Title:
Graph Neural Network-Based Finite Element Feature Recognition from B-rep Model

Authors:
Ryosuke Moriya, r-moriya@eis.hokudai.ac.jp, Hokkaido University
Satoshi Kanai, kanai@ssi.ist.hokudai.ac.jp, Hokkaido University
Hiroaki Date, hdate@ssi.ist.hokudai.ac.jp, Hokkaido University
Hideyoshi Takashima, hideyoshi_takashima@ais-hokkaido.co.jp, AIS Hokkaido, Inc.
Tetsufumi Taichi, tetsufumi_taichi@ais-hokkaido.co.jp, AIS Hokkaido, Inc.

Keywords:
B-Rep., Deep Learning, Feature Recognition, Graph Neural Network, Finite Element Meshing

DOI: 10.14733/cadconfP.2023.254-259

Introduction:
Finite element (FE) analysis requires mesh generation as a preprocess, which partitions a Boundary Representation (B-Rep) computer-aided design (CAD) model into FE meshes. Because mesh quality significantly affects analysis accuracy, manufacturers specify company-internal mesh generation rules for some types of FE features on CAD models, such as bosses and ribs, including free-form surfaces (Figure 1). However, at present, the recognition of the FE features from CAD models relies heavily on human eyes and hands, making it time-consuming, and error-prone. Therefore, a reliable, and versatile FE feature recognition method from CAD models is strongly required for efficient high-quality mesh generation.

Recently, several deep-neural-network (DNN)-based methods have been proposed for feature recognition from CAD models. They have an advantage in that they do not require algorithm design specific to each feature type, unlike classical methods. However, most DNN-based methods approximate input CAD model geometries with voxels [9] or point clouds [7] as an input of DNNs, causing discretization loss in model resolution or an increase in the data size, which results in more significant memory consumption or longer training time. To resolve these issues, other DNN-based feature recognition methods have been proposed in recent years [3, 5] that use “graphs” as an input of the DNN, which take advantage of high compatibility with standard B-Rep CAD models. Nevertheless, those methods also have issues. First, the recognition significantly depends on the model poses. Second, the recognition method targeted the machining features or geometric modeling procedures and was not tested with FE features that included free-form surfaces.

In this study, we propose an FE feature recognition method from a B-Rep CAD model using a graph neural network (GNN), which has a recognition ability invariant to model rotation or translation. The

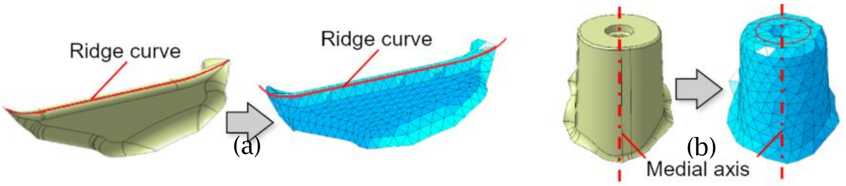

Fig. 1: Typical examples of mesh generation rules for features: (a) rib case and (b) boss case [3]. In a rib, mesh vertices are aligned with the ridge curve. In a boss, the vertices are positioned concentrically around the medial axis.

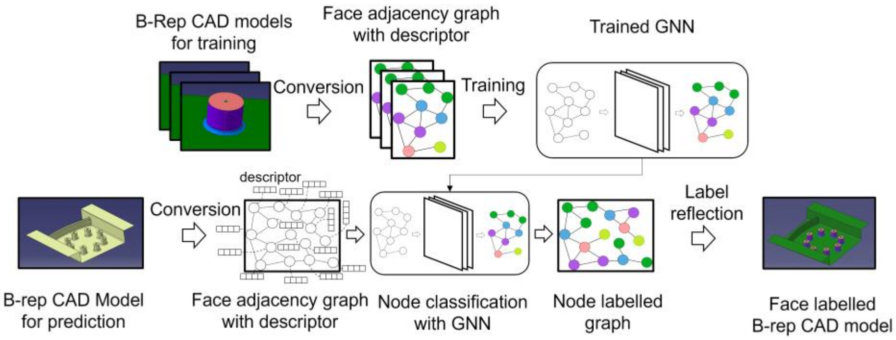

Fig. 2: The proposed FE feature recognition pipeline.

proposed method comprises a graph construction method with descriptors invariant to the translation or rotation, a neural network structure used for feature recognition, and data augmentation (DA) techniques for robust recognition. We tested our method with the original dataset of FE features, including bosses and ribs, and compared its performance with that of an existing method [1] using GNN [3,5], similar to ours.

GNN-Based FE Feature Recognition Method:

Overview

Figure 2 shows the recognition pipeline of the proposed method. The first step converts B-Rep CAD model data into a face adjacency graph (FAG) with descriptors. The FAG is a graph in which B-Rep faces and edges are represented as nodes and links, respectively. Each node and link in FAG have its descriptors, a multidimensional vector encoding the corresponding face or edge geometry. In the second step, the FAG is input into a GNN that classifies the labels of each node. The GNN is trained in advance with CAD models comprising FE features whose faces are labeled with feature labels. In the last step, the estimated face feature labels are extracted from the GNN output and reflected in the original CAD model.

Descriptors on the FAG

Our method encodes the topology and geometry of a B-Rep CAD model in terms of FAG connection relations and descriptors on the FAG nodes and links, respectively. To achieve pose-invariant feature recognition, we defined the following descriptors invariant to translation and rotation. Moreover, considering the class of geometries to be recognized, the descriptors that can discriminate the geometries of free-form surface and curve equations are included in the B-Rep CAD model.

The node descriptor $\boldsymbol{F}_n$ encodes the geometry of a face. It comprises the following two descriptors $\boldsymbol{F}_{SI}$ and $\boldsymbol{F}_{OBB}$ calculated from sampled points on the face.

1) Shape Index distribution $\boldsymbol{F}_{SI}$: $\boldsymbol{F}_{SI}$ represents a statistical distribution of the *Shape Index* [4] $SI(p)$ at a sampled point p on a face. $SI(p)$ is calculated with Eqn. (1), from principal curvature κ_1, κ_2 ($\kappa_1 \geq \kappa_2$) at p, and ranged from −1 to 1, except for the case that p is on planes.

$$SI(p) = \frac{\pi}{2}\tan^{-1}\frac{\kappa_2 + \kappa_1}{\kappa_2 - \kappa_1}, \tag{1}$$

The descriptor $\boldsymbol{F}_{SI}$ is a normalized histogram of $SI(p)$ over the face and is defined by Eqns. (2) and (3) with l as the number of intervals (we used $l = 7$):

$$\boldsymbol{F}_{SI} = \frac{1}{\sum_{i=0}^{l}|A_i|}[|A_0|, |A_1|, \cdots, |A_l|]^T, \tag{2}$$

$$A_i = \begin{cases} \left\{p \mid -1 + \frac{2i}{l} \leq SI(p) \leq -1 + \frac{2(i+1)}{l}\right\} & (i = 0, \cdots, l-1) \\ \{p \mid p \text{ is on a plane}\} & (i = l) \end{cases}. \tag{3}$$

2) Oriented bounding box (OBB) aspect ratio ($\boldsymbol{F}_{OBB}$): $\boldsymbol{F}_{OBB}$ is the proportions of three edge lengths l_1, l_2, l_3 ($l_1 \geq l_2 \geq l_3$) of the OBB that envelops the points sampled on a face:

$$\boldsymbol{F}_{OBB} = \frac{[l_1, l_2, l_3]^T}{\|[l_1, l_2, l_3]^T\|}. \tag{4}$$

Moreover, the link descriptor $\boldsymbol{F}_l$ represents the angle relation between two adjacent faces connected to an edge. The following two types of angles are adopted for the link descriptor: local face angle F_{la} and global face angle F_{ga}. Similar to the node descriptor, the link descriptor is calculated from the point sampled on the edge corresponding to a link.

3) Local face angle (F_{la}): F_{la} is an average of normalized signed angles between the normal vectors of the adjacent faces at the points sampled on an edge. The signed normalized angle $\theta_{la}(q)$ at a point q is defined as follows:

$$\theta_{la}(q) = \frac{1}{\pi} \operatorname{sgn}\left(\left(\boldsymbol{n}_i \times \boldsymbol{n}_j\right) \cdot \boldsymbol{t}_i\right) \cos^{-1}\left(\boldsymbol{n}_i \cdot \boldsymbol{n}_j\right), \tag{5}$$

where $\boldsymbol{n}_i, \boldsymbol{n}_j$ denote unit normal vectors of faces f_i, f_j adjacent to the edge, and $\boldsymbol{t}_i$ denotes a unit tangent vector of the corresponding half edge belonging to the face f_i. F_{la} is calculated by averaging $\theta_{la}(q)$ over the sampled point set Q on the edge as follows:

$$F_{la} = \frac{1}{|Q|} \sum_{q \in Q} \theta_{la}(q). \tag{6}$$

4) Global face angle (F_{ga}): F_{ga} is a normalized angle between the averaged normal vectors $\bar{\boldsymbol{n}}_i, \bar{\boldsymbol{n}}_j$ on two adjacent faces f_i, f_j connected to an edge. $\bar{\boldsymbol{n}}_i$ is calculated from the normal vectors at the sampled points on f_i. F_{ga} is calculated as follows:

$$F_{ga} = \frac{1}{\pi} \cos^{-1}\left(\bar{\boldsymbol{n}}_i \cdot \bar{\boldsymbol{n}}_j\right). \tag{7}$$

Node classification by GNN

Figure 3 shows the GNN structure used for the node classification for feature recognition. As shown in the figure, the network comprises two phases: convolution and multilayer perceptron (MLP). The convolution phase comprises three “MLP & Conv” layers. The layers perform three operations. First, each node/link descriptor is inputted into the affine layer, activation function, and batch normalization layer sequentially. Second, node descriptors are convoluted into link descriptors as follows:

$$\boldsymbol{e}'_{ij} = \sigma\left(\mathrm{BN}\left(\boldsymbol{W}\left(\boldsymbol{x}_i \oplus \boldsymbol{x}_j \oplus \boldsymbol{e}_{ij}\right)\right)\right), \tag{8}$$

where $\boldsymbol{x}_i$ denotes the descriptor of node i before the operation, $\boldsymbol{e}_{ij}$ and $\boldsymbol{e}'_{ij}$ denote the descriptors before and after the operation of the link between node i and j, respectively, BN denotes the batch normalization layer, σ denotes the activation function, $\oplus$ denotes vector concatenation, and $\boldsymbol{W}$ denotes

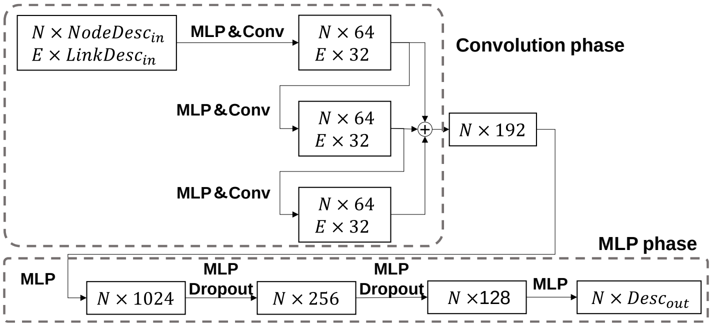

Fig. 3: GNN structure used for node classification. Each square describes the descriptor size attached to nodes and links. N and E denote the numbers of nodes and links, respectively, $NodeDesc_{in}$ and $LinkDesc_{in}$ denote the size of input descriptors of nodes and links, respectively, and $Desc_{out}$ denotes the output size of node descriptors.

learning parameters. Finally, the descriptors are convoluted between adjacent nodes using *graph attention networks* as follows [8]:

$$\boldsymbol{x}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i}\frac{\exp\left(\sigma(\boldsymbol{W}_n\boldsymbol{x}_i \oplus \boldsymbol{W}_n\boldsymbol{x}_j \oplus \boldsymbol{W}_e\boldsymbol{e}_{ij})\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\sigma(\boldsymbol{W}_n\boldsymbol{x}_i \oplus \boldsymbol{W}_n\boldsymbol{x}_k \oplus \boldsymbol{W}_e\boldsymbol{e}_{ik})\right)}\boldsymbol{W}\boldsymbol{x}_j\right), \tag{9}$$

where $\boldsymbol{x}_i'$ denotes the descriptor of node i after the operation, and $\boldsymbol{W}_n, \boldsymbol{W}_e$ denote learning parameters.

Data Augmentation

To improve recognition robustness, we conducted DA in every training epoch. As DA, we removed links of the input FAG and overwrote descriptor elements to 0 with a certain probability (0.15 in our experiment). DA was only performed on the training dataset and not on the validation dataset.

<u>Case Study:</u>

Dataset

We created a dataset "basic dataset" comprising representative FE features—boss and rib—for performance evaluation. As shown in Fig. 4 (a), the dataset includes the CAD models of simply shaped 17 boss and rib types. We varied the sizes of features such as radius or height to create a training dataset including 32,940 model instances. As shown in Fig. 4 (b), every face of a feature in the training dataset has seven labels: "BossTop," "BossSide," "BossHole," "RibTop," "RibSide," "Fillet," and "None." In addition, we prepared three CAD data "practical models" where bosses and ribs are arranged in combination to evaluate the performance of the FE feature recognition in a more practical scenario.

Experimental Condition

We implemented the proposed GNN-based feature recognition method using *PyTorch Geometric* [2] and *Open CASCADE Technology* [6], trained with the basic dataset and finally tested both with the basic dataset and practical models. We performed 10-fold cross-validation and compared our results with the recognition results of the combined method [1] of *BRepNet* [5] and *UV-Net* [3], which is a GNN-based feature recognition method like ours. To check the rotation invariance, the test was performed with both a posture similar to the training data and a randomly rotated posture.

Result

Table 1 summarizes the recognition performances, and Figure 5 shows an example of the recognition results for practical models. As shown in Table 1, our method and the comparison method showed high recognition performance for the basic dataset with the original pose, whereas their performance for the practical models was low. The reason could be that the practical models were far more complex and had much more interaction between bosses and ribs. Moreover, when the rotation was applied to the input, the performance of the comparison method degraded significantly. Meanwhile, our method did not show any degradation due to its rotational invariance. Finally, DA improved the performance of our method on the practical models by about 6%. This is because the recognition performance of input graphs with different topologies was improved.

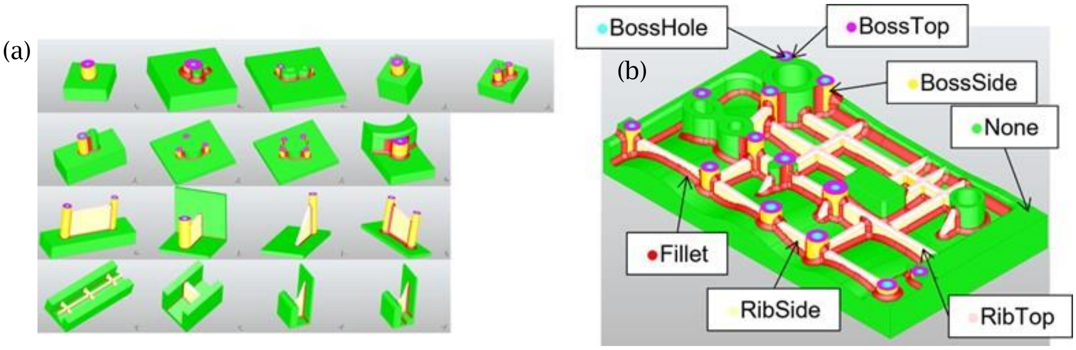

Fig. 4: Basic dataset: (a) 17 base models, (b) 7 face feature labels.

Method	Basic Dataset (Original Pose)		Basic Dataset (Rotated)		Practical Models (Original Pose)		Practical Models (Rotated)	
	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]	Acc [%]	mIoU [%]
Our method	99.43	98.85	**99.43**	98.83	76.30	47.51	76.56	47.80
Our method with DA	99.43	98.85	**99.43**	**98.84**	**82.32**	**53.54**	**82.42**	**53.92**
BRepNet + UV-Net [1]	**100.00**	**100.00**	62.70	28.33	71.92	47.46	48.09	14.23

Tab. 1: Recognition performance on the basic dataset and practical models; Acc denotes accuracy and mIoU denotes mean intersection over union.

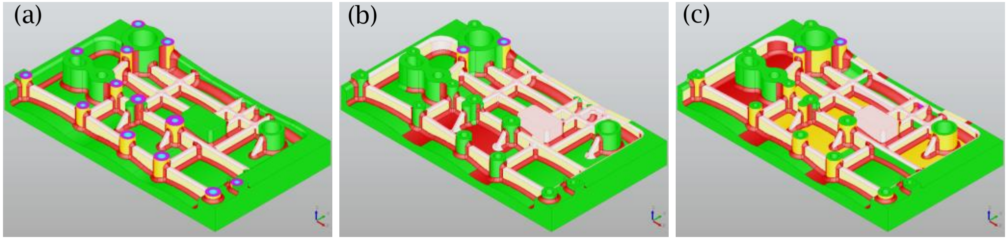

Fig. 5: Recognition results of a practical model: (a) ground truth, (b) our method, (c) our method + DA. Each face color represents the predicted feature label, same as Figure 4.

Conclusion:

A novel GNN-based FE feature recognition method from a B-rep CAD model was proposed. The recognition performance for boss and rib features were better than that of a similar existing method, and it significantly outperforms in rotated models. In addition, a data augmentation method was also proposed that could improve the performance against datasets slightly different from the training data. However, there is still room for improvement in recognition performance against complex models, which will be achieved by changing the training dataset generation or using transfer learning.

References:

[1] Autodesk AI Lab: https://github.com/AutodeskAILab/BRepNet. Referenced on 2023/1/28.

[2] Fey, M.; Lenssen, J. E.: Fast graph representation learning with PyTorch geometric, arXiv, 2019. https://doi.org/10.48550/arXiv.1903.02428

[3] Jayaraman, P. K.; Sanghi, A.; Lambourne, J. G.; Willis, K. D. D.; Davies, T.; Shayani, H.; Morris, N.: UV-net: Learning from boundary representations, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 11698-11707. https://doi.org/10.1109/CVPR46437.2021.01153

[4] Koenderink, J. J.; van Doorn, A. J.: Surface shape and curvature scales, Image and Vision Computing, 10(8), 1992, 557-564. https://doi.org/10.1016/0262-8856(92)90076-F

[5] Lambourne, J. G.; Willis, K. D. D.; Jayaraman, P. K.; Sanghi, A.; Meltzer, P.; Shayani, H.: BRepNet: A topological message passing system for solid models, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 12768-12777. https://doi.org/10.1109/CVPR46437.2021.01258

[6] Open Cascade Technology, https://dev.opencascade.org/doc/overview/html/index.html.

[7] Takashima, H.; Kanai, S.: Recognition of free-form features for finite element meshing using deep learning, Computer-Aided Design and Application, 19(4), 2022, 677-693. https://doi.org/10.14733/cadaps.2022.677-693

[8] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y.: Graph attention networks., arXiv, 2017. https://doi.org/10.48550/arXiv.1710.10903

[9] Zhang, Z.; Jaiswal P.; Rai R.: FeatureNet: Machining feature recognition based on 3D Convolution Neural Network, Computer-Aided Design, 101, 2018, 12-22. https://doi.org/10.1016/j.cad.2018.03.006