Free-Form Feature Classification for Finite Element Meshing based on Shape Descriptors and Machine Learning

Ippei Takaishi[1], Satoshi Kanai[2], Hiroaki Date[3], and Hideyoshi Takashima[4]

[1]Hokkaido University, i_takaishi@sdm.ssi.ist.hokudai.ac.jp
[2]Hokkaido University, kanai@ssi.ist.hokudai.ac.jp
[3]Hokkaido University, hdate@ssi.ist.hokudai.ac.jp
[4]AIS Hokkaido Inc., hideyoshi_takashima@ais-hokkaido.co.jp

Corresponding author: Satoshi Kanai, kanai@ssi.ist.hokudai.ac.jp

Abstract. Finite element mesh generation (FE meshing) from three-dimensional (3D) computer-aided design (CAD) models is generally the most critical process in the finite element analysis pipeline. In the FE meshing, several manufacturers strictly prescribe the meshing patterns for specific classes of free-form features such as "boss" or "rib" features on CAD models and, thus, establish company-specific FE meshing rules of where and how many node points of elements should be placed over and inside a form feature, to ensure the analysis accuracy. However, these features are currently recognized and extracted manually by experienced engineers. Therefore, it is crucial for manufacturers to develop software where features such as bosses or ribs with complex free-form surfaces can be extracted from CAD models and categorized based on prescribed meshing rules, where an FE mesh for the feature region can be automatically generated in accordance with the rules in order to realize a high-quality and reliable finite element analysis (FEA) pipeline. To this end, an algorithm of the free-form feature classification for FE meshing of a triangular surface mesh generated from a CAD model is proposed in this paper, which utilizes 3D shape descriptors, Bag-of-Features, and machine learning techniques. By using the triangular mesh and machine learning, the classification algorithm enables a uniform and expandable feature. Moreover, it employs shape descriptors of a point feature histogram as a local surface descriptor and a thickness histogram as a global volumetric descriptor. A combination of both descriptors yielded more excellent classification performance accuracy (92%) and recalls (95%–98%) than a single descriptor. Additionally, the classification performance is almost not affected by the key point sampling density and visual word length.

Keywords: Feature Classification, Feature Extraction, Machine Learning, Finite Element Method, Mesh Generation, Shape Descriptor, Bag-of-Features, Point Feature Histogram, Thickness Histogram.
DOI: https://doi.org/10.14733/cadaps.2020.1049-1066

1 INTRODUCTION

In recent times, finite element analysis (FEA) has enabled the manufacturing of efficient and reliable product designs. Finite element mesh generation (FE meshing) from three-dimensional (3D) computer-aided design (CAD) models is generally the most critical process in the FEA pipeline; therefore, fully automated meshing that can guarantee analysis accuracy is highly required to streamline the pipeline.

Several manufacturers strictly recommend FE meshing patterns for specific classes of free-form features on CAD models shown in Figure 1 (a) and, thus, established company-specific FE meshing rules of where and how many node points of elements should be placed over and inside a form feature, to ensure analysis accuracy. Meshing rules for "boss" or "rib" features, as illustrated in Figure 2, are often specially specified, as these play critical roles in securing strength for a part or transmitting forces between parts. As such, in Figure 1 (b), when an FE mesh is to be generated for a cylindrical boss feature, the node points of elements must be placed concentrically around a medial axis of the boss at an angle interval of 15 degrees. In the case of the rib feature shown in Figure 1 (c), the node points must be arranged along a ridge curve on top of the rib at a maximum interval of 3.0 mm. Therefore, it is crucial for manufacturers to develop software where features such as bosses or ribs with complex free-form surfaces can be extracted from CAD models and categorized under classes based on prescribed meshing rules such that an FE mesh for the feature region can be generated automatically in accordance with the rules to realize a high-quality and reliable FEA pipeline.

Although some feature recognition methods for FE meshing have been studied recently (see, e.g., [6], [17], [24]), these methods cannot be directly applied to the situation considered in this study for the following reasons.

First, the feature geometries discussed in previous studies (for example, in [6], [17], and [24]) are 2.5-dimensional, consisted only of simple planes and cylinders, and are bounded by sharply concave loops on a B-rep CAD model. In this study, however, based on Figure 1, a feature (i.e., boss and rib) that requires recognition is designed as a part of a cast or forged component's surface whose geometry is generally defined by 3D free-form surfaces. In addition, the feature is usually bounded by smooth free-form filet surfaces that are comparatively not discernible as the ones mentioned earlier. Therefore, conventional methods will not work efficiently for those free-form features.

Second, feature classes for FE meshing are usually defined subjectively based on the knowledge of skilled FEA engineers, and they often differ from one company to another. On the contrary, the recognition algorithms considered in previous studies, for example in [6], [17], and [24], are

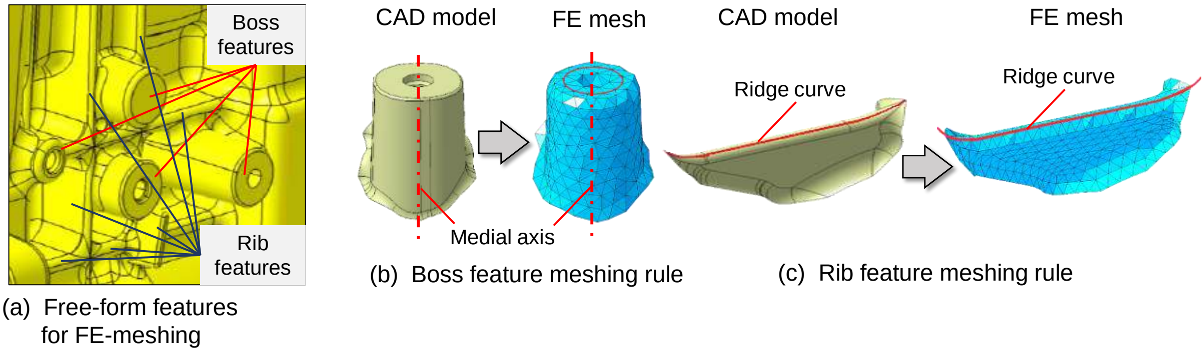


Figure 1: Free-form features and examples of FE meshing rules for the features.

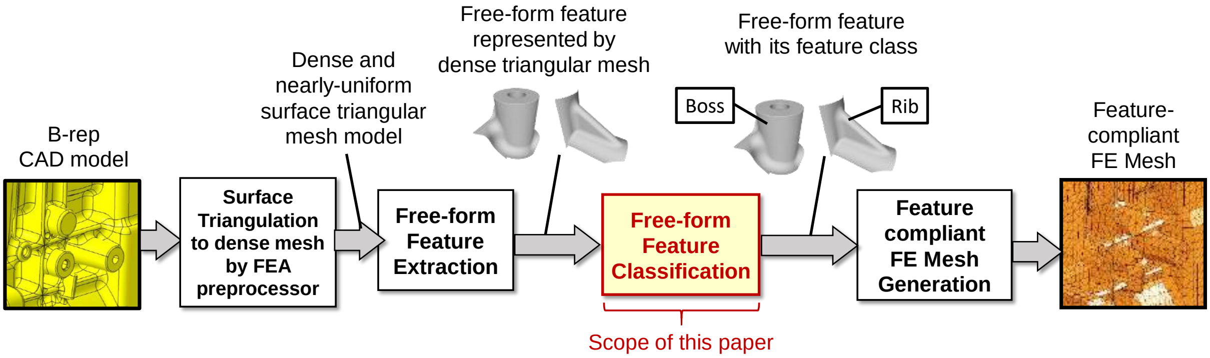


Figure 2: Overall feature-based FE meshing process and the scope of this paper.

designed for the elaborate procedural search of loops on a B-rep CAD model and coded in an ad hoc manner to fit the recognition of specific feature classes. Consequently, these algorithms are not easily expanded when a new feature class is added or a current feature class is to be modified.

Third, most previous studies assume that an input B-rep CAD model does not have any topological or geometric defect. However, it is well-known that the data quality of CAD models may degrade due to loss of information during the translation process and that some quality issues on the B-rep data (e.g., small cracks between faces) may be inducted. Therefore, a recognition algorithm relying mainly on the topological and geometrical search on the B-rep CAD model is more likely to fail.

To address the issues mentioned above, we propose, in this paper, an algorithm of the free-form feature classification for feature-based FE meshing, which, as shown in Figure 2, we regard to consist of three steps: feature extraction from the CAD model, feature classification, and feature-compliant mesh generation. In this paper, we focus only on feature classification. In principle, the proposed algorithm accepts a dense and nearly-uniform triangular mesh of a free-form feature generated from a B-rep CAD model by using a preprocessor of a commercial FEA software. Moreover, it identifies feature class labels, such as boss and rib, of the input mesh model via 3D shape descriptors, bag-of-features (BoF), and machine learning.

The advantages of the proposed algorithm are summarized as follows:

- It combines local and global shape descriptors, and the local shape descriptor is based on the local mean curvatures on a triangular mesh, which allows one to encode both the local and global features' geometry as a single multi-dimensional vector—even when a feature has complex free-form shapes bounded by smooth filet surfaces. Moreover, the BoF technique facilitates the application of the local shape descriptor representation to the machine learning scheme, and the control of the computational complexity of the classification process by limiting the number of visual words; thereby, addressing the first problem mentioned earlier.
- It employs machine learning, which makes the design of the feature classification algorithm uniform and portable regardless of the classes. As such, the algorithm can be easily expanded by the addition of newly labeled training feature samples; thereby, addressing the second problem.
- Instead of B-rep representation, it uses free-form features based on shape descriptors only defined at the vertices on a dense triangular mesh, and the descriptor evaluation is not directly affected by the topological defects on the B-rep CAD models; thus, avoiding unstable feature extraction and classification processes caused by product data quality issues, which solves the third problem. Of course, the conformal surface triangulation from B-rep CAD models is still a quite challenging task (e.g., [19],[32]); therefore, non-conformal mesh

topology sometimes might be included in our input dense triangular mesh. However, since we apply the uniform and sparse sampling strategy to the vertices for generating key points for the descriptors and they do not rely on the conformity of the mesh topology, our feature classification process is less affected by the non-conformal surface triangulation issue.

2 RELATED WORK

The feature classification method considered in this paper relates to two common topics in geometric modeling research interest: 3D feature recognition and 3D shape retrieval. We discuss their outlines and drawbacks based on free-form feature classification for FE meshing in Subsections 2.1 and 2.2.

2.1 Feature Recognition

Over the past 20 years, a great deal of research has been published on the feature recognition method from CAD models, especially for machining feature extraction. An outline of the method is well-discussed in [12]. There are several extraction approaches, such as the graph-based approach [9]. Also, as an extraction approach, a machine learning approach based on the traditional neural network has been introduced in machining feature extraction from CAD models (see, e.g., [18],[25],[26],[28], and [36]). Recently, Zhang et al. [41] proposed a machining feature recognition based on the novel and sophisticated 3D convolutional neural network. The method presented in [41] learns the distribution of various manufacturing feature shapes across public 3D model data sets and can recognize particular types of manufacturing features from low-level geometric data such as voxels. Moreover, their recognition algorithm based on the 3D convolutional neural network enables significant improvements over the state-of-the-art manufacturing feature detection techniques.

However, as in other previous studies, the feature geometries considered in [41] are basically 2.5-dimensional, consisted only of simple planes and cylinders, and are bounded by sharply concave loops on a B-rep CAD model. In our study, however, a feature to be recognized (i.e., boss and rib) is basically designed as a part of a cast or forged component's surface whose geometry is generally defined by 3D free-form surfaces. Moreover, the feature is usually bounded by smooth free-form filet surfaces that are comparatively not discernible as the ones considered in other studies. Therefore, it is doubtful whether the previous methods would work well or not for the free-form features.

On the other hand, several automatic recognition methods of free-form machining features from CAD models have also been studied (see, e.g., [7], [10], and [37]). For example, Sunil et al. [37] proposed a free-form machining feature recognition method based on a hybrid region segmentation algorithm that works on the triangulated STL mesh model generated from a B-rep CAD model. In their algorithm, a variety of protrusion and depression features such as bends, beads, and dimples on a sheet metal parts can be extracted based on the curvature properties of the triangulated model. Nevertheless, they validate their algorithm only using a few examples of sheet metal parts with simple shapes, and the boundaries of the feature they extracted are more easily-discernible than the ones obtained in this paper. Moreover, their feature recognition algorithm based on region identification and region merging is specifically coded in an ad hoc manner to fit specified feature classes to be recognized. Therefore, the recognition algorithm is not easily expanded when a new feature class is added; thus, it lacks portability. Recently, Cai et al. [7] proposed a free-form machining feature recognition algorithm similar to that considered in [37], and their feature segmentation algorithm is not only based on the curvature properties of a free-form surface but also its manufacturability. However, the authors only validated their algorithm using only one simple-shaped workpiece; besides, similar technical issues as in [37] also remain when the algorithm is applied to the free-form feature recognition for FE meshing.

Some feature recognition methods originally aimed for FE meshing have also been studied (see, e.g., [6], [17], and [24]). Lai et al. [17] proposed a method that recognizes rib features from a B-rep CAD model by finding specific topological and geometrical patterns of virtual loops around these features and then decomposes them into regions that can be meshed with hexahedral or prismatic FE meshes. Further, Lu et al. [24] introduced a feature-based hexahedral meshing method, which decomposes a B-rep CAD model into a set of hex meshable volumes by extracting protrusion features bounded by concave zones through the identification of three loop types in a CAD model to serve as feature boundaries. Moreover, Boussuge et al. [6] presented a method for recognizing protrusion features on a CAD model whose shape can be partitioned into plate and shell elements. However, the feature geometries considered in these studies are also 2.5-dimensional, consisting only simple planes and cylinders, and bounded by sharply concave loops on a B-rep CAD model. Therefore, the methods presented in these studies, also, cannot be applied to the free-form feature recognition for finite element meshing.

2.2 Shape Retrieval and Shape Descriptor

When a free-form feature has been extracted from a CAD model, the feature classification process into specific types (boss or rib) can be regarded as a kind of mesh classification problem, which often appears in content-based 3D shape retrieval. Recently, many researchers have investigated specific problems in content-based 3D shape retrieval. Also, an extensive amount of literature can be found in the related fields. Detailed reviews of these shape retrieval techniques are presented in [38] and [16]. The international competitions on the large-scale 3D shape retrieval are held periodically, where the performance of retrieval algorithms are evaluated and updated [31].

These studies provide solutions to the problem of quantifying the similarity between two geometries and classifying the geometry into semantically-reasonable classes based on the similarity measures. Shape descriptor approach has been recognized as one of the methods for providing solutions to this problem. Shape descriptor refers to a description method that utilizes a numeric descriptor (also known as a feature vector) of a given shape to characterize the shape uniquely [16]. This method has been studied widely in recent times. For example, Osada et al. [27] introduced shape distributions as a descriptor for content-based 3D shape retrieval and found that the histogram of distances between two randomly chosen points on the mesh surface yields a robust shape descriptor. Furthermore, Ankerst et al. [2], Ip et al. [15], and Wohlkinger et al. [40] proposed histogram-based shape descriptors that are similar to that presented in [27]. Unfortunately, most of these descriptors only deal with the classification of whole shapes that have relatively clear differences and do not contain free-form surfaces required in finite element meshing. Hence, they do not classify features such as bosses and ribs whose shape differences are not so obvious.

Recently, more sophisticated shape descriptors for feature recognition have been proposed. For example, Sun et al. [34] introduced a multi-scale spectral shape descriptor called heat kernel signature (HKS) based on the heat diffusion process and demonstrated its effectiveness for multi-scale shape matching. Further, Aubry et al. [4] proposed the wave kernel signature (WKS) for characterizing points on a 3D shape and applied it to non-rigid registration between deformable objects. Machine learning based shape descriptors have also been introduced. For example, Litman et al. [22] proposed a machine learning scheme for a generic family of spectral shape descriptors that generalized the HKS and WKS and applied the scheme to non-rigid registration between deformable objects. Sun et al. [35] develop descriptors for non-isometric registration by embedding the spectral shape descriptors into a different metric space, and a deep neural network to find such an embedding. While these descriptor studies based on the spectral shape descriptors have the potential to solve the free-form feature classification problem, most of them focus mainly on the shape correspondence or non-rigid registration between deformable objects and do not apply the descriptor to the feature extraction or classification problems.

Recently, applications of spectral shape descriptors in feature recognition were considered. For example, Harik et al. [13] utilize a multi-scale persistent heat signature to recognize traditional mechanical features; while Shi et al. [33] employ an HKS for a feature-based manufacturability analysis in additive manufacturing. However, the feature geometries treated in these two studies mainly consist of simple planes and cylinders and are bounded by sharply concave loops.

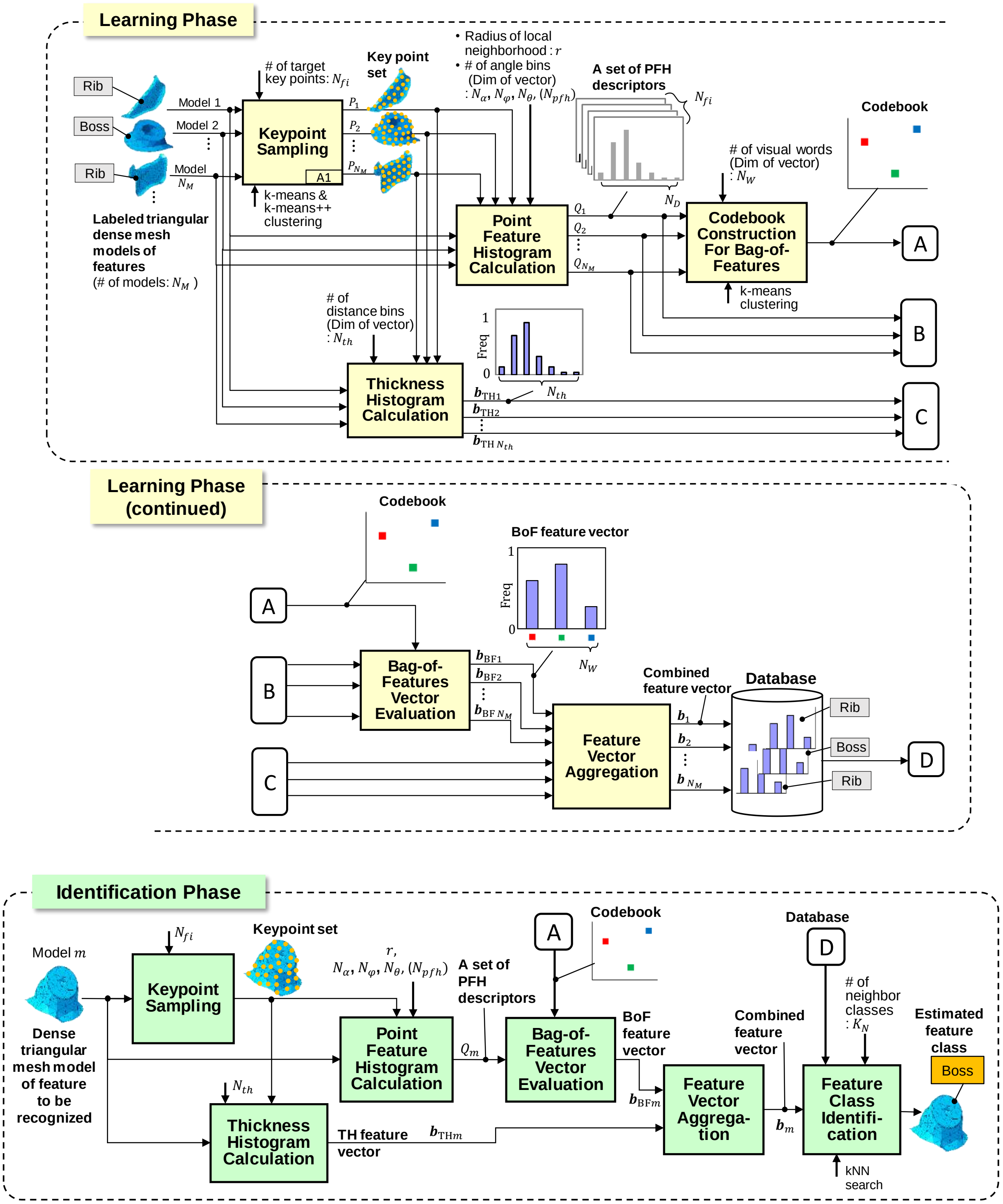


Figure 3: The proposed free-form feature classification process.

As a part of the efforts on 3D shape retrieval, algorithms for part-in-whole retrieval are also proposed in some researches (see, e.g., [11], [20], and [30]), where a part of the whole shape can be inputted as a query, and part-in-whole matching is performed. The part-in-whole retrieval algorithm has the potential to solve the free-form feature extraction problem for FE meshing directly. Furthermore, a sampling strategy for 3D part-in-whole retrieval algorithm using depth image rendering, SIFT features, and BoF, has also been studied [39]. However, these studies only deal with the classification of models whose shapes have relatively clear differences. Moreover, they cannot explicitly indicate which portions on a whole object are partially matched with the query shape.

3 FEATURE CLASSIFICATION METHOD FOR FE MESHING

3.1 Overview

The feature classification process proposed in this study accepts a triangular mesh model of the free-form feature as an input. We assumed that this input triangular mesh is the dense mesh shown in Figure 4(a), and its vertex density is relatively high. This dense triangular mesh is different from a CAD mesh, which is very sparse and mostly consists of highly non-uniform triangles. Robust algorithms for conformal triangulation from B-rep model have been gradually improved (see, e.g., [19] and [32]), and they can be used to generate the input triangular meshes in this study. However, alternatively, we utilized a preprocessor of a commercial FEA software (Altair HyperMesh [14]) to automatically generate the dense triangular mesh directly from a B-rep model. So, far, using the preprocessor, the conformal dense triangular meshes available for our feature classification can be obtained. Since the goal of this research is to accomplish a feature-compliant finite element mesh generation, the use of a commercial FEA software for preparing the input triangular mesh does not pose any problem in this study.

During classification, we identify one of the feature class labels that has been trained by supervised learning, in which a different FE meshing rule is prespecified. Currently, three feature classes (i.e., "rib," "boss," and "others") can be discriminated where several manufacturers often define company-specific FE meshing rules. Nevertheless, we can easily extend the feature classes to be distinguished by simply adding class labels for the training samples.

Figure 3 provides an overview of the proposed feature classification process; consisting mainly of learning and identification phases. In the learning phase, a large collection of labeled triangular mesh models of manually labeled free-form features class is provided as an input. Next, for each triangular mesh, two shape descriptors are computed at key points uniformly sampled on the mesh, namely, point feature histogram (PFH) [29] as a local shape descriptor, and thickness histogram (TH) [23] as a global volumetric descriptor.

Afterward, based on the BoF concept [5], a "codebook" is constructed via k-means clustering, from a set of the PFHs included in all labeled triangular meshes. Then, a BoF feature vector is evaluated for every labeled triangular mesh based on the codebook; whereas the TH descriptor is represented as a TH feature vector using all key points on the mesh. Both BoF and TH feature vectors form a combined feature vector that encodes the local surface and global volumetric geometry of a free-form feature included in the labeled triangular mesh input for learning. Finally, a set of the combined feature vectors is stored in a database for use in the identification phase.

A similar procedure is followed in the identification phase. Initially, PFH and TH descriptors are evaluated at every key point on a triangular mesh of an input free-form feature, and then a combined feature vector is calculated. Afterward, the distances between the feature vector of the input feature and the ones stored in the database are evaluated. Finally, the class of the input feature is determined using the k-nearest neighbor (k-NN) algorithm. Details of the classification algorithm are described in the subsequent sections.

As described in Subsection 2.2, there has been considerable research on 3D shape descriptors (see, e.g., [38] and [16]). The reasons why we chose the PFH and TH as shape descriptors in our feature classification are as follows. Firstly, the PFH descriptor can well encode the local curvature distributions around the key points on a free-form feature and is more sensitive to the curvature changes in the feature geometry than distance-based global descriptors such as shape distributions [27]. Secondly, since the PFH descriptor encodes the curvature distributions using a relatively large-dimensional (375-dimensional) vector, it can provide more local curvature information of the free-form surface than other descriptors. Thirdly, for the feature recognition process, the PFH descriptor, which encodes the local surface geometry on a mesh surface, is more suited than the global shape descriptor because we can simply evaluate the PFHs both at the key points on a free-form feature and those in the whole geometry of the model, and directly evaluate the similarity of the descriptor values between them. Lastly, as remarked in [1], the hybrid descriptor approach where both the local (PFH) and global shape descriptors (TH) are used at the same time generally outperforms—in terms of the recognition accuracy—their counterpart approaches that use only the local descriptor or global descriptor. However, the quantitative comparison of our proposed hybrid descriptor approach with other shape descriptors in terms of the classification accuracy will be considered in a future study.

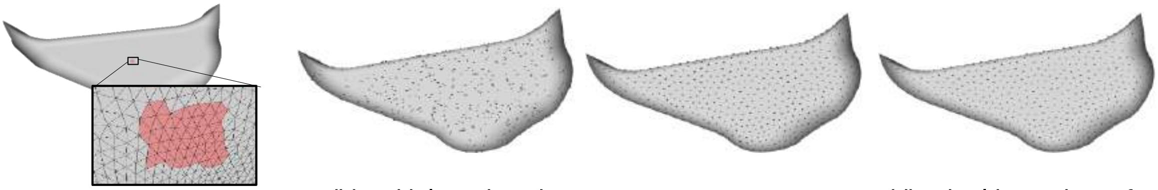
Since we applied the uniform and sparse sampling strategy to the vertices for evaluating the descriptors, and they do not rely on the conformity of the mesh topology; our feature classification process is less affected by the non-conformal triangulation issue.

3.2 Local Shape Descriptor using Point Feature Histogram (PFH)

First, for a labeled triangular mesh i ($\in I$), a set of key points $P_i = \{\boldsymbol{p}_i^j\}$ are sampled from the vertices on i, where I denotes a set of labeled triangular meshes for learning. The PFH [29] is then evaluated as a local shape descriptor $\boldsymbol{q}_i^j (\in Q_i, j \in J_i)$ at every key point $\boldsymbol{p}_i^j (\in P_i)$, where Q_i is a set of local shape descriptors for a mesh i and J_i is a set of descriptor indices for a mesh i. We assumed that an initial triangular mesh is almost uniformly triangulated, and its vertex density is relatively high (for example, the average edge length is less than 1 mm as shown in Figure 4(a)).

As described in [20], excellent classification results are achieved by sampling the key points on a mesh as uniformly as possible. Herein, we adopt the k-means clustering as the sampling method of the key points because the method is easy to implement and it can easily partition the set of mesh vertices into the specified number of uniformly distributed clusters. However, if we randomly select N_{f_i} vertices on the mesh (see, Figure 4(b)) as initial cluster centers of the k-means clustering, the resultant key points do not necessarily distribute uniformly on the mesh. To avoid this, we first perform the k-means++ clustering [3] for the mesh vertices to obtain more uniformly distributed N_{f_i} initial cluster centers (see, Figure 4(c)) than those randomly selected, and then apply the k-means clustering to the cluster centers. Finally, we took the N_{f_i} vertices on the mesh i of each point closest to a cluster centroid for adoption into a set of key points $P_i = \{\boldsymbol{p}_i^j\}$ (see, Figure 4(d)).

Since our initial triangular mesh is dense and almost uniformly triangulated, a cluster centroid obtained from k-means clustering is usually placed at a position very close to a mesh vertex such that the distance between the centroid and the vertex is negligible. Therefore, we can select the closest vertex as a key point location. This selection of the mesh vertex as a key point also



(a) Initial mesh density (b) Initial vertices by random sampling (c) Initial vertices by k-means++ clustering (d) Final keypoints after k-means clustering

Figure 4: Key point sampling process on the dense mesh.

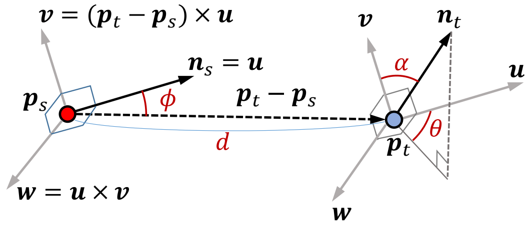


Figure 5: Point feature histogram (PFH) [29].

streamlines the local neighbor search and descriptor evaluation processes. Therefore, in our dense mesh setting, the use of k-means clustering so far works effectively and does not pose any problem in this study. However, to obtain non-uniform and sampling on the mesh for the key point selection, such as non-uniform or curvature-sensitive sampling, we introduced a more flexible and sophisticated method. In this case, Poisson-Disk sampling method on mesh domain (as utilized in [8] and [21]) will provide more impressive results than the k-means clustering; nevertheless, we shall consider this in a future study.

Next, at a key point $\boldsymbol{p}_i^j$, PFH is calculated as a local shape descriptor. The PFH encodes local geometric properties by generalizing the mean curvature around $\boldsymbol{p}_i^j$ by using a multi-dimensional vector. In the process, a set of connected local neighborhood vertices $V(\boldsymbol{p}_i^j, r)$ centered at a key point $\boldsymbol{p}_i^j$ is extracted on the mesh lying at a distance r from $\boldsymbol{p}_i^j$. If a non-conformal triangulation happens around the key point in the input triangular mesh, it is difficult to find the connected local neighborhood vertices by relying on the mesh topology only. However, in this case, we can switch the neighborhood search to a distance-based one. Then, for all pairs of vertices $(\boldsymbol{p}_s, \boldsymbol{p}_t)$ included in $V(\boldsymbol{p}_i^j, r) \cup \{\boldsymbol{p}_i^j\}$, the three angles $\alpha_{st}, \varphi_{st}$, and θ_{st} shown in Figure. 5 are obtained using equations (3.1) and (3.2) below:

$$\alpha_{st} = \cos^{-1}(\boldsymbol{v} \cdot \boldsymbol{n}_t); \ \varphi_{st} = \cos^{-1}\{\boldsymbol{u} \cdot (\boldsymbol{p}_t - \boldsymbol{p}_s)/d\}; \ \theta_{st} = \tan^{-1}\{(\boldsymbol{w} \cdot \boldsymbol{n}_t)/(\boldsymbol{u} \cdot \boldsymbol{n}_t)\}; \quad (3.1)$$

$$\boldsymbol{u} = \boldsymbol{n}_s; \ \boldsymbol{v} = \boldsymbol{u} \times (\boldsymbol{p}_t - \boldsymbol{p}_s)/\|\boldsymbol{p}_s - \boldsymbol{p}_t\|; \ \boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}, \quad (3.2)$$

where $\boldsymbol{n}_s$ and $\boldsymbol{n}_t$ are outward-directed unit normal vectors on the mesh at $\boldsymbol{p}_s$ and $\boldsymbol{p}_t$, respectively.

Based on the values of $\alpha_{st}, \varphi_{st}$, and θ_{st}, a vote is conducted for a corresponding bin in their quantized intervals. By summing up the votes for all pairs of vertices, we obtain a histogram of the votes, and the normalized histogram finally gives the PFH descriptor at $\boldsymbol{p}_i^j$. If we, respectively, partition the intervals of $\alpha_{st}, \varphi_{st}$, and θ_{st} into N_α, N_φ, and N_θ bins, the histogram will consist of N_{pfh} ($= N_\alpha N_\varphi N_\theta$) bins, which can be represented by a N_{pfh}-dimensional vector $\boldsymbol{q}_i^j$. In this study, we select $N_\alpha = 5, N_\varphi = 5$, and $N_\theta = 15$ based on a preliminary experiment; therefore, herein, the PFH is represented by a 375-dimensional vector $\boldsymbol{q}_i^j \in R^{375}$. The descriptor is based on the relationship between the points in the k-neighborhood and their estimated surface normals; making the PFH rotation and translation invariant.

3.3 Feature Vector Evaluation using Bag-of-Features (BoF)

Bag-of-features (BoF) is a machine learning classification scheme that has been extensively used in image classification [5]. The idea behind BoF is to represent an image as a set of *features* consisting of a key point and a descriptor. The *features* are quantized to construct a limited number of *visual words* (codes) into a *codebook*. Afterward, each *feature* of the image is assigned to its nearest code, and the image is represented as a frequency histogram of the codes. From the histogram, the image

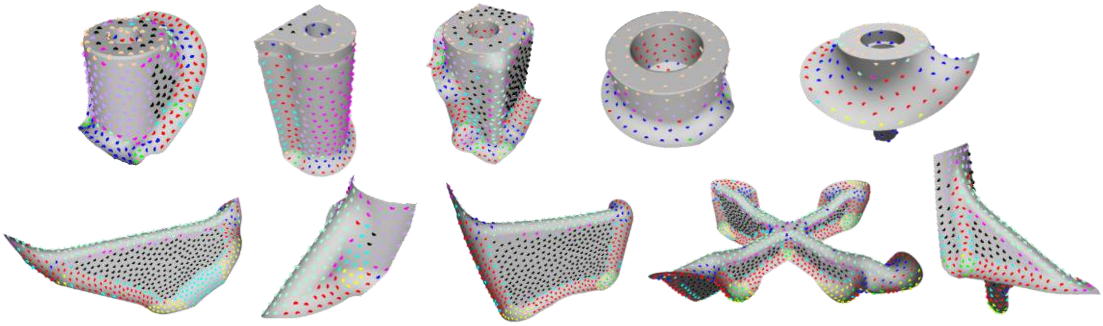


Figure 6: Visual words assignment to PFH descriptors on features (same-colored points indicate the same visual word assigned).

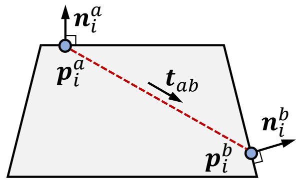


Figure 7: Thickness histogram (TH) [23].

can be categorized under the closest code. BoF enables a compact representation of the features for the classification and rapidity of search.

We applied BoF to the 3D free-form feature classification represented by a triangular mesh model. First, we performed *k*-means clustering for the set of PFH descriptors for all key points on the labeled triangular mesh $\{\boldsymbol{q}_i^j\}_{i\in I}^{j\in J_i}$ under a specified number of visual words N_w, and obtain N_w centroids of the clusters (visual words) as $\boldsymbol{c}_k$ $(\in R^{375}, k \in [1, N_w])$. Then, the set of centroids $\Gamma = \{\boldsymbol{c}_k\}_{k\in[1,N_w]}$ configures a codebook.

Subsequently, for all descriptors at all key points $\{\boldsymbol{q}_i^j\}^{j\in J_i}$ on a triangular mesh i, we identify which visual word $\boldsymbol{c}_k$ each descriptor $\boldsymbol{q}_i^j$ is closest to, and the appearance frequency of each word $\boldsymbol{c}_k$ $(k \in [1, N_w])$ in the codebook is represented as a histogram. Finally, the histogram is normalized to give a multi-dimensional BoF feature vector $\boldsymbol{b}_{\mathrm{BF}i} = [b_{\mathrm{BF}i}^1, b_{\mathrm{BF}i}^2, \ldots, b_{\mathrm{BF}i}^{N_w}]$ $(b_i^l \in [0,1], i \in I)$ representation that encodes the local surface geometry of the free-form feature represented by the mesh i.

Figure 6 provides examples describing the assignment of different visual words to PFH descriptors at key points in the case of N_w = 10. As shown in the figure, different words are loosely assigned to different local regions in a feature exhibiting similar geometries (planar or cylindrical regions).

3.4 Global Shape Descriptor using Thickness Histogram (TH)

While the BoF feature vector encodes and summarizes the geometry of a free-form feature, the PFH only encodes local surface geometries around a key point. Therefore, the BoF feature vector does not necessarily represent the global volumetric properties of free-form features.

To make up for the lack of volumetric properties of a feature, we considered a thickness histogram (TH) [23]. The TH descriptor encodes the statistical thickness distribution of an object as a histogram. When constructing the TH (see, Figure. 7), we selected a pair of different key points $\boldsymbol{p}_i^a$ and $\boldsymbol{p}_i^b$ on a triangular mesh i and evaluated a weight W_{ab} using the equation (3.3):

$$W_{ab} = \frac{(\boldsymbol{t}_{ab} \cdot \boldsymbol{n}_i^a)(\boldsymbol{t}_{ab} \cdot \boldsymbol{n}_i^b)}{d_{ab}^2} \tag{3.3}$$

where $\boldsymbol{t}_{ab} = (\boldsymbol{p}_i^b - \boldsymbol{p}_i^a)/\|\boldsymbol{p}_i^b - \boldsymbol{p}_i^a\|$, $\boldsymbol{n}_i^a$ and $\boldsymbol{n}_i^b$ are the outward-directed unit normal vectors on the mesh at $\boldsymbol{p}_i^a$ and $\boldsymbol{p}_i^b$, respectively, and $d_{ab} = \|\boldsymbol{p}_i^b - \boldsymbol{p}_i^a\|$. Weight W_{ab} is voted for one of the N_{th} bins, each of which corresponds to a quantized interval for d_{ab}. We performed this vote for all pairs of key points on a mesh i and obtained a histogram of votes for d_{ab}. By normalizing the cumulative frequency of the histogram to 1, we obtain a multi-dimensional TH feature vector $\boldsymbol{b}_{\mathrm{TH}i} = [b_{\mathrm{TH}i}^1, b_{\mathrm{TH}i}^2, \ldots, b_{\mathrm{TH}i}^{N_{th}}]$ for the mesh i.

Finally, we combined the BoF feature vector $\boldsymbol{b}_{\mathrm{BF}i}$ with the TH feature vector $\boldsymbol{b}_{\mathrm{TH}i}$ to construct a combined feature vector $\boldsymbol{b}_i = [\, \boldsymbol{b}_{\mathrm{BF}i} \mid \boldsymbol{b}_{\mathrm{TH}i} \,]$, which encodes both the local surface and the global volumetric geometry of a free-form feature represented by the mesh i. We utilized a set of combined feature vectors for all labeled triangular meshes $\{\boldsymbol{b}_i\}_{i \in I}$ for learning and class identification.

3.5 Feature Class Identification

As described earlier, the identification phase follows the same procedure as in the learning case. Here the PFH and TH descriptors are evaluated at a set of key points on a triangular mesh m of an input free-form feature, and their combined feature vector $\boldsymbol{b}_m$ is calculated. Next, the distance between the feature vectors $\boldsymbol{b}_m$ and $\boldsymbol{b}_i$ stored in the database is determined over $\{\boldsymbol{b}_i\}_{i \in I}$. Finally, the K_N classes for which K_N feature vectors closest to $\boldsymbol{b}_m$ in $\{\boldsymbol{b}_i\}_{i \in I}$ belong are identified through the k-NN algorithm, and the feature class of a triangular mesh m is determined by a majority vote of the K_N classes.

4 FEATURE CLASSIFICATION EXPERIMENT

4.1 Dataset and Parameters

As there was no publicly available data set of 3D free-form features, we prepared the labeled samples of boss and rib features. Under the direction of an FEA professional working at an engineering company, we then picked up a set of faces representing boss, rib, and the other classes of features from 30 solid models of forged automotive parts via a CAD system, CATIA-V5. Figure 8 shows a list of the 75 bosses, 87 ribs, and 23 other classes of features we collected. The longitudinal size of the features ranges roughly from 30 mm to 120 mm. Afterward, we constructed the dense triangular meshes of the samples using a FEM preprocessor (Altair HyperMesh [14]) and assigned a true feature class label for each mesh based on the direction of the professional. These labeled triangular meshes were used for the learning.

In the learning phase, for all mesh i included in a set of labeled triangular meshes I ($|I| = 185$), a set of key points $P_i = \{\boldsymbol{p}_i^j\}$ was first sampled, and a combined feature vector $\boldsymbol{b}_i = [\, \boldsymbol{b}_{\mathrm{BF}i} \mid \boldsymbol{b}_{\mathrm{TH}i} \,]$ was then evaluated based on the procedures described in Subsections 3.3 and 3.4. As we conducted 10-fold cross-validation for the classification experiment, first we partitioned a set of labeled triangular meshes I into ten disjoint subsets $I_1, I_2, \cdots, I_{10}$, each of which includes bosses, ribs, and other classes nearly uniformly. Then, we performed training on all but one of the subsets (9 subsets) to construct the database from their combined feature vectors. After that, in the identification phase, we estimated the class of features included in the subset that was not used for training, and evaluate the correctness of the class by comparing it with the mesh label. This process is repeated ten times with a different subset reserved for identification and excluded from training each time. We obtained the overall classification performance by computing the average of the accuracy metrics of each time.

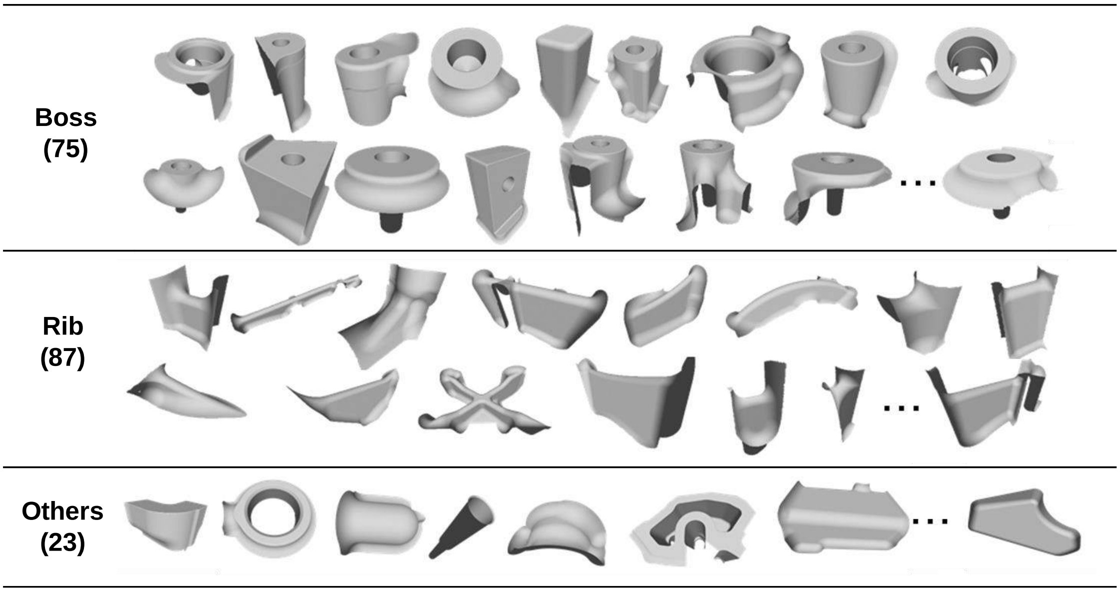


Figure 8: Labeled free-form feature examples.

Parameter	N_w	N_{fi}	N_v	r	N_{pfh}	b_α, b_φ	b_θ	N_{th}	K_N
Meaning	Number of visual words	Number of key points per one feature	Number of vertices on the mesh	Local neighbor radius in PFH calculation	Dimension size of PFH	Number of bins for α and φ in PFH	Number of bins for θ in PFH	Dimension size of TH	Number of the nearest neighbors used in class identification
Value	10	N_v/100	-	2.5[mm]	375	5	15	50	3

Table 1: Parameter settings for the classification.

Further, we compared the differences in feature classification performance of BoF, TH, and combined feature vectors. As the result slightly depended on the number of the visual words N_w, we performed the classification at different N_w settings and selected $N_w = 10$, which yielded the best result. Table 1 summarizes the final parameter settings for the descriptor calculation; hence, their optimum settings are determined experimentally.

4.2 Classification Results

Table 2 summarizes the confusion matrices, recalls, and accuracies of the classification. Using either the BoF or TH feature vector only, we achieved 89% or 84% accuracy in the classification, respectively. The accuracy increased to 92% with the combined feature vector, while the recall of “boss” and “rib” features further reached 95% and 98% accuracy, respectively. Based on this, PFH and TH proved to complement each other, and their combination, rather than standing alone, yielded excellent classification performance of geometries of the features.

PFH		Predicted Class			Recall
		Boss	Rib	Others	
True Class	Boss	70	5	0	0.93
	Rib	2	83	2	0.95
	Others	4	6	13	0.57
Accuracy		0.89			

(a) BoF feature vector

TH		Predicted Class			Recall
		Boss	Rib	Others	
True Class	Boss	61	11	3	0.81
	Rib	4	83	0	0.95
	Others	6	4	13	0.57
Accuracy		0.84			

(b) TH feature vector

PFH+TH		Predicted Class			Recall
		Boss	Rib	Others	
True Class	Boss	71	4	0	**0.95**
	Rib	1	85	1	**0.98**
	Others	4	4	15	**0.65**
Accuracy		**0.92**			

(c) Combined feature vector

Table 2: Classification performances using BoF, Th, and combined feature vectors.

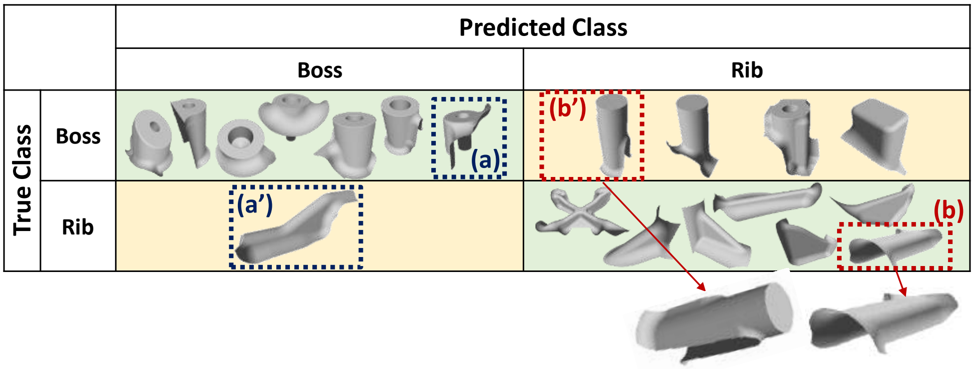


Figure 9: Examples of correctly and wrongly classified features: (a') Wrongly classified rib, (a) Boss feature nearest to (a') in the k-NN algorithm in the feature class identification, (b') Wrongly classified boss, (b) Rib feature nearest to (b') in the k-NN algorithm. The boss without a hole (b') closely resembled the rib (b).

Nevertheless, as shown in Figure 9, a small number of misclassifications were observed, especially with a boss feature without any hole improperly categorized as a rib, or with a rib whose thickness distribution is non-uniform incorrectly classified as a boss. For the former, the geometry of the boss without a hole (see, Figure 9(b')) closely resembled a rib (see, Figure 9(b)) if its thickness is close to that of a rib. Additionally, in the training set, there were a much lesser number of bosses with a hole than those without. For the latter, curvatures around the ridge of the rib (see, Figure 9(a')) closely resembled the curvature of the outer cylindrical regions of the boss (see, Figure 9 (a)), and the horizontal thickness of the rib was very close to most of the sampled bosses. Therefore, in both cases, PFH and TH failed to differentiate. The solution to these specified misclassification instances is open for future study.

Keypoint sampling	≈10s / model
PFH Calculation	≈20s / model
TH Calculation	≈1s / model
Codebook Construction	≈20s
BF Feature Vector Evaluation	0.1s / model

(a) Learning Phase

Keypoint sampling	≈10s
PFH Calculation	≈20s
TH Calculation	≈1s
BF Feature Vector Evaluation	0.1s
Feature Class Identification	50ms
Total / model	≈31s

(b) Identification Phase (per model)

Table 3: Processing time in the learning and identification phase (CPU: Core-i9-9700).

Table 3 summarizes the processing time for the learning and identification phase. In its present form, the algorithm spends the majority of time performing the key point sampling and PFH calculation. Thus, there is still a need to improve the efficiency of the proposed classification algorithm by introducing parallel processing strategies for the key point sampling and local neighbor vertex search in the descriptor calculation. Moreover, introducing a faster version of the PFH (FPFH) [29] will also improve the PFH calculation process.

4.3 Impact of Key Point Sampling Density and Visual Word Length

In the feature classification algorithm proposed in this study, the selection of the key point sampling density and visual word length N_w is critical to the performances. In the experiment discussed in Subsection 4.2, they were selected such that an optimum result is obtained in the other preliminary experiments. However, it is desirable for the algorithm to be relatively insensitive to these settings.

Therefore, we investigated in detail how the classification performance is affected by the change of the key point sampling density and visual word length used in the BoF feature. Figure 10 shows how the accuracies of the classification change as the visual word length N_w runs from 2 to 15, and the key point sampling density decreases tenfold. The size of the features ranges from 30 mm to 120 mm. We only evaluated the change in classification accuracy only by using the BoF feature vector based on the PFH descriptor because the TH feature vector is not affected by the visual word length.

As shown in Figure 10, it was found that the classification performance did not drastically drop; besides, the value was kept over 80% unchanged even when the key point sampling density decreased tenfold. Also, it was found that the key points were very sparsely populated on the feature surfaces at the lowest density. On the other hand, the accuracy gradually increased to about 89% until the visual word length reached 10; however, it was nearly saturated after that point. From this investigation, our proposed feature classification performance based on the BoF feature vector is less sensitive to the key point sampling density and visual word length.

5 CONCLUSIONS

In this paper, we presented an algorithm of the free-form feature classification for FE meshing of a triangular mesh, which utilizes 3D shape descriptors, BoF, and machine learning techniques. By using the triangular mesh and machine learning, the classification algorithm enables a uniform and expandable feature. Moreover, it employs shape descriptors of a PFH as a local surface descriptor and a TH as a global volumetric descriptor. A combination of both descriptors exhibited more efficient classification performance accuracy (92%) and recalls (95%–98%) than a single descriptor. Through an experiment, the effectiveness of the proposed free-form feature classification algorithm was confirmed. Besides, the classification performance is almost not affected by the key point sampling density and visual word length.

# of visual words	Key point sampling density [$\times$ 0.0173 keypoints$/mm^2$]										Average accuracy	Std. Deviation
	1	2	3	4	5	6	7	8	9	10		
2	0.51	0.55	0.52	0.49	0.44	0.51	0.51	0.47	0.48	0.47	0.50	0.03
3	0.65	0.66	0.69	0.58	0.71	0.7	0.55	0.55	0.72	0.74	0.66	0.07
4	0.72	0.71	0.7	0.73	0.7	0.71	0.74	0.71	0.68	0.7	0.71	0.02
5	0.8	0.75	0.78	0.78	0.75	0.76	0.74	0.76	0.81	0.7	0.76	0.03
6	0.79	0.75	0.77	0.82	0.78	0.81	0.81	0.81	0.83	0.8	0.80	0.02
7	0.82	0.8	0.77	0.84	0.77	0.83	0.87	0.8	0.8	0.83	0.81	0.03
8	0.83	0.75	0.84	0.82	0.84	0.78	0.82	0.88	0.83	0.81	0.82	0.03
9	0.85	0.84	0.83	0.77	0.87	0.87	0.88	0.88	0.85	0.83	0.85	0.03
10	0.87	0.85	0.87	0.92	0.85	0.88	0.85	0.82	0.84	0.83	0.86	0.03
11	0.84	0.87	0.83	0.88	0.83	0.89	0.88	0.9	0.88	0.87	0.87	0.02
12	0.86	0.87	0.85	0.89	0.88	0.87	0.9	0.9	0.88	0.89	0.88	0.02
13	0.82	0.89	0.88	0.9	0.88	0.89	0.89	0.88	0.89	0.9	0.88	0.02
14	0.85	0.89	0.88	0.89	0.89	0.88	0.91	0.9	0.88	0.89	0.89	0.01
15	0.85	0.88	0.85	0.87	0.87	0.89	0.9	0.9	0.89	0.88	0.88	0.02

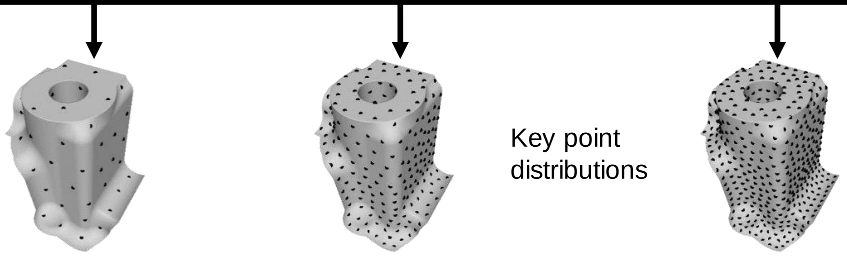


Figure 10: Feature classification accuracies with the change of key point sampling density and visual word length, and examples of key point distributions on a feature surface at different sampling densities.

In future studies, we will expand the approach to free-form feature extraction from a CAD model, which can be regarded as a part-in-whole retrieval problem (e.g., see, [11], [20], [30], and [39]). We will also improve the efficiency of the classification by introducing the parallel processing strategies and/or by a faster version of the PFH (FPFH) [29] and will evaluate our classification performances more objectively based on a comparison of our hybrid shape descriptors (PFH+TH) with other local and global shaped descriptors. Further, we will introduce a more efficient and flexible key point sampling strategy on the mesh, such as Poisson-Disk sampling (see, [8] and [21]). Moreover, we hope to develop a feature-based FE mesh generation framework from the feature extraction and classification results that conform to company-specific FE meshing rules.

ACKNOWLEDGMENTS

The authors greatly appreciate the support of the FEA professionals working at AIS Hokkaido Inc. for creating our training samples of free-form features on CAD models.

Ippei Takaishi, https://orcid.org/0000-0002-5356-5414
Satoshi Kanai, https://orcid.org/0000-0003-3570-1782
Hiroaki Date, https://orcid.org/0000-0002-6189-2044
Hideyoshi Takashima, https://orcid.org/0000-0002-7158-6059

REFERENCES

[1] Alhamzi, K.; Elmogy, M.; Barakat, S.: 3D object recognition based on local and global features using Point Cloud Library, International Journal of Advancements in Computing Technology, 7(3), 2015, 43 -54. ISSN 2233-9337.

[2] Ankerst, M.; Kastenmüller, G.; Kriegel, HP.; Seidl T.: 3D shape histograms for similarity search and classification in spatial databases, Lecture Notes in Computer Science, 1651, 1999, 207-226. https://doi.org/10.1007/3-540-48482-5_14

[3] Arthur, D.; Vassilvitskii, S.: k-means++: The advantages of careful seeding, SODA '07 Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, 2007, 1027–1035.

[4] Aubry, M.; Schlickewei, U.; D. Cremers, D.: The wave kernel signature: a quantum mechanical approach to shape analysis, Proceedings of 2011 IEEE International Conference on Computer Vision Workshops, 2011, 1626-1633. https://doi.org/10.1109/ICCVW.2011.6130444

[5] Bosch, A.; Muñoz, X.; Martí, R.: Which is the best way to organize/classify images by content?, Image and Vision Computing, 25(6), 2007, 778–791. https://doi.org/10.1016/j.imavis.2006.07.015.

[6] Boussuge, F.; Léon, J.C.; Hahmann, S.; Fine, L.: Idealized models for FEA derived from generative modeling processes based on extrusion primitives, Engineering with Computers, 31(3), 2015, 513–527. https://doi.org/10.1007/s00366-014-0382-x

[7] Cai, N.; Bendjebla, S.; Lavernhe, S.; Mehdi-Souzani, C.; Anwer, N.: Freeform machining feature recognition with manufacturability analysis, Procedia CIRP, 72, 2018, 1475-1480. https://doi.org/10.1016/j.procir.2018.03.261

[8] Corsini, M.; Cignoni, P.; Scopigno, R.: Efficient and flexible sampling with blue noise properties of triangular meshes, IEEE Transactions on Visualization and Computer Graphics, 18(6), 2012, 914-924. https://doi.org/10.1109/TVCG.2012.34

[9] Gao, S.; Shah, J.J.: Automatic recognition of interacting machining features based on minimal condition subgraph, Computer-Aided Design, 30(9), 1998, 727-739. https://doi.org/10.1016/S0010-4485(98)00033-5

[10] Gupta, R.K.; Gurumoorthy, B.: Automatic extraction of free-form surface features (FFSFs), Computer-Aided Design, 44(2), 2012, 99-112. https://doi.org/10.1016/j.cad.2011.09.012

[11] Ferreira, A.; Marini, S.; Attene, M.; Fonseca, M.J.; Spagnuolo, M.; Jorge, J.A.; Falcidieno, B.: Thesaurus-based 3D object retrieval with part-in-whole matching, International Journal of Computer Vision, 89, 2010, 327-347. https://doi.org/10.1007/s11263-009-0257-6

[12] Han, J.; 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. https://doi.org/10.1109/70.897789

[13] Harik, R.; Shi, Y.; Baek, S.: Shape Terra: mechanical feature recognition based on a persistent heat signature, Computer-Aided Design and Applications, 14(2), 2017, 206-218, https://doi.org/10.1080/16864360.2016.1223433

[14] HyperMesh, https://altairhyperworks.com/product/HyperMesh, Altair.

[15] Ip, C.Y.; Lapadat, D.; Sieger, L.; Regli, W.C.: Using shape distributions to compare solid models, Proceedings of the 7th ACM Symposium on Solid Modeling and Applications (SMA '02), 2002, 273-280. https://doi.org/10.1145/566282.566322

[16] Iyer, N.; Jayanti, S.; Lou, K.; Kalyanaraman, Y.; Ramani, K.: Three-dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design, 37(5), 2005, 509-530. https://doi.org/10.1016/j.cad.2004.07.002

[17] 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/10.1080/16864360.2017.1375678

[18] Lankalapalli, K.; Chatterjee, S.; Chang, T.C.: Feature recognition using ART2: A self-organizing neural network, Journal of Intelligent Manufacturing, 8(3), 1997, 203-214. https://doi.org/10.1023/A:1018521207901

[19] Laug, P.; Borouchaki, H.: Discrete CAD model for visualization and meshing, Procedia Engineering, 163, 2016, 149-161. https://doi.org/10.1016/j.proeng.2016.11.039

[20] Lavoué, G.: Bag of words and local spectral descriptor for 3D partial shape retrieval, Proceedings of the 4th Eurographics Workshop on 3D Object Retrieval, 2011, 41–48. http://dx.doi.org/10.2312/3DOR/3DOR11/041-048

[21] Li, H.; Lo, K.; Leung, M.; Fu, C.: Dual Poisson-disk tiling: an efficient method for distributing features on arbitrary surfaces, IEEE Transactions on Visualization and Computer Graphics, 14(5), 2008, 982-998. https://doi.org/10.1109/TVCG.2008.53

[22] Litman, R.; Bronstein, A.M.: Learning spectral descriptors for deformable shape correspondence, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(1), 2014, 171-180. https://doi.org/10.1109/TPAMI.2013.148

[23] Liu, Y.; Pu, J.; Zha, H.; Liu, W.; Uehara, Y.: Thickness histogram and statistical harmonic representation for 3D model retrieval, Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. https://doi.org/10.1109/TDPVT.2004.1335410

[24] 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

[25] Nezis, K.; Vosniakos, G.: Recognizing $2\frac{1}{2}$D shape features using a neural network and heuristics, Computer-Aided Design, 29(7), 1997, 523-539. https://doi.org/10.1016/S0010-4485(97)00003-1

[26] Onwubolu, G.C.: Manufacturing features recognition using backpropagation neural networks, Journal of Intelligent Manufacturing, 10(3), 1999, 289-299. https://doi.org/10.1023/A:1008904109029

[27] Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D.: Shape distributions, ACM Trans. Graph, 21(4), 2002, 807-832. https://doi.org/10.1145/571647.571648

[28] Prabhakar, S.; Henderson, M.R.: Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models, Computer-Aided Design, 24(7), 1992, 381-393. https://doi.org/10.1016/0010-4485(92)90064-H

[29] Rusu, R.B.: Semantic 3D object maps for everyday robot manipulation, Springer, 2013, 33-60.

[30] Savelonas, M.A.; Pratikakis, I.; Sfikas, K.: An overview of partial 3D object retrieval methodologies, Multimedia Tools and Applications, 74(24), 2015, 11783-11808. https://doi.org/10.1007/s11042-014-2267-9.

[31] Savva, M.; Yu, F.; Su, H.; Aono, M.; Chen, B.; Cohen-Or, D.; Deng, W.; Su, H.; Bai, S.; Bai, X.; Fish, N.; Han, J.; Kalogerakis, E.; Learned-Miller, E.G.; Li, Y.; Liao, M.; Maji, S.; Tatsuma, A.; Wang, Y.; Zhang, N.; Zhou, Z.: Large-scale 3D shape retrieval from ShapeNet core55. Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval (3DOR '16), 2016, 89-98. https://doi.org/10.2312/3dor.20161092

[32] Sheng, X.; Hirsch, B.E.: Triangulation of trimmed surfaces in parametric space, Computer-Aided Design, 24(8), 1992, 437-444. https://doi.org/10.1016/0010-4485(92)90011-X

[33] Shi, Y.; Zhang, Y.; Baek, S.; De Backer, W.; Harik, R.: Manufacturability analysis for additive manufacturing using a novel feature recognition technique, Computer-Aided Design and Applications, 15(6), 2018, 941-952. https://doi.org/10.1080/16864360.2018.1462574

[34] Sun, J.; Ovsjanikov, M.; Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion, Computer Graphics Forum, 28, 2009, 1383-1392. https://doi.org/10.1111/j.1467-8659.2009.01515.x

[35] Sun, Z.; He, Y.; Gritsenko, A.; Lendasse, A.; Baek, S.: Deep spectral descriptors: learning the point-wise correspondence metric via Siamese deep neural networks, 2017, arXiv:1710.06368.

[36] Sunil, V.B.; Pande, S.S.: Automatic recognition of machining features using artificial neural networks, International Journal of Advanced Manufacturing Technology, 41, 2009(9-10), 932-947. https://doi.org/10.1007/s00170-008-1536-z

[37] Sunil, V.B.; Pande, S.S.: Automatic recognition of features from freeform surface CAD models, Computer-Aided Design, 40(4), 2008, 502-517. https://doi.org/10.1016/j.cad.2008.01.006

[38] Tangelder, J.W.H.; Veltkamp, R.C.: A survey of content-based 3D shape retrieval methods, Multimedia Tools Applications, 39, 2008, 441-471. https://doi.org/10.1007/s11042-007-0181-0

[39] Wang, Y.; Lu, W.F.; Fuh, J. Y.H.: Sampling strategies for 3D partial shape matching and retrieval using Bag-of-Words model, Computer-Aided Design and Applications, 11(1), 2014, 43-48. https://doi.org/10.1080/16864360.2013.834137

[40] Wohlkinger, W.; Vincze, M.: Ensemble of shape functions for 3D object classification, Proceedings of the IEEE International Conference on Robotics and Biomimetics, 2011, 2987-2992. https://doi.org/10.1109/ROBIO.2011.6181760

[41] 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