs i.e. f_1–f_{p1}; f_2–f_{p2}; f_3–f_{p3}; f_4–f_{p4}; f_5–f_{p5}; f_6–f_{p6}, and f_7–f_{p7}.

The next step is to determine the number of *other* fan-shaped faces contained within each pair. For each pair of fan-shaped faces, there are two possible solutions. Consider pair f_3–f_{p3}. The first list of fan-shaped faces between these faces (going anti-clockwise) is f_3–f_4–f_{p4}–f_{p3} and the other list (going clockwise) is f_3–f_2–f_1–f_5–f_6–f_{p6}–f_{p5}–f_7–f_{p7}–f_{p1}–f_{p2}–f_{p3}. The longer list is discarded and the shorter list is associated with the pair.

This process is repeated for all the seven fan-shaped pairs and the retained lists are given below.

f_7–f_{p7}
f_6–f_{p6}
f_4–f_{p4}
f_5–f_6–f_{p6}–f_{p5}
f_3–f_4–f_{p4}–f_{p3}
f_2–f_3–f_4–f_{p4}–f_{p3}–f_{p2}
f_1–f_2–f_3–f_4–f_{p4}–f_{p3}–f_{p2}–f_{p1}

This procedure will select the longest list selected, thus ensuring that it does not start from the middle of a feature. Since the longest list (f_1–f_2–f_3–f_4–f_{p4}–f_{p3}–f_{p2}–f_{p1}) does not contain all the thickness faces, a second list is created. Since this list was obtained by traversing in an anti-clockwise direction, the second list is generated by traversing the boundary in a clockwise direction starting from, but not including, f_1, i.e. f_5–f_6–f_{p6}–f_{p5}–f_7–f_{p7}. Travelling in the opposite direction to generate the second list ensures that every fan-shaped face in the boundary thickness face chain appears in one of the two lists.

Next each face in these two lists is designated as either a 0 or a 1; a starting fan-shaped face is assigned a 0 whereas its partner face is assigned a 1. The resulting binary representation of these two lists is shown in Fig. 6, and these binary representations are further analysed in the next section for classifying the features they contain. Suffice to say that the list does not start in the middle of a feature i.e. it does not start at f_2 or f_6.

3.2.2. Concavity methods

The concavity and extended concavity methods rely on the presence of concave edges, concave faces and concave vertices [7,35]. Such entities are present in features such as reliefs, slots and notches. Sub-chains which represent these features are extracted from the boundary chain of thickness faces and they

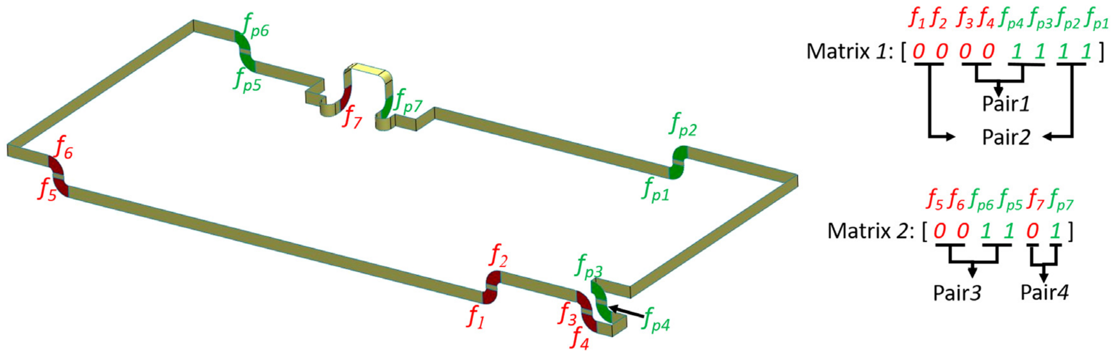

Fig. 6. Boundary thickness-face chain and its subdivision.

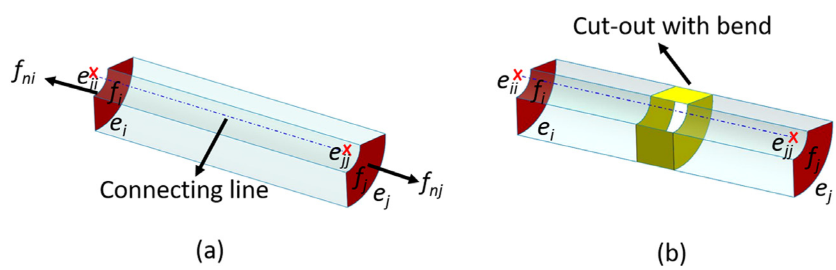

Fig. 7. Gluing fan-shaped faces into pairs.

consist either of planar faces as in Fig. 8(a) and (c) or a mixture of planar and circular faces as in Fig. 8(b). In Fig. 8(a), since edges e_i and e_j are concave, then, according to the concavity method, the three faces (f_i, f_j and f_k) that contain these edges constitute an open slot and are extracted from the boundary thickness face chain as a sub-chain. Similarly, the sub-chain representing the slot in Fig. 8(b) is extracted using the extended concavity method. According to this method, since f_j is part of a negative cylinder, it is deemed to be concave and, therefore, part of a feature [35]. Consider it to be the seed face for the sub-chain; since f_j is tangent to its neighbouring faces f_i and f_k, edges e_i and e_j are said to be concave; therefore, f_i, and f_k are attached to the seed face and the three faces (f_i, f_j and f_k) form an open sub-chain.

The concavity method is also used to extract sub-chains which represent a notch on the boundary (see Fig. 8(c)). Since edge e_i is concave, faces f_i and f_j belong to a feature, and since edge e_h and e_j are convex, the only two faces in the sub-chain are f_i and f_j.

Reliefs adjacent to fan-shaped faces are usually similar in shape to the features shown in Fig. 8(a) and (b). If the relief has a different shape, the concavity or extended concavity method will be able to extract the feature since it is bound to contain concave edges or faces. Reliefs which are adjacent to internal fan-shaped faces will be extracted as part of the closed chain of thickness faces, irrespective of their shape.

Once these sub-chains are obtained, they need further analysis to determine whether they should be considered as individual boundary cut-out features or combined with another feature to form composite features. If any face in a sub-chain shares a thickness edge with a fan-shaped face in a boundary bend feature (e.g. boundary bend or boundary jog/lance), then these two features are combined together and defined as a composite feature, such as a "Cut-out with bend" or a "Cut-out with jog".

3.2.3. Convexity method

This method extracts a sub-chain consisting of a single face representing a clip. Currently, face $f_j$in Fig. 8(d) is extracted from the boundary thickness face chain if the solid angles θ_1 between f_i and f_j and θ_2 between f_j and f_k are greater than 90° but smaller than 180°. Thus, any sub-chain that satisfies the angle pattern of $\theta = \{\theta_1, \theta_2\}$, where $90^\circ < \theta_1$ *and* $\theta_2 < 180^\circ$, is defined as a clip, see Fig. 8(d).

3.3. Inner loop method

This method extracts deformation features such as a dimple, bead and a flanged cut-out, and is similar but more concise and more efficient to the method suggested in [12]. These deformation features do not contain any thickness faces, and therefore the question of using them for extraction purposes does not arise. Instead they require the top s-face group for their extraction. The faces in the group are examined for the presence of an inner loop which is not part of a closed internal thickness chain. If these two conditions are satisfied, then this inner loop defines the external boundary of a deform feature and the s-face is referred to as the mother face (face f_i, Fig. 9(a)). A recursive procedure is used to collect all the faces which constitute a feature, as illustrated in Fig. 9.

This procedure is started by collecting all the partner faces connected to the edges of this inner loop (Fig. 9(b)), and these faces are included in the feature face set at level one. The procedure then continues to check for faces adjacent to level one faces. If it finds any faces, they are appended to the set at level two (Fig. 9(c)). This recursive procedure is continued until:

(1) the adjacent faces do not belong to top s-face group (Fig. 9(c)) (termination criterion 1), or
(2) no more new faces can be found (termination criterion 2) (Fig. 9(e)).

When the procedure is terminated, the faces in the list represent the extracted deform feature.

Fig. 10 shows the flowchart for extracting the sub-chains/chains representing features from the input data.

4. Feature classification

The feature classification method presented in this section is to identify specific feature categories and sub-classes of feature

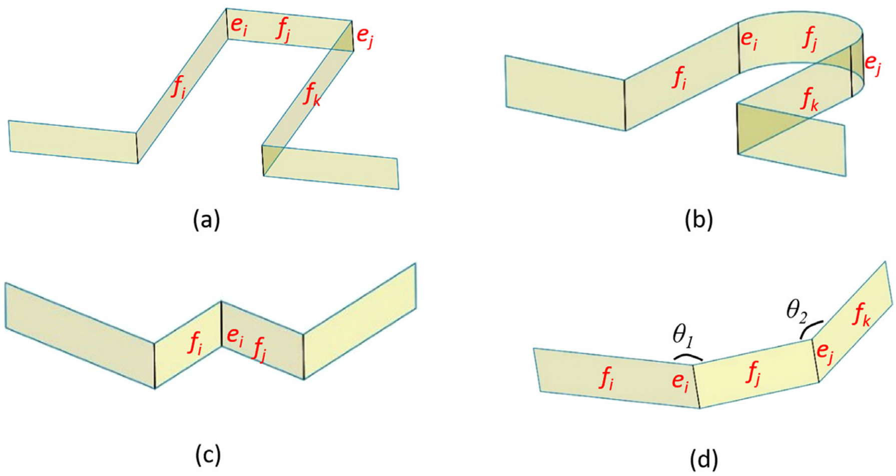

Fig. 8. Examples of boundary sub chains containing a notch, clip and slots.

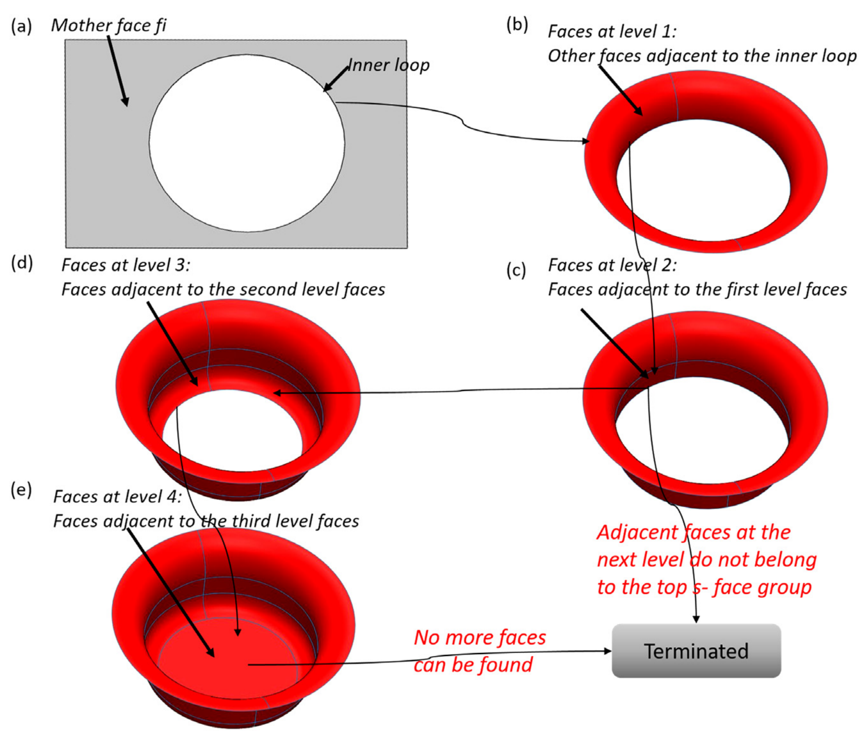

Fig. 9. Procedure for extracting deform features.

categories from the internal chains, boundary sub-face chains and sets of surface faces, which have been extracted based on the methods discussed in Section 3.

The feature classification is based on a set of properties derived from the geometrical and topological data of the face chains. For each feature type, i, one can identify a set of n properties $\{P_{i1}, P_{i2}, \ldots P_{in}\}$ which characterise the feature. The sets of properties for m feature types would be a $m \times n$ table or matrix $[P_{i,j}]$ referred to as a Characteristics Matrix (CM). The occurrence of a fan-shaped face in a thickness face chain makes a feature non-planar (referred to as 3D) and this helps to distinguish a 3D feature from a planar (2D) feature. Because the cut, bend and composite, and deform categories of features possess different characteristics, there are different matrices for each of these categories and they are considered separately in Sections 4.1–4.3. The feature classification is then considered in Section 4.4. The flowchart for feature classification is shown in Fig. 11.

4.1. Characteristics for cut features

The characteristic which distinguishes a cut feature from other feature categories is that every vertex in the thickness face chain lies on one of two parallel planar surfaces which are offset from each other by the thickness. Sub-classes of this feature are distinguished by the type of chain. For example, a “general hole” is characterised by a closed chain whereas boundary cut features such as a notch, clip and open slot are characterised by an open chain. The challenge is to determine the set of properties that uniquely and unambiguously characterises a feature type without subsuming others. It can be shown that such a set would consist of the following properties: (1) topological property of the chain-closure or openness of the chain; (2) total number of entities; (3) number of entities of each type; (4) geometric constraints between entities; (5) concavity condition between entities and (6) relative size of entities. Omitting any of these characterising

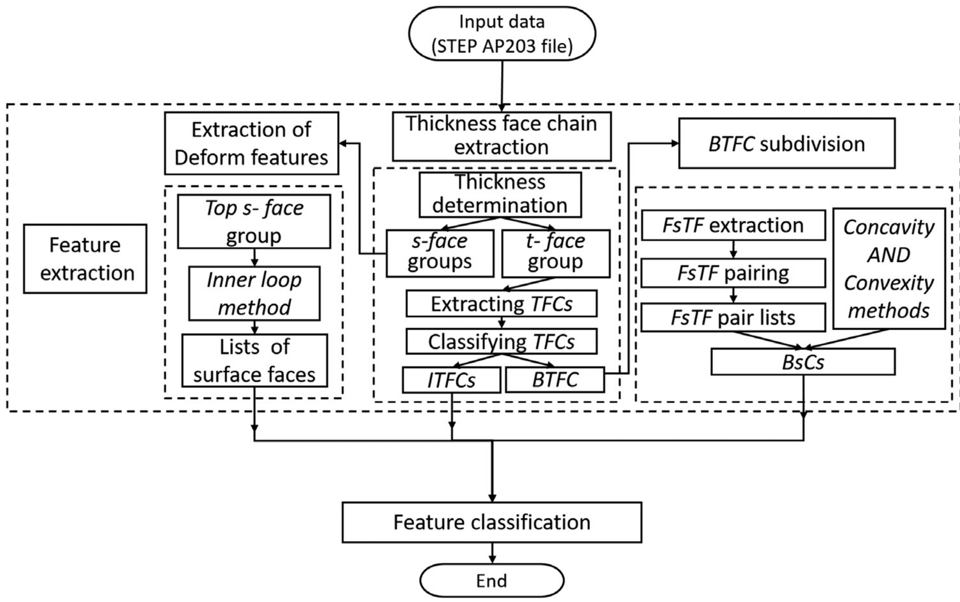

Fig. 10. Flowchart for feature extraction.

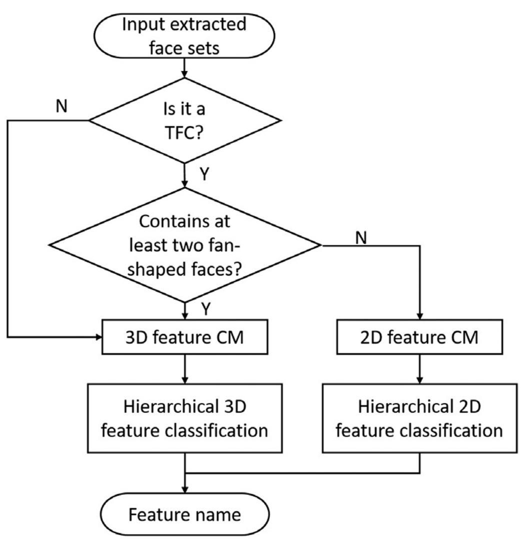

Fig. 11. Flowchart for classifying features from extracted face sets.

properties would result in various anomalies. For example, if the concavity between faces is omitted, a slot would have the ambiguity of representing the different shapes shown in Fig. 12. Properties satisfying the six conditions mentioned above have been derived by studying the geometry and topology of several thickness face chains. These properties are captured in a Characteristics Matrix, an example of which is given in Table 1 for some sub-classes of cut features.

Each row of the matrix contains a sufficient number of characteristics for identifying one type of cut feature. For instance, a slot feature is considered to have two cylindrical faces as well as one pair of planar faces which are parallel to each other and are of the same dimensions.

4.2. Characteristics of bend and composite features

The main characteristic of bend and composite features is the presence of at least two fan-shaped thickness faces in a chain, be it an internal or boundary sub-chain. The presence of such faces makes the chain 3D. In other words, the presence of a fan-shaped face in a chain confirms that the chain does not represent a cut-out category of features but a bend or a composite feature.

The next important characteristic in a composite feature is the number of fan-shaped thickness faces and the number of pairs they form; they help to further classify the thickness face chain. For example, a simple bend contains fan-shaped faces and they form one pair whereas a jog contains four such faces and they form two pairs (see columns 4 and 5 in Table 2).

The third characteristic is the number of chains required to represent a feature (column 2, Table 2). Generally, a feature is represented by one chain, but a bridge (with/without cut-out) requires two chains. However, these chains are geometrically related in the sense that they are mirror images of one another.

Fig. 7(a) shows a simple bend with edges e_i and e_j being part of the same face. But this is not true in Fig. 7(b), wherein e_i and e_j belong to two different faces. This characteristic is used to distinguish a cut-out with bend and louvre (with/without cut-out) from other features (see column 6, Table 2).

The direction of the normals to the fan-shaped faces helps to further distinguish a cut-out with bend from a louvre with or without cut-out. In the latter the normals are in the same direction whereas for a cut-out with bend, they are in opposite directions (see column 7, Table 2).

A composite feature (except a cut-out with bend) is subdivided into a flat part P_1 and deformed part P_2. The presence of a composite feature containing a cut-out is detected by comparing the width of the flat part D_1 with the width of the deformed part D_2, as shown in Fig. 13. If $D_1 > D_2$, it indicates that the thickness chain represents either a lance, jog, louvre or bridge with a cut-out.

4.3. Characteristics for deform features

The characteristic which distinguishes a deform feature from other feature categories is that none of the faces in the face chain belongs to a t-face group. One of the conditions to distinguish sub-classes of this feature is the termination criterion. For example, the extraction of a "dimple" is terminated when no more faces can be found at the next level (criterion 2) while a "flanged cut-out" is terminated when adjacent faces at the next level do

Fig. 12. Examples of ambiguous cases of slots when concavity is omitted in its feature definition.

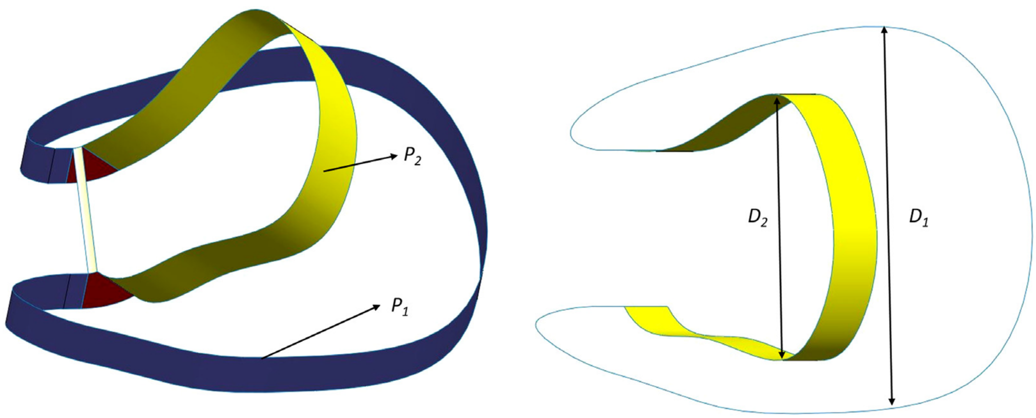

Fig. 13. Flat part P_1 and deformed part P_2 in a 3D feature.

Table 1
Characteristics matrix for different sub-classes of cut feature.

Sub-class category	Instances of Sub-classes	Type of chain	Total number of faces	Cylindrical surface number	Number of different cylindrical radius values	Total number of planar surfaces	Number of parallel planar face pairs
General hole	Circular hole	Closed	1	1	1	0	0
General hole	Slot	Closed	4	2	1	2	1
General hole	Key slot	Closed	4 or 5	2 or 3	2	2	1
General hole	Compound cut-out	Closed	≥2	Any	Any	Any	Any
Notch	None	Open	2	0	0	2	0
Clip	None	Open	1	0	0	1	0
Open slot	None	Open	≥3	≤1	≤1	≥2	≥1

Table 2
Characteristics matrix for composite feature.

Sub-class category	Number of internal thickness face chains	Number of boundary sub-chains	Number of fan-shaped faces	Number of fan-shaped face pairs	Fan-shaped faces share a common face	Normals to fan-shaped faces in opposite direction	$D_1 > D_2$
Shear with bridge	2	0	8	4	Yes	Yes	No
Cut-out with bridge	2	0	8	4	Yes	Yes	Yes
Shear with jog	1	0	≥4	≥2	Yes	Yes	No
Cut-out with jog	1	0	≥4	≥2	Yes	Yes	Yes
Shear with lance	1	0	2	1	Yes	Yes	No
Cut-out with lance	1	0	2	1	Yes	Yes	Yes
Shear with louvre	1	0	≥4	0	Yes	No	No
Cut-out with louvre	1	0	≥4	0	Yes	No	Yes
Cut-out with bend	1	0	≥2	0	No	Yes	N.A.
Bend	0	1	2	1	N/A	Yes	No
Cut-out with bend	0	1	2	1	N/A	Yes	Yes
jog	0	1	4	2	N/A	Yes	No
Cut-out with jog	0	1	4	2	N/A	Yes	Yes

not belong to the top s-face group (criterion 1). The shape of the inner loop on the mother face is another condition to distinguish these sub-categories. For example, the shape formed by the inner loop of edges associated with a dimple can be rectangular or circular; for a bead, the shape formed will be the same as a slot.

Properties satisfying the conditions mentioned above have been derived by studying the geometry and topology of surface face sets representing deform features. These properties are captured in a Characteristics Matrix, an example of which is given in Table 3 for some sub-categories of deform features. Since only some common deform features have been studied, the properties listed in this table may not be exhaustive. Suffice to say that the table in its present form serves as a template to characterise deform features.

4.4. Feature classification algorithm

In this section, the Characteristics Matrices can be applied to classify the face chains/face sets extracted by the different methods discussed in Section 3. For a face chain, F_i, if it is a cut feature, its characteristic properties are defined by:

$F_i\{P_1$ = Type of chain; P_2 = Total number of faces; P_3 = Cylindrical surface number; P_4 = Number of different cylindrical

Table 3
Characteristics matrix for deform feature.

Sub-class category	A set of interconnectedsurface faces	Termination criterion	Shape of starting inner loop
Flanged cut-out	Yes	1	Any
Dimple	Yes	2	Circular/rectangular
Bead	Yes	2	Slot-shaped

radius values; P_5 = Total number of planar surfaces; P_6 = Number of parallel planar face pairs}.

The face chain, F_i, can then be classified by matching this set of characteristic properties with every row of the Characteristics Matrix for cut features. For example, when it is being matched to the second row of the matrix for cut features (in Table 1), the format is:

If (P_1 = Closed AND P_2 = 4 AND P_3 = 2 AND P_4 = 1 AND P_5 = 2 AND P_6 = 1) then feature sub-category = Slot.

A similar process is followed for the bend and composite features. Thus, consider a face chain, F_j, determined to be a composite feature and having the characteristic properties obtained as:

F_j{P_1 = Number of internal thickness face chains; P_2 = Number of boundary sub-chains; P_3 = Number of fan-shaped faces; P_4 = Number of fan-shaped face pairs; P_5 = Do fan-shaped faces share a common face?; P_6 = Normals to fan-shaped faces in opposite directions; P_7 = D1 > D2}. The face chain, F_j, is classified by matching this set of characteristic properties to every row of the Characteristics Matrix or bend and composite features. For example, when it is being matched to the fifth row of the matrix for bend and composite features (in Table 2), the format is:

If (P_1 = 1 AND P_2 = 0 AND P_3 = 2 AND P_4 = 1 AND P_5 = Yes AND P_6 = Yes AND P_7 = No) then feature sub-category = Shear with lance.

The advantage of using this approach is that it facilitates extendibility as rows can be easily inserted into the Characteristics Matrix to represent new features. Whilst the four categories in the taxonomy are fixed, the sub-categories are extendible. For example, at the moment, slots, circular holes and keyhole slots have been cited as examples of the sub-category of the "cut" category. If there is a new feature in this sub-category, for example a T-slot, then the algorithm will not fail to extract it as it will be represented by a closed chain of thickness faces. However, it will not be able to characterise it as the geometrical properties of the T-slot will not match with those in any one of the existing rows in Table 1. The feature classification algorithm allows the user to append the properties of the chain of thickness faces, which represent the T-slot, as a new row in Table 1. This ensures that no feature (whether defined in the feature library or not) is missed during this classification process. This distinguishes the algorithm proposed in this paper from the approaches discussed in previous research [11,12,17,18].

5. Implementation and discussion

To test the proposed method for feature extraction and classification, a prototype system has been developed. The system has been developed in C++, and it runs in a Windows environment on a PC with an Intel i7 CPU and 16 GB RAM. Open CASCADE (OCC) is deployed as the geometric kernel [36] and the graphic user interface is built based on Microsoft Foundation Class package [37].

The workpiece in Fig. 14(a) was created by incorporating several different features listed in the proposed feature taxonomy shown in Fig. 2. The input data for this component was in STEP format AP203 and the computer program first extracted all the closed thickness face chains of which there were ten. One of them, i.e. the boundary thickness face chain, was further subdivided into five open sub-chains (*a*–*e*). The other nine closed thickness face chains were internal and are designated as *f*–*n*, see Fig. 14(b). In addition to these chains, two sets of surface faces were extracted, referred to as *p* and *q*.

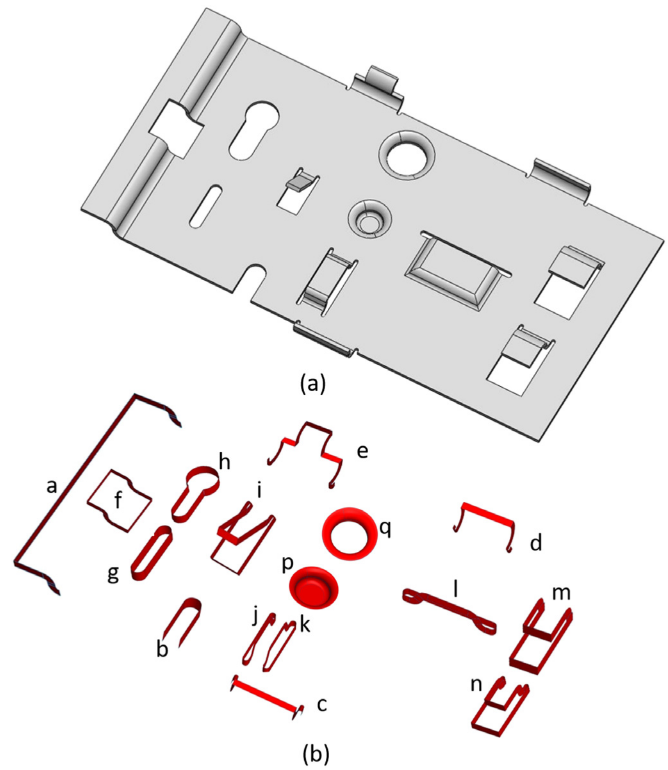

Fig. 14. Test workpiece, thickness face chains and face sets.

Next, the topological and geometrical properties of each chain were determined and compared with the characteristics listed in the matrices in Section 4, with a view to classifying them. To illustrate this procedure of feature classification, chains *f*, *j* and *k*, *n*, and face sets *p and q* are described in detail.

Chain *f*: As shown in Fig. 15(a), this internal chain contains four fan-shaped faces, f_1; f_2; f_3; f_4. However, these faces cannot be paired because they do not satisfy the criterion for pairing, i.e. they do not share a common face. Therefore, the characteristics of this chain can be represented as {1 0 4 0 No Yes N.A.} and they are identical to those in row 9 of the matrix in Table 2. Hence, this chain is labelled as representing a "Cut-out with bend".

Chains *j* and *k*: Both these chains (Fig. 15(b)) are also internal, have similar geometrical and topological characteristics and some of the edges share common faces. Therefore, these chains are paired together and they jointly represent one feature. This characteristic narrows the search to the first two rows of the matrix in Table 2. These two chains contain four pairs of fan-shaped faces, i.e. f_5–f_{p5}; f_6–f_{p6}; f_7–f_{p7}; f_8–f_{p8}, and these pairs share a common face. Next, the chain of thickness faces is split into two parts, a base part P_1 (the flat part between f_6–f_{p6} and f_7–f_{p7}) and a deformed part P_2 (the deformed part between f_6–f_{p6} and f_7–f_{p7}). If the width of these two parts is given by D_1 and D_2, and since in this feature $D_2 > D_1$, its characteristics can be

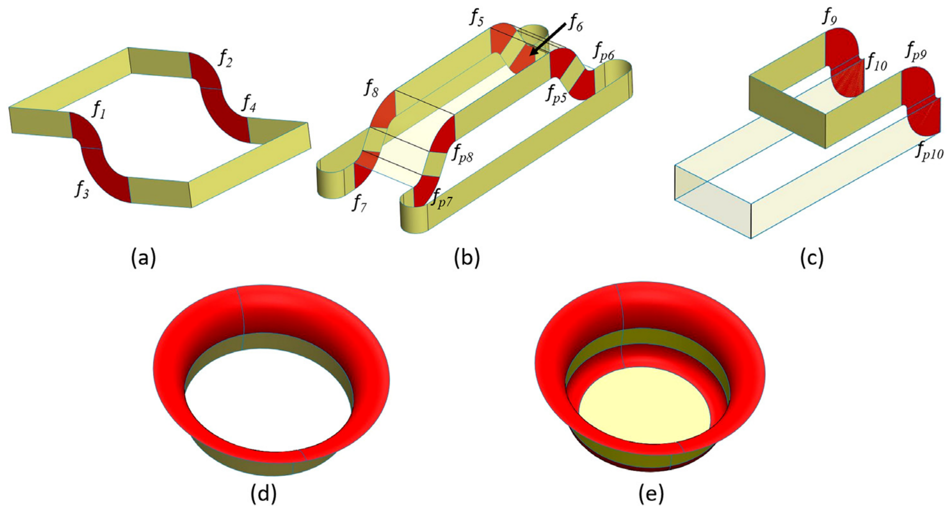

Fig. 15. Thickness-face chains *f*, *j* and *k* and face sets *p and q*.

represented in abbreviated form as {2 0 8 4 Yes Yes Yes} and the feature that these two chains represent is a "Cut-out with bridge".

Chain *n*: This is also a closed internal chain as shown in Fig. 15(c), and it contains two pairs of fan-shaped faces, i.e. f_9-f_{p9} and f_{10}-$f_{p10.}$ This chain is also split into two parts, P_1 and P_2, but in this case $D_2 = D_1$. Its characteristics are given by {1 0 4 2 Yes Yes No} which identifies it as "shear with jog".

Face set *p*: This set of interconnected surface faces (Fig. 15(d)) is terminated by criterion 1 because the faces at the next level did not belong to the top s-group. Furthermore, the shape of the loop of edges on the mother surface does not resemble that of a slot. Hence its characteristics are represented by {Yes 1 Any}, which classifies it as "flanged cut-out".

Face set *q*: This is also a set of interconnected surface faces (Fig. 15(e)). The extraction was terminated by criterion 1 because no more faces could be found. The shape of the loop of edges on the mother surface was circular and, therefore, its characteristics can be summarised by {Yes 2 Circular}, thus classifying it as a "dimple".

The topological and geometrical characteristics of the remaining chains were similarly determined and compared with those given in Tables 1 and 2, and they were classified as follows:

(i) chain *a* as a "boundary jog";
(ii) chain *b* as an "open slot";
(iii) chain *c* and *d* as "boundary cut-outs with bend";
(iv) chain *e* as a "boundary cut-out with jog";
(v) chain *g* as a "slot";
(vi) chain *h* as a "key slot";
(vii) chain *i* as a "shear with lance";
(viii) chain *l* as a "cut-out with louvre", and
(ix) chain *m* as a "shear with jog".

The results are graphically shown in Fig. 16; the left part of the window displays the workpiece with the extracted features shown in yellow. The classified list of features from which the user can select a feature is shown on the right. The selected feature is then displayed by the system in red.

To evaluate the efficacy of the proposed method, all the parts provided by the industrial partner were tested and Table 4 lists the computing times for nine of them, whereas Fig. 17 shows the computing times for different numbers of features and faces. The following observations can be made:

(i) The computing times for parts containing only 2D features are very small (parts 1 and 2).
(ii) It is not just the number of features that influences the computing time, but also the type of features and number of faces contained in the part. For example, comparing the computing times for parts 4 and 6, both contain a similar number of features, i.e. 14 and 15 features respectively, the computing time for part 6 is much less than that for part 4 because part 4 contains more faces and more 3D features than part 6.
(iii) The presence of 3D features has a greater effect on the computing time than the number of faces. This is understandable because such features contain fan-shaped faces which have not only more geometric parameters than 2D features but also require pairing. Consider the computing times required for parts 7, 8 and 9; it is the times for these parts which makes the latter half of the curve non-linear. This non-linearity is due to the increased number of 3D features as evidenced by the times required for processing parts 7, 8 and 9. These parts contain a similar number of faces (428 to 535) and 2D features (34 to 44), but the number of 3D features increases from 14 to 29. This increase causes the computing time to more than quadruple.

6. Conclusions and future work

This paper presents a feature recognition algorithm for sheet metal parts manufactured using progressive dies. From the work described in this paper, the following conclusions can be drawn.

(i) The feature recognition algorithm proposed in this paper relies on the topological and geometrical information extracted from a STEP AP203 format file. It is shown that thickness faces (t-faces), automatically extracted and clustered into chains, are sufficient to represent certain categories of sheet metal features. An important novelty of the proposed feature recognition approach is that it divides the recognition procedure into two stages i.e. feature extraction and classification, thus ensuring that all features can be extracted without considering what their name should be. Since the extraction of features does not rely on predefined patterns, it avoids the possibility of not detecting a feature which is not pre-defined and stored in the feature library.
(ii) Current feature recognition methods limit features that can be classified to those represented in fixed patterns of

Table 4
Computing times for different test parts.

No.	Entities					
	File size (kB)	Total faces	No of 2D features	No of 3D features	Total Features	Computing time (s)
1	29	22	12	0	12	0.347
2	144	100	26	0	26	1.453
3	270	168	10	7	17	1.981
4	358	246	3	12	15	4.584
5	375	251	23	8	31	8.816
6	532	138	5	9	14	2.353
7	682	428	44	14	58	10.614
8	804	533	34	19	43	11.745
9	1019	585	34	29	63	46.357

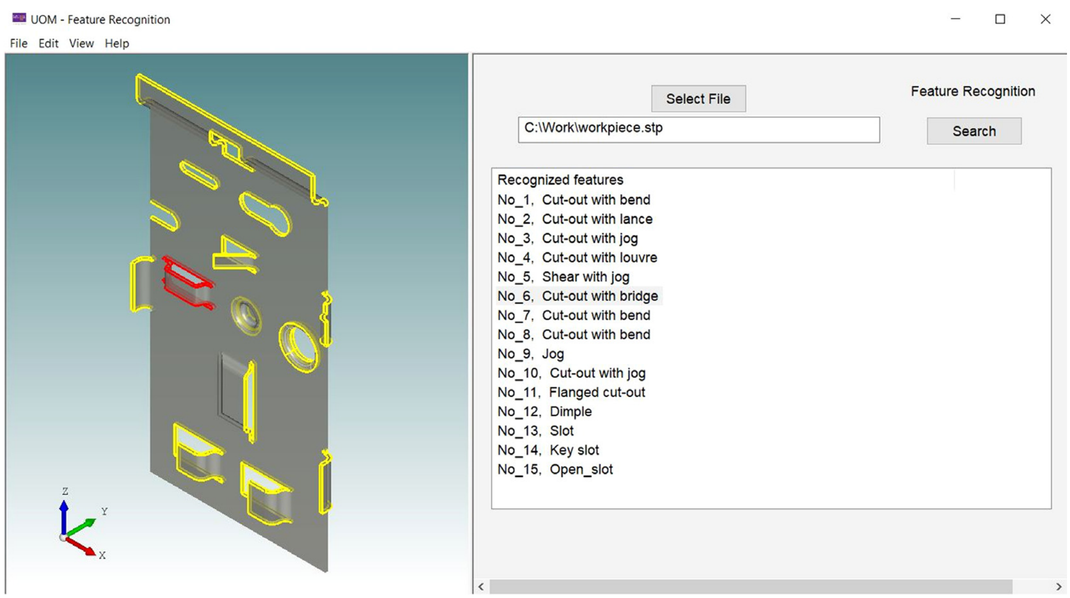

Fig. 16. Features displayed and listed in a graphic window. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

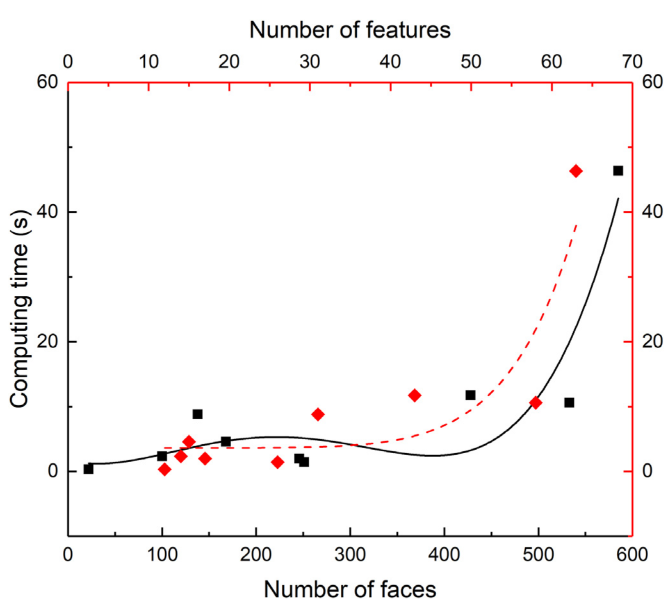

Fig. 17. Computing times for detection and classifying features in tested sheet metal components.

attributes. In the algorithm proposed herein, sets of properties sufficient to demarcate between feature categories have been identified, thus enabling any feature chain/set to be correctly classified. As a consequence, the approach proposed is more flexible and potentially extendible.

(iii) Using representative parts and their manufacturing technology, as an empirical grounding, an enhanced and flexible feature taxonomy has been proposed which can act as a template for a dialogue between content owners during alignment of diverse feature taxonomies. This allows the differences in feature taxonomies to be resolved without having to impose a unified feature taxonomy on content owners as proposed in some literature.

Several aspects of the feature recognition problem are proposed to be addressed in the next stage of this research project. Although the Characteristics Matrices developed are broadly invariant and flexible, they have heuristic elements and are currently hard-coded. It is necessary to develop a more universal and extendable library to classify the features from extracted face chains/sets, ideally having a self-learning capability.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors are thankful to CTC Pressings Ltd, Altrincham, UK and the Department of Mechanical, Aerospace and Civil Engineering, School of Engineering, The University of Manchester, UK, for financially supporting this project.

References

[1] Jackson C, Buxton M. The design reuse benchmark report. Seizing the opportunity to shorten product development, Boston, Aberdeen Group; 2007, p. 1–20.
[2] Liu J, Liu X, Ni Z, Zhou H. A new method of reusing the manufacturing information for the slightly changed 3D CAD model. J Intell Manuf 2018;29:1827–44. http://dx.doi.org/10.1007/s10845-016-1220-3.
[3] Vosniakos GC, Giannakakis T. A knowledge-based manufacturing advisor for pressworked sheet metal parts. J Intell Manuf 2013;24:1253–66. http://dx.doi.org/10.1007/s10845-012-0664-3.
[4] Kyprianou LK. Shape classification in computer-aided design (Ph.D. thesis), Cambridge, UK: University of Cambridge; 1980.
[5] Choi BK, Barash MM, Anderson DC. Automatic recognition of machined surfaces from a 3D solid model. Comput Aided Des 1984;16:81–6. http://dx.doi.org/10.1016/0010-4485(84)90164-7.
[6] Joshi S, Chang TC. Graph-based heuristics for recognition of machined features from a 3D solid model. Comput Aided Des 1988;20:58–66. http://dx.doi.org/10.1016/0010-4485(88)90050-4.
[7] Ferreira JCE, Hinduja S. Convex hull-based feature-recognition method for 2.5D components. Comput Aided Des 1990;22:41–9. http://dx.doi.org/10.1016/0010-4485(90)90028-B.
[8] Zhang CJ, Zhou XH, Li CX. Automatic recognition of intersecting features of freeform sheet metal parts. J Zhejiang Univ Sci A 2009;10:1439–49. http://dx.doi.org/10.1631/jzus.A0820705.
[9] Nnaji BO, Kang TS, Yeh S, Chen JP. Feature reasoning for sheet metal components. Int J Prod Res 1991;29:1867–96. http://dx.doi.org/10.1080/00207549108948055.
[10] Lentz DH, Sowerby R. Feature extraction of concave and convex regions and their intersections. Comput Aided Des 1993;25:421–37. http://dx.doi.org/10.1016/0010-4485(93)90004-8.
[11] Zhang WZ, Wang GX, Lu C, Nee AYC. A staged approach for feature extraction from sheet metal part models. Int J Prod Res 2007;45:3521–44. http://dx.doi.org/10.1080/00207540600806414.
[12] Kannan TR, Shunmugam MS. Processing of 3D sheet metal components in STEP AP-203 format. Part II: Feature reasoning system. Int J Prod Res 2009;47:1287–308. http://dx.doi.org/10.1080/00207540701510063.
[13] Jagirdar R, Jain VK, Batra JL. Characterization and identification of forming features for 3-D sheet metal components. Int J Mach Tools Manuf 2001;41:1295–322. http://dx.doi.org/10.1016/S0890-6955(01)00006-2.
[14] Kulkarni YH, Sahasrabudhe A, Kale M. Topological validation of midsurface computed from sheet metal part. Comput Aided Des Appl 2015;12:663–73. http://dx.doi.org/10.1080/16864360.2015.1033327.
[15] Salem AA, Abdelmaguid TF, Wifi AS, Elmokadem A. Towards an efficient process planning of the V-bending process: an enhanced automated feature recognition system. Int J Adv Manuf Technol 2017;91:4163–81. http://dx.doi.org/10.1007/s00170-017-0104-9.
[16] Woo Y. Abstraction of mid-surfaces from solid models of thin-walled parts: A divide-and-conquer approach. Comput Aided Des 2014;47:1–11. http://dx.doi.org/10.1016/j.cad.2013.08.010.
[17] Liu ZJ, Li JJ, Wang YL, Li CY, Xiao XZ. Automatically extracting sheet-metal features from solid model. J Zhejiang Univ Sci 2004;5:1456–65. http://dx.doi.org/10.1631/jzus.2004.1456.
[18] Gupta RK, Gurumoorthy B. Classification, representation, and automatic extraction of deformation features in sheet metal parts. Comput Aided Des 2013;45:1469–84. http://dx.doi.org/10.1016/j.cad.2013.06.010.
[19] Gupta RK, Gurumoorthy B. Automatic extraction of free-form surface features (FFSFs). Comput Aided Des 2012;44:99–112. http://dx.doi.org/10.1016/j.cad.2011.09.012.
[20] Sunil VB, Pande SS. Automatic recognition of features from freeform surface CAD models. Comput Aided Des 2008;40:502–17. http://dx.doi.org/10.1016/j.cad.2008.01.006.
[21] Nagarajan S, Reddy NV. STEP-based automatic system for recognising design and manufacturing features. Int J Prod Res 2010;48:117–44. http://dx.doi.org/10.1080/00207540701855419.
[22] Qi CR, Su H, Kaichun M, Guibas LJ. PointNet: Deep learning on point sets for 3D Classification and Segmentation. In: 2017 IEEE Conf. Comput. Vis. pattern Recognit.. IEEE; 2017, p. 77–85. http://dx.doi.org/10.1109/CVPR.2017.16.
[23] Zhang Z, Jaiswal P, Rai R. FeatureNet: Machining feature recognition based on 3D Convolution Neural Network. Comput Aided Des 2018;101:12–22. http://dx.doi.org/10.1016/j.cad.2018.03.006.
[24] Shi P, Qi Q, Qin Y, Scott PJ, Jiang X. A novel learning-based feature recognition method using multiple sectional view representation. J Intell Manuf 2020;31:1291–309. http://dx.doi.org/10.1007/s10845-020-01533-w.
[25] Cao W, Robinson T, Hua Y, Boussuge F, Colligan AR, Pan W. Graph representation of 3D CAD models for machining feature recognition with deep learning. In: Int. Des. Eng. Tech. Conf. Comput. Inf. Eng.. 2020, http://dx.doi.org/10.1115/DETC2020-22355.
[26] SolidWorks. 2020, http://help.solidworks.com/2020/english/SolidWorks/fworks/t_recognizing_features_automatically.htm?format=P&value= (accessed November 10, 2020).
[27] CATIA. 2020, https://www.3ds.com/products-services/catia/ (accessed November 10, 2020).
[28] Owodunni O, Hinduja S. Evaluation of existing and new feature recognition algorithms. Part 1: Theory and implementation. Proc Inst Mech Eng B 2002;216:839–51. http://dx.doi.org/10.1243/095440502320192978.
[29] Pratt MJ, Wilson PR. Requirements for support of form features in a solid modelling system. Comput Aided Manuf Int 1988.
[30] Gupta RK, Gurumoorthy B. Unified taxonomy for reference ontology of shape features in product model. IFIP Adv Inf Commun Technol 2013;409:295–307. http://dx.doi.org/10.1007/978-3-642-41501-2_30.
[31] Sanfilippo EM. Feature-based product modelling: an ontological approach. Int J Comput Integr Manuf 2018;31:1097–110. http://dx.doi.org/10.1080/0951192X.2018.1497814.
[32] Li H, Dragisic Z, Faria D, Ivanova V, Jiménez-Ruiz E, Lambrix P, et al. User validation in ontology alignment: functional assessment and impact. Knowl Eng Rev 2019;34. http://dx.doi.org/10.1017/s0269888919000080.
[33] Lai JY, Wang MH, Chiu YK, Hsu CH, Tsai YC, Huang CY. Recognition of depression and protrusion features on B-rep models based on virtual loops. Comput Aided Des Appl 2016;13:95–107. http://dx.doi.org/10.1080/16864360.2015.1059200.
[34] Dickinson HW, Rogers H. Origin of gauges for wire, sheets and strip. Trans Newcom Soc 1940;21:87–98. http://dx.doi.org/10.1179/tns.1940.007.
[35] Xu X, Hinduja S. Recognition of rough machining features in 2 1/2D components. Comput Aided Des 1998;30:503–16. http://dx.doi.org/10.1016/S0010-4485(97)00090-0.
[36] OPENCASCADE. 2020, https://www.opencascade.com/content/latest-release (accessed May 26, 2020).
[37] Prosise J. Programming windows with MFC. 2nd ed.. Redmond, WA: Microsoft Press; 1999.