Title:
Recognition of Free-form Features for Finite Element Meshing using Deep Learning

Authors:
Hideyoshi Takashima, hideyoshi_takashima@ais-hokkaido.co.jp, Hokkaido University, Japan
Satoshi Kanai, kanai@ssi.ist.hokudai.ac.jp, Hokkaido University, Japan

Keywords:
Deep Learning, PointNet++, Feature Recognition, Point Cloud, Triangular Mesh, Finite Element Method.

DOI: 10.14733/cadconfP.2021.31-35

Introduction:

Large-scale finite element models (FE models), having several million elements, require high-quality FE meshes in compliance with some company-specific meshing rules to ensure simulation accuracy. In these meshing rules, the meshing patterns, location and resolution are strictly defined for specific free-form features such as ribs and bosses. For example, as shown in Fig. 1, when an FE mesh is to be generated for a cylindrical boss feature, the node points of elements must be placed concentrically and evenly around a medial axis of the boss at specified number. However, the automatic feature-compliant finite element meshing for CAD models is not yet fully supported in commercial CAE software. And it still requires many manual operations resulting in a high person-hour ratio to the whole CAE process.

The feature-compliant finite element meshing mainly consists of a series of operations; 1) extracting the finite-element free-form feature shapes such as "ribs" from a given CAD model, 2) performing the segmentation of each feature shape into local feature areas such as "top", "side", and "fillet" areas, and 3) generating a finite-element mesh such that it complies with the specific meshing rules defined on the recognized feature shapes and areas. However, if the geometry of a given CAD model is large-scale and has high complexity, the operations relying on only the engineer's decisions become incredibly stressful, time-consuming, and error-prone. Therefore, an automated feature shape extraction and feature area recognition technique targeting a feature-compliant finite element meshing is strongly required.

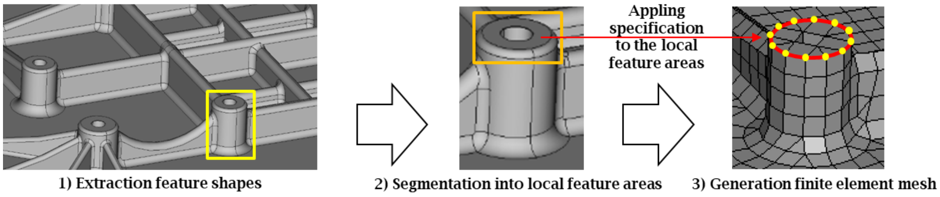

Fig. 1: Operations of the feature-compliant finite element meshing.

There has been some research on the feature extraction techniques from CAD models to generate meshes of FE models [1-3]. However, three main problems remain in them. First, the feature extraction algorithm does not work robustly when the CAD models include Product Data Quality (PDQ) issues such as cracked or degenerated geometries. Second, the free-form features surrounded by complicated and smooth boundaries, commonly found in casted or molded parts, remain challenging to detect by these techniques. Finally, the extraction algorithm must be designed in an ad-hoc way for different feature types and features with similar shapes. Thus, it is challenging to apply these previous feature extraction techniques when developing a feature-compliant finite element meshing.

As a solution to these problems, the 3D shape-descriptor-based finite-element feature classification [6] and feature extraction technique [7] using the dense point cloud representation have been proposed by our research group. However, they provide the solutions only to the first operations in the feature-compliant finite element meshing process. Unlike these previous studies, this paper proposes a deep-learning approach to extract free-form feature shapes and perform the segmentation of each feature shape into local feature areas for finite element meshing from CAD models of a product. PointNet++ [5] that is one of the popular convolutional neural network for 3D point cloud classification and segmentation is used to recognize our free-form feature shapes and local feature areas. The performance of the free-form feature recognition is experimentally verified in this paper.

Recognition of Free-Form Feature Shape and Local Feature Areas:

Basic Concept

The proposed feature recognition method first extracts the feature shape existing in a product shape represented by a solid model and then partitions each feature shape into local feature areas. The targeted features include ribs and bosses in casted or molded parts but are not limited to them. The proposed recognition method is designed and implemented based on the following concepts.

- In the feature extraction, the geometries of a product and finite element features are represented by 3D dense point clouds that are converted from solid models. This enables a stable feature extraction even when a solid model contains PDQ-degraded geometries and/or when the feature shape boundaries are filleted and ambiguous.
- The machine learning scheme enables a uniform and portable implementation of the feature extraction algorithm regardless of the feature types. If we want to add a new feature type to be recognized, we only have to prepare the new training examples. It avoids an ad-hoc and inefficient implementation of the algorithms for different feature types.

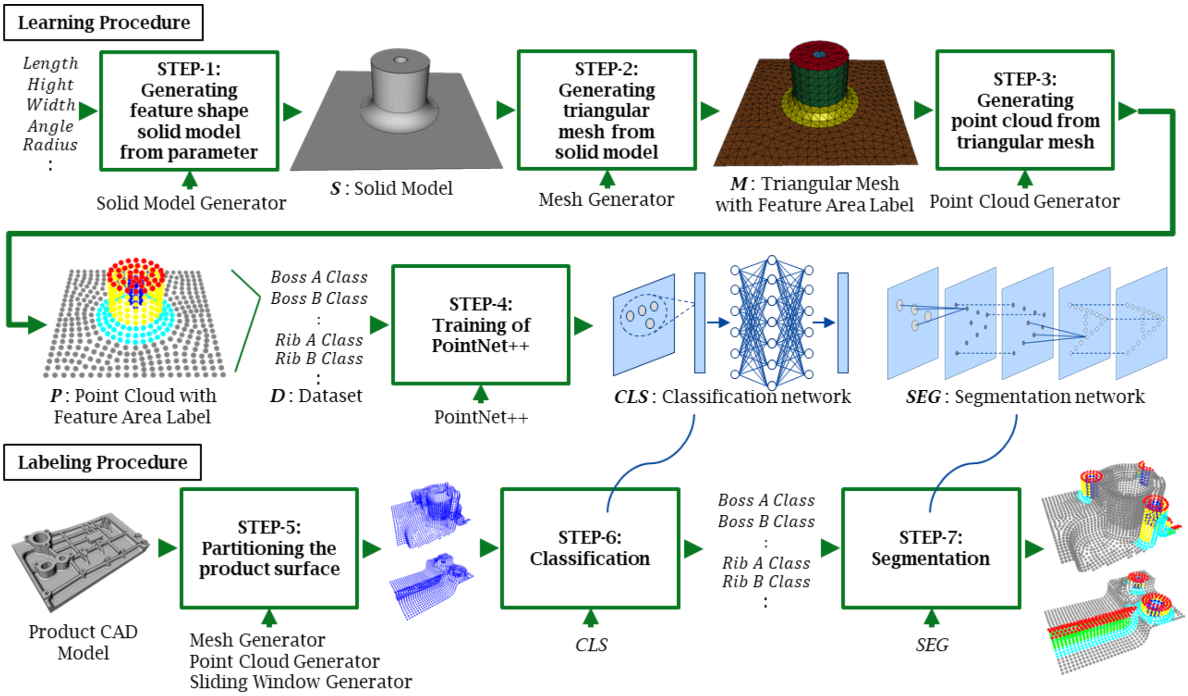

Fig. 2: Recognition Pipeline of Free-Form Feature Shape and Local Feature Areas.

- Using PointNet++ [5], which has a proven track record in classification and segmentation of 3D shapes using point clouds and deep learning, the general and unified machine learning procedures can be applied to extract the free-form feature shape and to perform the segmentation into local feature areas.

Feature Shape Extraction and Local Feature Area Segmentation Procedure

The proposed method comprises seven steps illustrated in Fig. 2. Steps 1 to 4 are the learning procedure to create a deep neural network for feature classification and segmentation, while steps 5 to 7 are the labeling procedure to label the features and perform the segmentation for each feature shape into local feature areas

Step 1: Generating the solid models of feature shapes using parametric deformation

For the feature recognition based on deep learning, a large amount of training set of feature variants must be prepared to secure the recognition accuracy. In this paper, relatively small number of feature types such as ribs and bosses that often appeared in automotive parts, and targeted. So we augmented the training feature shapes with many size variations using parametric deformation functions of the solid models of several reference rib and boss features shown in Fig. 3. To this end, we manually picked up typical rib and boss features from solid models of automotive parts, identified the local feature areas and size parameters that define the feature shapes on CAD system and generated a large number of solid models of the feature variants with different parameter settings.

Step 2: Generating the triangular mesh of a feature shape from the solid model

The solid model of a feature variant is converted into a 3D point cloud for learning and labelling process. To this end, a solid model is converted into a triangular mesh using commercial CAE preprocessor, then the surface normal vector and the label of the local feature area is automatically assigned to each vertex on the mesh.

Step 3: Generating a point cloud for training from the triangular mesh

For the deep learning, the position and size of the labeled point cloud of the feature valiant are normalized such that its centroid is positioned at the origin and the maximum length of AABB (Axis-aligned bounding box) becomes one. Finally, the point cloud for training by PointNet++ is generated. We repeat from Step 1 to Step 3 in different feature type to prepare the training point cloud set of feature variants.

Step 4: Training of PointNet++

PointNet++ [5] is a deep neural network for classification and segmentation of 3D point clouds. It achieves high recognition performances using the local feature learning architecture by a hierarchical neural network that applies PointNet [4] recursively. In this step, we use the training point cloud set of feature variants for PointNet++ to build two deep neural networks: the classification network for feature shape extraction, and the segmentation network for local feature area segmentation.

Step 5: Partitioning the Product Surface

In the labeling procedure, feature shapes and their local feature areas must be extracted from input product CAD model. Because the surface of a product CAD model includes many finite-element features, it is necessary to identify where and what kind of feature shapes exists on the product surface. For this purpose, we use the sliding window method, which has been frequently used in image recognition, is used. We partition the surface of a product CAD model into a collection of small subsets of the surfaces included in the window and provide the neural networks with the point cloud of the individual subset.

Step 6: Feature Shape Classification by PointNet++

In this step, the subset point cloud included in each window is input to the trained classification network. The network estimates a feature shape type such as rib and boss for the input point clouds.

Step 7: Local Feature Area Segmentation by PointNet++

Finally, the subset point cloud with feature shape type is further input to the trained segmentation network. The network assigns a label of local feature area such as *top*, *side* and *fillet* to each point.

By using the point-wise local feature area labels assigned by these steps, it is possible for mesh generation software to identify which meshing specifications should be applied to the feature geometry and to automatically generate FE meshes in compliance with meshing specifications.

Verification of the Feature Recognition:

Recognition of free-form feature of simple ribs and bosses

The variants of the two feature types (rib and boss) shown in Fig. 3 were used as learning and testing models in the verification. Totally 4188 different variants with different sizes were generated with various parameter settings such as widths and heights. The local feature areas labeled on these feature types are shown in Fig. 4. 3385 models were used as the training model, and 803 models were used as the testing model of PointNet++.

A part of the recognition results was shown in Fig. 5. The mean intersection-over-union (mIoU) of rib class was 0.988, boss class was 0.997, and total was 0.992. Therefore, the recognition accuracy of free-form feature types and local feature areas of proposed method was sufficient.

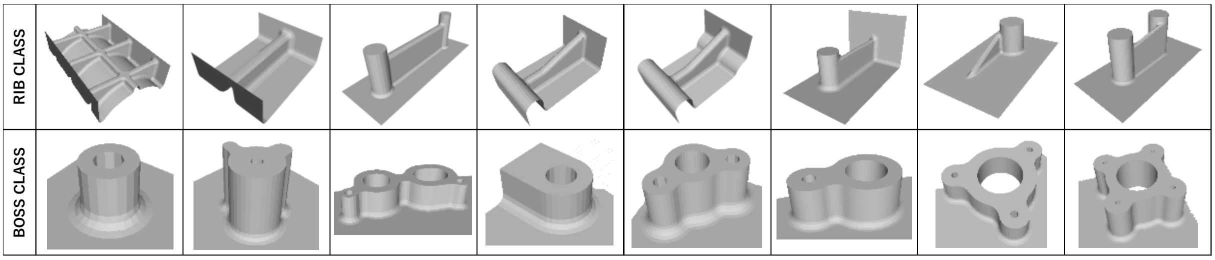

Fig. 3: Feature types for the verification.

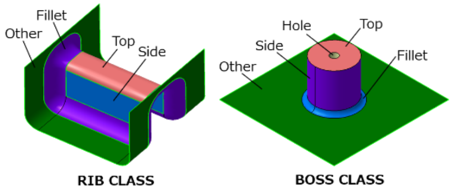

Fig. 4: Local feature areas on rib and boss.

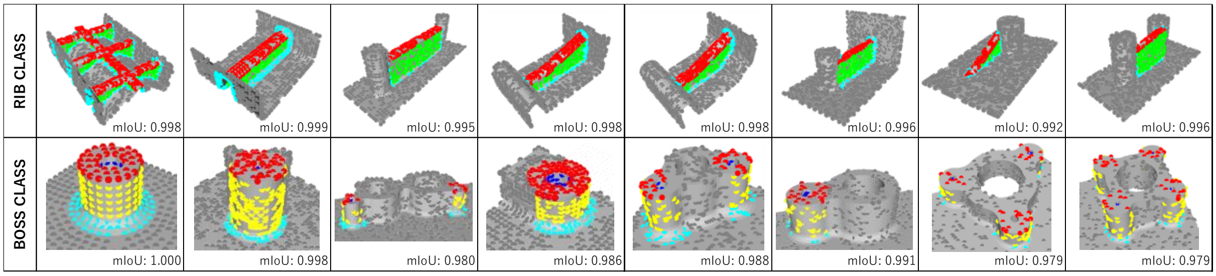

Fig. 5: Recognition examples of feature shapes and local feature areas.

Recognition free-form features on a complicated product model

In this verification, we attempted to recognize same free-form feature shapes and local feature areas on two complicated product models shown in Fig. 6. The recognition results were shown in Fig. 6. The "side" and "top" areas in the rib classes were well segmented. These results indicated a certain degree of effectiveness of the proposed deep-learning-based feature recognition. To the contrary, other local feature areas were not sufficiently segmented. Moreover, rib and boss feature shapes were not fully extracted when the original product shape contains multiple feature shapes and types.

There might be two causes of these missed recognition. First, the position and size of sliding windows might not be appropriate for the recognition. In this verification, two models were divided into 12 subsets and tested. However, the size of the sliding window appropriate for the recognition performance were not fully tested so far. Second, the feature areas selected as Fig. 2 may not be appropriate for recognition. In our learning phase, the "other" areas of the local feature are trained as a part of a feature shape as shown in Fig. 4. However, these areas are not strictly a part of the boss or rib feature geometries. This might have an adverse effect on the feature classification performance when different feature types are discriminated. These issues will be solved as our future work.

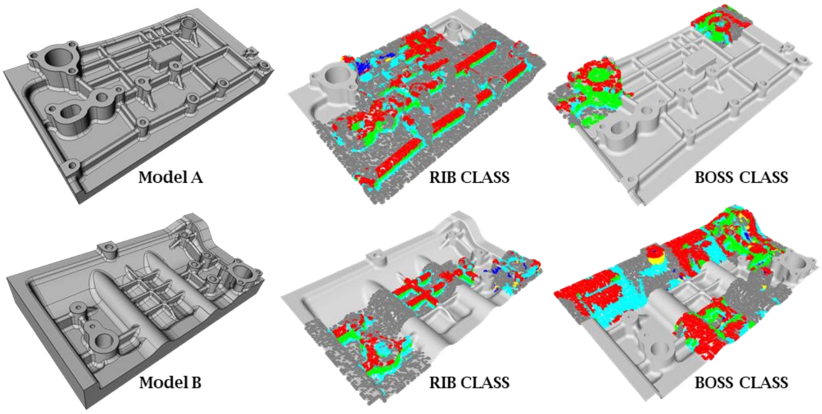

Fig. 6: The complicate solid models and the results of recognition of free-form features.

Conclusions:

A deep-learning approach was proposed for free-form feature recognition for FE meshing in this paper. PointNet++ that enables the classification and segmentation of 3D point clouds were utilized for the approach. It was shown that the proposed approach is effective for recognizing single free-form feature shape. However, the recognition performance still has a room for improvement when the original product shape contains multiple feature shapes and types. The improvement might be achieved by the appropriate settings of the sliding window size or learning dataset.

As the next step of this work, we will try to adjust the selection of the sliding window area and will modify the local feature areas of the learning datasets. We will also develop a FE meshing software that integrates the proposed feature extraction with the automated FE mesh generation in compliance with various company-specific meshing rules.

References:

[1] Lai, J.Y.; Wang, M.H.; Song, P.P.; Hsu, C.H.; Tsai, Y.C.: Recognition and decomposition of rib features in thin-shell plastic parts for finite element analysis, Computer-Aided Design and Applications, 15(2), 2018, 264-279. https://doi.org/10.10 80/16864360.2017.1375678

[2] Lu, Y.; Gadh, R.; Tautges, T.J.: Feature based hex meshing methodology: feature recognition and volume decomposition, Computer-Aided Design, 33(3), 2001, 221-232. https://doi.org/10.1016/S0010-4485(00)00122-6

[3] Onodera, M.; Hariya, M.; Kongo, C.; Shintani, M.; Ka, K.; Watanuki, K.: Development of high precise similar sub-part search technique for automatic mesh generation reusing proven models, Transactions of The Japan Society of Mechanical Engineers, 85(880), 2019, 19-00138. https://doi.org/10.1299/transjsme.19-00138

[4] Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, Proceedings of the IEEE Conference on CVPR, 2017, 652-660. https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.16

[5] Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, Proceedings of NIPS, 2017, 5105-5114. https://arxiv.org/abs/1706.02413

[6] Takaishi, I.; Kanai, S.; Date, H.; Takashima, H.: Free-Form feature classification for finite element meshing based on shape descriptors and machine learning, Computer-Aided Design and Applications, 17(5), 2020, 1049-1066. https://doi.org/10.14733/cadaps.2020.1049-1066

[7] Takashima, H.; Kanai, S.: Shape Descriptor-Based Similar Feature Extraction for Finite Element Meshing, Computer-Aided Design and Applications, 18(5), 2021, 1080-1095. https://doi.org/10.14733/cadaps.2021.1080-1095