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

Authors:
Ippei Takaishi, i_takaishi@sdm.ssi.ist.hokudai.ac.jp, Hokkaido University, Japan
Satoshi Kanai, kanai@ssi.ist.hokudai.ac.jp, Hokkaido University, Japan
Hiroaki Date, hdate@ssi.ist.hokudai.ac.jp, Hokkaido University, Japan
Hideyoshi Takashima, hideyoshi_takashima@ais-hokkaido.co.jp, AIS Hokkaido Inc., Japan

Keywords:
Feature Recognition, Boss, Rib, Shape Descriptor, Bag-of-Features, FEM, Mesh Generation

DOI: 10.14733/cadconfP.2019.414-419

Introduction:
Finite element analysis (FEA) has helped modern manufacturers create efficient and reliable product developments. Finite element mesh generation (FE meshing) from 3D-CAD) models is generally the most important process in the FEA pipeline, therefore fully automated meshing that can secure analysis accuracy is strongly required to streamline the pipeline.

Many manufacturers strictly prescribe FE meshing patterns for specific classes of free-form features on CAD models shown in Fig. 1., 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 secure the analysis accuracy. Meshing rules for “boss” or “rib” features, as illustrated in Fig. 2, are often specially specified, as these play critical roles in securing strength for a part or transmitting forces between parts. As such, in Fig. 2(a), 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 a rib feature shown in Fig. 2(b), the node points must be arranged along a ridge curve on top of the rib at a maximum interval of 3.0mm. 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 where meshing rules are prescribed and where an FE mesh for the feature region can be automatically generated according to rules realizing a high-quality and reliable FEA pipeline.

To date, some feature recognition methods aimed for FE meshing have been studied [3],[4],[7]. Lai et al. [4] proposed a method that recognizes rib features from a B-rep CAD model by finding specific

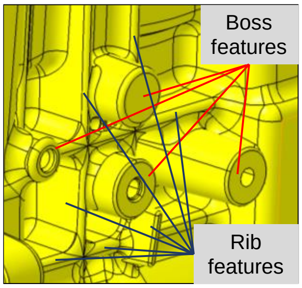


Fig. 1: Free-form features of FE meshing.

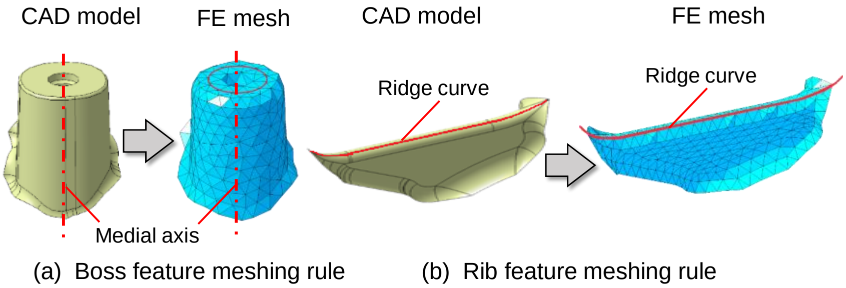


Fig. 2: Sample-specific FE meshing rules for free-form features.

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. Lu et al. [7] 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 the feature boundaries. Moreover, Boussuge et al. [3] presented a method for recognizing protrusion features on a CAD model whose shape can be partitioned into plate and shell elements.

Unfortunately, these feature recognition methods cannot be directly applied to our case for the following reasons. First, the feature geometries discussed in the previous studies [3],[4],[7] were basically 2.5-dimensional, consisted only of simple planes and cylinders, and were bounded by sharply concaved loops on a B-rep CAD model. In our study, however, based on Fig. 1, the feature (i.e., boss) that needs recognition is designed as a portion of a casted or forged part'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 mentioned above.

Second, feature classes for FE meshing are normally 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 of the previous studies were 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. This way, the algorithm is not easily expanded when a new feature class is added or a current feature class is to be modified.

Third, it was assumed in previous studies that an input B-rep CAD model is provided without any topological or geometric defect. However, it is well known that the data quality of CAD models may possibly degrade because of loss of information during the translation process and that some quality issues on the B-rep data (i.e., 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 solve the abovementioned issues, we propose an algorithm of the free-form feature classification for feature-based FE meshing, which we regard to consist of three steps: feature extraction from the

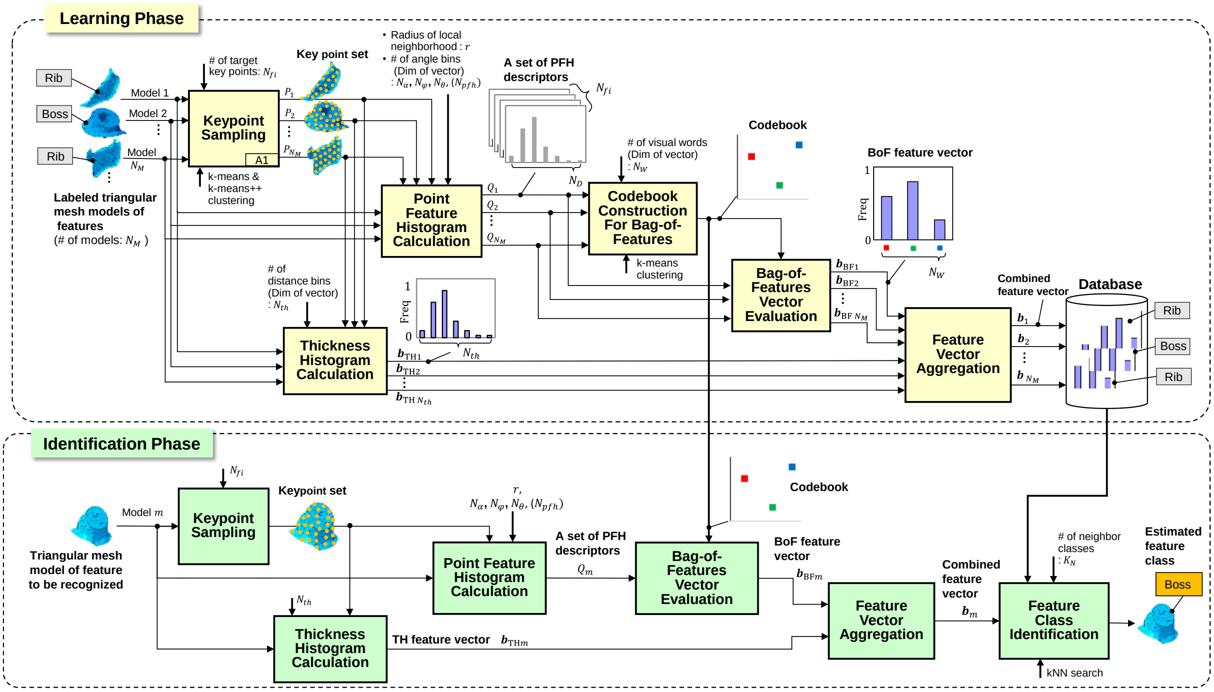


Fig. 3: The proposed free-form feature classification process.

CAD model, feature classification, and feature-compliant mesh generation. Our study focuses on the feature classification step. In principle, our algorithm accepts a triangular mesh of a free-form feature easily converted from a B-rep CAD model. Moreover, it identifies a feature class label, 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 approach are summarized as follows:

- The local and global shape descriptors allow us to encode both the local and the global features' geometry as a single multidimensional vector even when a feature has complex free-form shapes bounded by smooth filet surfaces. Moreover, the BoF technique [5] facilitates the application of the shape descriptor representation to the machine learning scheme, thereby solving the first problem.
- The machine learning technique 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 at the vertices on a triangular mesh, thus avoiding unstable feature extraction and classification processes caused by product data quality issues, which solves the third problem.

Feature Classification Method for FE Meshing:

Overview

The proposed feature classification process accepts a triangular mesh model of the free-form feature as an input, which can be easily created by triangulating a set of faces of a free-form feature on a B-rep CAD model. During classification, we identify one of the feature class labels that has been trained by supervised learning and in which a unique FE meshing rule is prespecified. Currently, three feature classes (i.e., "rib," "boss," and "others") can be discriminated where company-specific FE meshing rules are often defined in many manufacturers; nevertheless, we can easily extend the feature classes to be discriminated only by adding class labels for the training samples.

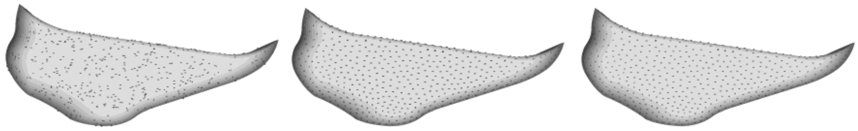
Fig. 3. provides an overview of the proposed feature classification process, consisting of learning and identification phases. In the learning phase, a large collection of labeled triangular mesh models of manually labeled free-form features class is provided as an input. Next, for each triangular mesh, two shape descriptors are computed at key points uniformly sampled on the mesh, namely, Point Feature Histogram (PFH) [8] as a local shape descriptor and Thickness Histogram (TH) [6] as a global volumetric descriptor. Afterwards, a BoF feature vector is evaluated from a set of the PFHs for every mesh, 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 and are 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 on an input free-form feature. Afterwards, distances between the combined feature vector of the input feature and the ones stored in the database are evaluated, and the class of the input feature is determined. Details of the classification algorithm are described in the subsequent sections.

Local Shape Descriptor Using 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 [8] 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.

As described earlier in [5], better classification results are achieved by sampling the key points on a mesh as uniformly as possible. We adopt the k-means clustering as the sampling method of the key



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

Fig. 4: Key point sampling strategy.

points because the method can easily partition the s set of mesh vertices into the specified number of uniformly distributed clusters. However, if we randomly select N_{f_i} vertices on the mesh (Fig. 4(a).) 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 [1] for the mesh vertices to obtain more uniformly distributed N_{f_i} initial cluster centers (Fig. 4(b).) than those randomly selected, and then apply the k-means clustering to the cluster centers. Finally, we take 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 $P_i = \{\boldsymbol{p}_i^j\}$ (Fig. 4(c).).

Next, at a key point $\boldsymbol{p}_i^j$, PFH [8] is calculated as a local shape descriptor. PFH encodes local geometric properties by generalizing the mean curvature around $\boldsymbol{p}_i^j$ as a N_{pfh}-dimensional vector $\boldsymbol{q}_i^j$. In this study, on the basis of a preliminary experiment, 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.

Feature Vector Evaluation Using BoF

Bag-of-Features (BoF) is a machine learning scheme that has been extensively used in image classification [2]. 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 then quantized to construct a limited number of *codes*. Afterwards, each *feature* of the image is assigned to its nearest code, and the image is represented as a histogram of the codes. From the histogram, the image can be categorized under the closest code. BoF enables a compact representation of the features for the classification and rapidity of search.

We employ BoF to the 3D free-form feature classification represented by a triangular mesh model. First, we perform 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 $\boldsymbol{c}_k$ $(\in R^{375}, k \in [1, N_w])$. 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 multidimensional BoF feature vector $\boldsymbol{b}_{\mathrm{BF}i} = \left[b_{\mathrm{BF}i}^1, b_{\mathrm{BF}i}^2, \dots, b_{\mathrm{BF}i}^{N_w}\right]$ $(b_i^l \in [0,1], i \in I)$ representation that encodes the local surface geometry of the free-form feature represented by the mesh i.

Fig. 5. provides examples describing the assignment of different visual words to PFH descriptors at key points in the case of $N_w = 10$. As shown, different words are loosely assigned to different local regions in a feature exhibiting similar geometries (planar or cylindrical regions).

Global Shape Descriptor Using TH and Combined Feature Vector

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. This way, the BoF feature vector does not necessarily represent global volumetric properties of free-form features.

In order to make up for the lack of volumetric properties of a feature, we introduce a Thickness-Histogram (TH) [6]. The TH descriptor encodes the statistical thickness distribution of an object as a

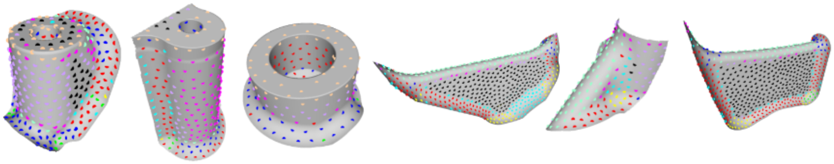


Fig. 5: Visual word assignment to PFH descriptors on features (same-colored points indicate the same visual word assigned).

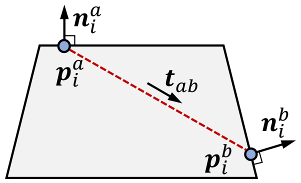


Fig. 6: Thickness histogram.

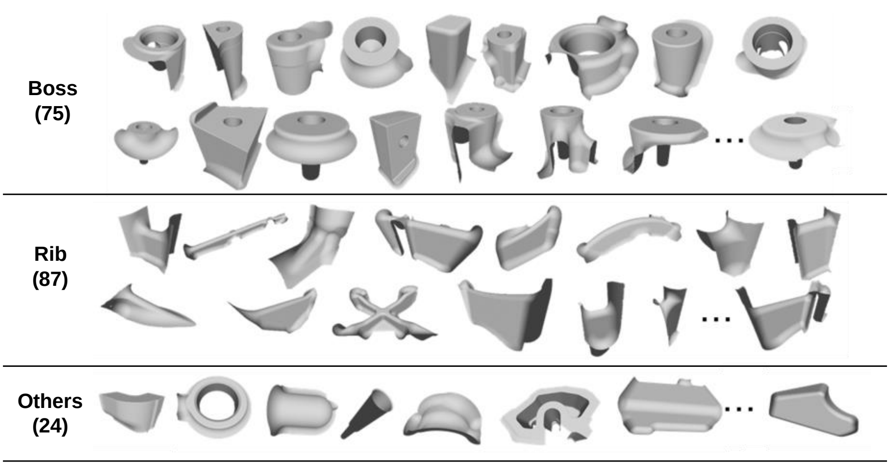


Fig. 7: Labeled free-form feature examples.

Parameter	Value
N_{fi}	N_v/100
N_v	# of vertices on the mesh
r	2.5 [mm]
N_{pfh}	375
b_α	5
b_φ	5
b_θ	15
N_{th}	50
N_w	10
K_N	3

Tab.1: Parameter settings.

histogram. When constructing the TH (Fig. 6.), we select a pair of different key points $\boldsymbol{p}_i^a$ and $\boldsymbol{p}_i^b$ on a triangular mesh i and evaluate a weight W_{ab} using the following Eqn. (1):

$$W_{ab} = (\boldsymbol{t}_{ab} \cdot \boldsymbol{n}_i^a)(\boldsymbol{t}_{ab} \cdot \boldsymbol{n}_i^b)/d_{ab}^2, \quad (1)$$

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 N_{th} bins, each of which corresponds to a quantized interval for d_{ab}. We perform this vote for all pairs of key points on a mesh i and obtain a histogram of votes for d_{ab}. By normalizing the cumulative frequency of the histogram to 1, we obtain a multidimensional 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 a mesh i.

Finally, we combine 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 a mesh i. A set of combined feature vectors for all labeled triangular meshes $\{\boldsymbol{b}_i\}_{i \in I}$ is used for learning and class identification.

Feature Class Identification

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.

Classification Results:

As there was no open data set of 3D free-form features, we personally 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 parts via a CAD system, CATIA-V5. Fig. 7. lists the 75 bosses, 87 ribs, and 23 other classes of features we collected. Afterwards, we constructed the triangular meshes of the samples using an FEM preprocessor (HyperMesh) and assigned a true feature class label for each mesh under the decision of the professional. These labeled triangular meshes were used for the learning.

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. A 10-fold

PFH		Predicated 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		Predicated 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		Predicated 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

Tab. 2: Classification performances using BoF, TH, and combined feature vectors.

cross-validation was used for the performance evaluation. Tab. 1. summarizes the other parameter settings for the descriptor calculation.

Tab. 2. summarizes the classification performances in the form of the confusion matrices, recalls, and accuracies of the classification. Using only either the BoF or the TH feature vector, we achieved 89% or 84% accuracy in the classification, respectively. Accuracy went up to 92% with the combined feature vector, whereas 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 superior classification performance of geometries of the features.

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. 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 proved superior classification performance accuracy (92%) and recalls (95%–98%) than for a single descriptor.

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[9]. Furthermore, 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.

References:

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

[2] 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.

[3] 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.

[4] 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.

[5] Lavoué, G.: Bag of words and local spectral descriptor for 3D partial shape retrieval, Proceedings of the 4th Eurographics Conference on 3D Object Retrieval, 2011, 41–48.

[6] 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.

[7] 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.

[8] Rusu, R. B.: Semantic 3D Object Maps for Everyday Robot Manipulation, Springer, 2013, 33–60.

[9] 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.