



## Machining Feature Recognition from In-Process Model of NC Simulation

Shixin Xu<sup>1,2</sup>, Nabil Anwer<sup>1,3</sup> and Charyar Mehdi-Souzani<sup>1,3</sup>

<sup>1</sup>LURPA, ENS de Cachan 94235, France, [sxu@ens-cachan.fr](mailto:sxu@ens-cachan.fr)

<sup>2</sup>SMEA, Beihang University, Beijing 100191, China, [anwer@lurpa.ens-cachan.fr](mailto:anwer@lurpa.ens-cachan.fr)

<sup>3</sup>University Paris XIII, France, [souzani@lurpa.ens-cachan.fr](mailto:souzani@lurpa.ens-cachan.fr)

### ABSTRACT

NC simulation emulates the CNC machine tool and the cutter move along its axis. It depicts the material removal to better visualize the machining process. Until now, its main goals are still confined to check an unproved NC program to avoid potential collisions, and to analyze undercuts and overcuts. In this work, by using the simulated output—the in-process model (IPM), and rebuilding a machining feature based model, applications of NC simulation are extended to establish the feedback link from CNC to CAM, which can automatically pass modifications of manufacturing engineers back through the digital chain. In cases when old CAD archives of parts are incompatible with new systems and machines, or are damaged or lost, the original CAD model is neither available nor usable, and the part program is the only data available, rebuilding the feature based model from the NC simulation is necessary. This paper proposes an approach for machining feature recognition from IPM. As per the IPM characteristics, the approach adopts a novel approach of curvature based region segmentation and valuated adjacency graph based feature recognition.

**Keywords:** machining feature recognition, in-process model, reverse engineering.

### 1. INTRODUCTION

Feature oriented approaches have been widely used in various phases of product life cycle, such as design, manufacture, inspection; and are adopted by many standards, such as STEP, and STEP-NC. Object oriented technology promotes greatly the application of feature modeling since the two combined can be better integrated in computerized systems, easing data adaptation and transfer. For deploying the feature oriented approach, it is necessary to establish a feature based model, such as the machining feature based model corresponding to the Computer-Aided Manufacturing (CAM) field. Machining feature recognition is one of the most important methods to establish a machining feature based model. It should be pointed out the differences between design features and machining features. Primary machining features are depressions in a part's geometry, where the material is machined off. A protrusion, such as a boss, in a part can be a normal design feature, whereas, it can only be an attached feature inside a primary machining feature, such as a pocket or a planar face.

Compared to the reverse engineering of the part's CAD model, which uses a coordinate measuring apparatus to acquire the shape, using the in-process model (IPM) of NC simulation can establish more accurate and reliable machining feature based model since there is little influence of simulation deviation. Conventionally, the data source of machining feature recognition is the part's CAD model, usually a B-Rep solid model. But the IPM of NC simulation is mostly collections of unordered set of triangles, usually in STL format.

NC simulation is the basis of virtual machining, it emulates the CNC machine tool and the cutter move along its axis, and depicts the material removal to better visualize the machining process. It can be performed on the CNC controller of a shopfloor machine, or on an offline computerized system. Until now, its main goals are still confined to check an unproved NC program to avoid potential collisions with fixture elements, making undercuts, or leaving excess material. In our work, by making use of the simulated output—the IPM, and re-establishing a machining feature based model, we extend the applications

of NC simulation to build the feedback link from CAM/CNC systems or shopfloor amendments back to CAM/CAPP/CAD systems. This feedback link is very useful because the part program is often changed by machining engineers according to the simulation results or the shopfloor situation. So the current unidirectional information flow from CAM to CNC can be changed into bidirectional one. The re-established feature model can serve for design improvements since we can compare this model with the original designed model, measure more reliable and accurate dimensions on this model, or use it for FEA, etc.

In addition, in some situations, the original CAD model is neither available nor usable, and the NC program is the only data available for producing a part. This could happen when old CAD archives are incompatible with new systems or machines, or they are damaged or lost, or when parts are designed and machined only at shopfloors without aid of CAX software, etc. [18]. Hence rebuilding the feature based CAD model from the IPM is necessary under these circumstances. It would save much resource than designing a feature based model from scratch.

Many research works have contributed to feature recognition (FR) [10]. Common FR methods consist in searching a solid's boundary for pattern of faces and edges that obey certain topological or geometrical relationships. These overlapping methods use graph matching, volume decomposition and attribute adjacency graph, to infer machining information. One common FR practice starts from importing a neutral data format (STEP, IGES, etc) file. The file is then translated into a part representation suitable for form FR. Bhandarkar [6] presented an FR system based on STEP AP224. Among graph-based FR methods, the graph pattern analysis method was first formalized by Joshi [12]. The part B-Rep is translated into a graph where, e.g., its nodes represent faces and its arcs represent edges. Additional information such as edge-convexity is incorporated into the graph. The graph is decomposed into subgraphs, which are then analyzed to determine their feature types. A graph search algorithm was proposed by Corney [10] for depression and protrusion recognition, where input is limited to "single-sided" machined parts which can be completely machined with one machining setup. The approach used by Anwer [2,4] is graph-based and process planning oriented. The subgraph structure of extracted form features allows a classification based on graph theory. Moreover, machinability analysis is considered for a classification based on characteristics of machining tool paths. Seven generic form feature classes are then proposed (passage, Depression, K-Protrusion, K-Slot, Chain, Compound and Degenerate). Graph-based methods are quite successful in recognizing isolated and compound features, but they have revealed difficulties when dealing with complex interacting features, and have few guarantees to manufacturability for recognized features. Convex

hull and cell-based decomposition, called volumetric decomposition approaches [10], are two approaches based on decomposing the input model into a set of intermediate volumes and manipulating the volumes in order to produce features. The problem with these approaches is that they are inherently based on polyhedral representations. No robust method, justifiable from a manufacturing point of view, has been developed to manipulate the intermediate volumes created by these initial steps. Hint-based FR approaches were proposed [15] to handle intersecting features, but their major problem is the difficulty in generating desirable interpretations. A method to tackle this problem is to pursue an optimal interpretation by incorporating some manufacturing knowledge into FR processes.

Feature recognition from NC programs relies on three different strategies, reported in only a few literatures: (a) FR based on machining simulation from G-code; (b) FR based on Z-map; and (c) FR based on G-code and cutting tools information. For the first strategy, a polyhedral CAD model is built from NC simulation data. Many researches focused on reconstruction of surfaces or B-Rep solid model from triangular meshes or polygons, octree representation, etc. But only few went further to recognize features from the built models. The work done by Sunil [14] enables to build a surface model from an STL file for feature recognition of sheet metal parts. The Z-map based approach was proposed by Yan [16]. This approach is only used for 3-axis milling and drilling. The types of recognized features are mainly 2 1/2D features. The other drawback is the considerable computing involved in the processing of Z-map updating for cut contour extraction. The approach based directly on G-code and cutting tools is realized according to machining operation types by analyzing G-code, in addition, types and geometry of the cutting tools [17] are also considered. Using the available informations, the feature profile can be obtained, and then the machining feature can be recognized. This approach is so far confined to handle turning features or 2 1/2D features for prismatic parts.

In the work presented here, the first strategy is adopted. The data source is IPM of NC simulation and the geometric model is a solid model. The remainder of the paper is organized as follows: Section 2 gives an overview of the developed method, followed by general strategies discussed in Sections 3-5. Finally, we draw conclusions in Section 6 and give some directions for further research.

## 2. MACHINING FEATURE RECOGNITION FROM NC SIMULATION DATA

The overall method is shown in Fig. 1.

The widely used NC simulation approach is the mesh-based virtual machining, whose output data of

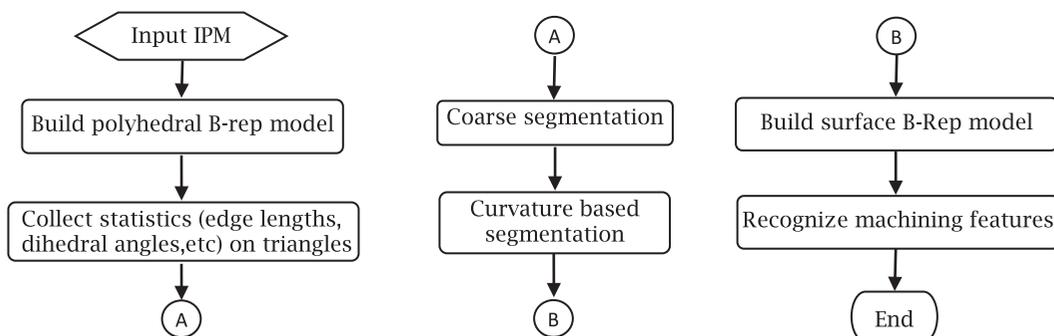


Fig. 1: Diagram of machining feature recognition from IPM.

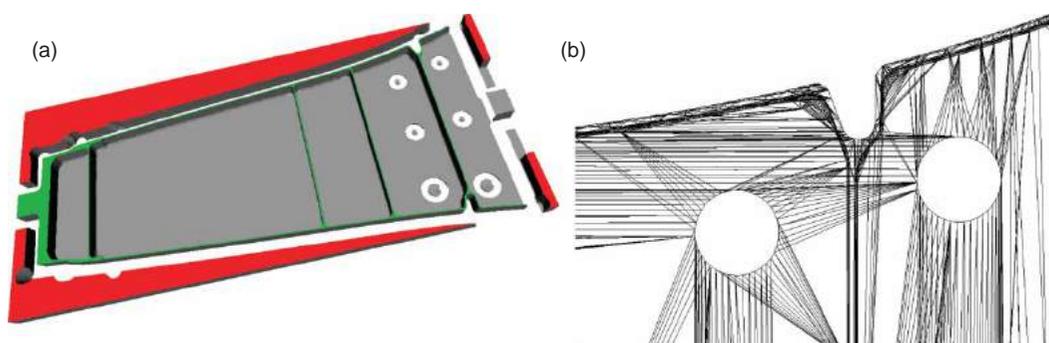


Fig. 2: The IPM of NC simulation (from left): (a) IPM with the offcut, (b) Partial view of the STL mesh.

the IPM is a collection of triangles, without topological information. The IPM contains unwanted data of the offcut (Fig. 2). The mesh data usually contains degenerated triangles (i.e. isolated vertices and edges) and incorrectly oriented triangles, or lacks of triangles data for building a solid. Therefore the first issue is to build a polyhedral solid model with a boundary representation from the IPM. The specific tasks include elimination of the offcut data, topology construction and model healing.

Because of the lack of adjacency relationships, the face based representation of polygonal mesh demonstrates its inconvenience and limitations for geometry computing. Instead, the boundary representation is given increasing considerations. A boundary representation of a polyhedral surface consists of a set of vertices, a set of edges, a set of facets and incidence relations on them. The half-Edge Data Structure (HDS), which is adopted in this work, provides an efficient way for boundary representation of an oriented manifold polygon surface and also provides a convenient way to query vertices, edges, and facets.

The second issue is to perform coarse region segmentation. A region is a set of connected triangles having certain geometrical attributes (curvature, surface normal, surface type). First, simple statistical computation on triangles, such as triangle area, edge length, dihedral angle between two triangles are computed. These data can be used to set some thresholds and seed triangle selection for subsequent searching.

Second, coarse segmentation is done by identifying the boundary sharp edges in the built model. The third issue is curvature based region segmentation, during which triangles are combined then segmented into meaningful regions using a discrete shape recognition and segmentation method. The method can be decomposed into three main phases [21]: discrete shape recognition, vertex clustering and connected region generation. The fourth issue is to build a surface B-Rep model from the reconstruction of the previous identified regions. Finally, machining feature recognition is based on face adjacency analysis.

### 3. MESH BASED MACHINING SIMULATION

Mesh based machining simulation refers to that the IPMs of NC simulation systems use triangle meshes for processing and dynamic display, and therefore the IPMs can be output as collection of triangles or polygons. Since this work concerns only geometrical aspects of machining simulation, we only consider the geometric NC simulation, which is used to check whether the cutting tools interfere with fixtures, workpieces and machine tools, gouge the part, or leave excess stock behind. Main methods among geometric simulations are B-Rep based, object space based (further divided into Z-map/vector/octree-based), and image space-based [19]. One popular type

of B-Rep based simulation is triangular polyhedral B-Rep (e.g. CATIA, UG, Pro/E adopt this approach). The objects' boundaries of the simulation environment are represented by a collection of triangles. In an object space-based simulation, parts are represented by a collection of discrete points (with vectors) or surfaces with vectors or certain volume elements. The IPMs of object space-based simulation can be obtained easily as collection of triangles or polygons directly or indirectly. Image space-based simulation is the fastest among all simulation methods (e.g. CGTech's VeriCut uses such an approach.). In this approach, the workpiece and the cutter are discretized along z-axis into small rectangular bars, each of which corresponds to a pixel in the display screen. A rectangular bar and its data structure together are called a dixel (depth element). The upper surface of the dixel is divided into two triangles along the diagonal line of the rectangle, and all these triangles form a mesh representing the machining surface of the workpiece. During the NC simulation, the heights of the vertices of a dixel are updated along the tool path, and the rendering of the machined surface is refreshed. In the course of simulation, the IPMs can be output as triangle meshes, usually in STL format or in VRML format. Nowadays most of commercial NC simulators provide such an interface.

The initial STL file has one or more components, of which one is the part, and the others are offcuts. We process the STL data to build a B-Rep triangular model. At the beginning of the processing, users are required to select the offcuts interactively to delete the corresponding offcut data from the STL file. Then, in our test, the CASCADE [7] toolkits, an open source geometry development platform, are used to build the B-Rep triangular model from the STL file. Each triangle in the STL file is transformed into a triangular face. When completed, a manifold B-Rep triangular solid model is generated. This model has the following characteristics: In ruled surface (cylinders/cones, etc) regions and in planar regions at its curved boundaries, the model has many long and thin triangles

resulting from the discretization; vertices of triangles in planar and ruled regions are mostly on the part's boundary edges; triangles in flat regions are sparse and relatively large, while dense and uniform in highly curved regions; triangles between two curved edges, typically of ruled regions, are usually connected in a sawtooth fashion (Fig. 2b).

## 4. SEGMENTATION

### 4.1. Coarse Segmentation

Discrete curvature calculation is based on vertex vicinity. In triangle sparse areas, most of vertices on boundary edges, belong to two or more regions. Curvature calculation would converge with large deviation, resulting mis-segmentation. Zhao [20] suggests increasing triangle density by adding vertices to avoid this problem. We use coarse segmentation by detecting the part's sharp edges. After the B-Rep triangular model construction (Fig. 3a), we compute the dihedral angle between two adjacent triangles. All the sharp boundary edges can be found by a threshold on dihedral angle (Fig. 3b, the red edges). Using a propagating approach, preliminary regions demarcated by sharp edges can be obtained. Follow-up curvature calculation will be confined to use vertices only within such a region. As thus curvature calculation is improved and is more reliable since the neighborhood of each vertex is not disturbed by vertices in other regions. This approach is easier and involves less data compared to the mesh densifying suggestion. It should be pointed out that in a preliminary region, there are smooth boundary edges, which will be separated from internal edges in further region segmentation.

### 4.2. Curvature Based Region Segmentation

On a surface the local shape around its point P is characterized by maximum and minimum principal curvatures ( $\kappa_{\max}$  and  $\kappa_{\min}$ ) and by the two

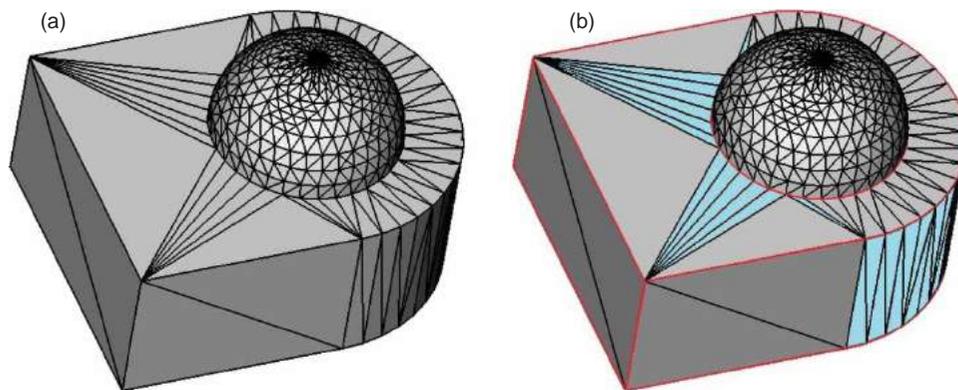


Fig. 3: Coarse segmentation (from left): (a) Example part, (b) Identified preliminary regions in red.

principal directions ( $d_{\max}$  and  $d_{\min}$ ) corresponding to the tangent vectors for which the principal curvatures are obtained. The method used here adopts Cohen-Steiner's [8] and Zhao's [20] to compute the curvatures of a discrete shape. Shape index and curvedness [9,13] are two indicators derived from the principal curvatures. Shape index, ranged  $[-1, 1]$ , is a quantitative measure of local surface type, and is defined as:

$$S = -\frac{2}{\pi} \arctan\left(\frac{\kappa_{\max} + \kappa_{\min}}{\kappa_{\max} - \kappa_{\min}}\right) \quad (1)$$

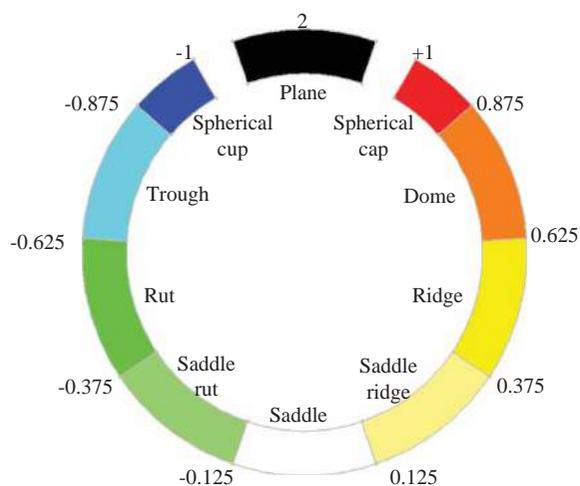


Fig. 4: Surface types, shape index scales and color scales.

Curvedness is a positive value that specifies the amount or intensity of the surface curvature.

$$C = \sqrt{\frac{\kappa_{\max}^2 + \kappa_{\min}^2}{2}} \quad (2)$$

Koenderink [13] defined 9 basic surface types in a continuous way using shape index. Here we add the planar shape a shape index equal to 2 [1].

10 surface types (spherical cup, trough, rut, saddle rut, saddle, spherical cap, dome, ridge, saddle ridge, and plane) are defined based on shape index and curvedness (Fig. 4). Every vertex, except vertices of sharp edges, of the shape is assigned a surface type label during local surface type recognition. We use the curvedness value to detect smooth edges within a preliminary region. Vertices on smooth edges and transition regions have higher curvedness value, as thus a threshold can be given to detect them. This method is easy to be realized and does not rely on triangles' shape [14].

Then connected regions are generated from these clustered vertices. Two operations are performed: connected region growing to generate initial segmentation result, and region refining, which aims to reduce over-segmented regions and to improve the segmentation result. When a vertex yet unassigned with a region label is met, the vertex is marked with a region label by its neighbor condition. With this associated region label, the vertex is added to an existing region or creates a new region. We adopt the approaches reported in [1,20,21] for the operations. Fig. 5 shows the flowchart of segmentation

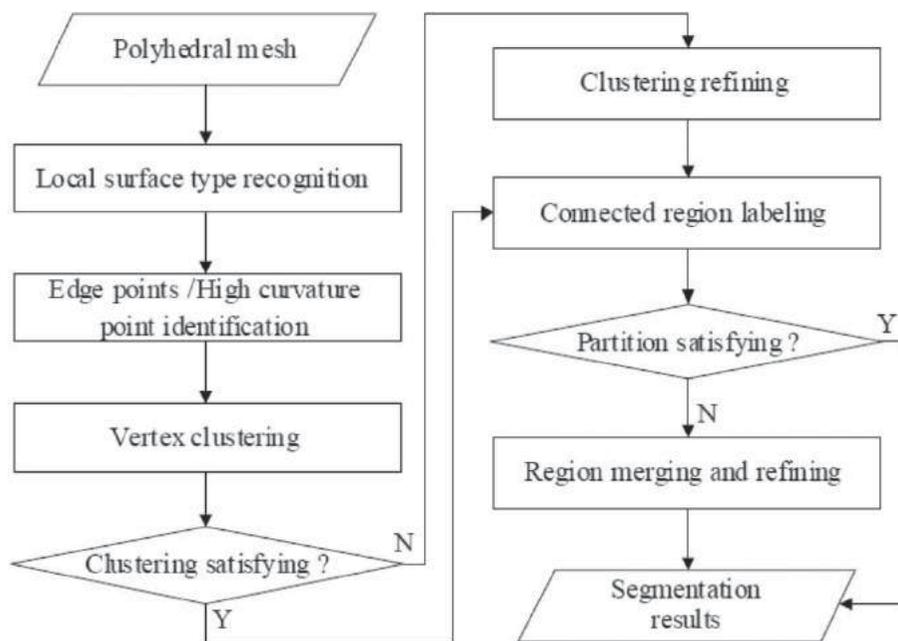


Fig. 5: Procedures of segmentation.

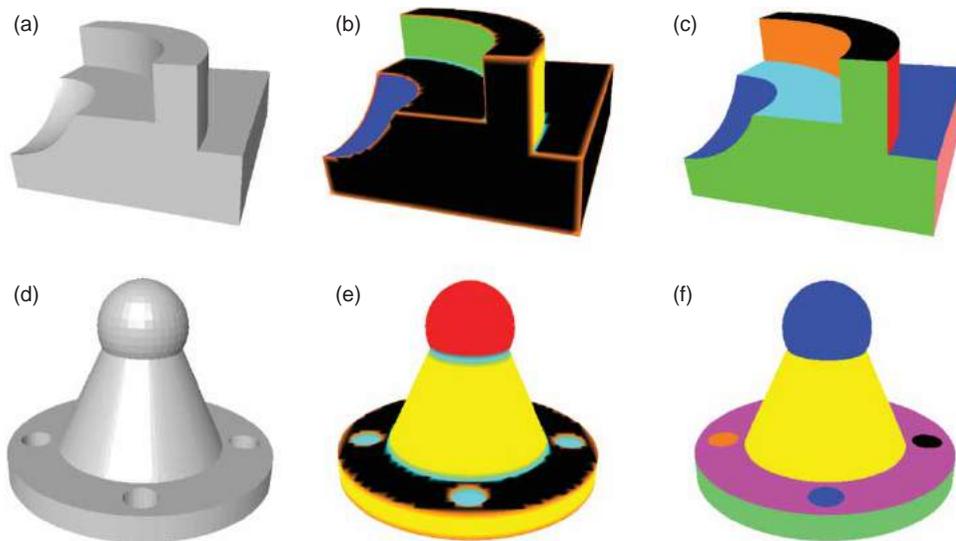


Fig. 6: Segmentation examples: (a) Discrete shapes, (b) Surface type recognition, (c) Segmentation.

procedures. Fig. 6 shows segmentation examples of two discrete shapes.

## 5. MACHINING FEATURE RECOGNITION

### 5.1. Surface B-Rep Model

Two methods can be used to obtain the boundary representation [1]. The first method enables construction of a surface for each mesh. It is simple while practical especially when the number of the triangles is small. The second method utilizes the additional information of the meshes. The additional information help to know which meshes belong to the same surface and what is the type of the surface to be constructed.

Some vertex regions satisfying certain geometric criteria can be merged in order to reduce the number of local regions. For instance, if two adjacent regions are of the same type and the common boundary is smooth, then they can be merged into one region. Then, surface association is used to fit a parametric or implicit surface to each region.

Edge loops that form the surface are identified and built. This can be done using the topology of the already built polyhedral B-Rep model. The process starts with finding a boundary edge whose vertices are on the surface. Following the first edge direction, find the next edge that links the first edge and is on the surface. We repeat the similar search until all edges that can form a loop are found. CASCADE [7] is used to build the curve from these edges, and all loops of the surface can be built in this way. Thus a continuous surface of the part is built. When all regions are exhaustively searched, the part's surface B-Rep model is built. At present the surface types of the surface B-Rep model can be planar faces, spheres, cones/cylinders and tori.

For planar regions the techniques of plane fitting are used. In a spherical cup/cap region (For true spheres,  $|s| = 1$ ), the sphere's center  $(x_c, y_c, z_c)$  and radius  $r$  are approximated using equation (3). A regression with a least squares method is also carried out in this case.

$$x^2 + y^2 + z^2 + xA_1 + yA_2 + zA_3 + A_0 = 0 \quad (3)$$

where:  $A_0 = x_c^2 + y_c^2 + z_c^2 - r^2$ ,  $A_1 = -2x_c$ ,  $A_2 = -2y_c$ ,  $A_3 = -2z_c$

The rut/ridge regions can be fit with a cylinder/cone (For true cylinder/cones,  $|s| = 0.5$ ). For each region of this type, a cylinder/cone is approximated by using a Gaussian image. A cylinder/cone's Gaussian image is a circle. The normal of the plane which fits this Gaussian image corresponds to the direction of the axis. It allows us to obtain the axis and also the angle for the cone. To determine the cylinder radius, the region points are projected onto the approximated plane and form a circle which corresponds to a cylinder circle, i.e. with the same radius and the center equivalent to an axis point.

Other vertex regions might be a portion of a torus. Take an outside torus for example, it has saddle regions, saddle ridge regions, dome regions, etc. But on ordinary CAD parts, usually a portion of a torus surface is used for blending cylinders and planes. So a vertex region should correspond to a portion of a torus, not a whole torus. All vertices have a common principal curvature corresponding to minor radius. The second principal can be used to calculate major radius and the torus center [5].

The remaining regions, which are hard to be represented by regular quadric surfaces, could be fitted using general parametric surfaces such as NURBS [5].

## 5.2. Adjacency Graph Based Feature Recognition

Feature recognition is performed on the surface B-Rep model obtained above. Here we adopt the valued-adjacency graph approach [2-4]. In this approach, the B-Rep model is converted into a planar graph where, e.g., its nodes represent faces and its arcs represent edges. Additional information, such as edge-xity (a measure of concavity/convexity), is incorporated into the graph. The xity  $m$  is defined as follows:

$$m = (\vec{n}_1 \times \vec{n}_2) \cdot \vec{a}_1 \quad (4)$$

where:  $\vec{a}_1$  is the orientation of the common edge  $e$  of face  $F_1$ .  $\vec{n}_1$  and  $\vec{n}_2$  are the normals at point  $P$  of  $e$  for  $F_1$  and  $F_2$ , respectively (see Fig. 7).

In this work, a feature is a set of faces of the boundary representation of a workpiece. These faces

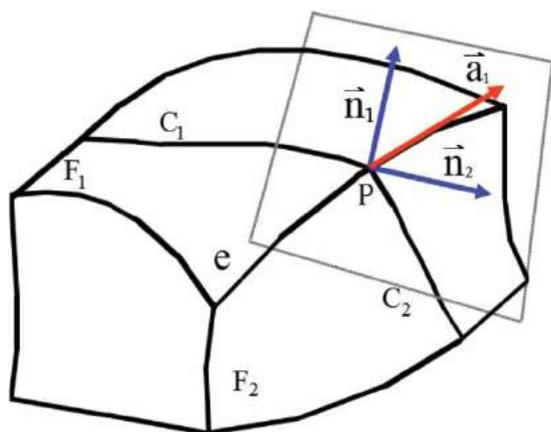


Fig. 7: Definition of the edge-xity.

have topological links of concavity that form cavities or protrusions and are related to a machining volume. For each form feature, it is possible to define a set of convex edges that are the limits of the feature: boundary edges. Note that the definition of a feature is only based on its topology.

In order to reduce the complexity of the graph, some general rules are applied. For example, a face whose incident edges are all convex does not form part of a feature, and it can be deleted from the graph. Anwer [2] pointed out that the convexity value of a curved edge is variable, and this property can be applied to recognize complex features.

Seven generic classes for form feature classification have been proposed [2-4]. General and specific rules to define the classes are based on properties of planar graphs [2].

Many research works have concerned feature classification for process planning [4,10]. The approach used in this work is manufacturing oriented. The subgraph structure of extracted features allows a classification based on graph theory. A machinability analysis is considered for a first classification based upon characteristics of machining tool paths (machining operation, accessibility, cutting tool geometry, etc.) and five generic classes have been proposed (Fig. 8):

- a) Passage: It consists of wall faces connected to each other by concave or convex edges and two or more entrance faces.
- b) Depression: It consists of a bottom face connected to all the wall faces by concave edges. The wall faces are connected to each other with either convex or concave edges.

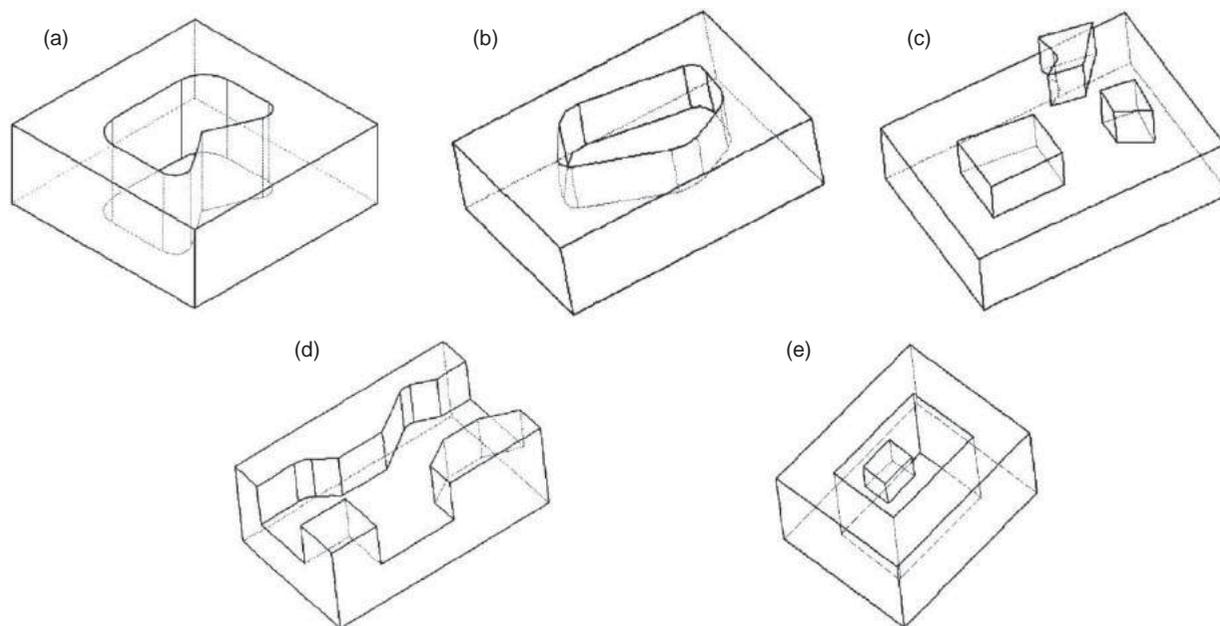


Fig. 8: Form feature generic classes: (a) Passage, (b) Depression, (c) K-Protrusion, (d) K-Slot, (e) Compound.

- c) K-Protrusion: It consists of a base face connected to all the wall faces by concave edges. The wall faces are linked to each other with either convex or concave edges that form inner protruding volumes.
- d) K-Slot: It consists of one or more base face(s) which is/are connected to all the wall faces by concave edges. The wall faces are linked to each other with either convex or concave edges that form sides around the base face(s).
- e) Compound: It gathers features that do not belong to any of the above classes and can be broken down into them.

K-Protrusion class definition depicts formal rules (Fig. 9c) that have been developed with strong emphasis in graph theory. The classification rule consists of the identification of a base face which is connected to all the wall faces by concave edges. The wall faces are linked to each other with either convex or concave edges that form inner protruding volumes (Fig. 9a). The rule can be easily described using cyclomatic and co-cyclomatic numbers of extracted planar graphs [2]. Using this approach machining features can be recognized. Their types and data structures adopt STEP-NC's definitions (also referred as STEP-NC features), so that the recognized feature data can be

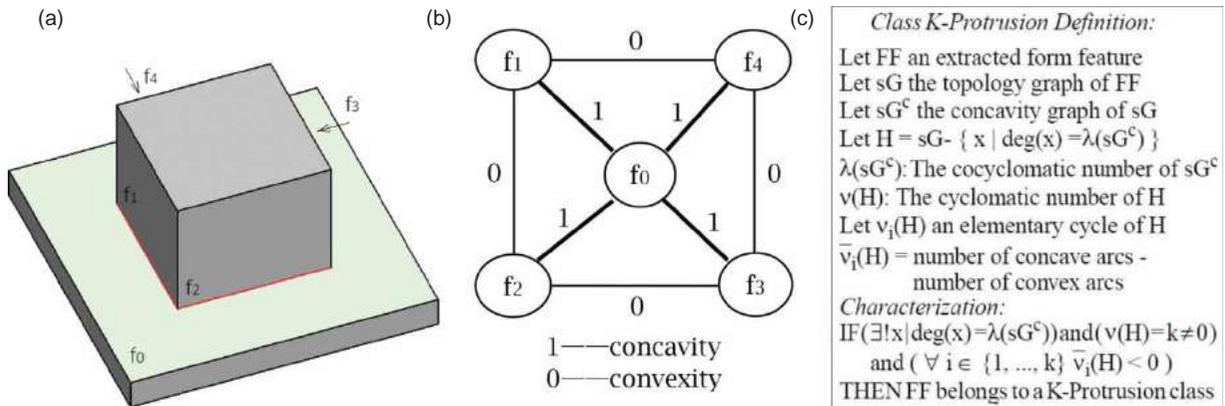


Fig. 9: Adjacency graph (from left): (a) A part, (b) The adjacency graph, (c) K-Protrusion classification rule.

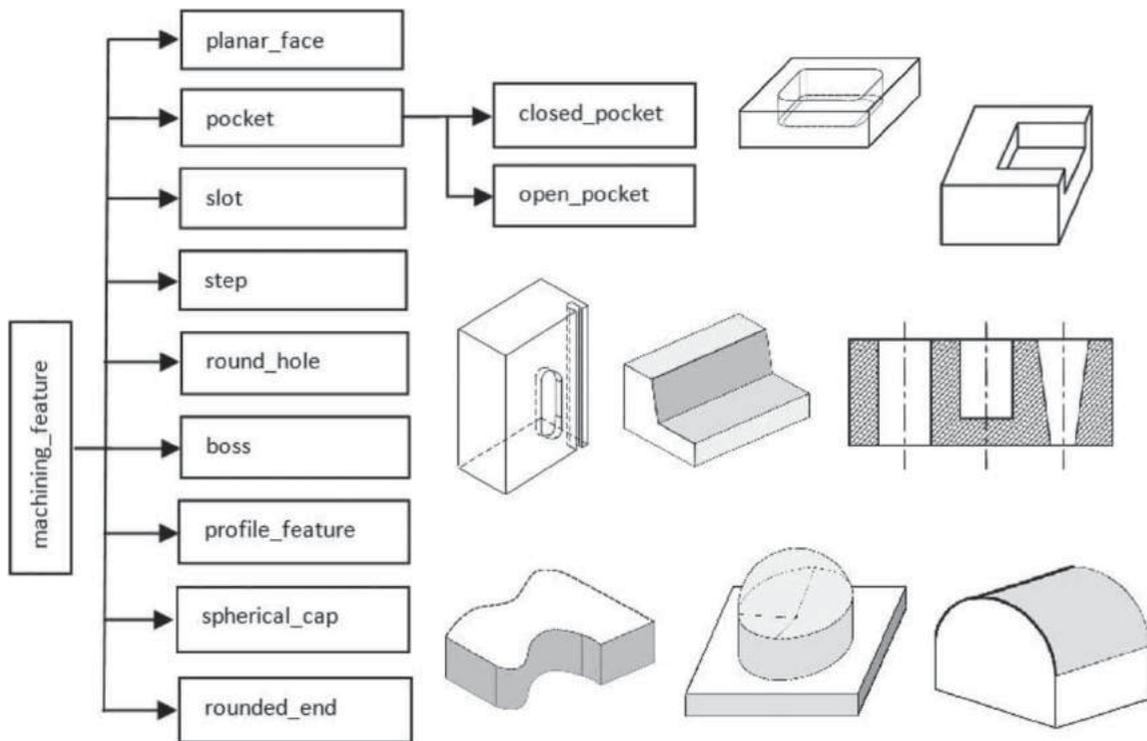


Fig. 10: STEP-NC features (adapted from [11]).

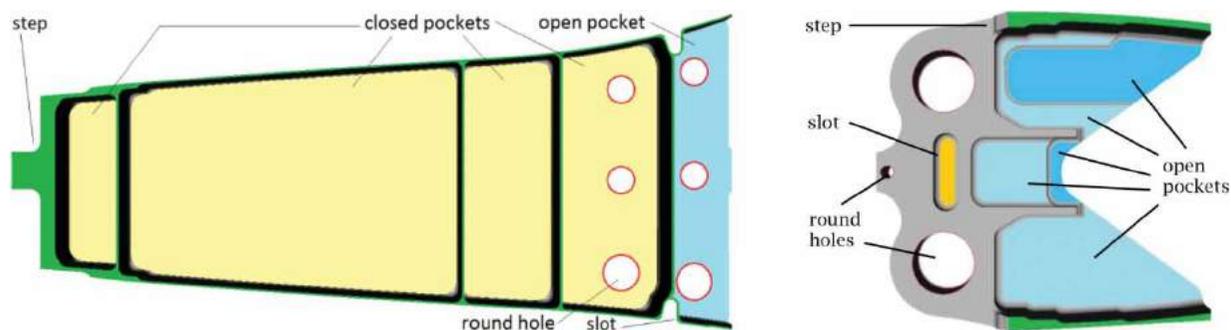


Fig. 11: Test parts.

stored in conformance to STEP-NC [11]. So far the types of machining features that can be recognized are closed pockets, open pockets, slots, steps, holes (Fig. 10).

A machining feature is then regarded as a set of connected faces satisfying certain geometrical and topological conditions. For each feature type, its feature template is defined. A graph-based and rule based manufacturability analysis approach is used for validating the results. For example, as for the manufacturability of a step, multiple interpretations due to more than one valid machining direction are available in the extracted feature data.

Figure 11 shows some feature recognition results for two test parts.

## 6. CONCLUSION

In this research an approach has been proposed for machining feature recognition from the In-Process Model of NC simulation. This approach involves the following procedures: (a) B-Rep polyhedral model construction from the In-Process Model, an STL file; (b) Curvature based region segmentation; (c) Surface B-Rep model construction; (d) Adjacency graph based machining feature recognition. This approach is designed according to the characteristics of the In-Process Model. The preliminary work shows that the roadmap that is “curvature-based segmentation, classification, then machining feature recognition” for this type of meshes is viable and worth trying. An improvement of the approach could be to fit general parametric surfaces such as NURBS in the dense triangle region of the In-Process Model, meanwhile performing the freeform volumetric feature recognition based on the discrete curvatures of those vertices in this type of regions. Future works include recognition of freeform volumetric features; developing specific approaches of region segmentation for other quadric surfaces and improving the robustness of region segmentation algorithms and extending rules to recognize more complex features and testing the approach on more complex parts (five axis and mill-turn). The potential applications of this work are also to bridge

the gap between solid model feature recognition techniques and surface reconstruction approaches for the purpose of reverse engineering.

## ACKNOWLEDGEMENTS

This work is a part of the ANGEL FUI project recently funded by the French Inter-ministerial Fund and endorsed by top French competitiveness clusters (SYSTEMATIC PARIS REGION, VIAMECA and ASTECH).

## REFERENCES

- [1] Anwer, N.; Yang, Y.; Zhao, H.; Coma, O.; Paul, J.: Reverse engineering for NC machining simulation, In IDMME'2010-Visual Concept 2010, Bordeaux, France, 2010.
- [2] Anwer, N.: Méthodologie d'analyse de raisonnement pour la génération automatique des gammes d'usinage en fraisage, PhD thesis, ENS de Cachan, France, 2000.
- [3] Anwer, N.; Chep, A.; Anselmetti, B.: Feature modeling and machinability analysis for CAD/CAPP Integration, IEEE Int. Conf. on Intelligent Engineering Systems, INES98, Vienna, Austria, 1998.
- [4] Anwer, N.; Chep, A.: HERA: A knowledge modeling scheme for CAPP, Second International workshop on intelligent manufacturing systems, IMS'99, Leuven, Belgium, 1999.
- [5] Bénéière, R.; Subsol, G.; Gesquière, G.; Breton, F.; Puech, W.: A comprehensive process of reverse engineering from 3D meshes to CAD models, *Computer-Aided Design*, 45, 2013, 1382-1393. <http://dx.doi.org/10.1016/j.cad.2013.06.004>
- [6] Bhandarkar, M.P.; Nagi, R.: STEP-based feature extraction from STEP geometry for agile manufacturing, *Computers in Industry*, 41, 2000, 3-24. [http://dx.doi.org/10.1016/S0166-3615\(99\)00040-8](http://dx.doi.org/10.1016/S0166-3615(99)00040-8)
- [7] CASCADE, <http://www.opencascade.org>, Open CASCADE Technology.
- [8] Cohen-Steiner, D.; Morvan, J.-M.: Restricted Delaunay triangulation and normal cycle, In 19th Annual ACM Symposium on Computational Geometry'03, 2003, 312-321.

- [9] Digne, J.; Morel, J.-M.; Mehdi-Souzani, C.; Lartigues, C.: Mesh segmentation and model extraction. *Curves and Surfaces; Lecture Notes in Computer Science*, Vol.6920, 2012, 236–252
- [10] Han, J.-H.; Pratt, M.; Regli, W.C.: Manufacturing feature recognition from solid models: a status report, *IEEE Transactions on Robotics and Automation*, 16(6), 2000, 782–796. <http://dx.doi.org/10.1109/70.897789>
- [11] ISO 14649 Part 10: General process data, 2002.
- [12] Joshi, S.: Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer-Aided Design*, 20(2), 1988, 58–66. [http://dx.doi.org/10.1016/0010-4485\(88\)90050-4](http://dx.doi.org/10.1016/0010-4485(88)90050-4)
- [13] Koenderink, J.J.; Doorn, A.J.: Surface shape and curvature scales, *Imaging and Vision Computing*, 10(8), 1992, 557–565. [http://dx.doi.org/10.1016/0262-8856\(92\)90076-F](http://dx.doi.org/10.1016/0262-8856(92)90076-F)
- [14] Sunil, V.B.; Pande, S.S.: Automatic recognition of features from freeform surface CAD models, *Computer-Aided Design*, 40, 2008, 502–517. <http://dx.doi.org/10.1016/j.cad.2008.01.006>
- [15] Vandenbrande, J.H.; Requicha, A.A.G.: Spatial reasoning for the automatic recognition of machinable features in solid models, *IEEE Transactions on P. A. & M. I.*, 15(12), 1993, 1–17. <http://dx.doi.org/10.1109/34.250845>
- [16] Yan, X.; Yamazaki, K.; Liu, J.: Recognition of machining features and feature topologies from NC programs, *Computer-Aided Design*, 32, 2000, 605–616. [http://dx.doi.org/10.1016/S0010-4485\(00\)00045-2](http://dx.doi.org/10.1016/S0010-4485(00)00045-2)
- [17] Zhang, X.; Nassehi, A.; Newman, S.T.: Feature recognition from CNC part programs for milling operations, *Int J Adv Manuf Technol*, 70, 2014, 397–412. <http://dx.doi.org/10.1007/s00170-013-5275-4>
- [18] Zhang, X.; Nassehi, A.; Safaieh, M.; Newman S.T.: Process comprehension for shopfloor manufacturing knowledge reuse, *International Journal of Production Research*, 2013, 1–15. <http://dx.doi.org/10.1080/00207543.2012.757669>
- [19] Zhang, Y.; Xu, X.; Liu, Y.: Numerical control machining simulation: a comprehensive survey, *International Journal of Computer Integrated Manufacturing*, 24(7), 2011, 593–609. <http://dx.doi.org/10.1080/0951192X.2011.566283>
- [20] Zhao, H.: Multisensor integration and discrete geometry processing for coordinate metrology, Ph.D thesis, ENS de Cachan, France, 2010.
- [21] Zhao, H.; Anwer, N.; Bourdet, P.: Curvature-based registration and segmentation for multisensor coordinate metrology, *Procedia CIRP*, 10, 2013, 112–118. <http://dx.doi.org/10.1016/j.procir.2013.08.020>