3D Shape Retrieval Focused on Holes and Surface Roughness

Masaki Aono, Hitoshi Koyanagi, and Atsushi Tatsuma
*Department of Computer Science and Engineering, Toyohashi University of Technology, Hibarigaoka, Toyohashi, Aichi, Japan
E-mail: aono@tut.jp Tel: +81-532-44-6764

***Abstract*—Although quite a few 3D shape descriptors have been proposed for more than a decade, 3D shape retrieval has remained a still challenging research. No single 3D shape descriptor has been known to outperform all the different types of 3D shape geometries. In this paper, we propose a new 3D shape descriptor that focuses on 3D mechanical parts having holes and surface roughness by using Fourier spectra computed from multiple projections of distinct images.**

Our proposed method makes it possible to explore potential real applications of 3D shape retrieval to manufacturing industries where the cost reduction of creating a new 3D shape from scratch is greatly appreciated.

Our method explicitly attempts to extract holes and surface roughness, as well as contours, lines, and circular edges as our proposed features from multiple projections of a given 3D shape model, and use them to retrieve 3D shape objects. We demonstrate the effectiveness of our method by using several 3D shape benchmarks, one of which is composed of many mechanical parts, and compare our proposed method with several previously known methods. The results are very encouraging and promising.

I. Introduction

3D shape retrieval has become popular for more than a decade, ever since Osada et al [1] published their work on shape distributions. More and more 3D data are available on the Internet as well as inside allied companies for manufacturing, architectural planning, medical treatment, education, entertainment, and many other miscellaneous purposes. Yet no single 3D shape descriptor to date has excelled proposed descriptors, given arbitrary collections of 3D shape models.

In this paper, we propose a 3D shape descriptor suited to search 3D mechanical parts, having *holes* and *surface roughness*. *Holes* are literally the holes often observed in small mechanical parts. *Surface roughness* represents the convexities and concavities of the surface that might account for sudden intensity variations, when rendering, located inside the contour of the projected shapes.

In what follows, we first describe related work in the next section. We then introduce our new features focused on holes and surface roughness. Specifically, we describe five base feature images, including Contour, Hole, Surface Roughness, Line, and Circle images, and how we generate them, followed by when and how Fourier transform is applied to compute our composite feature vector. Subsequently we introduce the dissimilarity computation under the assumption of our proposed features. Finally, we demonstrate our method through experiments in Experiment and Evaluation section, and summarize our results in the Conclusion.

II. Related Work

Research on 3D shape retrieval has gained much popularity these days, ever since Osada et al [1] published one of the pioneering studies on 3D shape retrieval. Among their proposed models, D2 (a shape descriptor based on the histogram of distances between random two points on the surface) has served as a "Baseline" method for 3D shape retrieval. Kazhdan et al [2] propose a method in which a 3D shape is first transformed into a voxel representation, and then the voxel data is transformed into a power spectrum by means of spherical harmonics called a Spherical Harmonic Descriptor (SHD). The above two methods are just examples of 3D shape descriptors that have rotational invariance.

To achieve similar effects, there have been approaches to define features by rendering a large collection of projected images of each 3D shape viewed from almost all the imaginable directions [3][4]. Inspired by these early efforts, 3D shape descriptors based on a composite of several features such as DESIRE [5] and MFSD (Multi-Fourier Spectra Descriptor) [6] have been proposed. Recently, local 3D shape features such as *Bag of Visual Words (BoVW)* (a.k.a. *Bag of Features (BoF)*) have been proposed, along with supervised learning [7][8].

However, to our knowledge, most of the previous 3D shape descriptors failed to behave very well for 3D mechanical shapes such as Engineering Shape Benchmark (ESB) [9]. One reason for being incapable of capturing the good shape features of this type of mechanical part lies in the fact that minor characteristics (such as *holes* and *surface roughness*) in the original shape might have disappeared during the computation of features.

With respect to *holes*, there has been no previous research, to our knowledge, which explicitly employs holes as one of the shape features. Meanwhile, there has been research on detecting holes from 3D models. An example of such research has been done by Wang et al [10]. They constructed a connected graph of vertices, applied clustering of vertices to form sub-graphs, and then used the relationship between the planes enclosing holes and sub-graphs to detect holes.

A 3D shape descriptor using *Hough transform* was first reported by Zaharia et al [11] in 2001. They defined a

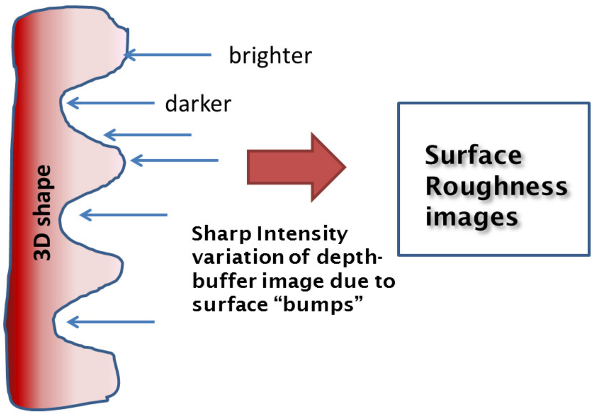

Fig. 1. Illustration of surface roughness

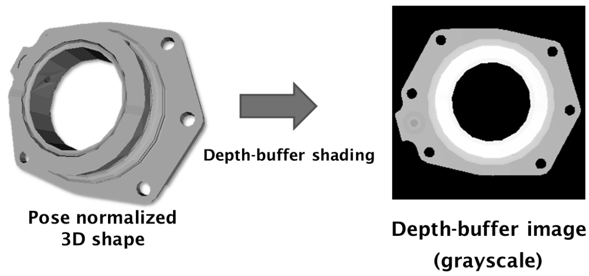

Fig. 2. Depth-buffer image generation from multiple viewpoints after pose normalization of a given 3D shape

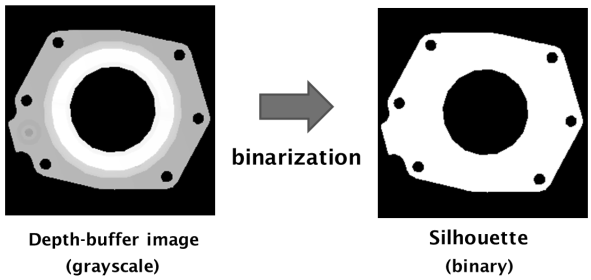

Fig. 3. Silhouette image generation by binarization from Depth-buffer image

shape feature intrinsically invariant with respect to topological decriptions and geometric transforms. They define 48 dimensional feature vectors based on Hough transform, and demonstrate the validity of their model using 3D Cafe data set [12] and other miscellaneous data.

In this paper, we propose a new composite of features, explicitly incorporating *holes* and *surface roughness*, as well as features extracted from *Hough transform*.

III. New Feature Focused on Holes and Surface Roughness

In this section, we introduce our proposed features, and how we create them for an arbitrary 3D shape. The basic idea is to extract the vector-based features frequently observed in 3D CAD shapes of mechanical parts. Primary interests include not only straight lines as vectors, but curved edges rendered for *holes* and *surface roughness*. *Surface roughness* can be represented by a set of pixels, usually a set of line segments, where sharp intensity change occurs inside the contour of the object as convexities and concavities, once the view projection is determined. The idea of incorporating the surface roughness into 3D shape features is to attempt to extract the region of surface "bumpiness" as illustrated in Fig. 1.

In summary, we extract five edge-based images, namely, Contour, Hole, Surface Roughness, Line, and Circle images, followed by transforming them into polar coordinates, applying *peripheral intensity enhancement* [6], in order to emphasize the shape perimeter and converting them to Fourier spectra as our composite features.

In the following, we elaborate the process of obtaining our proposed features.

A. Computing Five Base Feature Images

First of all, the ideal 3D features must capture inherent 3D shapes, which are invariant to position, size, and orientation. Making 3D shapes invariant to position, size, and orientation requires the process called *pose normalization*. For this purpose, we first employ our previously published pose normalization methods called *Point SVD* and *Normal SVD* [13]. After pose normalization is carried out, we compute Depth-buffer images from multiple viewpoints as shown in Fig. 2. Then, Depth-buffer images are converted to Silhouette images by binarization as shown in Fig. 3. The binary Silhouette images are converted to ternary *Flood Fill* images using Flood Fill algorithm (a.k.a.Seed Fill algorithm) [14] by changing the background color from black to gray, as shown in Fig. 4.

Binary Contour images, as the first of our five base feature images, are computed from ternary Flood Fill images by way of Hole FF (Flood Fill) images. Specifically Hole FF images are generated by removing holes, if any, from ternary Flood Fill images. On the other hand, binary Contour images are generated by applying a *Canny edge detector* [15], known in edge detection in image recognition for computer vision. This process for obtaining binary Contour images is illustrated in Fig. 5. Binary Hole images as the second of our five base feature images are computed from ternary Flood Fill images by way of binary Background FF images, where a binary Background FF image is generated by converting the background color into white from gray. Binary Hole images

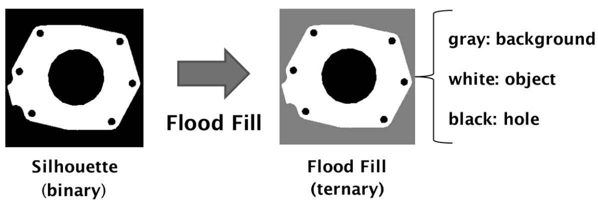

Fig. 4. Ternary Flood-Fill image computation by applying Flood Fill algorithm

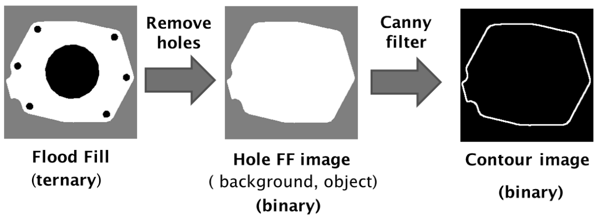

Fig. 5. Generation of a binary Contour image from ternary Flood-Fill (FF) image by removing holes (if any) in binary Hole FF images, followed by applying Canny edge detector

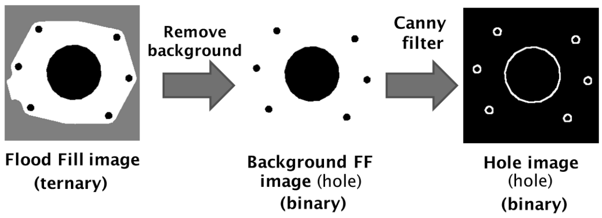

Fig. 6. Generation of a binary Hole image from ternary Flood-Fill (FF) image via removing background, followed by applying canny edge detector

are generated by applying the *Canny edge detector* to binary Background FF image, similar to the generation of binary Contour images. Incidentally, the generation of these two images (Contour and Hole images) can be done independently. It should be noted that binary Hole images might end up just black images if there are no holes in the projection of the 3D shape object concerned, but they are harmless in our proposed base features.

Binary Surface Roughness images, as the third of our five base feature images, are computed in two steps. In the first step, we extract binary Edge images from Depth-buffer images by applying the *Canny edge detector*. In the second step, Edge images are subtracted by Contour and Hole images, ending up binary Surface Roughness images. This process is illustrated in Figs. 7 and 8. The reason for showing two examples here is because the Surface Roughness image in the first example results in a circular shape, which might give the false impression that it might be the same as a Circle image to be discussed later. Surface Roughness images are introduced in order to represent boundaries where there are sharp intensity variations in rendering with depth-buffering.

The remaining two base features are Line and Circle images by applying the*Hough transform* [15] to Edge images, which are the intermediate products when computing Surface Roughness images. This process is illustrated in Fig. 9. It should be also noted that Circle images might be just black unless Edge images have circular shapes.

B. Fourier Spectra computed from Five Base Feature Images

For the above five binary images composed of different types of Edge-based images, we apply a *dilation* operator [15], one of the well-known morphological operators. Then, we convert dilated images into images in polar coordinates with

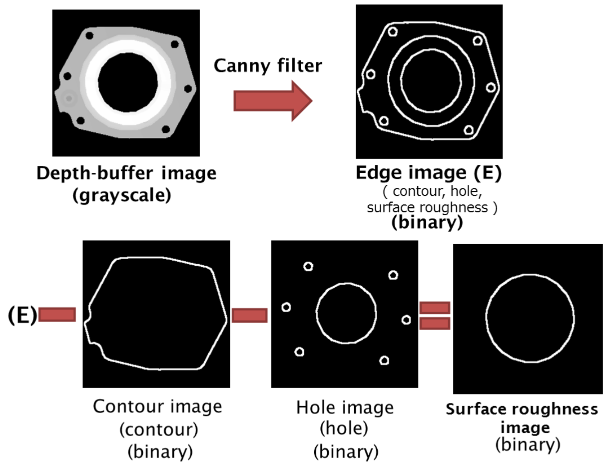

Fig. 7. Generation of a binary Surface Roughness image from Depth-buffer images in two steps via Edge images subtracted by Contour and Hole images

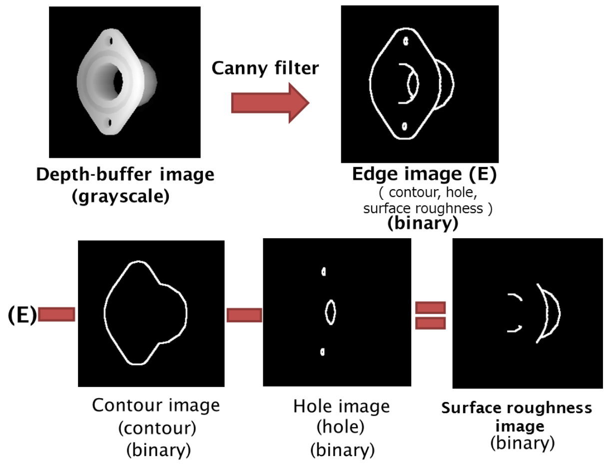

Fig. 8. A different example of derivation of a Surface Roughness image

parameters r and θ, to be robust against rotation. This is because in polar coordinate, a rotational difference is absorbed in a translational difference, which is later absorbed by a Fourier transform. After images are represented by polar coordinates, we separately apply Fourier transform to convert them into spectra, and construct our proposed features by filtering out high frequency components. The final stage is illustrated in Fig. 10.

IV. Feature Alignment between a given Query and 3D Shape Models

The 3D shape retrieval system with our proposed features is depicted in Fig. 11. As pre-processing we perform feature extraction for every 3D shape data in the database. In other words, all the 3D shape data are converted to Fourier spectra computed from polar coordinates of Contour, Hole, Surface Roughness, Line, and Circle images.

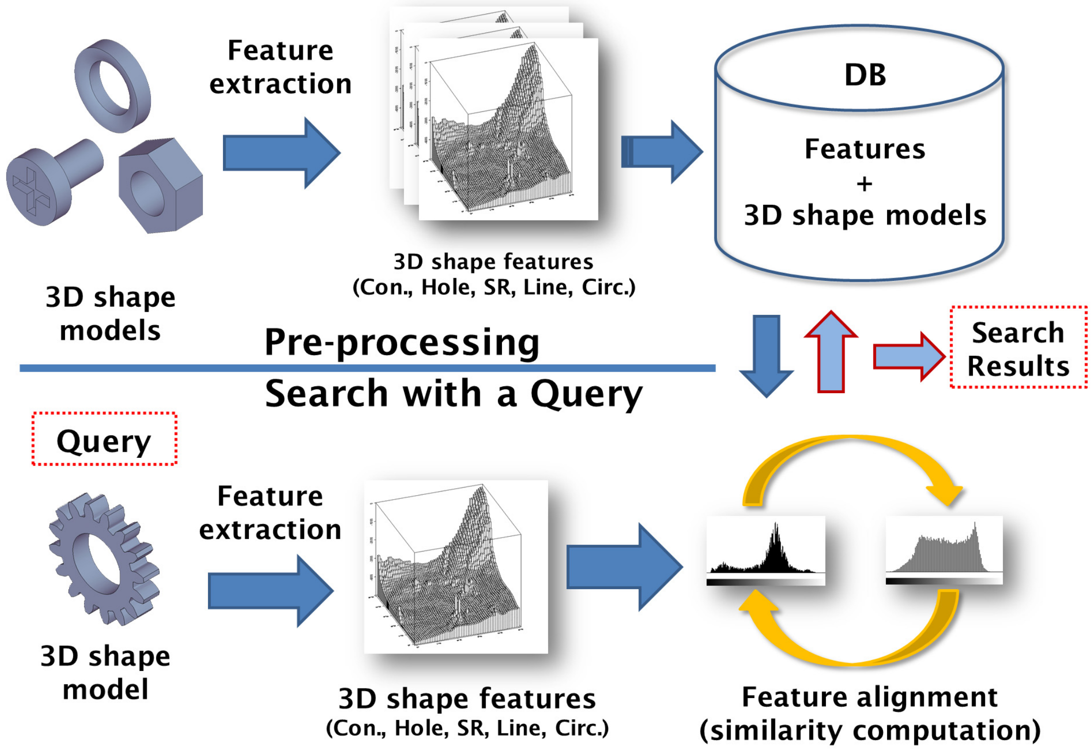

Fig. 11. System Flow: 3D shape retrieval using new shape features based on different Fourier spectra from five different sets of images (Contour images (Con.), Hole images, Surface roughness images (SR), Line images, and Circle images (Circ.))

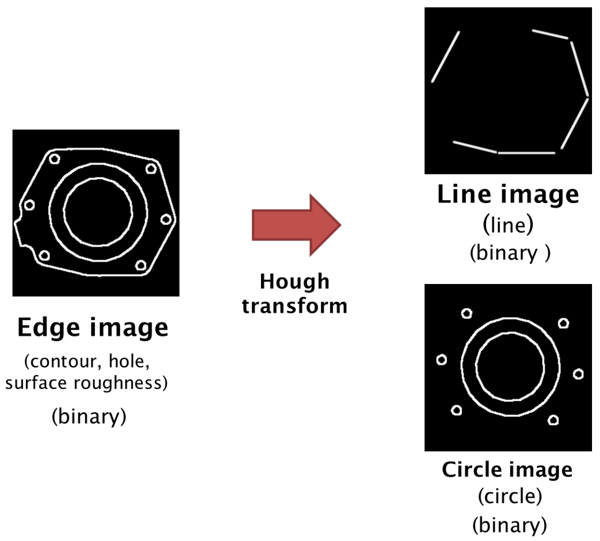

Fig. 9. Generation of Line and Circle images from Edge images

A. Dissimilarity Computation

Given an arbitrary 3D query shape model, we compute the Fourier spectra, and start feature alignment as dissimilarity computation.

Dissimilarity computation is carried out such that we compute Manhattan distance between features extracted from a query and features of 3D shape data stored in our database. We have tested a collection of a different number of views and a different combination of polar coordinate discretization parameters. Table I summarizes the result.

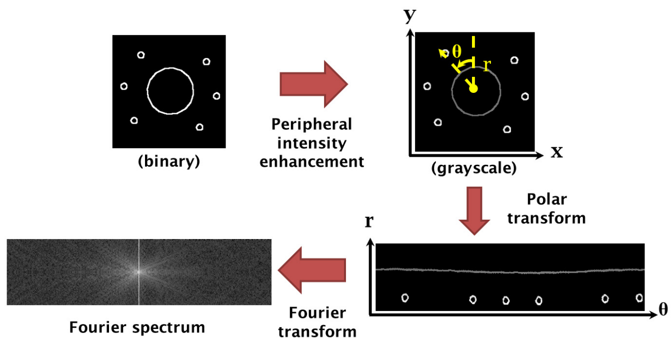

Fig. 10. Generation of Fourier spectra after converting each image to polar coordinate

As is easily computed from Table I, the total number of dimensions of our proposed features is 1,536. Assume that we have n number of dimensions of a 3D feature vector $f_1, f_2, ..., f_n$, and assume that a given 3D query is converted to a 3D feature vector $q_1, q_2, ..., q_n$, the dissimilarity between 3D shape model M and a 3D query shape model Q is denoted

TABLE I
NUMBER OF VIEWS, POLAR COORDINATE PARAMETERS, AND DIMENSIONS FOR EACH FEATURE

Feature	$\boldsymbol{\theta}$	$\boldsymbol{r}$	views	dimensions
Contour	8	8	3	192
Hole	8	8	3	192
Surface Roughness	4	4	24	384
Line	8	2	24	384
Circle	8	2	24	384

by the following equation:

$$dissimilarity(M_k, Q_k) = \sum_{i=1}^{n} |f_{k,i} - q_{k,i}|,$$

where k is either *Point SVD* or *Normal SVD*. Thus, the final dissimilarity is given by the following equation:

$$\begin{aligned} &dissimilarity(M, Q) = \min(\\ &\quad dissimilarity(M_{\text{PointSVD}}, Q_{\text{PointSVD}}), \\ &\quad dissimilarity(M_{\text{NormalSVD}}, Q_{\text{NormalSVD}})) \end{aligned}$$

V. EXPERIMENTS AND EVALUATIONS

In this section, we describe comparative experiments of our proposed method with some of the selected previous methods for 3D shape retrieval. To demonstrate the effectiveness of our proposed methods, we employ Princeton Shape Benchmark (PSB) [16], Architectural Shape Benchmark (ASB) [17], and Engineering Shape Benchmark (ESB) [9] as 3D shape data. The previous methods to compare are D2 [1], SHD [2], LFD [3], DESIRE [5], and MFSD [6], partly because we have access to these programs and partly because they are all categorized as unsupervised learning methods without relevance feedbacks, which do not require a training stage.

A. Evaluation Measures

The evaluation measures we selected include Recall, Precision, First Tier (1-Tier), Second Tier (2-Tier), and P@1 (Nearest Neighbor (NN)) [16], [18]. Let $rel(x)$ be the number of objects that are *relevant* among the top x rankings, let K be the number of closest matches returned, and let C be the number of objects in the category belonging to a query. Then, the evaluation measures are given by the following formula:

$$\text{Recall} = \frac{rel(K)}{C}$$

$$\text{Precision} = \frac{rel(K)}{K}$$

$$\text{First Tier} = \frac{rel(C-1)}{C-1}$$

$$\text{Second Tier} = \frac{rel(2(C-1))}{C-1}$$

$$\text{P@1 (Nearest Neighbor)} = rel(1)$$

$$DCG(i) = \begin{cases} G(1) & i = 1 \\ DCG(i-1) + \frac{G(i)}{\log_2(i)} & \text{otherwise} \end{cases}$$

Discounted Cumulative Gain (DCG) [19] represents how well a retrieval system works at the top-*k* ranked lists. Generally speaking, if relevant results appear at the first top-*k* search results, the DCG is larger. In the above equation, $G(i)$ stands for a gain vector, where the element of the vector represents the degree of relevance.

There are two ways of computing the degree of relevance, depending on how we compute the average of accuracy, given ground truth data with different categories. *Micro-averaged* values [20] are calculated by constructing a global contingency table and then calculating evaluation measures using these sums. We avoided macro-average, since macro-averaged scores are calculated by first computing evaluation measures for each category and then using their average. Some of the categories in PSB have only a few shape models, which ends up with the macro-average having a large variance across different categories.

B. Result with Princeton Shape Benchmark

The Princeton Shape Benchmark (PSB) is designed for general 3D shape objects, consisting of 907 training data with 90 classes, and another 907 testing data with 92 classes [16]. This benchmark is not particularly suited to our proposed method, because no explicit mechanical parts having holes are included. Nonetheless, our proposed method exhibits reasonably good search performance.

Fig. 12 shows averaged recall-precision graphs including the proposed method and several previous methods by using PSB. Our proposed method is almost equal to MFSD when "Recall" is smaller, and slightly inferior to MFSD when "Recall" is larger. Table IV summarizes the average behavior of other evaluation measures, including 1-Tier, 2-Tier, P@1, and DCG. It is noted that in terms of P@1, or the nearest neighbor, our proposed method turns out to be the best.

TABLE II
COMPARISON OF OUR PROPOSED METHOD AND PREVIOUS METHODS IN TERMS OF 1-TIER, 2-TIER, P@1, AND DCG FOR PSB

Method	1-Tier	2-Tier	P@1	DCG
Proposed	44.0%	56.6%	**72.5**%	69.9%
MFSD	**45.3**%	**56.6**%	71.5%	**70.3**%
DESIRE	40.4%	52.2%	65.8%	66.3%
LFD	37.8%	49.2%	65.9%	64.4%
SHD	30.0%	42.0%	55.3%	58.4%
D2	18.7%	27.9%	35.7%	46.2%

C. Architectural Shape Benchmark (ASB)

The Architectural Shape benchmark (ASB) [17] is composed of 3D shape objects related to architectural design, including "arm chairs," "book shelves" and "double bed"

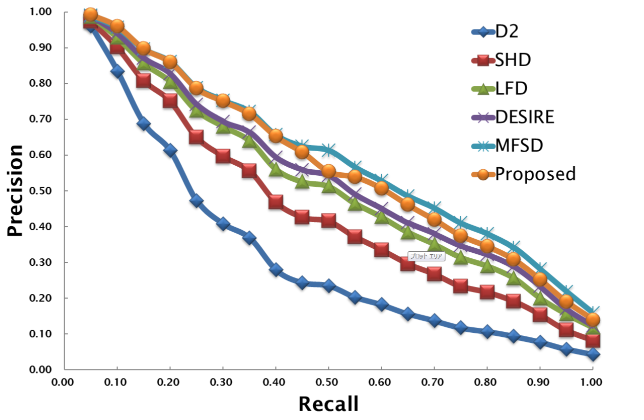

Fig. 12. Average recall-precision graph of our proposed method and previous methods for comparison using PSB

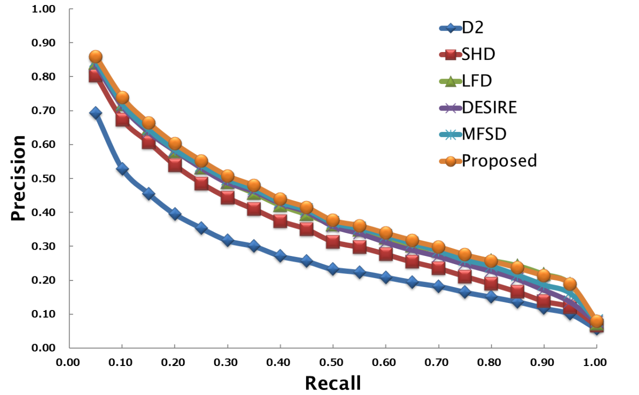

Fig. 13. Average recall-precision graph of our proposed method and previous methods for comparison using ASB

classes. ASB itself consists of two large classes; "form" class and "funtion" class. The former class categorizes 3D objects by *shapes*, while the latter class does so by *functions*. Naturally, we chose the former class for shape retrieval, where 2,257 models are included with 95 classes. Fig. 13 shows the averaged comparison result of our proposed method against several previous methods. Table III summarizes the average behavior of other evaluation measures, including 1-Tier, 2-Tier, P@1, and DCG. It is noted that our proposed method proves to be the best, except for the 2-Tier.

TABLE III
COMPARISON OF OUR PROPOSED METHOD AND PREVIOUS METHODS IN TERMS OF 1-TIER, 2-TIER, P@1, AND DCG FOR ASB

Method	1-Tier	2-Tier	P@1	DCG
Proposed	**37.8%**	47.6%	**76.7%**	**70.2%**
MFSD	35.9%	**48.3%**	74.3%	69.8%
DESIRE	34.6%	46.6%	75.5%	68.9%
LFD	35.9%	45.2%	75.6%	69.0%
SHD	31.3%	41.7%	71.4%	66.4%
D2	23.4%	34.5%	59.6%	59.9%

D. Engineering Shape Benchmark (ESB)

The Engineering Shape Benchmark (ESB) [9] is known for its unique 3D data sets, and is a proven standard benchmark for 3D CAD models. It consists of 801 models classified into 42 categories of similar parts such as "Discs," "T-shaped parts" and "Bracket-like parts." Fig. 14 shows the averaged comparison result of our proposed method against several previous methods. It is obvious our proposed method outperforms other methods over all "Recall" values. This can be a proof that our proposed method is particularly suited to 3D mechanical parts with holes and surface roughness.

In our comparative experiments in Fig. 14, MFSD [6] and our proposed method are the two top methods in Recall-Precision graph. To demonstrate the effectiveness of our

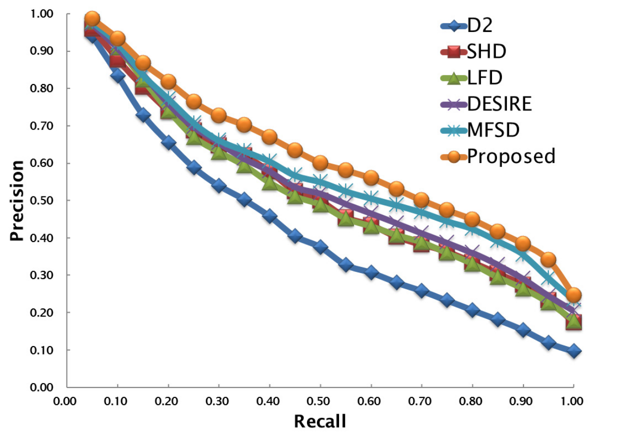

Fig. 14. Average recall-precision graph of our proposed method and previous methods for comparison using ESB

method, we take two example where *holes* and *surface roughness* have played important roles. Fig. 15 illustrates an example showing the top 10 search results with our proposed method and MFSD, given a query belonging to gear-like class objects. It should be noted that the proposed method has not omitted the gear-like jagged perimeters for all the top 10 search results, while MFSD has omitted the gear-like jagged perimeters by 70%. Fig. 16 illustrates another example, given a query belonging to slender-links class objects. In this query example, small holes are observed in the query, and the proposed method has not omitted objects with small holes for all the top 10 search results, while MFSD has omitted such objects by 50%.

Table IV summarizes the average behavior of other evaluation measures, including 1-Tier, 2-Tier, P@1, and DCG (Discounted Cumulative Gain). It turns out that in every evaluation measure, our method outperforms the previous methods. In addition, Table V summarizes the result with ESB, showing how each independent feature behaves in terms of these evaluation measures. It appears that as far as DCG is concerned, Contour, Surface Roughness, and Line images contribute to the search efficiency. Although Hole and Circle

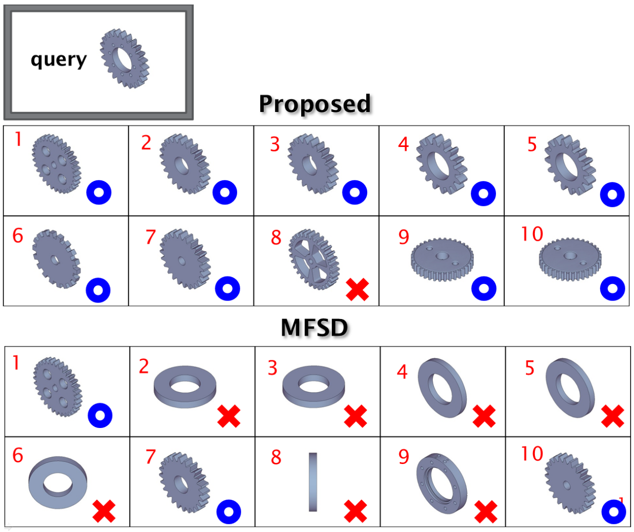

Fig. 15. Top 10 search results from MFSD and our proposed method, given the same query in the gear-like objects of ESB. Blue circles indicates the resulting object is relevant to the query, while a red cross indicates otherwise.

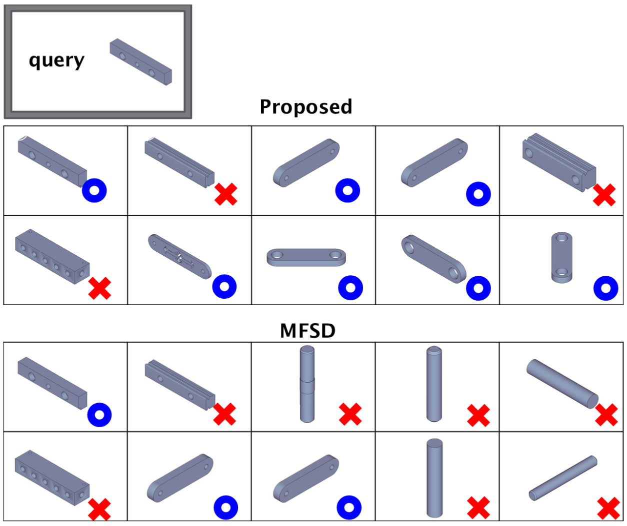

Fig. 16. Top 10 search results from MFSD and our proposed method, given the same query in the slender-links objects of ESB.

images alone do not contribute to the search efficiency, they play a major role whenever circular shapes including holes are observed at a certain projection of a given 3D mechanical part.

VI. Conclusions

We have proposed a new 3D shape feature focused on holes and surface roughness, suited to 3D mechanical CAD parts. We defined five base feature images including Contour, Hole, Surface Roughness, Line, and Circle images. All five are binary images. Then we applied peripheral intensity enhancement to change them into grayscale images. After peripheral intensity enhancement, we converted them into polar coordinates, then applied Fourier transform to obtain Fourier

TABLE IV
Comparison of our proposed method and previous methods in terms of 1-Tier, 2-Tier, P@1, and DCG for ESB

Method	1-Tier	2-Tier	P@1	DCG
Proposed	**52.8%**	**69.7%**	**90.2%**	**81.3%**
MFSD	49.4%	65.7%	87.5%	78.8%
DESIRE	45.1%	59.9%	86.0%	77.1%
LFD	43.8%	58.8%	85.5%	75.6%
SHD	43.9%	58.4%	83.5%	75.5%
D2	34.4%	47.6%	79.7%	69.0%

TABLE V
Effectiveness of Proposed Features independently when applied to ESB

Feature	1-Tier	2-Tier	P@1	DCG
Proposed(All)	**52.8%**	**69.7%**	**90.2%**	**81.3%**
Contour	46.0%	62.9%	85.1%	76.3%
Hole	42.7%	56.3%	86.1%	74.8%
Surface Roughness	44.3%	58.1%	85.2%	74.8%
Line	22.2%	31.2%	51.2%	56.5%
Circle	11.6%	21.2%	26.6%	46.5%

spectra. Through comparative experiments, we demonstrated that our proposed method outperformed the previous methods in ESB (Engineering Shape Benchmark), which consists of 3D mechanical parts. In the future, we plan to investigate more about the optimal combination of features for 3D shape retrieval suited to another particular collection of 3D shape objects such as architectural and human data collections.

Acknowledgment

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-In-Aid (C) 23500119, and was partially supported by the A-step FS-stage program (AS242Z01713H), Japan Science and Technology Agency (JST).

References

[1] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.
[2] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," in *Proc. Eurographics/ACM SIGGRAPH Symp. on Geometric Processing*, pp. 156–164, 2003.
[3] D.-Y. Chen, *Three-Dimensional Model Shape Description and Retrieval Based on Lightfield Descriptors*. PhD thesis, National Taiwan University, June 2003.
[4] R. Ohbuchi, T. Nakazawa, and T. Takei, "Retrieving 3D Shape Based On Their Appearance," in *Proc. 5th ACM SIGMM Workshop on Multimedia Information Retrieval (MIR2003)*, pp. 1–8, 2003.
[5] D. V. Vranic, "DESIRE: a Composite 3D-Shape Descriptor," in *Multimedia and Expo, IEEE International Conference on*, p. 4, 2005.
[6] A. Tatsuma and M. Aono, "Multi-Fourier Spectra Descriptor and Augmentation with Spectral Clustering for 3D Shape Retrieval," *Visual Computer*, vol. 25, pp. 785–804, June 2009.
[7] Z. Lian, A. Godil, and X. Sun, "Visual Similarity based 3D Shape Retrieval Using Bag-of-Features," in *IEEE International Conference on Shape Modeling and Applications (SMI2010)*, 2010.

[8] R. Ohbuchi and T. Furuya, "Distance Metric Learning and Feature Combination for Shape-Based 3D Model Retrieval," in *Proceedings of the ACM Workshop on 3D object retrieval 2010*, 2010.

[9] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani, "Developing an engineering shape benchmark for CAD models," *Computer-Aided Design*, vol. 38, no. 9, pp. 939–953, 2006.

[10] Y. Wang, R. Liu, F. Li, S. Endo, T. Baba, and Y. Uehara, "An Effective Hole Detection Method for 3D Models," in *20th European Signal Processing Conference (EUSIPCO 2012)*, pp. 1940–1944, 2012.

[11] T. Zaharia and F. Preteux, "Hough Transform-based 3D Mesh Retrieval ," in *Proc. SPIE 4476, Vision Geometry X, 175*, p. 11, 2001.

[12] 3DCafe. http://www.3dcafe.com/.

[13] M. Aono and H. Iwabuchi, "3D Shape Retrieval from a 2D Image as Query," in *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC 2012),*, 2012.

[14] P. Heckbert, "A Seed Fill Algorithm," in *Graphics Gems*, pp. 84–86, 1990.

[15] M. Nixon and A. S. Aquedo, *Feature Extraction and Image Processing for Computer Vision, Third Edition*. Academic Press, 2012.

[16] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *SMI '04 Proceedings of the Shape Modeling International*, pp. 167–178, 2004.

[17] R. Wessel, I. Blümel, and R. Klein, "A 3D Shape Benchmark for Retrieval and Automatic Classification of Architectural Data," in *Eurographics 2009 Workshop on 3D Object Retrieval*, pp. 53–56, Mar. 2009.

[18] S. Y. Tang, "An Evaluation of Local Shape Descriptors for 3D Shape Retrieval," 2012. http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909219.

[19] R. Baeza-Yates and B. Riberiro-Neto, *Modern Information Retrieval, Second Edition*. Addison Wesley, 2011.

[20] P. Min, *A 3D Model Search Engine*. PhD thesis, Princeton University, Jan. 2004.