GIFT: Towards Scalable 3D Shape Retrieval

Song Bai, Xiang Bai, *Senior Member, IEEE*, Zhichao Zhou, Zhaoxiang Zhang, *Senior Member, IEEE*, Qi Tian, *Fellow, IEEE*, and Longin Jan Latecki, *Member, IEEE*

***Abstract*—Projective analysis is an important solution in three-dimensional (3D) shape retrieval, since human visual perceptions of 3D shapes rely on various 2D observations from different viewpoints. Although multiple informative and discriminative views are utilized, most projection-based retrieval systems suffer from heavy computational cost, and thus cannot satisfy the basic requirement of scalability for search engines. In the past three years, shape retrieval contest (SHREC) pays much attention to the scalability of 3D shape retrieval algorithms, and organizes several large scale tracks accordingly [1]–[3]. However, the experimental results indicate that conventional algorithms cannot be directly applied to large datasets. In this paper, we present a real-time 3D shape search engine based on the projective images of 3D shapes. The real-time property of our search engine results from the following aspects: 1) efficient projection and view feature extraction using GPU acceleration; 2) the first inverted file, called F-IF, is utilized to speed up the procedure of multiview matching; and 3) the second inverted file, which captures a local distribution of 3D shapes in the feature manifold, is adopted for efficient context-based reranking. As a result, for each query the retrieval task can be finished within one second despite the necessary cost of IO overhead. We name the proposed 3D shape search engine, which combines GPU acceleration and inverted file (twice), as GIFT. Besides its high efficiency, GIFT also outperforms state-of-the-art methods significantly in retrieval accuracy on various shape benchmarks (ModelNet40 dataset, ModelNet10 dataset, PSB dataset, McGill dataset) and competitions (SHREC14LSGTB, ShapeNet Core55, WM-SHREC07).**

***Index Terms*—3D shape retrieval, CNN, shape retrieval contest (SHREC).**

Manuscript received July 31, 2016; revised November 28, 2016; accepted December 29, 2016. Date of publication January 11, 2017; date of current version May 13, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61231010, Grant 61573160, and Grant 61429201, in part by the China Scholarship Council, and in part by the National Science Foundation under Grant IIS-1302164. The work of Q. Tian was supported by the ARO Grant W911NF-15-1-0290, and by the Faculty Research Gift Awards by NEC Laboratories of America and Blippar. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yap-Peng Tan. *(Corresponding author: Xiang Bai.)*

S. Bai, X. Bai, and Z. Zhou are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: songbai@hust.edu.cn; xbai@hust.edu.cn; zzc@hust.edu.cn).

Z. Zhang is with the Research Center for Brain-Inspired Intelligence, CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhaoxiang.zhang@ia.ac.cn).

Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249-1604 USA (e-mail: qitian@cs.utsa.edu).

L. J. Latecki is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: latecki@temple.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2017.2652071

I. Introduction

3D shape retrieval is a fundamental issue in many fields, including multimedia computing, computer vision, graphics and pattern recognition. Given a query shape, the goal of retrieval is to return a list of shapes which share similar geometric characteristic or semantic meaning with the query. Thus, one crucial part of 3D shape retrieval is to design informative and discriminative features, so that visually similar shapes will have small dissimilarities. Enormous efforts [4]–[8] have been devoted to retrieval *effectiveness*, that is to say, to boost the retrieval accuracy.

With the rapid development of large scale public 3D repositories, e.g., Google 3D Warehouse or TurboSquid, and large scale shape benchmarks, e.g., ModelNet [9], SHape REtrieval Contest (SHREC) [1]–[3], the scalability of 3D shape retrieval algorithms becomes increasingly important for practical applications. However, as suggested in [1], plenty of those conventional algorithms cannot scale up to large 3D shape databases due to their high time complexity. It indicates that retrieval *efficiency* issue has been more or less ignored by previous works.

Meanwhile, owing to the fact that human visual perception of 3D shapes depends upon 2D observations, projective analysis has become a basic and inherent tool in 3D shape domain for a long time, with applications to segmentation [10], matching [11], reconstruction, recognition [12], [13], etc. Specifically in 3D shape retrieval, projection-based methods demonstrate impressive performances. Especially in recent years, the success of planar image representation [14], makes it easier to describe 3D models using depth or silhouette projections.

Generally, a typical 3D shape search engine is comprised of the following four components:

1) *Projection rendering.* With a 3D model as input, the output of this component is a collection of projections in depth buffers, binary masks or RGB images. Most methods set an array of virtual cameras at pre-defined viewpoints to capture views. These viewpoints can be the vertices of a dodecahedron [15], located on the unit sphere [16], or around the lateral surface of a cylinder [11]. In most cases, pose normalization [17] is needed for the sake of invariance to translation, rotation and scale changes.
2) *View feature extraction.* The role of this component is to obtain multiple view representations, which affects the retrieval quality largely. A widely-used paradigm is Bag-of-Words (BoW) [14] model. BoW has shown its superiority as natural image descriptors in many fields, such as image search [18], [19] and classification [20]. However, in order to get better performances, many features [1]

1520-9210 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

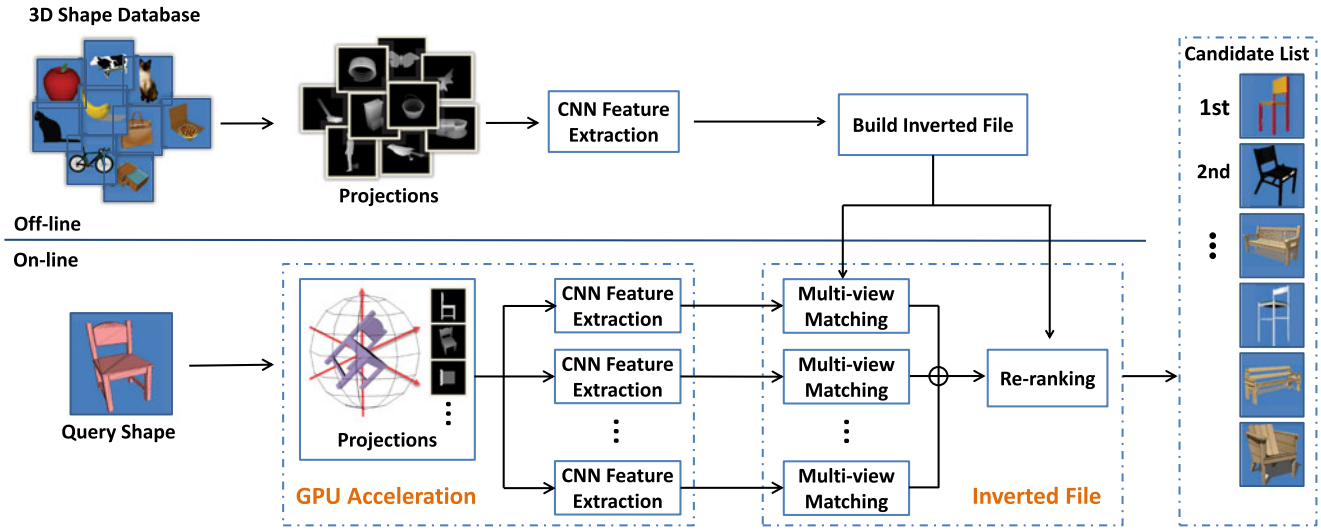


Fig. 1. Structure of the proposed 3D shape search engine GIFT.

are of extremely high dimension. As a consequence, raw descriptor extraction (e.g., SIFT [21]), quantization and distance calculation are all time-consuming.

3) *Multi-view matching.* This component establishes the correspondence between two sets of view features, and returns a matching cost between two 3D models. Since at least a set-to-set matching strategy [22] is required, this stage suffers from high time complexity even when using the simplest Hausdorff matching. Hence, the usage of some more sophisticated matching strategies on large scale 3D datasets is limited due to their heavy computational cost.
4) *Re-ranking.* It aims at refining the initial ranking list by using some extra information. For retrieval problems, since no prior or supervised information is available, contextual similarity measure is usually utilized. A classic context-based re-ranking methodology for shape retrieval is diffusion process [23], which exhibits outstanding performances on various datasets. However, as graph-based and iterative algorithms, many variants of diffusion process (e.g., locally constrained diffusion process [24], [25]), generally require the computational complexity of $O(TN^3)$, where N is the total number of shapes in the database and T is the number of iterations. In this sense, diffusion process does not seem to be applicable for real-time analysis.

In this paper, we present a real-time 3D shape search engine (see Fig. 1) that includes all the aforementioned components. It combines Graphics Processing Unit (GPU) acceleration and Inverted File (Twice), hence we name it GIFT. In on-line processing, once a user submits a query shape, GIFT can react and present the retrieved shapes within one second (the off-line preprocessing operations, such as CNN model training and inverted file establishment, are excluded). GIFT is evaluated on several popular 3D benchmark datasets, especially on two tracks of SHape REtrieval Contest (SHREC) which focuses on scalable 3D retrieval. The experimental results on retrieval accuracy and query time demonstrate the potential of GIFT in handling large scale data.

In summary, our main contributions lie in three aspects.

1) GPU is used to speed up the procedure of projection rendering and feature extraction.
2) In multi-view matching procedure, a robust version of Hausdorff distance for noise data is approximated with an inverted file, which allows for extremely efficient matching between two view sets without impairing the retrieval performances too much.
3) In the re-ranking component, a new feature fusion algorithm based on fuzzy set theory is proposed, which can fuse hierarchical activations of neural network using fuzzy aggregation operator. Different from diffusion processes of high time complexity, our re-ranking here is quite time efficient on account of using inverted file again.

Compared with the previous conference version [26], this article gives a deeper analysis about the evolution of related algorithms. To make GIFT suitable to tackle more than two similarity measures, an approach called "neighbor multi-augmentation" is proposed. By doing so, we also improve the retrieval performances on all the shape benchmarks by combining the deep features of GIFT with handcrafted features. Meanwhile, GIFT reports excellent retrieval performances on the latest ShapeNet Core55 large scale competition. Afterwards, we show that GIFT, with some simple modifications, can be applied to shape classification task, and achieves comparable classification accuracies on ModelNet dataset. Promising future topics that can be investigated in the proposed framework are summarized at last.

The rest of paper is organized as follows. In Section II, we briefly introduce some related works. The details of the proposed search engine are given in Section III. In Section IV, comprehensive experiments and comparisons with other state-of-the-art algorithms are conducted on various shape benchmarks and competitions. Future work is discussed in Section V and conclusions are given in Section VI.

II. Related Work

3D shape retrieval has been extensively investigated for a long time, and plenty of algorithms were proposed for 3D model preprocessing, feature extraction, shape matching, etc. A thorough and exhausted review of those algorithms is unrealistic. Therefore, we mainly focus on projection-based methods which have a close relationship with our work.

Light Field Descriptor (LFD) [15], composed of Zernike moments and Fourier descriptors, is one of the most representative projection-based algorithms. Its basic assumption is that if two 3D shapes are similar, they also look similar from all viewpoints. Vranic *et al.* [27] define a composite shape descriptor, which is generated using depth buffer images, silhouettes, and ray-extents of a polygonal mesh. In [11], a novel descriptor called PANORAMA is proposed. It projects 3D shapes to the lateral surface of a cylinder, and describes the obtained panoramic view by 2D Discrete Fourier Transform and 2D Discrete Wavelet Transform. To ensure the rotation invariance as far as possible, Continuous PCA (CPCA) and Normals PCA (NPCA) [17] are both applied to 3D shapes before rendering the projection. Daras *et al.* [28] propose Compact Multi-view Descriptor (CMVD), where 18 characteristic views are described by 2D Polar-Fourier Transform, 2D Zernike Moments, and 2D Krawtchouk Moments.

Meanwhile, some researchers consider borrowing the development of feature learning in natural image analysis, so as to attain discriminative representations of projections. For example, Furuya *et al.* [29] introduce the Bag of visual Words (BoW) [14] to 3D shape retrieval, where local descriptors [21] are extracted on depth projections of 3D shapes and encoded into histogram feature via vector quantization. By putting the visual descriptors from different projections in one bag, Vectors of Locally Aggregated Tensors (VLAT) [16] is investigated to produce an equal-sized feature for each 3D shape. Tabia *et al.* [30], [31] firstly explore the usage of covariance matrices of descriptors, instead of the descriptors themselves, in 3D shape analysis. Bai *et al.* [32] introduce a two layer coding framework which jointly encodes a pair of views. By doing so, the spatial arrangement of multiple views is captured which is shown to be rotation-invariant.

Since deep learning has been proven to be a powerful tool in many computer vision and pattern recognition topics, there is an growing interest to leverage this popular paradigm in 3D shape community. As an extension of PANORAMA [11], Shi *et al.* [33] choose to pool the response of each row of feature map so that the deep panoramic representation remains unchanged when the 3D shape rotates with regard to its principal axis. Multi-view Convolutional Neural Networks (MVCNN) [34] sets a view pooling layer in the architecture of CNN to aggregate the multiple view representations. Note that some deep-learning-based algorithms do not learn from projections of shapes. For example, Wu *et al.* [9] perform 3D Convolution on voxel grid of shapes with Deep Belief Network. They also construct a large scale 3D shape repository called ModelNet. In [35]–[38], deep learning is applied to mid-level shape descriptors, instead of raw shape data.

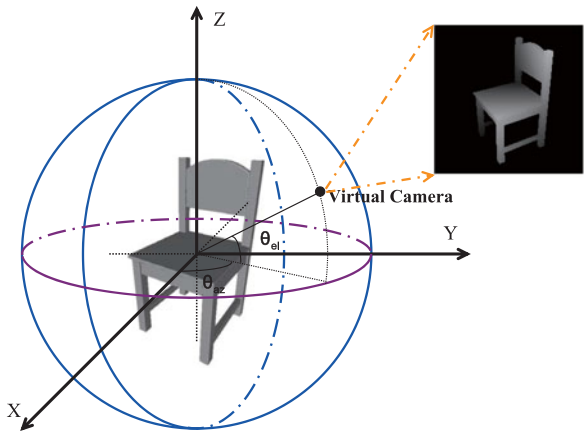


Fig. 2. Illustration of projection rendering. θ_{az} is the polar angle in xy plane and θ_{el} is the angle between the camera and xy plane.

Besides, there are also some works which focus on the optimal matching strategy (e.g., clock matching [39], vector extrapolation matching [40], random forest [41], elastic net matching [42]), discriminative view selection (e.g., adaptive views clustering [43]), feature fusion (e.g., 2D/3D Hybrid [44], Hybrid BoW [45], ZFDR [46]) and re-ranking (Multi-Feature Anchor Manifold Ranking [47], diffusion process [23]).

As opposed to the above algorithms concerning retrieval accuracy only, we establish a shape search system which attaches more importance to retrieval efficiency.

III. Proposed Search Engine

In this section, the details of each component of the proposed search engine are given.

A. Projection Rendering

Prior to projection rendering, pose normalization for each 3D shape is needed in order to attain invariance to some common geometrical transformations. However, unlike many previous algorithms [11], [17], [44] that require rotation normalization using some Principal Component Analysis (PCA) techniques, we only normalize the scale and the translation in our system. Our concerns are two-fold: 1) PCA techniques are not always stable, especially when dealing with some specific geometrical characteristics such as symmetries, large planar or bumpy surfaces; 2) the view feature used in our system can tolerate the rotation issue to a certain extent, though cannot be completely invariant to such changes. In fact, we observe that if enough projections (more than 25 in our experiments) are used, one can already achieve reliable retrieval performances.

The projection procedure is as follows. Firstly, as illustrated in Fig. 2, we place the centroid of each 3D shape at the origin of a spherical coordinate system, and resize the maximum polar distance of the points on the surface of the shape to unit length. Then, we evenly divide $[0, 2\pi]$ into 8 parts to get the values of θ_{az}, and divide $[0, \pi]$ into 8 parts to get the values of θ_{el}. For each pair $(\theta_{az}, \theta_{el})$, a virtual camera is set on the unit sphere.

At last, we render one projected view in depth buffer at each combination of θ_{az} and θ_{el}. Therefore, we will have $N_v = 64$ depth projections for each 3D shape. For the sake of speed, GPU is utilized here so that for each 3D shape, the average time cost of rendering 64 projections is only 30 ms.

B. Feature Extraction via GPU Acceleration

Feature design has been a crucial problem in 3D shape retrieval for a long time owing to its great influence on the retrieval accuracy. Though extensively studied, almost all the existing algorithms ignore the efficiency of the feature extraction.

To this end, our search engine adopts GPU to accelerate the procedure of feature extraction. Impressed by the superior performance of deep learning approaches in various visual tasks, we propose to use the activation of a Convolutional Neural Network (CNN). The CNN used here takes depth images as input. Specifically, let $\mathcal{Y} = \{y_1, y_2, \ldots, y_{N_r}\}$ denote the training data with N_r shapes. For each $y_i \in \mathcal{Y}$ with label l_i, we can obtain its projective image set $\mathcal{P}(y_i) = \{y_{i,1}, y_{i,2}, \ldots, y_{i,N_v}\}$. So, the labeled training images in N_c-th category are

$$\mathcal{P}_{N_c} = \{y_{i,j} | y_{i,j} \in \mathcal{P}(y_i), y_i \in \mathcal{Y}, l_i = N_c\}. \tag{1}$$

Equation (1) suggests that projections are assigned to the labels of their corresponding 3D shape.

The CNN architecture used in this paper is VGG-S as defined in [48], which consists of five successive convolutional layers and three fully connected layers. The last SoftMax layer produces the probability distribution over the label space. We finetune the CNN model (pretrained on ImageNet) with projections, by minimizing the classification error for $\mathcal{P}_{N_c}$ using back-propagation.

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ denote the testing database, where retrieval evaluation is performed. In the testing phase, each shape $x_p \in \mathcal{X}$ is rendered with N_v projections $\mathcal{P}(x_p) = \{x_{p,1}, x_{p,1}, \ldots, x_{p,N_v}\}$. By feeding each projection $x_{p,j} \in \mathcal{P}(x_p)$ into the trained network in the forward direction, we gain its activation with regard to the N_l-th layer of CNN as

$$p_j = \mathcal{F}(x_{p,j}, N_l) \tag{2}$$

where function $\mathcal{F}(\cdot)$ is the feature extractor associated with the trained CNN model. We normalize each activation in its Euclidean norm to avoid scale changes. It only takes 56 ms on average to extract the view features for a 3D shape.

Since no prior information is available to judge the discriminative power of activations of different layers, we propose a robust feature fusion algorithm described in Section III-D. It can fuse those homogenous features efficiently based on fuzzy set theory in the re-ranking component.

C. Inverted File for Multiview Matching

Consider a query shape x_q and a shape x_p from the database $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. Through (2), we can obtain two feature sets $\mathcal{V}(x_q) = \{q_1, q_2, \ldots, q_{N_v}\}$ and $\mathcal{V}(x_p) = \{p_1, p_2, \ldots, p_{N_v}\}$ respectively, where N_v is the number of views. q_i (or p_i) denotes the view feature assigned to the i-th view of shape x_q (or x_p).

A 3D shape search engine requires a multi-view matching component to establish a correspondence between two sets of view features. These matching strategies are usually metrics defined on sets (e.g., Hausdorff distance) or graph matching algorithms (e.g., Hungarian method, Dynamic Programming, clock-matching). However, these pairwise strategies are time-consuming for a real-time search engine. Among them, Hausdorff distance may be the most efficient one, since it only involves some basic algebraic operations without sophisticated optimizations.

Recall that the standard Hausdorff distance measures the difference between two sets, and it is defined as

$$D(x_q, x_p) = \max_{q_i \in \mathcal{V}(x_q)} \min_{p_j \in \mathcal{V}(x_p)} d(q_i, p_j) \tag{3}$$

where function $d(\cdot)$ measures the distance between two input vectors. In order to eliminate the disturbance of isolated views in the query view set, a more robust version of Hausdorff distance is given by

$$D(x_q, x_p) = \frac{1}{N_v} \sum_{q_i \in \mathcal{V}(x_q)} \min_{p_j \in \mathcal{V}(x_p)} d(q_i, p_j). \tag{4}$$

For the convenience of analysis, we consider its dual form in the similarity space as

$$S(x_q, x_p) = \frac{1}{N_v} \sum_{q_i \in \mathcal{V}(x_q)} \max_{p_j \in \mathcal{V}(x_p)} s(q_i, p_j) \tag{5}$$

where $s(\cdot)$ measures the similarity between the two input vectors. In this paper, we adopt the cosine similarity.

As can be seen from (4) and (5), Hausdorff matching requires the time complexity $O(N \times {N_v}^2)$ for retrieving a given query (assuming that there are N shapes in the database). Though the complexity grows linearly with respect to the database size, it is still intolerable when N gets larger. However, by analyzing (5), we can make several observations: 1) let $s^*(q_i) = \max_{1 \le j \le N_v} s(q_i, p_j)$, the similarity calculations of $s(q_i, p_j)$ are unnecessary when $s(q_i, p_j) < s^*(q_i)$, since these similarity values are unused due to the max operation, i.e., only $s^*(q_i)$ is kept; 2) when considering from the query side, we can find that $s^*(q_i)$ counts little to the final matching cost if $s^*(q_i) < \xi$ and ξ is a small threshold. Those observations suggest that although the matching function in (5) requires calculating all the pairwise similarities between two view sets, some similarity calculations, which generate small values, can be eliminated without impairing the retrieval performance too much.

In order to avoid these unnecessary operations and improve the efficiency of multi-view matching procedure, we adopt inverted file for an approximation by adding the Kronecker delta response

$$\delta_{x,y} = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases} \tag{6}$$

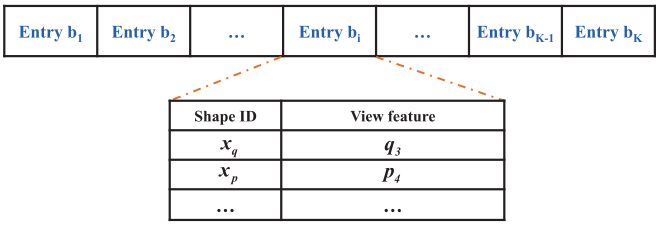


Fig. 3. Structure of the first inverted file.

into (5) as

$$S(x_q, x_p) = \frac{1}{N_v} \sum_{q_i \in \mathcal{V}(x_q)} \max_{p_j \in \mathcal{V}(x_p)} s(q_i, p_j) \cdot \delta_{c(q_i), c(p_j)} \quad (7)$$

where the quantizer $c(x) = \arg\min_{1 \le i \le K} \|x - b_i\|^2$ maps the input feature into an integer index that corresponds to the nearest codeword of the given vocabulary $B = \{b_1, b_2, \ldots, b_K\}$. As a result, the contribution of p_j to the similarity measure, which satisfies $c(q_i) \neq c(p_j)$, can be directly set to zero, without estimating $s(q_i, p_j)$ explicitly.

In conclusion, our inverted file for multi-view matching is built as illustrated in Fig. 3. For each view feature, we store it and its corresponding shape ID in the nearest codeword. It should be mentioned that we can also use Multiple Assignment (MA), i.e., assigning each view to multiple codewords, to improve the matching precision at the sacrifice of memory cost and on-line query time. In this scenario, the definition of δ is not changed and each view q_i will have more than one assignment $c(q_i)$.

D. Inverted File for Reranking

A typical search engine usually involves a re-ranking component [49], aiming at refining the initial candidate list by using some contextual information. In GIFT, we present a re-ranking algorithm called Aggregated Contextual Activation (ACA), which can integrate activations of multiple layers of neural network. It follows the same principles as diffusion process [23], [50], i.e., the similarity between two shapes should go beyond their pairwise formulation and is influenced by their contextual distributions along the underlying data manifold. However, different from diffusion process which has high time complexity, ACA enables real-time re-ranking, which can be potentially applied to large scale data.

Let $\mathcal{N}_k(x_q)$ denote the neighbor set of x_q. The elements in $\mathcal{N}_k(x_q)$ are determined by k-nearest neighbors (kNN) rule, i.e., we select those $x_p \in \mathcal{X}$, which have the top-k largest similarity values to x_q computed using (7). Similar to [51], [52], our basic idea is that the similarity between two shapes can be more reliably measured by comparing their neighbors using Jaccard similarity as

$$S'(x_q, x_p) = \frac{|\mathcal{N}_k(x_q) \cap \mathcal{N}_k(x_p)|}{|\mathcal{N}_k(x_q) \cup \mathcal{N}_k(x_p)|}. \quad (8)$$

One can find that the neighbors are treated equally in (8). However, the top-ranked neighbors are more likely to be true positives. So a more proper behavior is increasing the weights of top-ranked neighbors.

To achieve this, we define the neighbor set using fuzzy set theory. Different from classical (crisp) set theory where each element either belongs or does not belong to the set, fuzzy set theory allows a gradual assessment of the membership of elements in a set. We utilize $S(x_q, x_i)$ to measure the membership grade of x_i in the neighbor set of x_q. Accordingly, (8) is rewritten as

$$S'(x_q, x_p) = \frac{\sum_{x_i \in \mathcal{N}_k(x_q) \cap \mathcal{N}_k(x_p)} \min\left(S(x_q, x_i), S(x_p, x_i)\right)}{\sum_{x_i \in \mathcal{N}_k(x_q) \bigcup \mathcal{N}_k(x_p)} \max\left(S(x_q, x_i), S(x_p, x_i)\right)}. \quad (9)$$

Since considering equal-sized vector comparison is more convenient in real computational applications, we use $F \in \mathbb{R}^N$ to encode the membership values. The i-th element in F_q is given as

$$F_q[i] = \begin{cases} S(x_q, x_i) & \text{if } x_i \in \mathcal{N}_k(x_q) \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Based on this definition we replace (9) with

$$S'(x_q, x_p) = \frac{\sum_{i=1}^{N} \min\left(F_q[i], F_p[i]\right)}{\sum_{i=1}^{N} \max\left(F_q[i], F_p[i]\right)}. \quad (11)$$

Considering vector F_q is sparse, we can view it as sparse activation of shape x_q, where the activation at coordinate i is the membership grade of x_i in the neighbor set $\mathcal{N}_k(x_q)$. Equation (11) utilizes the sparse activations F_q and F_p to define the new contextual shape similarity measure.

Note that all the above analysis is carried out for only one similarity measure. However, in our specific scenario, the outputs of different layers of CNN are usually at different abstraction resolutions. That is, through (2), we can have multiple representations for each projective image of 3D shapes by selecting different layers N_l. Two different layers of CNN lead to two different similarities $S^{(1)}$ and $S^{(2)}$ by (7), which in turn yield two different sparse activations $F_q^{(1)}$ and $F_q^{(2)}$ by (10). Since no prior information is available to assess their discriminative power, our goal now is to fuse them in an unsupervised manner. To achieve this, we utilize the *fuzzy aggregation operator* in fuzzy set theory, by which several fuzzy sets are combined in a desirable way to produce a single fuzzy set. We consider two fuzzy sets represented by the sparse activations $F_q^{(1)}$ and $F_q^{(2)}$ (the extension to more than two activations is similar). Their aggregation is then defined as

$$F_q = \left(\frac{(F_q^{(1)})^\alpha + (F_q^{(2)})^\alpha}{2} \right)^{\frac{1}{\alpha}} \quad (12)$$

which computes the element-wise generalized means with exponent α of $F_q^{(1)}$ and $F_q^{(2)}$. Instead of using arithmetic mean, we use this generalized means (α is set to 0.5 throughout our experiments). Our concern for this is to avoid the problem that some artificially large elements in F_q dominate the similarity measure. This motivation is very similar to handling bursty visual elements in Bag-of-Words (BoW) model (see [53] for examples).

In summary, we call the feature in (12) Aggregated Contextual Activation (ACA). Next, we will introduce some improvements of (12) concerning its retrieval accuracy and computational efficiency.

1) Improving Accuracy: Similar to diffusion process, the proposed ACA requires an accurate estimation of the context in the data manifold. Here we provide three alternative ways to improve the retrieval performance of ACA without depriving its efficiency.

Neighbor augmentation: The first one is to augment F_q using the neighbors of the second order, i.e., the neighbors of the neighbors of x_q. Inspired by query expansion [11], the second order neighbors are added as

$$F_q^{(l)} := \frac{1}{|\mathcal{N}_k^{(l)}(x_q)|} \sum_{x_i \in \mathcal{N}_k^{(l)}(x_q)} F_i^{(l)}. \tag{13}$$

Neighbor co-augmentation: Our second improvement is to use a so-called "neighbor co-augmentation". Specifically, the neighbors generated by one similarity measure are used to augment contextual activations of the other similarity measure, formally defined as

$$\begin{aligned} F_q^{(1)} &:= \frac{1}{|\mathcal{N}_k^{(2)}(x_q)|} \sum_{x_i \in \mathcal{N}_k^{(2)}(x_q)} F_i^{(1)} \\ F_q^{(2)} &:= \frac{1}{|\mathcal{N}_k^{(1)}(x_q)|} \sum_{x_i \in \mathcal{N}_k^{(1)}(x_q)} F_i^{(2)}. \end{aligned} \tag{14}$$

This formula is inspired by "co-training" [54]. Essentially, one similarity measure tells the other one that "I think these neighbors to be true positives, and lend them to you such that you can improve your own discriminative power".

Neighbor multi-augmentation: In case that more than two similarity measures are accessible, neighbor co-augmentation cannot be directly used. To make the proposed system suitable to tackle M ($M > 2$) similarity measures, we propose to use neighbor multi-augmentation. $F_q^{(j)}$, the j-th ($1 \le j \le M$) activation of shape x_q, is augmented as

$$F_q^{(j)} := \frac{1}{|\mathcal{N}_k^{\Delta}(x_q)|} \sum_{x_i \in \mathcal{N}_k^{\Delta}(x_q)} F_i^{(j)} \tag{15}$$

where

$$\mathcal{N}_k^{\Delta}(x_q) = \bigcup_{j'=1, j' \neq j}^{M} \mathcal{N}_k^{(j')}(x_q). \tag{16}$$

Equation (16) suggests that the union of the other $M-1$ neighbor set is used to augment $F_q^{(j)}$.

Note that the size of neighbor set used here may be different from that used in (10). In order to distinguish them, we denote the size of neighbor set in (10) as k_1, while that used in (13), (14) and (15) as k_2.

2) Improving Efficiency: Considering that the length of F_q is N, one may doubt the efficiency of similarity computation in (11), especially when the database size N is large. In fact, F_q is a sparse vector, since F_q only encodes the neighborhood structure of x_q. This observation motivates us to utilize an inverted file again to leverage the sparsity of F_q. Now we derive the feasibility of applying inverted file in Jaccard similarity theoretically.

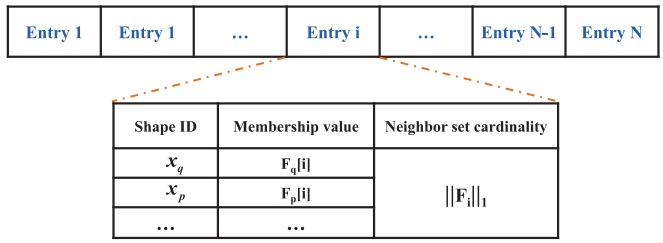


Fig. 4. Structure of the second inverted file.

The numerator in (11) is computed as

$$\begin{aligned} \sum_i \min\left(F_q[i], F_p[i]\right) &= \sum_{i|F_q[i]\neq 0, F_p[i]\neq 0} \min(F_q[i], F_p[i]) \\ &+ \sum_{i|F_q[i]=0} \min(F_q[i], F_p[i]) + \sum_{i|F_p[i]=0} \min(F_q[i], F_p[i]). \end{aligned} \tag{17}$$

Since all values of the aggregated contextual activation are non-negative, the last two items in (17) are equal to zero. Consequently, (17) can be simplified as

$$\sum_i \min\left(F_q[i], F_p[i]\right) = \sum_{i|F_q[i]\neq 0, F_p[i]\neq 0} \min(F_q[i], F_p[i]) \tag{18}$$

which only requires accessing non-zero entries of the query, and hence can be computed efficiently on-the-fly.

Although the calculation of the denominator in (11) seems sophisticated, it can be expressed as

$$\begin{aligned} &\sum_i \max\left(F_q[i], F_p[i]\right) \\ &= \|F_q\|_1 + \|F_p\|_1 - \sum_i \min\left(F_q[i], F_p[i]\right) \\ &= \|F_q\|_1 + \|F_p\|_1 - \sum_{i|F_q[i]\neq 0, F_p[i]\neq 0} \min(F_q[i], F_p[i]). \end{aligned} \tag{19}$$

Besides the query-dependent operations (the first and the last items), (19) only involves an operation of L_1 norm calculation of F_p, which is simply equal to the cardinality of the fuzzy set $\mathcal{N}_k(x_p)$ and can be pre-computed off-line.

Our inverted file for re-ranking is built as illustrated in Fig. 4. It has exactly N entries, and each entry corresponds to one shape in the database. For each entry, we first store the cardinality of its fuzzy neighbor set. Then, we find those shapes which have non-negative membership values in this entry. Those shape IDs and the membership values are stored in this entry.

IV. Experiments

In this section, we evaluate the performances of GIFT on various shape benchmarks and competitions, including ModelNet40 dataset [9], ModelNet10 dataset [9], SHape REtrieval

Contest 2014 Large Scale Comprehensive Track Benchmark (SHREC14LSGTB) [1], ShapeNet Core55 dataset [3], Princeton Shape Benchmark (PSB), Watertight Models track of SHape REtrieval Contest 2007 (WM-SHREC07) dataset [55] and McGill dataset [56].

If not specified, we adopt the following setup throughout our experiments. The projection rendered for each shape is $N_v = 64$. In multi-view matching procedure, the approximate Hausdorff matching defined in (7) with an inverted file of 256 entries is used. Multiple Assignment is set to 2. We use two pairwise similarity measures, which are calculated using activations from $N_l = 5$ layer (denoted as L_5 below) and $N_l = 7$ layer (denoted as L_7 below), respectively. In re-ranking component, each similarity measure generates one sparse activation F_q to capture the contextual information for the 3D shape x_q, and neighbor co-augmentation in (14) is used to produce $F_q^{(1)}$ and $F_q^{(2)}$. Finally, both $F_q^{(1)}$ and $F_q^{(2)}$ are integrated by (12) with exponent $\alpha = 0.5$. All the experiments are done on a server with an Intel (R) Xeon (R) CPU (3.50 GHz), 64GB RAM memory and 4 GTX NVIDIA TITAN X.

To quantify the retrieval performance, the following evaluation metrics are employed:

1) *Mean Average Precision (MAP):* The average precision where a positive shape is returned.
2) *Area Under Curve (AUC):* The mean area under the precision-recall curves.
3) *Nearest Neighbor (NN):* The percentage of the closest matches that belongs to the same class as the query.
4) *First Tier (FT):* The recall for the top $(C-1)$ matches in the ranking list, where C is the number of shapes in the category which contains the query shape.
5) *Second Tier (ST):* The recall for the top $2 \times (C-1)$ matches in the ranking list, where C is the number of shapes in the category which contains the query shape.
6) *F-measure:* The harmonic mean of precision and recall.
7) *Discounted Cumulative Gain (DCG):* A statistic that attaches more importance to the correct results near the front of the ranked list than the correct results at the end of the ranked list, under the assumption that a user is less likely to consider elements near the end of the list.

All the aforementioned evaluation metrics range from 0 to 1, and larger values indicate better performances. We refer to [3], [9], [57] for their detailed definitions.

Note that different 3D shape datasets and competitions favor different evaluation metrics, and we will follow their convention in our experiments. For example, we use MAP and AUC on ModelNet 40 dataset and Modelnet 10 dataset, use MAP and F-measure on ShapeNet CORE55 dataset. NN, FT and ST are used on the SHREC14LSGTB, PSB dataset, WM-SHREC07 and McGill dataset. Moreover, we also leverage Precision-Recall curves and confusion matrix to visualize the performances of the proposed GIFT against other state-of-the-art algorithms.

A. ModelNet

ModelNet is a large-scale 3D CAD model dataset introduced by Wu *et al.* [9] recently, which contains 151,128 3D CAD models divided into 660 object categories. Two subsets are used for evaluation, i.e., ModelNet40 and ModelNet10. The former one contains 12,311 models, and the latter one contains 4,899 models. We evaluate the performance of GIFT on both subsets and adopt the same training and test split as in [9], namely randomly selecting 100 unique models per category from the subset, in which 80 models are used for training the CNN model and the rest for testing the retrieval performance.

TABLE I
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART ON MODELNET40 DATASET AND MODELNET10 DATASET

Methods	ModelNet40		ModelNet10	
	AUC	MAP	AUC	MAP
SPH [58]	34.47%	33.26%	45.97%	44.05%
LFD [15]	42.04%	40.91%	51.70%	49.82%
PANORAMA [11]	45.00%	46.13%	60.72%	60.32%
ShapeNets [9]	49.94%	49.23%	69.28%	68.26%
DeepPano [33]	77.63%	76.81%	85.45%	84.18%
MVCNN [34]	–	78.90%	–	–
L_5	63.70%	63.07%	78.19%	77.25%
L_7	77.28%	76.63%	89.03%	88.05%
GIFT	**83.10%**	**81.94%**	**92.35%**	**91.12%**

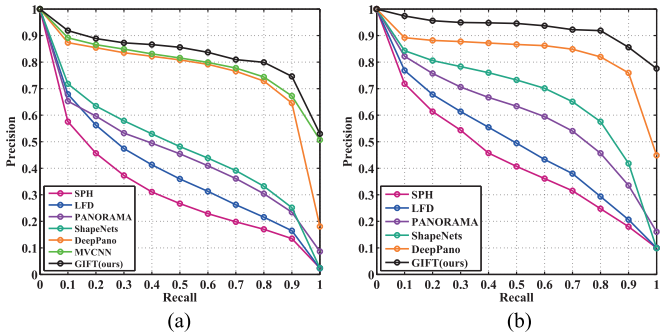


Fig. 5. Precision-recall curves on (a) ModelNet40 dataset and (b) ModelNet10 dataset.

For comparison, we collect all the retrieval results publicly available. The chosen methods are (Spherical Harmonic) SPH [58], Light Field descriptor (LFD) [15], PANORAMA [11], 3D ShapeNet [9], DeepPano [33] and MVCNN [34]. As Table I shows, GIFT outperforms all the state-of-the-art methods remarkably on both evaluation metrics. We also present the performances of two baseline methods, i.e., feature L_5 or L_7 with exact Hausdorff matching. As can be seen, L_7 achieves a better performance than L_5, and GIFT leads to a significant improvement over L_7 of 5.82% in AUC, 5.31% in MAP on ModelNet40 dataset, and 3.32% in AUC, 3.07% in MAP on ModelNet10 dataset. Fig. 5 compares the precision-recall curves. It demonstrates again the discriminative power of the proposed search engine in 3D shape retrieval.

ModelNet also defines two 3D shape classification tracks, where classifiers learned in a supervised way can be used. Although GIFT is initially developed for real-time retrieval, GIFT can be also applied to 3D shape classification with some modifications. We follow the default parameter setup as

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY ON MODELNET40 DATASET AND MODELNET10 DATASET

Methods	ModelNet40	ModelNet10
SPH [58]	68.23%	79.79%
LFD [15]	75.47%	79.87%
ShapeNets [9]	77.32%	83.54%
DeepPano [33]	82.54%	88.66%
Voxnet [60]	83.00%	92.00%
3D-GAN [61]	83.30%	91.00%
MVCNN [34]	90.10%	–
Pairwise [59]	**90.70%**	**92.40%**
L_5	89.50%	91.50%
L_7	87.38%	91.50%

introduced before, except that re-ranking stage is not used since it is not proper to deal with supervised classification. Instead, we utilize a linear SVM classifier learned on training shapes to predict the labels of test shapes.

The quantitative comparison of classification accuracy is given in Table II. As the table presents, the best performance is achieved by Pairwise [59], which is 90.70% on ModelNet40 dataset and 92.40% on ModelNet10 dataset, respectively. GIFT also achieves competitive performances with Pairwise [59] on both datasets. The confusion matrices generated by our system on ModelNet40 dataset and ModelNet10 dataset are given in Fig. 6.

B. SHREC14LSGTB

As the most authoritative 3D retrieval competition held each year, SHape REtrieval Contest (SHREC) pays much attention to the development of scalable algorithms gradually. Especially in recent years, several large scale tracks, such as SHREC14LSGTB [1], are organized to test the scalability of algorithms. However, most algorithms that the participants submit are of high time complexity, and cannot be applied when the dataset becomes larger (millions or more). Here we choose SHREC14LSGTB dataset for a comprehensive evaluation. This dataset contains 8,987 3D models classified into 171 classes, and each 3D shape is taken in turn as the query. As for the feature extractor, we collected 54,728 unrelated models from ModelNet [9] divided into 461 categories to train a CNN model.

To keep the comparison fair, we choose two types of results from the survey paper [1] to present in Table III. The first type consists of the top-5 best-performing methods on retrieval accuracy, including PANORAMA [11], DBSVC, MR-BF-DSIFT, MR-D1SIFT and LCDR-DBSVC. The second type is the most efficient one, i.e., ZFDR [46].

As can be seen from the table, excluding GIFT, the best performance is achieved by LCDR-DBSVC. However, it requires 668.6 s to return the retrieval results per query, which means that 69 days are needed to finish the query task on the whole dataset. The reason behind such a high complexity lies in two aspects: 1) its visual feature is $270K$ dimensional, which is time-consuming to compute, store and compare; 2) it adopts locally constrained diffusion process (LCDP) [24] for re-ranking, while it is known that LCDP is an iterative graph-based algorithm of high time complexity. As for ZFDR, its average query time is shortened to 1.77 s by computing parallel on 12 cores. Unfortunately, ZFDR achieves much less accurate retrieval performance, and its FT is 13% smaller than LCDR-DBSVC. In summary, a conclusion can be drawn that no method can achieve a good enough performance at a low time complexity.

By contrast, GIFT outperforms all these methods and sets a new state-of-the-art performance on this challenging competition. More importantly, GIFT can provide the retrieval results within 63.14 ms, which is 4 orders of magnitude faster than LCDR-DBSVC. Meanwhile, the two baseline methods L_5 and L_7 incur heavy query cost due to the usage of exact Hausdorff matching, which testifies the advantage of the proposed F-IF.

We also compare the results of GIFT to other recent algorithms, Two Layer Coding (TLC) [32] and Covariance [31]. TLC reports NN 0.879, FT 0.456 and ST 0.585. Tabia *et al.* extends the covariance descriptor [30] to its spatially-sensitive version, and report NN 0.775, FT 0.460 and ST 0.501. Those results are still inferior to GIFT.

C. ShapeNet Core55

ShapeNet Core55 [3] from SHape REtrieval Contest (SHREC) 2016 is the latest competition track about the scalability of 3D shape retrieval. It consists of 51,190 3D shapes categorized into 55 categories and 204 sub-categories. Each shape is assigned to a category label indicating a coarse division, and a sub-category label indicating a fine division. To make the supervised training possible, 70% (35,765) shapes serve as training data, and 10% (5,159) shapes serve as validation data. The rest 20% (10,266) shapes are testing data. All the metadata has two versions. On the "normal" dataset, all the shapes are upright and front orientated. On the "perturbed" dataset, all the shapes are randomly rotated.

Since each shape has two groundtruth labels, the competition organizers also define a modified version of normalized discounted cumulative gain (NDCG) for evaluation. Specially, NDCG defined here uses the following graded relevance: 3 for perfect category and subcategory match in query and returned shape, 2 for category and subcategory both being same as the category, 1 for correct category but a sibling subcategory, and 0 for no match. Meanwhile, since the number of shapes in different categories is not the same, all the used evaluation metrics (MAP, F-measure and NDCG) will have two versions. Macro-averaged version is used to give an unweighted average over the entire dataset. Micro-averaged version is used to adjust for model category sizes giving a representative performance metric averaged across categories. To ensure a straightforward comparison, the arithmetic mean of macro version and micro version is also used.

Table IV presents the performance comparison on the normal dataset, and Table V presents the performance comparison on the perturbed dataset. As the two tables show, our method wins the 2nd place on the normal dataset and 1st place on the perturbed dataset.

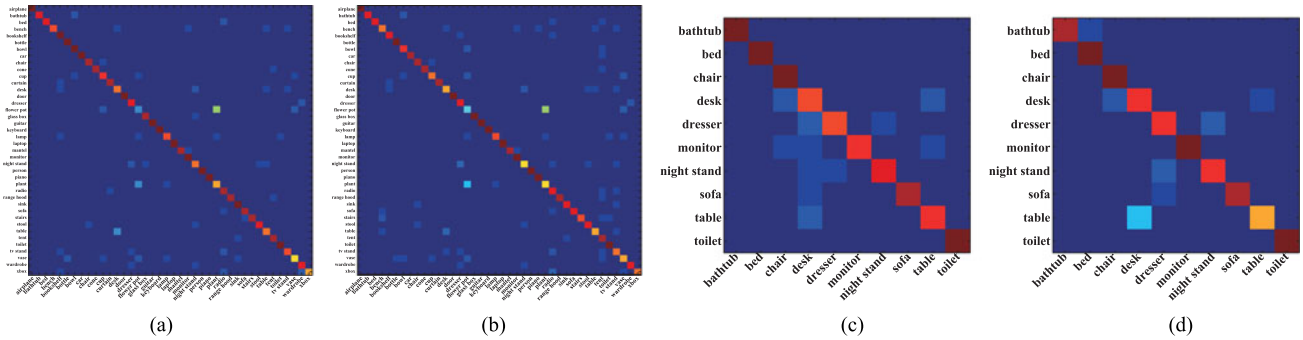


Fig. 6. Confusion matrices of GIFT on (a), (b) ModelNet40 dataset and (c), (d) Modelnet10 dataset. The features used are (a), (c) L_5 and (b), (d) L_7, respectively.

TABLE III
PERFORMANCE COMPARISON ON SHREC14LSGTB

Methods	Accuracy			Query time
	NN	FT	ST	
ZFDR	0.879	0.398	0.535	1.77 s
PANORAMA	0.859	0.436	0.560	370.2 s
DBSVC	0.868	0.438	0.563	62.66 s
MR-BF-DSIFT	0.845	0.455	0.567	65.17 s
MR-D1SIFT	0.856	0.465	0.578	131.04 s
LCDR-DBSVC	0.864	0.528	0.661	668.6 s
L_5	0.879	0.460	0.592	22.73 s
L_7	0.884	0.507	0.642	4.82 s
GIFT	**0.889**	**0.567**	**0.689**	**63.14 ms**

D. Generic 3D Retrieval

Following [30], [31], we select three popular datasets for a generic evaluation, including PSB dataset [57], WM-SHREC07 [55] and McGill dataset [56]. Among them, PSB dataset is probably the first widely-used generic shape benchmark, and it consists of 907 polygonal models divided into 92 categories. WM-SHREC07 contains 400 watertight models evenly distributed in 20 classes, and is a representative competition held by SHREC community. McGill dataset focuses on non-rigid analysis, and contains 255 articulated objects classified into 10 classes. We train CNN on an independent TSB dataset [63], and then use the trained CNN to extract view features for the shapes on all the three testing datasets.

In Table VI, a comprehensive comparison between GIFT and various state-of-the-art methods is presented, including LFD [15], the curve-based method of Tabia *et al.* [64], DESIRE descriptor [27], total Bregman Divergences (tBD) [65], Covariance descriptor [30], the Hybrid of 2D and 3D descriptor [44], Two Layer Coding (TLC) [32] and PANORAMA [11]. As can be seen, GIFT exhibits encouraging discriminative ability and achieves state-of-the-art performances consistently in all the three evaluation metrics.

E. Execution Time

In addition to state-of-the-art performances on several datasets and competitions, the most important property of GIFT is the "real-time" performance with the potential of handling large scale shape corpora. In Table VII, we give a deeper analysis of the time cost. The off-line operations mainly include projection rendering and feature extraction for database shapes, training CNN, and building two inverted files. As the table shows, the time cost of off-line operations varies significantly for different datasets. Among them, the most time-consuming operation is training CNN, followed by building the first inverted file with k-means. However, the average query time on different datasets can be controlled within one second, even for the biggest SHREC14LSGTB dataset.

F. Qualitative Evaluation

Typical retrieval results on PSB dataset are presented in Fig. 7. Three query shapes are used. For each query, we present the ranking lists of the baselines L_5 and L_7 in the first two rows, and those of GIFT in the last row.

First, we observe that although L_7 yields better overall performances than L_5 as the previous experiments show, it is not the case when we consider an individual query shape. For instance, L_5 is more discriminative for the query shape "flying saucer". Second, some works (e.g., [67]) suggest that middle level (e.g., L_5) activations are more suitable to capture low-level patterns, while high level (e.g., L_7) activations carry more information about semantic attributes. For example, when L_5 is used for indexing the "bird" query, it tends to return "airplanes", since both birds and airplanes have similar wings. Therefore, one can clearly find the complementary nature between L_5 and L_7, and the feature fusion mechanism in GIFT makes full use of such complementarity to produce more reliable retrieval results.

At last, we also illustrate a case (the query is "bench seat"), where both L_5 and L_7 lead to unsatisfactory results. Both of them return many false positives from dining chair or desk chair, two sibling sub-categories of bench seat. In this case, the context information seems to be unreliable for context-based re-ranking. However, we also observe the robustness of the re-ranking component. This is because two shapes from the same category are expected to have more common neighbors, but the neighbors themselves do not necessarily come from the category of the query shape. Even if those neighbors are false positives, they can still be helpful in describing the contextual distribution of the query shape to a certain extent.

TABLE IV
PERFORMANCE COMPARISON ON SHAPENET CORE55 NORMAL DATASET

Methods	Micro			Macro			Micro + Macro		
	F-measure	MAP	NDCG	F-measure	MAP	NDCG	F-measure	MAP	NDCG
Tatsuma *et al.* [62]	0.472	0.728	0.875	0.203	0.596	0.806	0.338	0.662	0.841
Wang *et al.*	0.391	0.823	0.886	0.286	0.661	0.820	0.338	0.742	0.853
Li *et al.*	0.582	0.829	**0.904**	0.201	0.711	0.846	0.392	0.770	0.875
MVCNN [34]	**0.764**	**0.873**	0.899	**0.575**	**0.817**	**0.880**	**0.669**	**0.845**	**0.890**
GIFT	0.689	0.825	0.896	0.454	0.740	0.850	0.572	0.783	0.873

TABLE V
PERFORMANCE COMPARISON ON SHAPENET CORE55 PERTURBED DATASET

Methods	Micro			Macro			Micro + Macro		
	F-measure	MAP	NDCG	F-measure	MAP	NDCG	F-measure	MAP	NDCG
Tatsuma *et al.* [62]	0.413	0.638	0.838	0.166	0.493	0.743	0.290	0.566	0.791
Wang *et al.*	0.246	0.600	0.776	0.163	0.478	0.695	0.205	0.539	0.736
Li *et al.*	0.534	0.749	0.865	0.182	0.579	0.767	0.358	0.664	0.816
MVCNN [34]	0.612	0.734	0.843	0.416	0.662	0.793	0.514	0.698	0.818
GIFT	**0.661**	**0.811**	**0.889**	**0.423**	**0.730**	**0.843**	**0.542**	**0.770**	**0.866**

TABLE VI
PERFORMANCE COMPARISON ON PSB DATASET, WM-SHREC07 COMPETITION, AND MCGILL DATASET

Methods	PSB dataset			WM-SHREC07 competition			McGill dataset		
	NN	FT	ST	NN	FT	ST	NN	FT	ST
LFD [15]	0.657	0.380	0.487	0.923	0.526	0.662	–	–	–
Tabia *et al.* [64]	–	–	–	0.853	0.527	0.639	–	–	–
DESIRE [27]	0.665	0.403	0.512	0.917	0.535	0.673	–	–	–
tBD [65]	0.723	–	–	–	–	–	–	–	–
Covariance [30]	–	–	–	0.930	0.623	0.737	0.977	0.732	0.818
2D/3D Hybrid [44]	0.742	0.473	0.606	0.955	0.642	0.773	0.925	0.557	0.698
PANORAMA [11]	0.753	0.479	0.603	0.957	0.673	0.784	0.929	0.589	0.732
Shape Vocabulary [66]	0.717	0.484	0.609	–	–	–	–	–	–
PANORAMA + LRF [11]	0.752	0.531	0.659	0.957	0.743	0.839	0.910	0.693	0.812
TLC [32]	0.763	0.562	0.705	0.988	0.831	0.935	0.980	0.807	0.933
L_5	0.849	0.588	0.721	0.980	0.777	0.877	0.984	0.747	0.881
L_7	0.837	0.653	0.784	0.980	0.805	0.898	0.980	0.763	0.897
GIFT	**0.849**	**0.712**	**0.830**	**0.990**	**0.949**	**0.990**	**0.984**	**0.905**	**0.973**

TABLE VII
TIME COST ANALYSIS OF GIFT

Datasets	Off-line	On-line Indexing
ModelNet40	≈ 0.7 h	27.02 ms
ModelNet10	≈ 0.3 h	10.25 ms
SHREC14LSGTB	≈ 8.5 h	63.14 ms
PSB		16.25 ms
WMSHREC07	≈ 1.8 h	16.05 ms
McGill		9.38 ms

G. Combination With Handcrafted Features

As illustrated above, one can easily observe the complementary nature between activations from convolutional layer L_5 and fully-connected layer L_7. This phenomenon also inspires us to fuse more complementary features to enable more accurate retrieval. In this subsection, we consider a fusion of GIFT and handcrafted features for an additional evaluation.

Following the pipeline in [16], we extract SIFT [21] descriptors on Hessian-affine interest points of depth projections. Then for each shape, we put the SIFT descriptors of all its projections into one bag, and encode them via Vector of Locally Aggregated Descriptors (VLAD) [68], [69] with improvements RootSIFT [70] and power normalization [71]. The codebook size is 1,024. The Euclidean distance between two VLADs is essentially based on the matching kernel [72] between two sets of SIFT descriptors. Hence, compared with deep features used in the original GIFT, VLAD representation is more robust to rotation, non-rigid deformation, part missing, etc.

In Table VIII, we present the performances of VLAD, the original GIFT [26] and a fusion of them through Neighbor

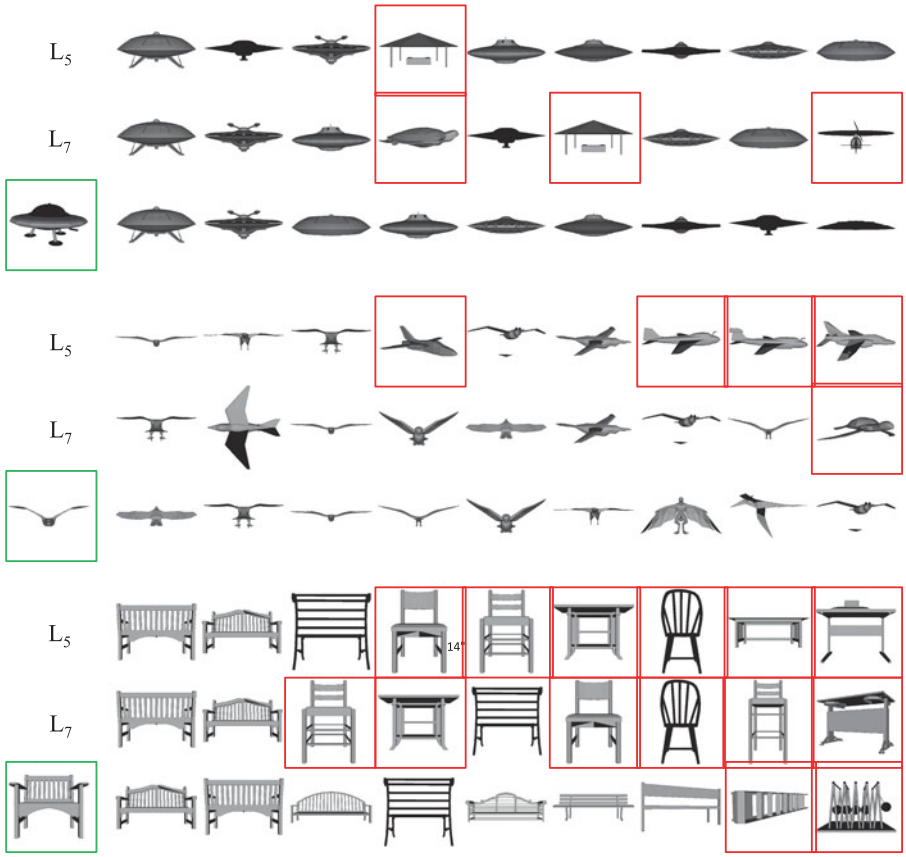


Fig. 7. Query shapes are in green boxes, and the returned false positives are in red boxes. For each query, we present the ranking lists of the baselines L_5 and L_7 in the first two rows. The results of GIFT are in the last row.

TABLE VIII
PERFORMANCES OF COMBINING GIFT WITH HANDCRAFTED FEATURES

Dataset	Metric	VLAD	GIFT [26]	GIFT+VLAD
ModelNet40	AUC	55.13%	83.10%	**85.49%**
	MAP	54.52%	81.94%	**84.54%**
ModelNet10	AUC	62.50%	92.35%	**93.44%**
	MAP	61.57%	91.12%	**92.59%**
SHREC14LSGTB	NN	0.873	**0.889**	0.874
	FT	0.435	**0.567**	0.566
	ST	0.563	0.689	**0.698**
PSB	NN	0.804	**0.849**	**0.849**
	FT	0.575	0.712	**0.723**
	ST	0.715	0.830	**0.834**
WM-SHREC07	NN	0.978	**0.990**	0.988
	FT	0.801	0.949	**0.960**
	ST	0.906	0.990	**0.994**
McGill	NN	**0.992**	0.984	0.984
	FT	0.789	0.905	**0.932**
	ST	0.922	0.973	**0.984**

Multi-augmentation defined in (15). As can be seen from the table, the retrieval performances are further improved on all the datasets, which testifies the complementarity between handcrafted features and deep-learned features. We draw the reader's attention that the distance calculation between VLADs is of great computational cost since VLAD representation is generally compact and high-dimensional (128×1024 in our implementation). Though this problem can be solved using approximation nearest neighbor (ANN) search (e.g., [73]), most of those techniques lead to considerable performance drop in real retrieval system. Therefore, we omit those techniques in this experiment.

H. Parameter Discussion

All the discussions are conducted on PSB dataset, if not specified otherwise. Improvements over Baseline. In Table IX, a thorough discussion is given about the influence of various components of GIFT. We can observe a consistent performance boost by those improvements. The performance jumps a lot especially when the re-ranking component is embedded. A linear combination of L_5 and L_7 with equal weights only achieves FT 0.657, which demonstrates that GIFT provides a better way for multiple feature fusion in the re-ranking level. One should also note a slight performance decrease when approximate Hausdorff matching with F-IF is used compared with its exact version. However, as discussed below, the embedding with inverted file does not necessarily result in a poorer performance, but shortens the query time significantly.

Discussion on F-IF: In Fig. 8, we plot the retrieval performance and the average query time using feature L_7, as the

TABLE IX
PERFORMANCE IMPROVEMENTS BROUGHT BY VARIOUS COMPONENTS IN GIFT OVER BASELINE. IN COLUMN "HAUSDORFF", $\surd$ DENOTES APPROXIMATE HAUSDORFF MATCHING IN (7), WHILE $\times$ DENOTES EXACT MATCHING IN (5). COLUMN "α" PRESENTS THE VALUE OF EXPONENT IN (12). COLUMN "NA" DESCRIBES THE PROCEDURE OF NEIGHBOR AUGMENTATION IN SECTION III-D.1: $\surd$ IS ASSOCIATED WITH (14) AND $\times$ IS ASSOCIATED WITH (13). THE BLANKS MEAN THAT THIS IMPROVEMENT IS NOT USED

Feature	Hausdorff	Re-ranking		First Tier
		α	NA	
L_5	$\times$			0.588
L_7	$\times$			0.653
$L_5 + L_7$	$\times$			0.657
$L_5 + L_7$	$\times$	1		0.688
$L_5 + L_7$	$\times$	0.5		0.692
$L_5 + L_7$	$\times$	0.5	$\times$	0.710
$L_5 + L_7$	$\times$	0.5	$\surd$	0.717
$L_5 + L_7$	$\surd$	0.5	$\surd$	0.712

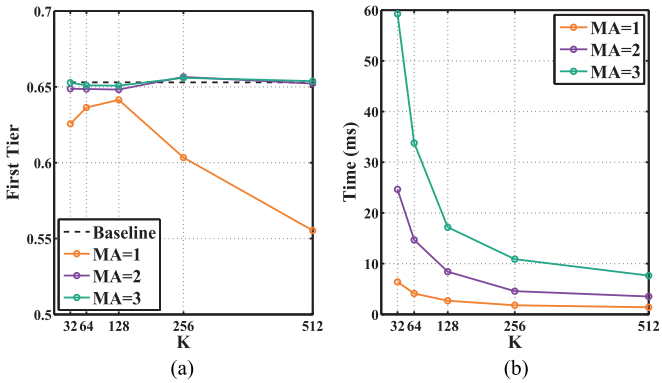


Fig. 8. Performance difference between Hausdorff matching and its approximate version in terms of (a) retrieval accuracy and (b) average query time.

number of entries used in the first inverted file changes. As Fig. 8(a) shows, the retrieval performance generally decreases with more entries, and multiple assignment can boost the retrieval performance significantly. However, it should be addressed that a better approximation to (5) using fewer entries (decreasing K) or larger multiple assignments (increasing MA) does not necessarily imply a better retrieval performance. For example, when $K = 256$ and MA$= 2$, the performance of approximate Hausdorff matching using inverted file surpasses the baseline using exact Hausdorff matching. The reason for this "abnormal" observation is that the principle of inverted file here is to reject those view matching operations that lead to smaller similarities, and sometimes they are noisy and false matching pairs which can be harmful to the retrieval performance.

As can be seen from Fig. 8(b), the average query time is higher at smaller K and larger MA, since the two cases both increase the number of candidate matchings in each entry. The baseline query time using exact Hausdorff matching is 0.69 s, which is at least one order of magnitude larger than the approximate one.

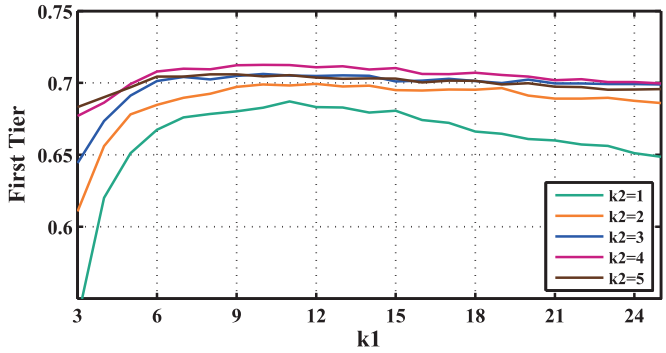


Fig. 9. Influence of neighbor set sizes $k1$ and $k2$ used in the second inverted file.

TABLE X
PERFORMANCE COMPARISON BETWEEN FINETUNING, DENOTED BY $\surd$, AND TRAINING FROM SCRATCH, DENOTED BY $\times$

Methods	NN		FT		ST	
	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$
L_5	0.849	0.642	0.588	0.370	0.721	0.473
L_7	0.837	0.628	0.653	0.380	0.784	0.512
GIFT	0.849	0.622	0.712	0.427	0.830	0.563

Discussion on S-IF: Two parameters, k_1 and k_2, are involved in the second inverted file, which are determined empirically. We plot the influence of them in Fig. 9. As can be drawn from the figure, when k_1 increases, the retrieval performance increases at first. Since noise contextual information can be included at a larger k_1, we can observe the performance decreases after $k_1 > 10$. Meanwhile, neighbor augmentation can boost the performance further. For example, the best performance is achieved when $k_2 = 4$. However, when $k_2 = 5$, the performance tends to decrease. One may find that the optimal value of k_2 is much smaller than that of k_1. The reason for this is that k_2 defines the size of the second order neighbor, which is more likely to return noise context compared with the first order neighbor defined by k_1.

Finetuning or Not: By default, GIFT finetunes the CNN model which is pre-trained on natural images from ImageNet. In this experiment, we evaluate whether training the neural network from scratch will affect the retrieval performance. As Table X shows, finetuning achieves much better performances than training from scratch. It suggests that although depth images have very different appearances from natural images, finetuning the pretrained model is a better choice to learn more discriminative filters for them.

Transferring Capacity: In the phase of feature extracting, the proposed GIFT needs labeled shapes to train the network. The testing categories usually appear in the training categories, so that the category-specific cues can be captured by the feature extractor. In this experiment, we investigate the transferring capacity of GIFT, that is, the testing categories have non-overlap with the training categories. Therefore, we construct a

TABLE XI
PERFORMANCES OF GIFT WHEN VARYING THE TRAINING AND TESTING SOURCES

Training	Testing	AUC	MAP
ModelNet40	ModelNet40	83.10%	81.94%
ModelNet10	ModelNet40	64.73%	63.65%
ModelNet10	ModelNet30	64.10%	62.94%

new dataset called ModelNet30, which is comprised of the 30 categories that appear in ModelNet40 but do not appear in ModelNet10. By doing so, when we use neural network trained on ModelNet10 to test the retrieval performance on ModelNet30, all the queries are outside the training categories.

Table XI presents the performances of GIFT with different training and testing sources. As can be drawn, when the training source is changed from ModelNet40 to ModelNet10, the retrieval performance on ModelNet40 decreases by around 19 percent. When testing on the new constructed ModelNet30, the performance slightly decreases further. Nevertheless, those performances of GIFT are still higher than SPH [58], LFD [15], PANORAMA [11] and ShapeNets [9] presented in Table I, which testifies the generalization ability of GIFT.

V. FUTURE WORK

There are still many interesting issues that can be studied further, such as:

1) *Rotation invariance:* In the proposed method, the used shape descriptor is not completely invariant to rotation. Learning rotation-invariant representation for 3D shape has been a challenging topic for a long time. Some classic algorithms (e.g., [11]) leverage PCA techniques as a preprocessing step before feature extraction. However, as suggested above, PCA is not always stable. Recent trends [74] show that the robustness of image representations to common geometric transformations (e.g., translation, scale, rotation, warping) can be learned in deep learning framework. By analogy, it is promising to learn the transformations of 3D shapes in a neural network, so that the preprocessing of pose normalization and feature extraction can be done simultaneously.
2) *Query by sketch:* Note that the query in the proposed system is 3D shape. SHREC community also starts to attach importance to large scale sketch-based 3D shape retrieval, which aims to retrieve relevant 3D shapes using sketch as input. To facilitate this research area, several competition tracks [75], [76] are organized. The competition results demonstrate that the scalability of sketch-based 3D shape retrieval is also badly required.
3) *Spatial topology:* Human perceptions of 3D shapes depend on 2D projections. These projections actually constitute a spatial topology around the 3D shape. It can be expected that in the multi-view matching procedure, the relative spatial arrangement of these projections is helpful to establish a more robust correspondence between two shapes. So, how to efficiently utilize the spatial information can be investigated in large scale 3D shape retrieval.

VI. CONCLUSION

In the past years, 3D shape retrieval was evaluated with only small numbers of shapes. In this sense, the problem of 3D shape retrieval has stagnated for a long time. Only recently, shape community starts to pay more attention to the scalable retrieval issue gradually. However, as suggested in [1], most classical methods encounter severe obstacles when dealing with larger databases.

In this paper, we focus on the scalability of 3D shape retrieval algorithms, and build a well-designed 3D shape search engine called GIFT. In our retrieval system, GPU is utilized to accelerate the speed of projection rendering and view feature extraction, and two inverted files are embedded to enable real-time multi-view matching and re-ranking. As a result, the average query time is controlled within one second, which clearly demonstrates the potential of GIFT for large scale 3D shape retrieval. What is more impressive is that while preserving the high time efficiency, GIFT outperforms state-of-the-art methods in retrieval accuracy by a large margin. Therefore, we view the proposed search engine as a promising step towards larger 3D shape corpora.

REFERENCES

[1] B. Li *et al.*, "A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries," *Comput. Vis. Image Understand.*, vol. 131, pp. 1–27, 2015.
[2] I. Sipiran *et al.*, "Scalability of non-rigid 3D shape retrieval," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2015, pp. 121–128.
[3] M. Savva *et al.*, "SHREC16 track large-scale 3D shape retrieval from shapenet Core55," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2016.
[4] I. Kokkinos, M. M. Bronstein, R. Litman, and A. M. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 159–166.
[5] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.
[6] C. Li, M. Ovsjanikov, and F. Chazal, "Persistence-based structural recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 2003–2010.
[7] R. Litman, A. Bronstein, M. Bronstein, and U. Castellani, "Supervised learning of bag-of-features shape descriptors using sparse coding," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 127–136, 2014.
[8] J. Assfalg, M. Bertini, A. Del Bimbo, and P. Pala, "Content-based retrieval of 3D objects using spin image signatures," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 589–599, Apr. 2007.
[9] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shape modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 1912–1920.
[10] Y. Wang *et al.*, "Projective analysis for 3D shape segmentation," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 192:1–192:12, 2013.
[11] P. Papadakis, I. Pratikakis, T. Theoharis, and S. J. Perantonis, "Panorama: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval," *Int. J. Comput. Vis.*, vol. 89, no. 2/3, pp. 177–192, 2010.
[12] Y. Guo *et al.*, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, pp. 66–89, 2015.
[13] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013.

[14] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2005, vol. 2, pp. 524–531.

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

[16] H. Tabia, D. Picard, H. Laga, and P. H. Gosselin, "Compact vectors of locally aggregated tensors for 3D shape retrieval," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2013, pp. 17–24.

[17] P. Papadakis, I. Pratikakis, S. J. Perantonis, and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recog.*, vol. 40, no. 9, pp. 2437–2452, 2007.

[18] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Fast image retrieval: Query pruning and early termination," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 648–659, May 2015.

[19] L. Zheng *et al.*, "Query-adaptive late fusion for image search and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 1741–1750.

[20] S. Bai, X. Bai, and W. Liu, "Multiple stage residual model for image classification and vector compression," *IEEE Trans. Multimedia*, vol. 18, no. 7, pp. 1351–1362, Jul. 2016.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[22] M. Havlena and K. Schindler, "VocMatch: Efficient multiview correspondence for structure from motion," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 46–60.

[23] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1320–1327.

[24] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 357–364.

[25] L. Luo, C. Shen, C. Zhang, and A. van den Hengel, "Shape similarity analysis by self-tuning locally constrained mixed-diffusion," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1174–1183, Aug. 2013.

[26] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "GIFT: A real-time and scalable 3D shape search engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2016, pp. 5023–5032.

[27] D. V. Vranic, "Desire: A composite 3D-shape descriptor," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, pp. 962–965.

[28] P. Daras and A. Axenopoulos, "A 3D shape retrieval framework supporting multimodal queries," *Int. J. Comput. Vis.*, vol. 89, no. 2/3, pp. 229–247, 2010.

[29] T. Furuya and R. Ohbuchi, "Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features," in *Proc. Int. Conf. Image Video Retrieval*, 2009, p. 26.

[30] H. Tabia, H. Laga, D. Picard, and P.-H. Gosselin, "Covariance descriptors for 3D shape matching and retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 4185–4192.

[31] H. Tabia and H. Laga, "Covariance-based descriptors for efficient 3D shape matching, retrieval, and classification," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1591–1603, Sep. 2015.

[32] X. Bai, S. Bai, Z. Zhu, and L. J. Latecki, "3D shape matching via two layer coding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2361–2373, Dec. 2015.

[33] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3D shape recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, Dec. 2015.

[34] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 945–953.

[35] Y. Fang *et al.*, "3D deep shape descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 2319–2328.

[36] J. Xie, Y. Fang, F. Zhu, and E. Wong, "Deepshape: Deep learned shape descriptor for 3D shape matching and retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 1275–1283.

[37] M. W. Jin Xie and Y. Fang, "Learned binary spectral shape descriptor for 3D shape correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2016, pp. 3309–3317.

[38] C. Li *et al.*, "Learning weight uncertainty with stochastic gradient MCMC for shape classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2016, pp. 5666–5675.

[39] Z. Lian, A. Godil, X. Sun, and J. Xiao, "CM-BOF: Visual similarity-based 3D shape retrieval using clock matching and bag-of-features," *Mach. Vis. Appl.*, vol. 24, no. 8, pp. 1685–1704, 2013.

[40] E. Rodolà, T. Harada, Y. Kuniyoshi, and D. Cremers, "Efficient shape matching using vector extrapolation," in *Proc. Brit. Mach. Vis. Conf.*, 2013, vol. 1, pp. 91.1–91.11.

[41] E. Rodola, S. Rota Bulò, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 4177–4184.

[42] E. Rodola, A. Torsello, T. Harada, Y. Kuniyoshi, and D. Cremers, "Elastic net constraints for shape matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1169–1176.

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

[44] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. J. Perantonis, "3D object retrieval using an efficient and compact hybrid shape descriptor," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2008, pp. 9–16.

[45] G. Lavoué, "Combination of bag-of-words descriptors for robust partial shape retrieval," *Visual Comput.*, vol. 28, no. 9, pp. 931–942, 2012.

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

[47] T. Furuya and R. Ohbuchi, "Fusing multiple features for shape-based 3D model retrieval," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.

[48] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014.

[49] T. Mei, Y. Rui, S. Li, and Q. Tian, "Multimedia search reranking: A literature survey," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 38:1–38:38, 2014.

[50] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki, "Similarity fusion for visual tracking," *Int. J. Comput. Vis.*, vol. 118, no. 3, pp. 337–363, 2016.

[51] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific rank fusion for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 4, pp. 803–815, Apr. 2015.

[52] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1056–1069, Mar. 2016.

[53] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 1169–1176.

[54] Z.-H. Zhou and M. Li, "Semi-supervised regression with co-training," in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 908–913.

[55] D. Giorgi, S. Biasotti, and L. Paraboschi, "Shape retrieval contest 2007: Watertight models track," *SHREC Competition*, vol. 8, pp. 1–8, 2007.

[56] K. Siddiqi *et al.*, "Retrieving articulated 3D models using medial surfaces," *Mach. Vis. Appl.*, vol. 19, no. 4, pp. 261–275, 2008.

[57] P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser, "The princeton shape benchmark," in *Proc. Shape Model. Appl.*, 2004, pp. 167–178.

[58] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Eurograph./ACM SIGGRAPH Symp. Geom. Process.*, 2003, pp. 156–164.

[59] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2016, pp. 3813–3822.

[60] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep.-Oct. 2015, pp. 922–928.

[61] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

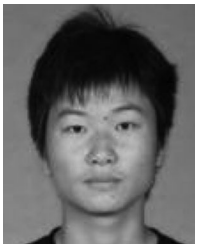
[62] A. Tatsuma and A. Masaki, "Food image recognition using covariance of convolutional layer feature maps," *IEICE Trans. Inform. Syst.*, vol. 99, no. 6, pp. 1711–1715, 2016.

[63] A. Tatsuma, H. Koyanagi, and M. Aono, "A large-scale shape benchmark for 3D object retrieval: Toyohashi shape benchmark," in *Proc. Asia Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2012, pp. 1–10.

[64] H. Tabia, M. Daoudi, J.-P. Vandeborre, and O. Colot, "A new 3D-matching method of nonrigid and partially similar models using curve analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 852–858, Apr. 2011.

[65] M. Liu, B. C. Vemuri, S. ichi Amari, and F. Nielsen, "Shape retrieval using hierarchical total Bregman soft clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2407–2419, Dec. 2012.

[66] X. Bai, C. Rao, and X. Wang, "Shape vocabulary: A robust and efficient shape representation for shape matching," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3935–3949, Sep. 2014.
[67] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian, "Good practice in CNN feature transfer," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1604.00133
[68] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 3304–3311.
[69] H. Jegou *et al.*, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, 2012.
[70] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2911–2918.
[71] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 653–656.
[72] G. Tolias, Y. Avrithis, and H. Jégou, "Image search with selective match kernels: Aggregation across single and multiple images," *Int. J. Comput. Vis.*, vol. 116, no. 3, pp. 247–261, 2016.
[73] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[74] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[75] B. Li *et al.*, "SHREC'14 track: Extended large scale sketch-based 3D shape retrieval," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2014, pp. 121–130.
[76] B. Li *et al.*''SHREC'13 track: Large scale sketch-based 3d shape retrieval," in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2013, pp. 89–96.



Song Bai received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013, and is currently working toward the Ph.D. degree at the School of Electronic Information and Communications, HUST.

His research interests include shape analysis, image classification, and retrieval.

Xiang Bai (S'07–M'09–SM'14) received the B.S., M.S., and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively.

He is currently a Professor with the School of Electronic Information and Communications, HUST. He is also the Vice-Director of National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition, and intelligent systems.

Zhichao Zhou received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2015, and is currently working toward the M.S. degree at the School of Electronic Information and Communications, HUST.

His research interests include shape analysis, deep learning and its applications.

Zhaoxiang Zhang (S'08–M'09–SM'15) received the B.S. degree in electronic science and technology from the University of Science and Technology of China (HUST), Hefei, China, in 2004, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009.

In 2009, he joined the School of Computer Science and Engineering, Beihang University, Beijing, China, as an Assistant Professor from 2009 to 2011, an Associate Professor from 2012 to 2015, and as the Vice-Director of the Department of Computer Application Technology from 2014 to 2015. In 2015, he returned to the Institute of Automation, Chinese Academy of Sciences, as a Full Professor. His current research interests include computer vision, pattern recognition, machine learning, and brain-inspired neural network and brain-inspired learning.

Prof. Zhang is the Associate Editor or Guest Editor of some internal journals, like *Neurocomputing*, *Pattern Recognition Letters*, and IEEE ACCESS.

Qi Tian (S'95–M'96–SM'03–F'16) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 1992, and the M.S. degree in ECE from Drexel University, Philadelphia, PA, USA, in 1996, and the Ph.D. degree in ECE from University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2002.

He is currently a Full Professor with the Department of Computer Science, the University of Texas at San Antonio (UTSA), San Antonio, TX, USA. He was a tenured Associate Professor from 2008 to 2012 and a tenure-track Assistant Professor from 2002 to 2008. During 2008 and 2009, he took one-year Faculty Leave at Microsoft Research Asia, Beijing, China, as a Lead Researcher in the Media Computing Group. He has authored or coauthored more than 340 refereed journal and conference papers. His research projects are funded by ARO, NSF, DHS, Google, FX-PAL, NEC, SALSI, CIAS, Akiira Media Systems, HP, Blippar, and UTSA. His research interests include multimedia information retrieval, computer vision, pattern recognition, and bioinformatics.

Dr. Tian is the Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the *Multimedia System Journal*. He is the Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and the *Journal of Computer Vision and Image Understanding*. He is on the Editorial Board of the *Journal of Multimedia* and the *Journal of Machine Vision and Applications*. He was the recipient of the 2014 Research Achievement Award from the College of Science, UTSA, and the 2010 ACM Service Award. He was coauthor of a Best Paper in ACM ICMR 2015, a Best Paper in PCM 2013, a Best Paper in MMM 2013, a Best Paper in ACM ICIMCS 2012, a Top 10% Paper Award in MMSP 2011, a Best Student Paper in ICASSP 2006, a Best Student Paper Candidate in ICME 2015, and a Best Paper Candidate in PCM 2007.

Longin Jan Latecki (M'03) is a Professor with Temple University, Philadelphia, PA, USA. He has authored or coauthored 200 research papers and books. His research interests include computer vision and pattern recognition.

Prof. Latecki is an Editorial Board Member of *Pattern Recognition* and *International Journal of Mathematical Imaging*. He was the recipient of the annual Pattern Recognition Society Award, together with Azriel Rosenfeld, for the best article published in the journal *Pattern Recognition* in 1998.