Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 6049750, 12 pages
https://doi.org/10.1155/2017/6049750

Research Article

A Novel 3D CAD Model Retrieval Method Based on Vertices Classification and Weights Combination Optimization

Ting Zhuang,[1] Xutang Zhang,[1] Zhenxiu Hou,[1] Wangmeng Zuo,[2] and Yan Liu[3]

[1] *School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, China*
[2] *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China*
[3] *School of Science, Harbin Institute of Technology, Harbin, China*

Correspondence should be addressed to Ting Zhuang; zt_hit@126.com

Received 18 June 2016; Revised 8 November 2016; Accepted 17 November 2016; Published 18 January 2017

Academic Editor: Paolo Crippa

3D shape retrieval is a problem of current interest in different fields, especially in the mechanical engineering domain. According to our knowledge, multifeature based techniques achieve the best performance at present. However, the practicability of those methods is badly limited due to the high computational cost. To improve the retrieval efficiency of 3D CAD model, we propose a novel 3D CAD model retrieval algorithm called VSC_WCO which consists of a new 3D shape descriptor named VSC and Weights Combination Optimization scheme WCO. VSC represents a 3D model with three distance distribution histograms based on vertices classification. The weighted sum of $L1$ norm distances between corresponding distance histograms of two VSC descriptors is regarded as dissimilarity of two models. For higher retrieval accuracy on a classified 3D model database, WCO is proposed based on Particle Swarm Optimization and existing class information. Experimental results on ESB, PSB, and NTU databases show that the discriminative power of VSC is already comparable to or better than several typical shape descriptors. After WCO is employed, the performance of VSC_WCO is similar to the leading methods by all performance metrics and is much better by computational efficiency.

1. Introduction

The development of computer graphics technology gives a birth to the explosion in the number of 3D models which are now widely used in many diverse applications. Meanwhile, more and more 3D model databases are available on the web, such as Princeton Shape Benchmark (PSB) [1], National Taiwan University Shape Benchmark (NTU) [2], and Engineering Shape Benchmark (ESB) [3]. As Funkhouser et al. [4] predicted, the key question about 3D model has shifted from “How do we construct them?” to “How do we find them?” There is no doubt that searching suitable 3D engineering models from existing resource is extremely beneficial for mechanical design, because (1) more than 75% of design activity comprises reuse of previous design knowledge to address a new design problem [5] and (2) modeling a complex 3D model, such as mechanical parts, is still error-prone and time-consuming. Therefore, it is very necessary to develop efficient tools for 3D CAD model retrieval. As a promising technique, the so-called content-based 3D model retrieval, using the characteristics of 3D model itself, has attracted a lot of research interests [6–13]. The content-based approaches are broadly classified into three categories: shape-based, view-based, and multifeature based approaches.

Shape-Based Approaches. This kind of approaches uses the distribution of 3D features to characterize the geometric properties of a 3D model. D2 [14] is a well known shape-based 3D model retrieval method. D2 took samples on the distances between two points on the model surface and then accumulated them into bins of different intervals to form a distance distribution histogram as the model shape descriptor. D2 has many advantages: robustness, rotation invariance, and computational efficiency. However, the discriminative capability of D2 is limited for complex 3D model such as 3D engineering models. Inspired by D2, several new shape-based methods were proposed. Shih et al. [15] investigated a descriptor named Grid D2 (GD2). In GD2, the 3D model is decomposed by a voxel grid. A voxel is regarded as valid if there is a polygonal surface located within this voxel and invalid otherwise. The distribution of distance between two valid voxels instead of two points on the surface is calculated. Different from GD2, the shape distributions in Volume D2 (VD2) [16] are generated from Euclidean distance between randomly selected voxels. Not only surface voxels but also internal ones are involved. In order to reduce the computational complexity of D2, Wu et al. [17] proposed an algorithm called Quick D2 which modifies the sampling phase of D2. Pan et al. [18] proposed a new shape descriptor called Poisson Histogram which can be defined by the following two steps. Firstly, a 3D shape signature was defined by Poisson equation. Secondly, a histogram was constructed by accumulating the values of the defined signature in bins.

View-Based Approaches. The main concept of visual representation in 3D model retrieval is to first convert the 3D model into 2D projection image and then utilize mature image processing techniques to extract various features [19]. The original work was accomplished by Chen et al. [2], who proposed a method named Light Field (LF) descriptor which is a typical view-based approach. The main idea of LF is that if two 3D models are similar, they also look similar from all viewing angles. LF represents a 3D model as a collection of 2D images rendered from uniformly sampled positions on a viewing sphere located around the model. The distance between two descriptors is defined as the minimum $L1$ distance, taken over all rotations and all pairing of vertices on two dodecahedra. LF had been shown to perform well on the ESB [3] database. In [20], adaptive views clustering (AVC) selected the best characteristic views from more than 320 projected views. The characteristic view selection algorithm is based on an adaptive clustering algorithm and uses statistical model distribution scores to select the optimal number of views. The view-based methods achieved better retrieval performance compared to shape-based ones because they capture more of shape content. However, as pointed out in [2], view-based methods involve more computation compared to the shape-based ones.

Multifeature Based Approaches. According to [3], the shape representations that hold more shape content are better at retrieving more relevant models. Therefore, combining different 3D shape features is an efficient way to enhance the retrieval accuracy [21]. Papadakis et al. [22] presented a novel hybrid 3D model shape descriptor named PANORAMA using a set of panoramic views of a 3D model. A panoramic view is generated by projecting the model to the lateral surface of a cylinder parallel to one of its three principal axes and centered at the centroid of the model. The 3D model is projected to three perpendicular cylinders, each one aligned with one of its principal axes in order to capture the global shape of the model. They used Fourier and wavelet transforms to extract the features for each panoramic view. Leng and Xiong [23] proposed a composite shape descriptor called TUGE which is obtained from a combination of two-view version of depth buffer-based shape descriptor in [24] and GEDT shape descriptor in [4]. Li and Johan [25] proposed a hybrid 3D shape descriptor ZFDR which comprises four features: Zernike moments, Fourier descriptor, depth information and ray-based features.

Multifeatures based methods have more powerful retrieval capability than shape-based and view-based techniques; however, the time consumption is much higher than the latter because of involving more computations. In ZFDR example [25], the average retrieval time on ESB database (contains 866 3D CAD models) is 1.37 sec. If the target database is 100 times bigger than ESB, the average retrieval time will be more than 10 sec. High time cost limits the practicability of multifeature based techniques seriously.

In addition to those methods mentioned above, machine learning technologies are also used to address 3D models retrieval. In [26], 3D models are represented as probability distributions of binary variables on a 3D voxel grid and the proposed method uses a Convolutional Deep Belief Network to learn the distribution of complex 3D shapes across different categories and arbitrary poses from raw CAD data. In [27], the distance between 3D models is computed based on distance histogram features and 3D moment features. Using this distance measure, the relationships between all 3D models in dataset are formulated as a graph structure. A semisupervised learning process is then conducted to estimate the relevance among the 3D models. Leng et al. [28] proposed a 3D model recognition mechanism based on Deep Boltzmann Machines (DBM). The high-level abstraction representation can be obtained from a DBM which is trained by depth images of 3D model and the feature is used in semisupervised classification method. Qin et al. [29] proposed a deep learning approach to automatically classify 3D CAD models according to the mechanical part catalogue. The designed deep neural network classifier is based on the latest machine learning technique, deep learning.

In this paper, to improve the retrieval efficiency of 3D CAD model, we propose a novel retrieval algorithm named VSC_WCO which consists of a new shape-based 3D model descriptor called VSC and Weights Combination Optimization scheme WCO. The VSC descriptor represents a 3D model with three Euclidean distance distribution histograms which are calculated based on vertices classification. The weighted sum of $L1$ norm distances of corresponding histograms in two VSC descriptors is regarded as the dissimilarity of two models. To avoid risks caused by using fixed weights in VSC distance computation and further improve the retrieval performance on classified 3D model database, WCO is proposed based on Particle Swarm Optimization (PSO) and existing class information of 3D model database. The objective of WCO is searching optimal weights combinations for different query models.

The rest of this paper is organized as follows. The proposed 3D shape descriptor VSC is presented in Section 2. In Section 3, we describe the details of WCO and our 3D CAD models retrieval method VSC_WCO. Experimental results are demonstrated in Section 4. Finally, conclusions and future work are presented in Section 5.

2. 3D Shape Descriptor VSC Based on Vertices Classification

In this section, we define a new shape-based 3D model descriptor named VSC based on classification of vertices. A VSC descriptor consists of three distance distribution histograms which are computed by the following steps: (i) according to Tensor Voting Theory, the vertices of a 3D model are divided into three categories: Face, Sharp Edge, and Corner; (ii) the Euclidean distances between pairs of vertices in the same category are calculated; (iii) the distance distribution histograms of distinct categories are formed.

2.1. Vertices Classification Based on Tensor Voting Theory. Tensor voting theory has great advantages in computer vision tasks such as segmentation and object recognition. According to [30], the normal voting tensor is defined as

$$T_v = \sum_{t_i \in N_t(v)} \mu_{t_i} \vec{n}_{t_i} \vec{n}_{t_i}^T, \tag{1}$$

where v is vertex of 3D mesh model; T_v is the normal voting tensor of v; $N_t(v)$ denotes a collection of 1-ring neighbor triangles of v; t_i is ith 1-ring neighbor triangle in $N_i(v)$; $\vec{n}_{t_i}$ represents the unit normal vector of t_i; μ_{t_i} is calculated based on [30]:

$$\mu_{t_i} = \left(\frac{\text{area}(t_i)}{\max(\text{area}(N_t(v)))} \right) \exp\left(- \frac{\left\| \vec{c}_{t_i} - \vec{p}_v \right\|}{\max\left(\left\| \vec{c}_{t_i} - \vec{p}_v \right\| \right)} \right), \tag{2}$$

where area (t_i) is area of triangle t_i; $\max(\text{area}(N_t(v)))$ is maximum area among $N_t(v)$; $\vec{c}_{t_i}$ denotes barycenter of triangle t_i; $\vec{p}_v$ represents the position of v.

Because T_v is a rank-3 positive semidefinite matrix, it can be diagonalized as follows:

$$T_v = \lambda_1 \vec{e}_1 \vec{e}_1^T + \lambda_2 \vec{e}_2 \vec{e}_2^T + \lambda_3 \vec{e}_3 \vec{e}_3^T, \tag{3}$$

where $\vec{e}_1$, $\vec{e}_2$, and $\vec{e}_3$ are the corresponding unit eigenvectors of λ_1, λ_2, and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$), respectively. According to the eigenvalues, vertices can be divided into Face, Sharp Edge, and Corner as follows [31]:

(i) Face: λ_1 is dominant; λ_2 and λ_3 are close to 0.

(ii) Sharp Edge: λ_1 and λ_2 are dominant; λ_3 is close to 0.

(iii) Corner: λ_1, λ_2, and λ_3 are approximately equal.

Figure 1 shows an example of vertices classification; (a) is a 3D CAD model called advgr01 in ESB. Marked points in (b), (c), and (d) denote the vertices of Face, Sharp Edge, and Corner, respectively. The numbers of vertices in the distinct categories are 670, 127, and 16, respectively.

2.2. Distance Distribution Histogram. After classification of vertices, the Euclidean distances between any two vertices in the distinct categories are measured. To eliminate the influence of quantity of points, the distances histogram containing n bins is defined as

$$h = \frac{\{b_1, b_2, b_3, \ldots, b_n\}}{N(N-1)/2}, \tag{4}$$

where b_i is the number of distances within the range of the ith bin; N is the quantity of each type of vertices; h denotes a histogram constructed by counting how many distances fall into each bin. The width of bin is determined by

$$\text{width} = \frac{|d_{\max} - d_{\min}|}{n}, \tag{5}$$

where $d_{\max}$ and $d_{\min}$ are maximum and minimum distance between pairs of vertices in the same class; n is the number of bins.

Three distance distribution histograms compose the 3D shape descriptor VSC which is defined as follows:

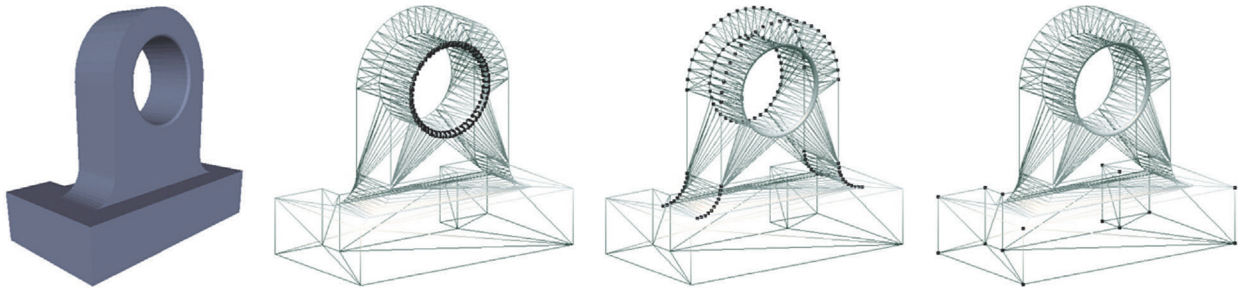
$$\text{VSC} = \left\{ h^f, h^e, h^c \right\}, \tag{6}$$

where h^f, h^e, and h^c denote distance distribution histograms of vertices in Face, Sharp Edge, and Corner, respectively. Table 1 is a comparison between feature extraction results of D2 and VSC for six models of ESB database, the first two models are taken from different classes of ESB; the third and fourth models are taken from *ESB\Flat-Thin Wall Components\Back Doors;* and the fifth and sixth are taken from *ESB\Solid Of Revolution\Gear-like Parts*. As shown in Table 1, the first and second models are completely different in shape, but their D2 descriptors are very similar. Different to D2, those two models are quite different in the view of VSC. The third and fourth models are taken from the same class and their shapes are very similar. There are almost no differences between the VSC descriptors of those two models, so do the D2 descriptors. The fifth and sixth models are taken from the same class also, but there is a little difference between their shapes. Through comparing two VSC descriptors of those two models, it is clear that the VSC descriptor reflects differences between two models successfully. Generally speaking, due to the subdivision of distances between sample points, the shape discrimination capacity of VSC is better than D2.

2.3. Distance between Two VSC Descriptors. After VSC descriptors are constructed, the similarity comparison between two 3D models is mapped into the comparison of VSC descriptors. The distance between two VSC descriptors is defined as the weighted sum of $L1$ norm distances of corresponding histograms:

$$\begin{aligned} \text{Dis}\left(\text{VSC}_{m_a}, \text{VSC}_{m_b}\right) = {} & w_1 \times L_1\left(h^f_{m_a}, h^f_{m_b}\right) + w_2 \\ & \times L_1\left(h^e_{m_a}, h^e_{m_b}\right) + w_3 \\ & \times L_1\left(h^c_{m_a}, h^c_{m_b}\right), \end{aligned} \tag{7}$$

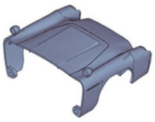
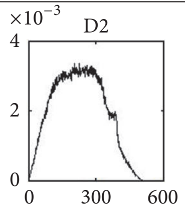
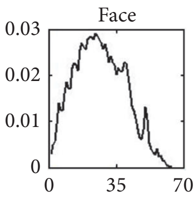
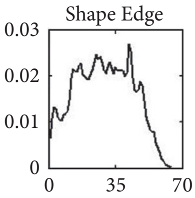
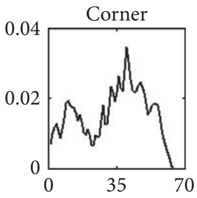
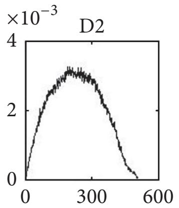
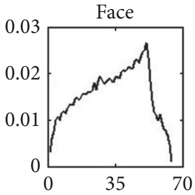
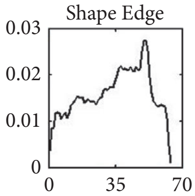
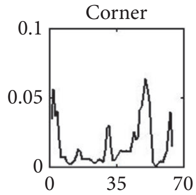
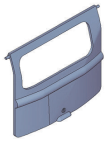
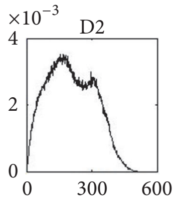
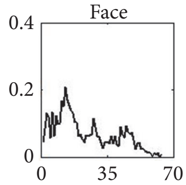
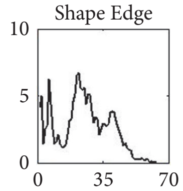
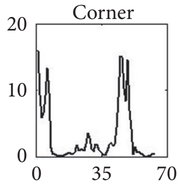
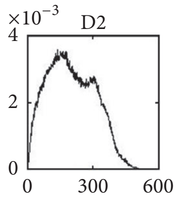
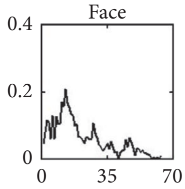
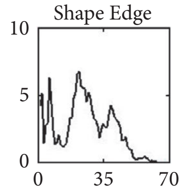
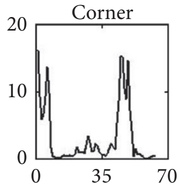
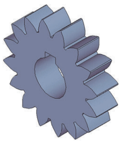
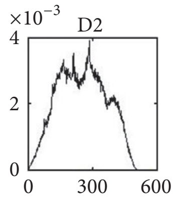
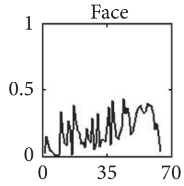
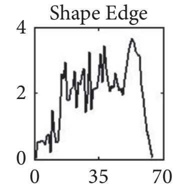
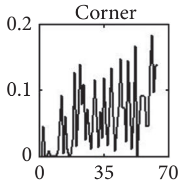
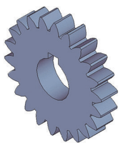
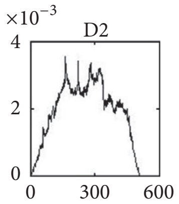
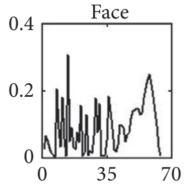
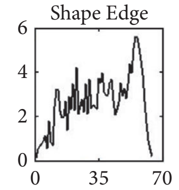
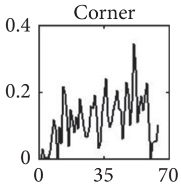
where m_a and m_b denote two 3D models; VSC_{m_a} and VSC_{m_b} are VSC descriptors of m_a and m_b, respectively; L_1 is the $L1$ norm distance measure; w_1, w_2, and w_3 are weights. The



(a) (b) (c) (d)

Figure 1: An example of vertices classification: (a) 3D CAD model: ESB\Rectangular-Cubic Prism\Bearing Blocks\advgr01, 813 vertices; (b) vertices of Face; (c) vertices of Sharp Edge; (d) vertices of Corner.

Table 1: Comparison of feature extraction results of D2 and VSC on two ESB models.

Model	D2	VSC		
 Front				
DEMO_REF				
 Back Door 1				
Back Door 2				
 16t_2d_05w				
 22t_275d_05w				

three distance histograms reflect the different shape characteristics of a model and they have the same contribution for VSC distance computation. Therefore, we linearly combine them. In addition, the distances between the corresponding histograms fall in the same range; as such, we assign the same weight for each histogram: $w_1 = w_2 = w_3 = 1/3$. For two 3D models, the distance between corresponding VSC descriptor is regarded as the dissimilarity, and the smaller the VSC distance is, the more similar they are. According to (7), the VSC distance between first two models shown in Table 1 is 1.71.

For a query model q, different weights combinations used in (7) will produce different dissimilarities between q and models in database and lead to different retrieval results eventually. The influence of weights combination against retrieval result provides an opportunity to achieve more desirable retrieval result on a classified 3D model database (details are discussed in the next section).

3. 3D CAD Model Retrieval Algorithm VSC_WCO Based on VSC and Weights Combination Optimization Scheme WCO

In this section, to further improve retrieval performance in a classified 3D model database, we propose a 3D CAD model retrieval algorithm called VSC_WCO based on 3D shape descriptor VSC presented in Section 2 and Weights Combination Optimization scheme WCO. In order to avoid the risks caused by using fixed weights combination in VSC distance computation, WCO takes into account the class information of 3D model database and utilizes Particle Swarm Optimization (PSO) to search optimal weights combination for each class.

3.1. Weights Combination Optimization. In order to avoid the risks caused by fixed weights and improve retrieval performance, we define Weights Combination Optimization method named WCO which takes into account the class information of 3D model database and utilizes Particle Swarm Optimization (PSO) [32] to search the optimal weights combination for each class. The motivations of WCO are described as follows:

(1) For a query model q, different weights combinations used in (7) will produce different dissimilarities between q and models in database and lead to different retrieval results eventually. The influence of weights combination against retrieval result provides an opportunity to achieve more desirable retrieval result by searching more suitable weights combination for q.

(2) In fact, those three distance histograms h^f, h^e, and h^c in the proposed shape descriptor VSC depict a 3D model from different aspects, respectively, and they can be regarded as three single 3D shape descriptors. It is demonstrated in [3] that no single 3D model descriptor performs well on all models and different descriptors have different strengths and weaknesses. According to this conclusion, fixed weights combination used in (7) for VSC distance computation is not suitable for all models.

In this subsection, we introduce the PSO first and then describe the details of WCO.

3.1.1. Particle Swarm Optimization (PSO). PSO has several advantages over other artificial intelligence algorithms. For example, it is better at global optimization and is easier to apply to multiple-objective problems [28]. PSO is initialized with a population of random potential solutions and the algorithm searches the optimal solution according to its performance. The goodness of a particle is determined by a function called fitness function in PSO. The fitness function takes position of a particle as input and returns a single number which denotes the goodness of the particle for the optimization problem.

Consider a group of N particles that are searching a global optimal solution in a D dimensions space. The position and velocity of P_i (ith particle) can be expressed as

$$X_i = \left(x_i^1, x_i^2, \ldots, x_i^D\right), \tag{8}$$

$$V_i = \left(v_i^1, v_i^2, \ldots, v_i^D\right), \tag{9}$$

$$\begin{aligned} V_i^{t+1} &= w \times V_i^t + c_1 \times r_1 \times \left(p\text{best}_i - X_i^t\right) + c_2 \times r_2 \\ &\quad \times \left(g\text{best}^t - X_i^t\right), \end{aligned} \tag{10}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \tag{11}$$

where V_i^t and V_i^{t+1} denote the velocity of P_i in iterations t and $t+1$; X_i^t and X_i^{t+1} represent the positions of P_i in iterations t and $t+1$; $p\text{best}_i$ is the best position of P_i until iteration t; $g\text{best}^t$ is the best position among all the particles until iteration t; c_1 and c_2 are the personal learning coefficient and the social learning factor, respectively; r_1 and r_2 are random numbers in the range of $(0, 1)$; w is an inertia factor from 0.8 to 1.2.

In WCO, PSO is used for searching the optimal weights combination for each class in a classified 3D model database; thus the position and velocity of each particle consist of three numbers, respectively. In order to avoid negative numbers during the optimization process, the relationship between positions and weights is expressed as

$$w_i = \exp\left(x_i\right) \quad (i = 1, 2, 3), \tag{12}$$

where w_i is the ith weight and x_i is the ith component of particle's position.

It is obvious that if the weights combination used in (7) is more reasonable to query model q, the VSC distance between q and all models in database will be more accurate and the retrieval performance will be better. In other words, the more reasonable weights combination, the better retrieval performance. Thus, the retrieval performance achieved by a particle can be used as its fitness. We employ the performance metrics First Tier (FT) [1] to evaluate the retrieval performance of each particle. FT is expressed as

$$\text{FT}(R) = \frac{n_r}{|C| - 1}, \tag{13}$$

where R represents a retrieval result which is formed by sorting all models in ascending order based on their VSC distances from query model q; C denotes the class of q and $|C|$ is the cardinality of C; n_r is the number of models of C in top $|C| - 1$ list of R.

3.1.2. Implementation Process of WCO. By controlling weight alterations, one ensures the evaluation of retrieval result better, which is the main idea of WCO. The optimization process of WCO is summarized in Algorithm 1 and notations used in Algorithm 1 are explained in the Notations. According to Algorithm 1, the time and space complexity of WCO are $O(N \times T \times P \times |C| \times |M|)$ and $O(1)$.

3.2. Implementation of VSC_WCO. To improve the retrieval performance on a classified 3D CAD model database, we propose a new 3D CAD model retrieval algorithm named VSC_WCO based on VSC descriptor described in Section 2 and Weights Combination Optimization scheme WCO presented in Section 3.1. Our VSC_WCO is composed of two parts: online and offline parts, which are described as follows and the whole process of VSC_WCO is illustrated in Figure 2.

Offline

(1) *VSC Feature Extraction*. We extract the VSC shape descriptors of all the models in Train and Test sets, based on the method in Section 2.

(2) *Optimal Weights Calculation in Train Set*. We calculate the optimal weights combinations (OWC) for all classes of Train set, based on WCO described in Section 3.1. All optimal weights combinations are stored in a database with the corresponding class information.

Online

(1) *VSC Feature Extraction*. We extract the VSC shape descriptor of query model q, based on the method in Section 2.

(2) *Determine the Most Similar Class (MSC) of q in Train Set*. First, we compute the VSC distance between q and models in Train set according to (7) using weights combination: {1/3, 1/3, and 1/3}, and then we select the nearest model's class as the most similar class of q, which is denoted as MSC_q.

(3) *Determine the Optimal Weights Combination (OWC) of q according to MSC_q*. The optimal weights combination corresponding to q's MSC is selected from database, which is denoted as OWC_q.

(4) *VSC Distances Computation in Test Set*. We compute the VSC distances between q and models in Test set according to (7) using weights combination OWC_q.

(5) *Ranking and Output*. Sort all the models in Test set in ascending order based on their VSC distances computed in step (4) and output retrieval lists accordingly.

4. Experiments

In this paper, all retrieval experiments are implemented in MATLAB R2011b and performed in a PC with configuration: CPU: Intel Pentium Dual-Core E5400@2.7 GHz; memory: 2.0 GB; OS: Windows XP. To investigate the performance of VSC_WCO for 3D engineering and generic models, we selected the following three representative standard benchmark databases:

(i) *Engineer Shape Benchmark (ESB) [3].* It is developed by Purdue University for evaluating the search methods relevant to the mechanical engineering domain. There are 866 3D engineering models in ESB and those models are classified into three superclasses, namely, Flat-Thin Wall Component, Rectangular-Cubic, and Solids of Revolution. Within each superclass, models are further classified into clusters of similar shapes. Figure 3 shows some example views of 3D models in ESB. In order to ensure the usefulness of weights combinations calculated by WCO, we equally divide ESB database into two parts: Train and Test sets, the former is used for WCO and the latter is used as target database for retrieval experiments.

(ii) *Princeton Shape Benchmark (PSB) [1].* It contains 1814 3D models totally, which are classified into Test and Train parts. In our experiments, Train database is used for WCO and Test database is used for retrieval experiments.

(iii) *National Taiwan University (NTU) Database [2].* NTU database contains 1,833 3D models and only 549 3D models are grouped into 47 classes and the remaining 1,284 models are assigned as the "miscellaneous." Thus, we only use the 549 classified ones for our experiments.

4.1. Retrieval Performance Evaluation Metrics. To comprehensively evaluate the 3D model retrieval results, we employ five metrics including precision-recall curve, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), and Discounted Cumulative Gain (DCG) [1]. Precision indicates how much percentage of the top K models belongs to the same class as the query model while recall means how much percentage of a class has been retrieved among the top K retrieval list. The precision-recall curve comprehensively demonstrates retrieval performance, which is assessed in terms of average recall and average precision. NN is defined as the percentage of the closest matches that are relevant models. FT is the recall of the top $C - 1$ list, where C is the cardinality of the relevant class of the query model. ST is the recall of the top $2(C - 1)$ list. DCG is defined as the summed weighted value related to it which combines precision and recall as well as ranking positions.

4.2. Comparison Algorithms. To compare the performance of our retrieval algorithm VSC_WCO, we consider the following four algorithms:

```
**Input**: A classified 3D model database $M$;
**Output**: Optimal weights combinations for all classes in $M$;
Initialize a population of particles with random positions and velocities;
FOR $(i = 1;\ i \le N;\ i++)$
  WHILE $(t \le T \parallel \text{fitness}_{gbest} \ne 1)$
    FOR $(j = 1;\ j \le P;\ j++)$
      FOR $(k = 1;\ k \le |C_i|;\ k++)$
        $q = m_k^{C_i}$; //the $k$th model in $C_i$ is used as query model
        FOR $(l = 1;\ l \le |M|;\ l++)$
          $d_l = \exp(x_1^{P_j}) \times L_1(h_q^f, h_{m_l}^f) + \exp(x_2^{P_j}) \times L_1(h_q^e, h_{m_l}^e) + \exp(x_3^{P_j}) \times L_1(h_q^c, h_{m_l}^c)$;
        End FOR
        $R_k = \text{Ascending}\ (d_1, d_2, \ldots, d_{|M|-1})$;
      End FOR
      $\text{fitness}_{p_j}^t = \sum_{k=1}^{|C_i|} \text{FT}(R_k)/|C_i|$;
      IF $(\text{fitness}_{p_j}^t > \text{fitness}_{pbest_{p_j}})$
        $\{x_1^{pbest_{P_j}}, x_2^{pbest_{P_j}}, x_3^{pbest_{P_j}}\} = \{x_1^{P_j}, x_2^{P_j}, x_3^{P_j}\}$;
        $\text{fitness}_{pbest_{p_j}} = \text{fitness}_{P_j}^t$;
      End IF
      IF $(\text{fitness}_{p_j}^t > \text{fitness}_{gbest})$
        $\{x_1^{gbest}, x_2^{gbest}, x_3^{gbest}\} = \{x_1^{P_j}, x_2^{P_j}, x_3^{P_j}\}$;
        $\text{fitness}_{gbest} = \text{fitness}_{P_j}^t$;
      End IF
    End FOR
    $t++$;
    Update all particles according to Equation (10) and Equation (11);
  End WHILE
  Output $\{x_1^{gbest}, x_2^{gbest}, x_3^{gbest}\}$ as optimal weights combination for $C_i$;
End FOR
```

ALGORITHM 1: Search optimal weighs combinations.

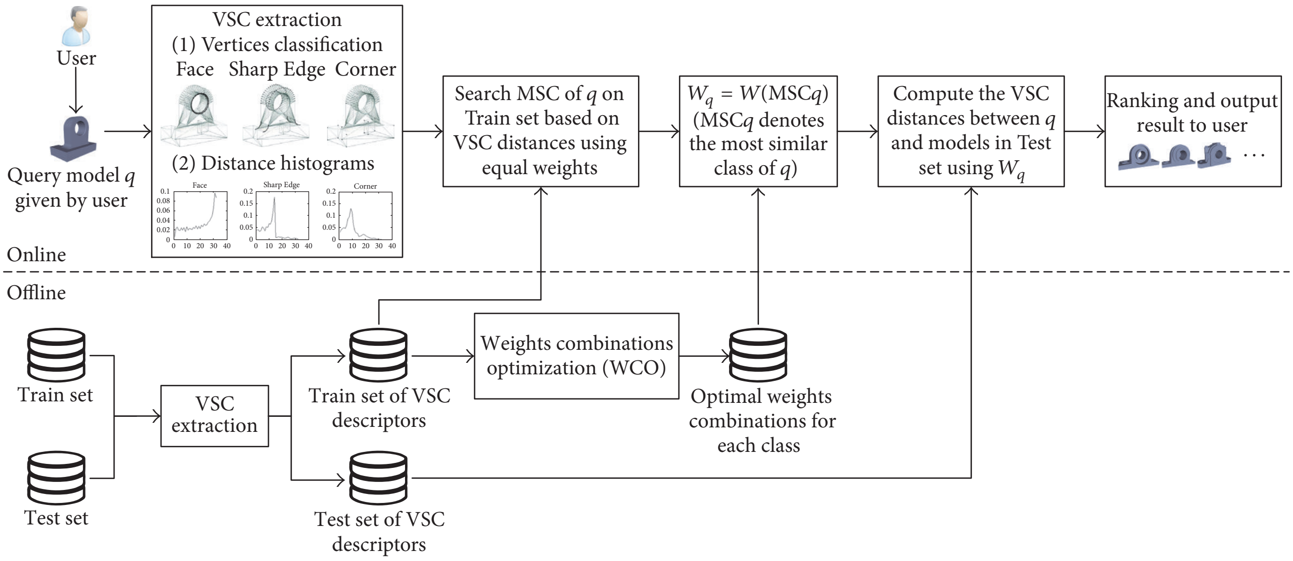

User
Query model q given by user
VSC extraction
(1) Vertices classification
Face Sharp Edge Corner
(2) Distance histograms
Face
Sharp Edge
Corner
Search MSC of q on Train set based on VSC distances using equal weights
$W_q = W(\text{MSC}q)$ (MSCq denotes the most similar class of q)
Compute the VSC distances between q and models in Test set using W_q
Ranking and output result to user
Online
Offline
Train set
Test set
VSC extraction
Train set of VSC descriptors
Test set of VSC descriptors
Weights combinations optimization (WCO)
Optimal weights combinations for each class

FIGURE 2: The process of the proposed algorithm VSC_WCO.

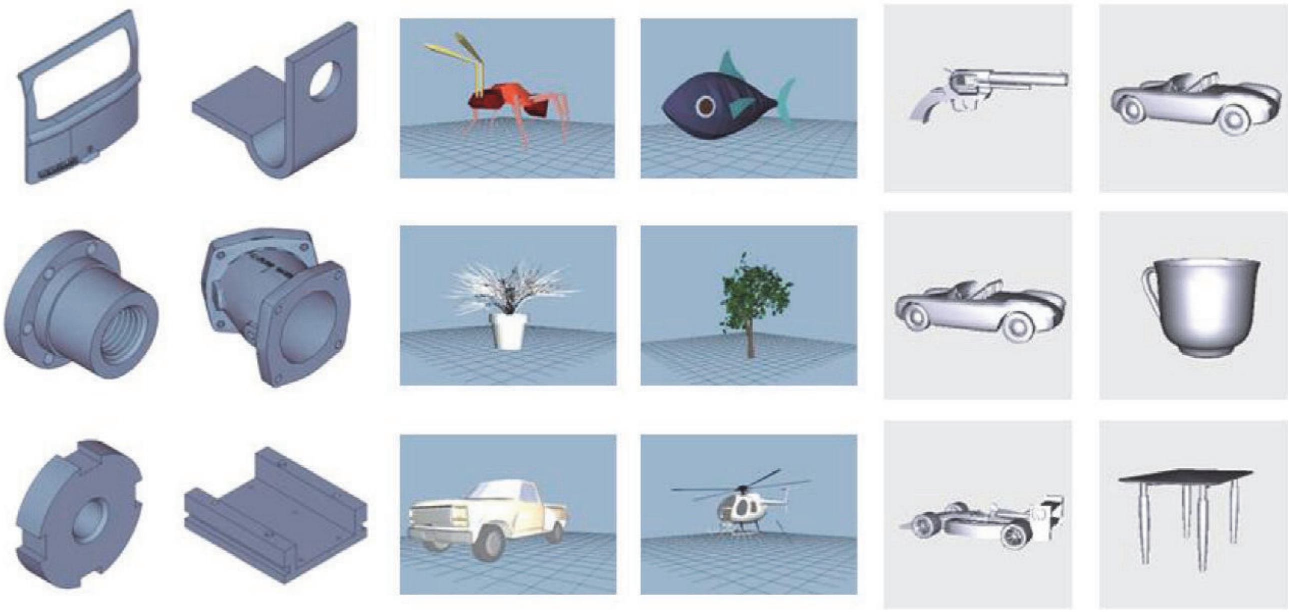


Figure 3: Example views of 3D models in ESB, PSB, and NTU.

(i) D2: a classic shape-based retrieval method and the most related method of VSC. We implemented D2 based on the original paper [14].

(ii) LF: a typical view-based retrieval method, which performed well on ESB [3]. We performed experiments based on their provided execution files [2].

(iii) ZFDR: a state-of-the-art hybrid retrieval algorithm, which integrates a 3D model's both visual and geometric information from different aspects. For the retrieval performance of ZFDR on ESB, PSB, and NTU, we refer to [25].

(iv) PANORAMA: a state-of-the-art hybrid retrieval algorithm, which utilizes a set of panoramic views. According to Li and Johan [25], PANORAMA is arguably the best retrieval method reported to date. For the retrieval performance of PANORAMA on ESB, PSB, and NTU, we refer to [22, 25].

4.3. Experiments Results and Analysis. In our experiments, every model in Test set is used as query model and to avoid bias we exclude this model from Test set when computing the VSC distance. Figures 4 and 5 and Table 2 compare the performance of our VSC_WCO algorithm and the above-mentioned four methods. To evaluate the effectiveness of WCO, for comparison, we also list the performances when using only the VSC shape descriptor. As can be seen in Figures 4 and 5 and Table 2, firstly, our shape descriptor VSC is better than D2 and comparable to LF with lower computational complexity. Secondly, after applying the WCO, VSC_WCO achieves significantly better performance than VSC. On ESB, VSC_WCO exceeds VSC in NN, FT, ST, and DCG by 2.4%, 13.4%, 12.2%, and 6.1%. On PSB, those metrics were increased by 2.9%, 16.7%, 14.8%, and 16.1%. The corresponding increments are 3.1%, 16.2%, 17.4%, and 13.1% on NTU. This increase indicates that fixed weights combination for VSC distance computation is not suitable for all query models and the optimal weights combinations produced by WCO are useful to improve the retrieval performance of VSC. Finally, the performance of VSC_WCO is substantially the same as that achieved by the state-of-the-art algorithms, such as PANORAMA and ZFDR. Furthermore, the average retrieval time of VSC_WCO on ESB, PSB, and NTU database are less than the latter. Table 3 lists the timings of VSC_WCO on

Table 2: Performance comparison: retrieval performance metrics of VSC, VSC_WCO, D2, LF, ZFDR, and PANORAMA on ESB, PSB, and NTU databases.

Database	Methods	NN	FT	ST	DCG
ESB	PANORAMA	0.86	0.49	0.64	0.79
	ZFDR	0.84	0.47	0.61	0.77
	VSC_WCO	0.84	0.51	0.58	0.81
	VSC	0.82	0.40	0.48	0.76
	LF	0.82	0.41	0.54	0.75
	D2	0.73	0.27	0.37	0.70
PSB	PANORAMA	0.75	0.48	0.60	—
	ZFDR	0.70	0.44	0.55	0.70
	VSC_WCO	0.69	0.42	0.54	0.72
	VSC	0.67	0.36	0.47	0.62
	LF	0.66	0.38	0.49	0.63
NTU	PANORAMA	0.80	0.49	0.61	0.76
	ZFDR	0.75	0.45	0.58	0.73
	VSC_WCO	0.67	0.43	0.54	0.69
	VSC	0.65	0.37	0.46	0.61
	LF	0.61	0.34	0.42	0.55

Table 3: Run time information of VSC_WCO on ESB, PSB, and NTU.

Database	t_1	t_2	t_3
ESB	0.293	0.016	0.017
PSB	0.364	0.034	0.038
NTU	0.416	0.001	0.001

ESB, PSB, and NTU databases. In Table 3, t_1 denotes the time consumption of VSC extraction of query model; t_2 denotes the time consumption of finding the most similar class of query model in Train set; t_3 is time consumption of searching similar models in Test set. On a common configuration computer (CPU: Intel Pentium Dual-Core E5400@2.7 GHz; memory: 2.0 GB), the average retrieval time of VSC_WCO is less than 0.5 s which enables real-time retrieval from large repositories. This is a significant advantage of the VSC_WCO as it can be rendered more efficient by reducing its storage requirements and time complexity. Altogether, VSC_WCO has considerable retrieval performance and is very efficient in computation and comparison time.

In Table 4, we provide a few examples of queries and the retrieved results to first 8 retrieval models from the ESB database using D2, VSC, and VSC_WCO. As illustrated in Table 4, the performance of VSC_WCO is better than VSC and D2 in terms of both quantity and rankings of relevant models.

4.4. Limitations. Our 3D CAD model retrieval algorithm VSC_WCO has achieved good performance on ESB database. However, there are still some limitations. Firstly, since the class information of 3D model database is a key condition for the proposed Weights Combination Optimization method WCO, VSC_WCO only can be used to the already classified 3D model database. Secondly, the optimal weights combination to a query model q is determined by the most similar class of q; therefore, if the most similar class is not the target class, then our method VSC_WCO will fail to get the right models.

5. Conclusions and Future Work

In this paper, to improve the retrieval efficiency for 3D CAD models in a classified 3D model database, we have proposed a 3D model retrieval method called VSC_WCO which is based on the proposed 3D shape descriptor VSC and Weights Combination Optimization scheme WCO. VSC descriptor represents a 3D model with three distance histograms which is computed by vertices classification based on Tensor Theory. WCO utilizes the existing class information and Particle Swarm Optimization (PSO) to search optimal weights combinations for all classes. The retrieval experiments are conducted on Engineering Shape Benchmark (ESB). We equally divide the ESB database into two parts, one is used as Train set for WCO to compute the optimal weights combinations for every class in ESB, and the other one is used as Test set for retrieval experiment. Several metrics are selected to evaluate the retrieval capacity of our algorithm. Retrieval results on ESB demonstrated that (1) the retrieval performance of the proposed 3D shape descriptor VSC is much better than shape-based descriptor D2 and comparable to view-based descriptor Light Field with lower computational complexity and, (2) after WCO is employed, the performance of VSC_WCO is close to the state-of-the-art algorithms, such as PANORAMA and ZFDR. Furthermore, the average retrieval time of VSC_WCO on ESB database is much lower than the latter.

Further research will focus on the following aspects: (1) in order to improve the accuracy of the most similar class of query model, we need to enhance the discrimination capacity of proposed shape descriptor VSC; (2) develop a 3D model clustering algorithm for applying our method VSC_WCO on unclassified 3D model databases.

Notations

M:	Classified 3D models database which has several classes
$\lvert M\rvert$:	Number of models in M
N:	Number of classes in M
C_i:	The ith class of M
$\lvert C_i\rvert$:	Number of models in C_i
$m_k^{C_i}$:	The kth model of C_i
P:	Number of particles in PSO
Pj:	The jth particle of PSO
Q:	Query model
d_l:	VSC distance between q and m_l
Ascending $(d_1, d_2, \dots, d_{\lvert M\rvert-1})$:	Sort all the models in M in ascending order based on VSC distances
R_k:	Retrieval result when $m_k^{C_i}$ is used as query model
T:	Maximum number of iterations
t:	Current number of iterations
$\text{fitness}_{P_j}^{t}$:	Fitness of P_j in the tth iteration
$\text{FT}(R_k)$:	Retrieval performance of P_j when $m_k^{C_i}$ is used as query model, measured by metrics First Tier which is expressed as (13)
$p\text{best}_{P_j}$:	The best position achieved by P_j so far
$\{x_1^{p\text{best}_{P_j}}, x_2^{p\text{best}_{P_j}}, x_3^{p\text{best}_{P_j}}\}$:	Three components of $p\text{best}_{P_j}$'s position
$\text{fitness}_{p\text{best}_{P_j}}$:	Fitness of $p\text{best}_{P_j}$
$g\text{best}$:	The best position achieved by any particle so far
$\{x_1^{g\text{best}}, x_2^{g\text{best}}, x_3^{g\text{best}}\}$:	Three components of $g$best's position
$\text{fitness}_{g\text{best}}$:	Fitness of $g$best.

Competing Interests

The authors declare that they have no competing interests.

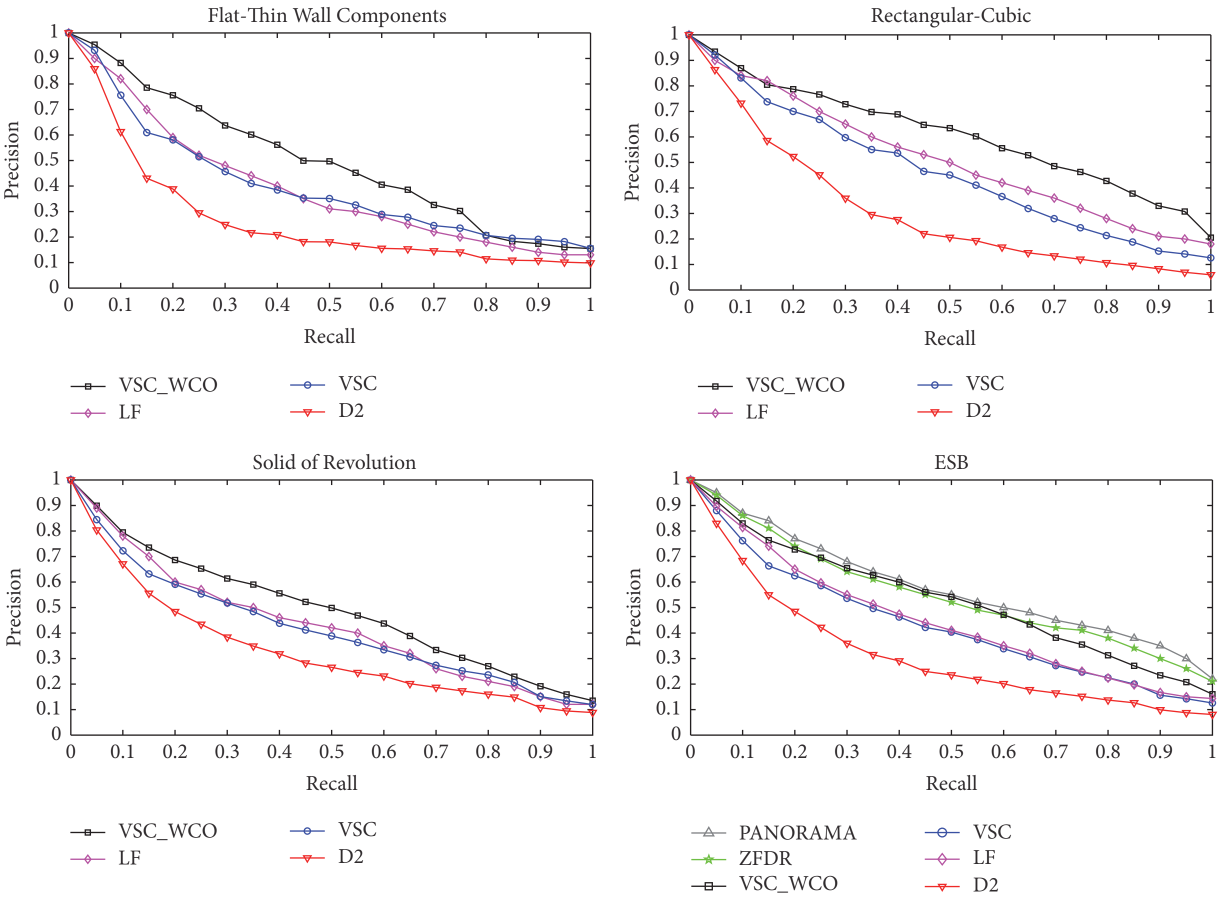


FIGURE 4: Performance comparison: precision-recall curve of VSC, VSC_WCO, and other methods on each superclass of ESB and whole ESB database.

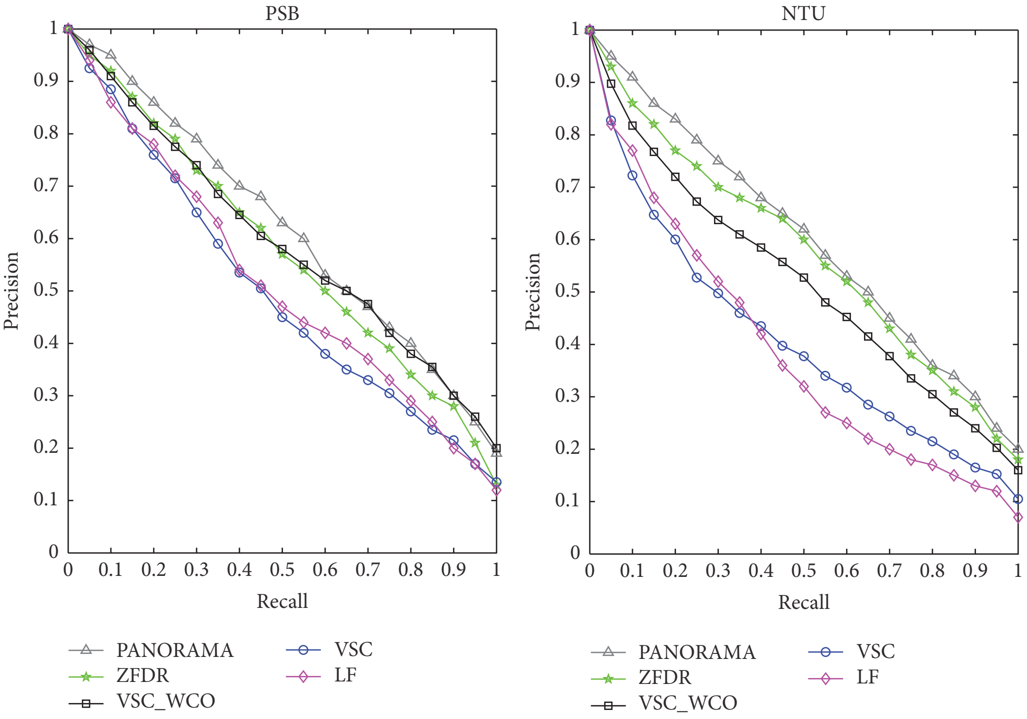
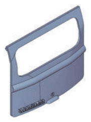
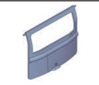
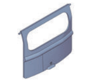
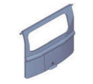
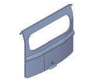
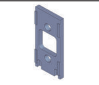
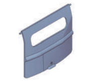
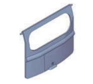
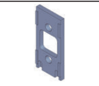
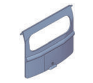
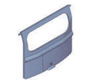
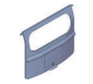
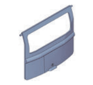
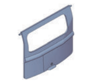
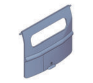
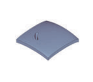
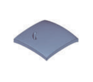
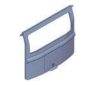
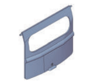
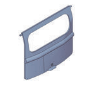
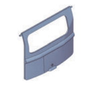
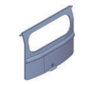
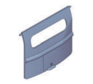
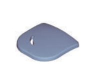
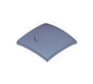
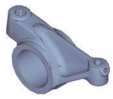
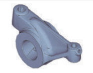
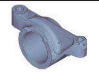
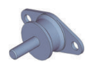
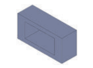
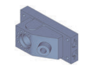
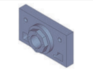
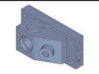
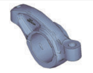
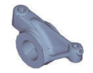
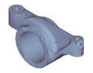
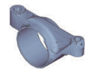
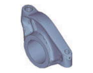
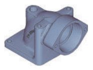
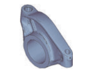
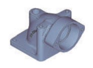
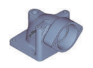
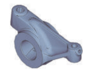
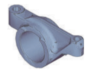
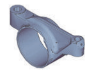
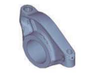
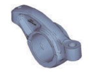
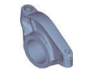
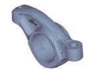
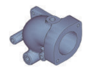
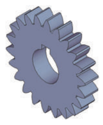
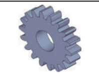
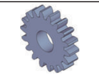
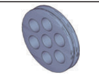
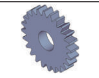
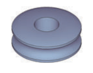
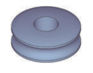
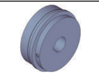
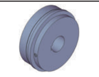
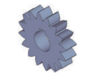
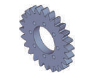
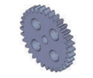
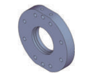
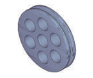
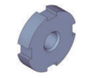
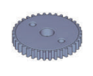
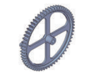
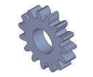
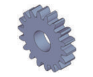
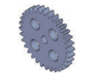
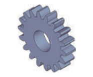
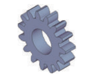
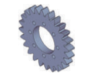
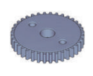
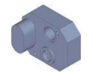


FIGURE 5: Performance comparison: precision-recall curve of VSC, VSC_WCO, and other methods on PSB and NTU databases.

Table 4: Some retrieval examples on ESB database using D2, VSC, and VSC_WCO.

Query model	Method	First 8 similar models							
		1	2	3	4	5	6	7	8
	D2								
	VSC								
	VSC_WCO								
	D2								
	VSC								
	VSC_WCO								
	D2								
	VSC								
	VSC_WCO								

Acknowledgments

This study was supported by Natural Science Foundation of China (51001121), Fundamental Research Funds for the Center Universities and Program for Innovation Research of Science in Harbin Institute of Technology (PIRS OF HIT Q201503), China Postdoctoral Science Foundation funded project (2015M581450), and Program for Innovation Research of Harbin City (2015RAQXJ090).

References

[1] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton shape benchmark," in *Proceedings of the Shape Modeling International (SMI '04)*, pp. 167–178, Genova, Italy, June 2004.

[2] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

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

[4] T. Funkhouser, P. Min, M. Kazhdan et al., "A search engine for 3D models," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105, 2003.

[5] W. C. Regli and V. A. Cicirello, "Managing digital libraries for computer-aided design," *CAD Computer Aided Design*, vol. 32, no. 2, pp. 119–132, 2000.

[6] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić, "Feature-based similarity search in 3D object databases," *ACM Computing Surveys*, vol. 37, no. 4, pp. 345–387, 2005.

[7] T. Funkhouser, M. Kazhdan, P. Min, and P. Shilane, "Shape-based retrival and analysis of 3D model," *Communications of the ACM*, vol. 48, no. 6, pp. 58–64, 2005.

[8] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, "Three-dimensional shape searching: state-of-the-art review and future trends," *CAD Computer Aided Design*, vol. 37, no. 5, pp. 509–530, 2005.

[9] A. Del Bimbo and P. Pala, "Content-based retrieval of 3D models," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 2, no. 1, pp. 20–43, 2006.

[10] Y. Yang, H. Lin, and Y. Zhang, "Content-based 3-D model retrieval: a survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1081–1098, 2007.

[11] B. Bustos, D. Keim, D. Saupe, and T. Schreck, "Content-based 3D object retrieval," *IEEE Computer Graphics and Applications*, vol. 27, no. 4, pp. 22–27, 2007.

[12] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, 2008.

[13] A. Foncubierta-Rodríguez, H. Müller, and A. Depeursinge, "Retrieval of high-dimensional visual data: current state, trends and challenges ahead," *Multimedia Tools and Applications*, vol. 69, no. 2, pp. 539–567, 2014.

[14] R. Osada, T. Funkhouser, B. Chazelle, and D. Bobkin, "Shape distribution," *ACM Trans Graphics*, vol. 4, pp. 807–832, 2002.

[15] J.-L. Shih, C.-H. Lee, and J. T. Wang, "3D object retrieval system based on grid D2," *Electronics Letters*, vol. 41, no. 4, pp. 179–181, 2005.

[16] Q. W. Bian, J. L. Wang, and Y. J. He, "Retrieval of 3D shapes using volume D2," *Electronics Letters*, vol. 45, no. 23, pp. 1163–1165, 2009.

[17] Y. Wu, L. Tian, and C. Li, "High efficient methods of content-based 3D model retrieval," *Chinese Journal of Mechanical Engineering*, vol. 26, no. 2, pp. 248–256, 2013.

[18] X. Pan, Q. You, Z. Liu, and Q. H. Chen, "3D shape retrieval by Poisson histogram," *Pattern Recognition Letters*, vol. 32, no. 6, pp. 787–794, 2011.

[19] Y. Gao and Q. Dai, "View-based 3D object retrieval: challenges and approaches," *IEEE Multimedia*, vol. 21, no. 3, pp. 52–57, 2014.

[20] T. F. Ansary, M. Daoudi, and J.-P. Vandeborre, "A bayesian 3-D search engine using adaptive views clustering," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 78–88, 2007.

[21] P. Daras, A. Axenopoulos, and G. Litos, "Investigating the effects of multiple factors towards more accurate 3-D object retrieval," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 374–388, 2012.

[22] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis, "Panorama: a 3D shape descriptor based on panoramic views for unsupervised 3d object retrieval," *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 177–192, 2010.

[23] B. Leng and Z. Xiong, "ModelSeek: an effective 3D model retrieval system," *Multimedia Tools and Applications*, vol. 51, no. 3, pp. 935–962, 2011.

[24] D. V. Vranic, *3d model retrieval [Ph.D. thesis]*, University of Leipzig, Leipzig, Germany, 2004.

[25] B. Li and H. Johan, "3D model retrieval using hybrid features and class information," *Multimedia Tools and Applications*, vol. 62, no. 3, pp. 821–846, 2013.

[26] Z. Wu, S. Song, A. Khosla et al., "3D ShapeNets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1912–1920, Boston, Mass, USA, June 2015.

[27] K. Lu, Q. Wang, J. Xue, and W. Pan, "3D model retrieval and classification by semi-supervised learning with content-based similarity," *Information Sciences*, vol. 281, pp. 703–713, 2014.

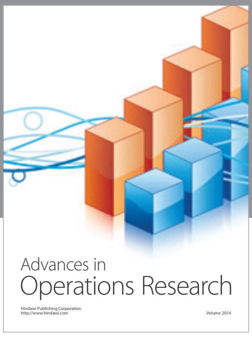
[28] B. Leng, X. Zhang, M. Yao, and Z. Xiong, "A 3D model recognition mechanism based on deep Boltzmann machines," *Neurocomputing*, vol. 151, no. 2, pp. 593–602, 2015.

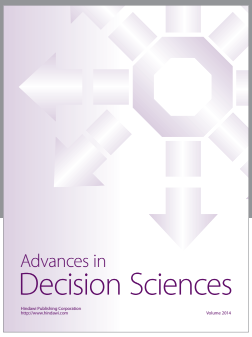
[29] F.-W. Qin, L.-Y. Li, S.-M. Gao, X.-L. Yang, and X. Chen, "A deep learning approach to the classification of 3D CAD models," *Journal of Zhejiang University: Science C*, vol. 15, no. 2, pp. 91–106, 2014.

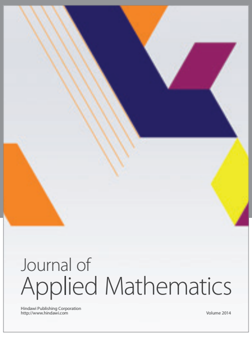
[30] H. S. Kim, H. K. Choi, and K. H. Lee, "Feature detection of triangular meshes based on tensor voting theory," *Computer-Aided Design*, vol. 41, no. 1, pp. 47–58, 2009.

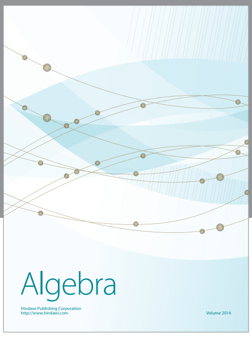
[31] T. Shimizu, H. Date, S. Kanai, and T. Kishinami, "A new bilateral mesh smoothing method by recognizing features," in *Proceedings of the 9th International Conference on Computer Aided Design and Computer Graphics (CAD/CG '05)*, pp. 281–286, Hong Kong, China, December 2005.

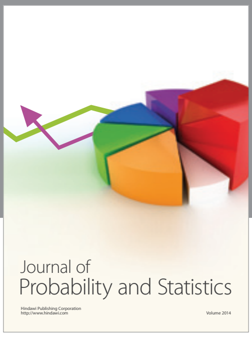
[32] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Conference on Evolutionary Computation*, pp. 81–86, Soul, Korea, May 2001.


Advances in
Operations Research


Advances in
Decision Sciences

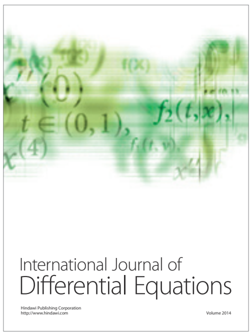

Journal of
Applied Mathematics

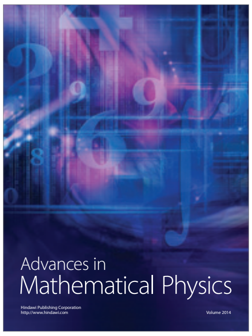

Algebra

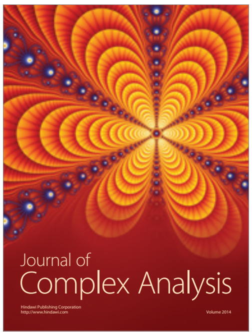

Journal of
Probability and Statistics

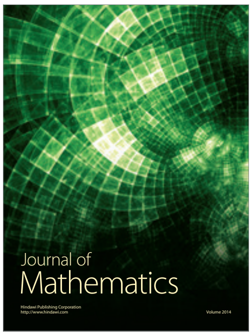
The Scientific
World Journal

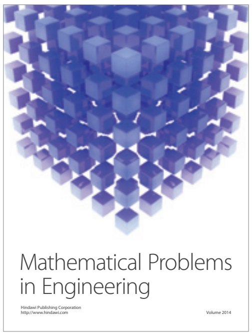
International Journal of
Combinatorics

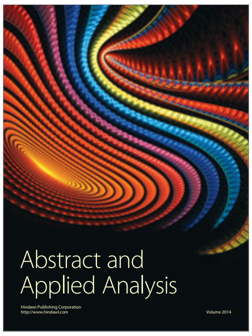

International Journal of
Differential Equations

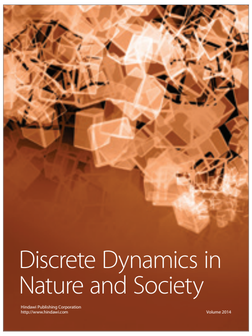

Advances in
Mathematical Physics

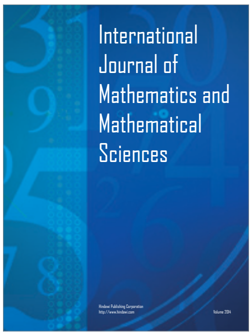

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

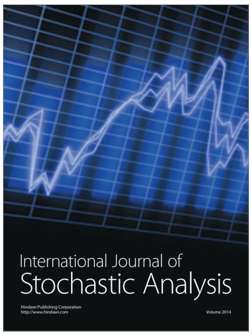

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization