Graph-based feature recognition and suppression on the solid models

A. Malyshev[1], S. Slyadnev[1], V. Turlapov[1]
al.s.malyshev@rambler.ru|sergey.slyadnev@gmail.com|vadim.turlapov@itmm.unn.ru
[1]Lobachevsky State University of Nizhny Novgorod, Nizhny Novgorod, Russia

We consider one of the main problems in the field of CAD / CAM, which is feature recognition for CAD models represented in boundary representation form. We take into account volumetric features only. We use the recognition results for CAD model simplification. Our recognition procedures are based on detection of specific patterns in adjacency graphs constructed for the input CAD parts. Recognition procedures for bosses, holes, pockets, and for arbitrary cavities are proposed. Simplification is conducted by suppression of the detected features using a dedicated topological operator.

Keywords: *feature recognition, feature suppression, boundary representation, CAD model simplification, graph-based methods.*

1. Introduction

The problem of feature recognition and suppression from the so-called "dumb" model is one of the most frequently encountered problems in the CAD field. “Dumb” means that digital input model does not contain a history of construction: only pure geometric and topological data are present.

The vast majority of CAD-systems implement history-based modeling. However, a way of storing this information is vendor-specific and is not readily available for integration with third-party software. A “dumb” model may emerge in various scenarios like a lossy translation from one format to another or reverse engineering. Although the international STEP standard (ISO 10303) supports features as a part of AP224 protocol, usually, CAD-systems do not fully use this possibility.

Features are widely covered by academic literature. However, a “feature” term may have a different meaning in different engineering domains. In our paper, the “feature” term is used in a way proposed by Shah and Mantyla [9]. The description, properties, and semantics of the most commonly used features can be found in their fundamental book.

Feature-based modeling is close in nature to the *constructive solid geometry (CSG)*.The main difference between the two techniques is how a building tree is organized. In the feature-based modeling the building tree consists of various features, while in the CSG, it consists of Boolean operators and primitives. The concept of the building tree can be found in the article [1], where the problem of reconstructing three-dimensional models from two-dimensional drawings is considered.

Feature-based modeling is a more advanced design technique compared to other design methods. In feature-based modeling paradigm, an engineer works in terms of meaningful groups of geometric elements (features), and not in terms of *vertices, edges, and faces.* Feature recognition is opposed to design-by-features paradigm in sense that it does not rely on a construction tree. Feature recognition allows performing the following conversions:

- Defeaturing (feature suppression) on a model.
- Creation of feature model associated with a “dumb” model.

The defeatured model can be used in the following contexts:

- Protection of the intellectual property when exchanging data.
- Preparation of a model to engineering analysis.
- Creating levels of details for efficient visualization of massive data (e.g. large mechanical assemblies).
- Preparation of a model for visualization on mobile devices.

2. Related work

A general overview of recognition methods is given by Han, Pratt, and Regli in [3]. Joshi and Chang proposed *graph-based methods of recognition* in [5]. The main idea besides the graph-based methods is to analyze connectivity between the boundary elements of a model, these methods convert the input model to one or several graphs, assign some attributes to the vertices and edges of these graphs and after that perform recognition procedure.

Joshi and Chang propose to use *attributed adjacency graph* (AAG). Its structure is presented below:

- **Vertices -** represent faces of the input model.
- **Edges -** represent adjacency relationship between faces.
- **Attributes -** represent convexity of dihedral angles.

Gao and Shah [2] introduce new graphs into recognition process. They use manufacturing face adjacency graph, minimal condition sub-graph, concave adjacency graph, and other medium structures. The main goal of their work is to recognize interacting features which is the key problem in feature recognition field.

Verma and Rajotia in [11] propose to use a matrix representation of a graph instead of isomorphism in the recognition process. The so-called feature vector is calculated for each feature of interest. The feature vector is then used in recognition heuristic. This transformation decreases recognition complexity from $o(n^k)$ to$o(n^3)$.

Lockett and Guenov in [6] propose to build AAG on the *mid-surface representation* of a geometric model. The main problem of their approach is to adapt AAG concept to the mid-surface representation. This technique significantly reduces a number of nodes and vertices in the graph and improves recognition speed.

Hayasi and Asiabanpour [4] consider the platform-dependent approach of conversion design features into manufacturing features.

Nasr and others [7] unite different recognition procedures into the sequential workflow to construct building tree for a model.

The major drawback of graph-based methods is the impossibility of guaranteed separation of interacting features. This problem is partially solved by Gao and Shah, but it remains actual.

Usually, the result of recognition is presented as a set of faces, which is an advantage, since it allows using this method for recognition of *volumetric* and *surface* features. The concept of volumetric and surface features can be found in [10].

The method, which is proposed in this paper, is based on the original attributed adjacency graph. Therefore it can be implemented in any CAD-system since no specific functionality is required. The recognition procedures presented below are based on the *rule-based* approach, initially invented in [10]. This approach reduces complexity of the problem and allows flexible tuning of the *recognition rules*.

3. Recognition procedures

The development of efficient feature recognition algorithm is hardly possible without constraining the input. In this work, the following assumptions are made:

- **Input model consists of one or more valid solids.** This restriction enables convexity analysis for dihedral angles. Such analysis relies on proper orientations of faces in a solid.
- **Face maximization was applied to the input model.** The definition of the maximization procedure can be found in [8].
- **All features are placed in interior of faces.** It is a principal constraint which is actively used in the recognition and suppression procedures.
- **The host geometry of faces is canonical.** This limitation is used in the first three procedures and does not apply to the last method.

The common stage which is shared by all recognition procedures proposed in our paper is detection of a *base face*. In general, a base face is a face where features can exist. In our study, the base faces are the faces with inner wires (Figure 1).

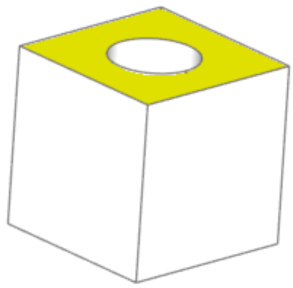

Figure 1: Cube model and base face (yellow color).

Here and later, it is assumed that attributed adjacency graph is built. Particular features can be recognized using additional constraints, which are proposed for each feature type.

3.1 Holes recognition

Hole is cavity feature, so the dihedral angle between base face and feature faces should be convex. Feature faces which are adjacent to the base face should have cylindrical host geometry. After that, the list of feature faces can be extended by adjacent faces with non-cylindrical host geometry. For instance, it is possible to allow transition from cylindrical geometry to the planar to handle nested holes case. For clarity, the simplest case is considered – the stopping criterion for the recognition procedure is another base face. The main parameter determining the hole is radius. It seems reasonable to filter out all holes with radius bigger than some given value. Modified recognition procedures can be proposed using the requirements of a particular subject area.

The figure below shows the detected holes on the ANC101 test model with a radius not exceeding a given threshold (Figure 2):

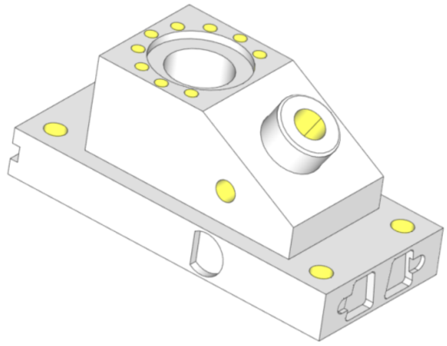

Figure 2: ANC101 model and recognized holes (yellow color).

3.2 Pockets recognition

Pockets are another machining features, the main difference between holes and pockets is that the pocket is always *blind*. The dihedral angles on the edges of the inner contour must also be convex. The underlying geometry on the feature faces which are adjacent to the base face can have any type. The feature is constructed by adding adjacent faces. The recognition process stops when a plane face without internal contours is detected, parallel to the base face. The pocket can be defined by two parameters – the depth and the contour that forms it. For example, the filtering can be carried out using the area of the figure bounded by the contour of the pocket. The following illustration shows the result of this recognition rule (Figure 3):

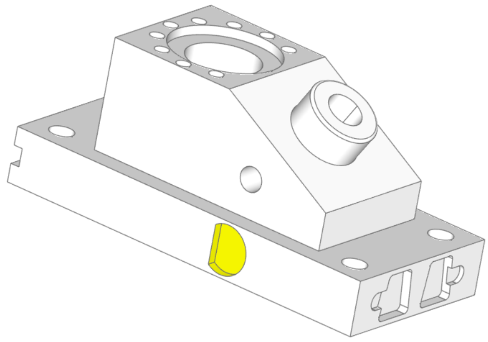

Figure 3: ANC101 model and recognized pockets (yellow color).

3.3 Bosses recognition

A boss is a protrusion feature in a work piece. Therefore, the dihedral angles between the boss faces and the base face are concave. The faces constituting a boss feature yield an isolated connected component in the adjacency graph with eliminated base face (as in the case of a pocket). New faces are added to the feature similarly to the case with pockets. Filtering bosses by size can be conducted using the diagonals of their corresponding bounding boxes.

Recognition speed can be improved by limiting the search depth in the adjacency graph. For instance, the search depth equal to two can be used to find the simplest bosses (the search begins from the base face). Figure 4 shows the boss feature found in the MBB Gehause Rohteil model.

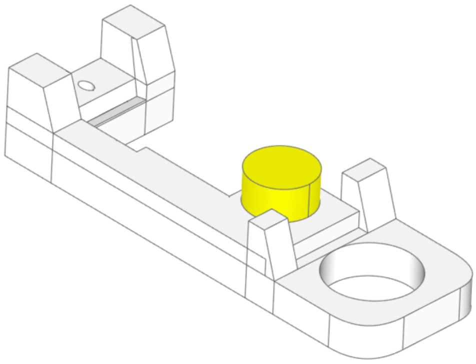

Figure 4: Modified MBB Gehause rohteil model and recognized boss (yellow color).

3.4 General holes recognition

All the procedures considered earlier are designed to operate on canonical geometry. In this section, a more general method is proposed. The features detected by the following procedure do not correspond to simple machining operations such as drilling or milling. We call this new feature type a "general hole" since it represents an arbitrary cavity. Like in the case with a pocket or a hole feature, the dihedral angles for general holes are convex. More faces are added to the feature according to the procedure already described. There are two stopping criteria:

- For through general holes, the feature faces are located between two base faces.
- For blind general holes, the feature faces yield an isolated connectivity component in the adjacency graph.

Filtering by size can be based on diagonal of a bounding box calculated for the detected feature faces.

The described recognition procedure covers both the procedure for holes and the procedure for pockets. Figure 5 and Figure 6 show the recognition result for ANC101 model.

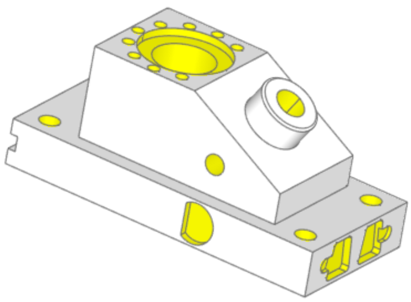

Figure 5: ANC101 model and recognized general holes.

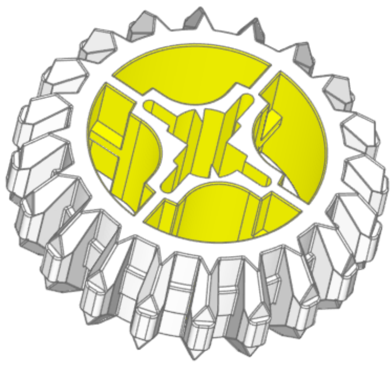

Figure 6: Gear model and recognized general holes (yellow color).

4. Feature suppression

When all the features that should be suppressed are recognized, they can be suppressed to simplify a model. This process involves two main steps:

1. Removing of the feature faces from B-Rep representation.
2. Updating all affected faces to get valid B-Rep model after suppression.

In general, the update procedure is very difficult since it requires both topology and geometry modification. Restrictions introduced earlier significantly reduce the complexity of this step. It is necessary to delete only inner wires on the base faces, and no additional actions are needed to bring geometry into valid state. It is the reason to detect isolated non-interacting features which represented as isolated connectivity component on the graph.

5. Conclusion and further work

Feature recognition and suppression procedures were proposed for holes, pockets, bosses, and general holes. These methods are implemented using OpenCascade geometric modeling kernel.

The algorithms were tested on a wide set of cases, including the well-known benchmarking models (MBB Gehause Rohteil, ANC101) and real industrial parts.

Figure 7 and Figure 8 demonstrate the defeaturing process based on the proposed recognition procedures.

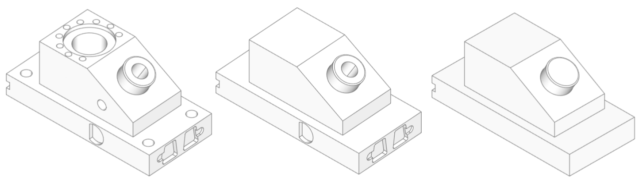

Figure 7: Feature suppression on the ANC101 model.

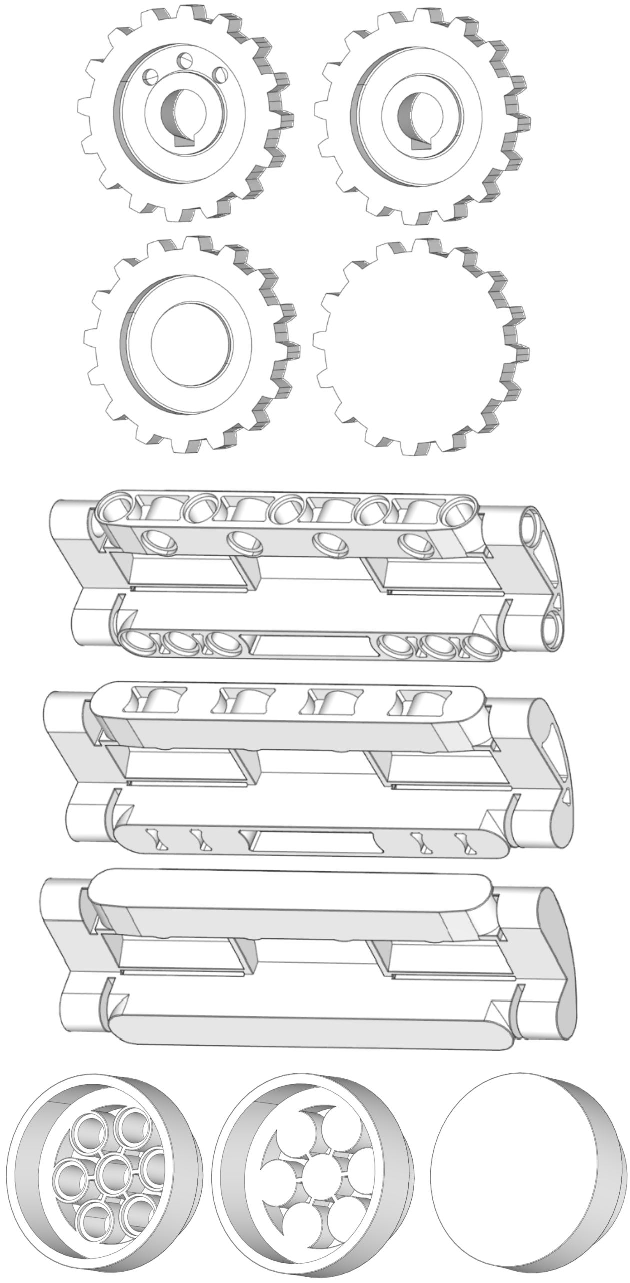

Figure 8: Feature suppression demonstration, from left to right, from top to bottom.

In future, we plan to improve the recognition methods in the following aspects:

- Extension of the supported feature types.
- Recognition of non-isolated features.
- Migration from explicit-based rules to the parameterized rules.
- Automatic generation of the building tree.

6. References

[1] Жилина, Н., Лагунова, М., Мошкова, Т., Ротков, С. и Тюрина, В. 2014. Автоматизация процесса чтения чертежа с использованием системы Компас -3D. ГрафиКон 2014: труды 24 междунар. конф. по компьютерной графике и зрению, 17–19.

[2] Gao, S. and Shah, J.J. 1998. Automatic recognition of interacting machining features based on minimal condition subgraph. Computer-Aided Design 30, 9, 727–739.

[3] Han JungHyun, Pratt, M., and Regli, W.C. 2000. Manufacturing feature recognition from solid models: a status report. IEEE Transactions on Robotics and Automation 16, 6, 782–796.АвторД. Источник.

[4] Hayasi, M.T. and Asiabanpour, B. 2009. Extraction of manufacturing information from Design-by-feature solid model through feature recognition. International Journal of Advanced Manufacturing Technology 44, 11–12, 1191–1203.

[5] Joshi, S. and Chang, T.C. 1988. Graph-based heuristics for recognition of machined features from a 3D solid model. Computer-Aided Design 20, 2, 58–66.

[6] Lockett, H.L. and Guenov, M.D. 2005. Graph-based feature recognition for injection moulding based on a mid-surface approach. CAD Computer Aided Design 37, 2, 251–262.

[7] Nasr, E.S.A., Khan, A.A., Alahmari, A.M., and Hussein, H.M.A. 2014. A Feature Recognition System Using Geometric Reasoning. Procedia CIRP 18, 238–243.

[8] Requicha, A.A.G. and Voelcker, H.B. 1985. Boolean operations in solid modeling: Boundary evaluation and merging algorithms. Proceedings of the IEEE 73, 1, 30–44.

[9] Shah, J.J. and Mantyla, M. 1995. Parametric and Feature Based CAD/Cam: Concepts, Techniques, and Applications.

[10] Vandenbrande, J.H. and Requicha, A.A.G. 1993. Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 15, 12, 1269–1285.

[11] Verma, A. K. and Rajotia, S. 2004. Feature vector: a graph-based feature recognition methodology. International Journal of Production Research 42, 16, 3219–3234.

About authors

Vadim Turlapov (vadim.turlapov@itmm.unn.ru): professor of Computer Science, Lobachevsky Nizhny Novgorod University.

Alexander Malyshev (al.s.malyshev@rambler.ru): PhD student, Lobachevsky Nizhny Novgorod University.

Sergey Slyadnev (sergey.slyadnev@gmail.com): PhD student, Lobachevsky Nizhny Novgorod University.