Appl. Math. J. Chinese Univ.
2018, 33(1): 88-106

Data driven composite shape descriptor design for shape retrieval with a VoR-Tree

WANG Zi-hao LIN Hong-wei* XU Chen-kai

Abstract. We develop a data driven method (probability model) to construct a composite shape descriptor by combining a pair of scale-based shape descriptors. The selection of a pair of scale-based shape descriptors is modeled as the computation of the union of two events, i.e., retrieving similar shapes by using a single scale-based shape descriptor. The pair of scale-based shape descriptors with the highest probability forms the composite shape descriptor. Given a shape database, the composite shape descriptors for the shapes constitute a planar point set. A VoR-Tree of the planar point set is then used as an indexing structure for efficient query operation. Experiments and comparisons show the effectiveness and efficiency of the proposed composite shape descriptor.

§1 Introduction

Shape descriptors are key values in shape retrieval. Several different kinds of shape descriptors have been developed. They can be divided into three categories based on the dimension of the descriptor, i.e., scale-based, vector-based, and matrix-based descriptors. Scale-based shape descriptors employ scales to measure various properties of shapes, such as orientability [32], compactness [3], rectilinearity [17], etc. Similarly, vector-based and matrix-based shape descriptors describe shapes using vectors and matrices, respectively. For example, Shape-DNA [22] is a vector-based descriptor, and DeepSD [8] is a matrix-based descriptor. The computations of the vector and matrix-based descriptors are generally much more complicated than those of the scale-based descriptors, while the shape retrieval precision of the scale-based descriptors is much lower than that of the vector and matrix-based descriptors. Because the scale-based descriptors measure different properties of a shape, a composition of several scale-based descriptors can improve retrieval precision with less computational complexity than vector and matrix-based descriptors.

Received: 2017-06-03. Revised: 2017-10-29.
MR Subject Classification: 68P20, 68U05.
Keywords: shape descriptor, shape retrieval, shape analysis, data-driven model.
Digital Object Identifier(DOI): https://doi.org/10.1007/s11766-018-3536-6.
This work is supported by the National Key R&D Plan of China (2016YFB1001501).
* Corresponding author.

However, one must decide how many scale-based descriptors should be used in the composite shape descriptor (CSD). When taking both shape retrieval precision and computational complexity into account, it is most appropriate for the CSD to contain two scale-based shape descriptors. The reasoning for this is as follows. If the CSD includes exactly two scale-based descriptors, all of the CSDs of the shapes in a database constitute a planar data point set. Not only can a Voronoi diagram of the planar data point set be constructed using efficient algorithms, but a Voronoi diagram-based query is also very fast and efficient. However, if three or more scale-based descriptors are included in the composite descriptor, the computation of the shape query will become much more complicated when compared to queries of the CSD using two scale-based descriptors. Therefore, it is most appropriate for the CSD to contain exactly two scale-based descriptors.

Next, one must choose the two scale-based descriptors to be used for the CSD. We employ a data-driven method to select the two scale-based descriptors and create the CSD with the best performance. Data-driven methods are used to extract knowledge or insights from data in various forms, either structured or unstructured [6]. With the dawn of the big data era, data-driven methods have been quickly become increasingly powerful. Because there are many existing geometric models, and the computations of scale-based shape descriptors are relatively simply, it is easy to acquire a large amount of scale-based descriptor data. This makes the selection of the two best scale-based descriptors using data-driven methods possible.

We first develop an algorithm using the MapReduce framework to calculate scale-based descriptors for each geometric shape model. With this data, the Pearson correlation coefficients [15] are calculated for each pair of scale-based descriptors. We then consider an event to be the retrieval of similar shapes using the scale-based descriptor as an inquiry, and test the retrieval performance of the scale-based descriptor using four measurements: nearest neighbor (NN), first-tier (1-tier), second-tier (2-tier), and discounted cumulative gain(DCG) [25] as the probabilities of the event. Next, in order to choose the best pair of scale-based descriptors, we model the problem as solving for the joint probability of two events. The pair of scale-based descriptors with the highest joint probability are chosen as the two descriptors for creating the CSD. Finally, for a given database of polygonal models, a CSD is calculated for every model in the database. Using these CSDs as planar points, a VoR-Tree [24] is constructed with the aid of the MapReduce framework, which can be employed to retrieve similar shapes for efficient shape queries. The experiments presented in this paper validate the effectiveness and efficiency of the proposed CSD.

The rest of this paper is arranged as follows. In Section 2, some related work is briefly surveyed. The method for constructing the CSD is discussed in Section 3. The shape retrieval data structure and algorithm are introduced in Section 4. After some experimental results and discussion are presented in Section 5, we give our conclusions in Section 6.

§2 Related work

In this section, we briefly review related work on shape descriptors and Voronoi diagram based query methods.

2.1 Shape descriptors

Many different shape descriptors have been developed. As stated previously, they can be divided into scale-based, vector-based, and matrix-based descriptors. Because the composite shape descriptor designed in this paper is a combination of two scale-based descriptors, we focus on the work related to scale-based shape descriptors for 3D polygonal models in this section, including eccentricity $\mathcal{E}$ [20], rectilinearity $\mathcal{R}$ [17], convexity $\mathcal{C}_1$ [30] and $\mathcal{C}_2$ [16], compactness $\mathcal{C}_{st}$ and $\mathcal{C}_d$ [3], and fractal dimension $\mathcal{F}$ [12].

Eccentricity. Eccentricity is the ratio of the length of the minor axis to that of the major axis, which can be calculated using the principal axis method CPCA [20]. First, a covariance matrix of the model's surface is calculated:

$$\begin{aligned} C &= \int\int_{v\in V} (v-m)(v-m)^T ds \\ &= \frac{1}{12E}\sum_{i=1}^{N} E_i[f(A_i)+f(B_i)+f(C_i)+9\cdot f((A_i+B_i+C_i)/3)], \end{aligned}$$

where V is the set of vertices of the mesh, m denotes the center of mass, $f(v)=(v-m)(v-m)^T$, E is the total surface area of the model, and E_i is the area of a triangle with vertices A_i, B_i, and $C_i, i=1,2,\cdots,N$. Because the lengths of the three principal axes equal the eigenvalues of the covariance matrix C, the eccentricity can be represented as:

$$\mathcal{E} = \frac{\lambda_{min}}{\lambda_{max}}, \tag{1}$$

where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of matrix C [23].

Rectilinearity. Rectilinearity is a shape descriptor derived from the concept of rectilinear polygons, all of whose interior angles belong to the set $\{\frac{\pi}{2}, \frac{3\pi}{2}\}$ [31]. Lian et al. [17] extended the rectilinearity measurement from 2D polygons to 3D meshes. The rectilinearity of a 3D mesh M is defined as:

$$\mathcal{R}(M) = 3\times\left(\max_{\alpha,\beta,\gamma\in[0,2\pi]} \frac{S(M)}{P(M,\alpha,\beta,\gamma)} - \frac{2}{3}\right), \tag{2}$$

where $S(M)$ is the surface of M,

$$P(M,\alpha,\beta,\gamma) = \sum_i P(T_i,\alpha,\beta,\gamma),$$

and $P(T_i,\alpha,\beta,\gamma)$ denotes the sum of three projected areas on the YOZ, ZOX, and XOY planes of the triangular face T_i after rotating the coordinate frame around the x,y,z axes by the angles α,β,γ.

Convexity. As proposed in [21,30], the convexity of a shape M is defined as:

$$\mathcal{C}_1(M) = P(\alpha X_1 + (1-\alpha)X_2 \in M_+),\ \forall\alpha\in[0,1], \tag{3}$$

where $P(\cdot)$ denotes the probability of the event that for randomly chosen points A and B from

the shape M, all points in the segment AB also belong to M. Recently, Lian et al. [16] proposed a new convexity for 3D shapes, which is given by:

$$\mathcal{C}_2(M) = \min_{\alpha,\beta,\gamma\in[0,2\pi]} \frac{Pview(M,\alpha,\beta,\gamma)}{Pface(M,\alpha,\beta,\gamma)}, \tag{4}$$

where $Pface(M,\alpha,\beta,\gamma)$ is the sum of the projected areas of triangles on the planes YOZ, ZOX, and XOY after rotating the coordinate system around its x,y,z axes by the angles α,β,γ. $Pview(M,\alpha,\beta,\gamma)$ denotes the sum of the areas of the valid regions in six views projected onto the faces of the bounding box whose edges are parallel to the coordinate axes.

Compactness. Compactness is an intrinsic property that describes the degree to which a shape is compact. The common compactness measurement for a 3D shape M is also called sphericity, which is the ratio of the surface area of a sphere (with the same volume as the given shape) to the surface area of M:

$$\mathcal{C}_{st}(M) = \frac{\pi^{\frac{1}{3}}(6\cdot volume\_of\_M)^{\frac{2}{3}}}{area\_of\_M}. \tag{5}$$

In 2008, Bribiesca [3] proposed a new method to measure the compactness of a 3D shape. This descriptor is based on volumetric reconstruction of a 3D mesh. Solids are discretized as voxel representations. In this manner, compactness is formulated as:

$$\mathcal{C}_d = \frac{A_c}{A_{c_{max}}} = \frac{n - A/6}{n - (\sqrt[3]{n})^2}, \tag{6}$$

where the contact area A_c corresponds to the sum of the areas of the contact surfaces which are common to both voxels, the maximum contact surface area $A_{c_{max}}$ occurs when the solid is a perfect cube, and A denotes the area of the enclosing surface of the solid composed of n voxels.

Fractal dimension. The fractal dimension (FD), first proposed by Mandlebrot [18], can be used to measure the degree of self-similarity of a shape. In this paper, we compute FD by using the box-counting method [12]. This method places the shape M onto a cubic grid of size r, then calculates the number of boxes occupied by the object, specifically, $N(r)$. The slope of the fitted r-$N(r)$ curve is then taken as the value of FD. Additionally, in order to normalize the attribute value to [0,1], we define the FD measure as:

$$\mathcal{F}(M) = \frac{1}{3}\cdot\frac{\mathrm{d}\ln N(r)}{\mathrm{d}\ln 1/r}. \tag{7}$$

Descriptor fusion. Although the scale-based descriptors outlined above are useful in some specific cases, no single scale-based descriptor is capable of providing satisfactory retrieval performance for a broad class of shapes [25,26]. In order to improve the retrieval performance, some shape descriptor fusion methods were developed in previous literature. Chahooki et al. [4] proposed a fusion method for contour and region based shape descriptors. The retrieval results showed significant improvement after fusion. Akgül et al. [2] proposed a supervised fusion method based on a linear combination of shape similarity scores provided by different descriptors, aiming to find the optimal combination of weights by minimizing the empirical ranking risk criterion. Additionally, Zhang et al. [29] investigated an unsupervised graph-based fusion of retrieval sets using multiple methods to enhance retrieval precision.

Note that the outputs of the methods developed in [2,4] are a linear combination of several shape descriptors, and the method proposed in [29] integrates the retrieval results retrieved by several descriptors into a better results using an ensemble learning technique. However, in our method, a pair of scale based shape descriptors are selected by a data driven method, i.e., a probability model, to form a vector for improving the retrieval results.

2.2 MapReduce

Hadoop MapReduce is an efficient offline, parallel, and distributed computing framework for processing large datasets. It is the open source implementation of Google's MapReduce [5] and runs on HDFS, a distributed storage system. Using MapReduce, complex processes can be reduced to two operations: Map and Reduce. The files to be processed are stored in HDFS and broken into multiple blocks, called splits. The Map task first parses splits into <key, value> pairs, and then uses a map() function defined by user to process the pairs. The intermediate output of the Mapper is stored on a local disk and divided into several partitions, each of which maps to a Reduce Task. The Reduce task can be divided into three stages:

- Shuffle stage: read the intermediate results of the Map task from the remote computing node.
- Sort stage: sort pairs according to their keys.
- Reduce stage: call a user-defined reduce() function to process the corresponding intermediate <key, value> pairs and, finally, write the results to HDFS.

2.3 Voronoi diagram, VoR-Tree and query

Consider a given set of points (called sites)

$$P = \{p_1, p_2, \ldots, p_n\}.$$

A voronoi diagram of P is a partition of the space into disjoint regions $R(p_i), i = 1, \ldots, n$, with a specified distance metric. Each region corresponds to one site and the region $R(p_i)$ corresponding to the site p_i covers points in space that are closer to p_i than to any other site. Many methods have been developed to construct Voronoi diagrams; they can be roughly divided into four types: divide-and-conquer methods [7], incremental methods [10], Delaunay triangle transform methods, and sweep-line-based methods [9].

Most existing Voronoi diagram parallel construction algorithms are based on the divide-and-conquer strategy. Parallel frameworks, like MapReduce, are suitable for the construction of Voronoi diagrams because they can be obtained by merging multiple partial Voronoi diagrams [1]. Using this idea, Akdogan et al. [1] proposed a Voronoi diagram construction algorithm using MapReduce for geospatial query processing. In their method, data points are first separated into several subsets. Then, the Map task is designed to generate partial Voronoi diagrams for each subset, and the Reduce process aggregates and combines them into a single Voronoi diagram.

Additionally, Voronoi diagrams have been shown to be efficient at solving spatial data query problems like *kNN* [13] with the help of extra indexing structures, such as the Voronoi history graph D-tree [28]. Thus, the parallel algorithm should take not only the Voronoi diagram, but also the indexing structure into consideration. In 2010, Sharifzadeh et al. [24] proposed a new structure, VoR-Tree, which incorporates Voronoi diagrams into R-tree index structures. Thus, it benefits from both the coarse granularity hierarchical grouping of R-trees and the fine granularity exploration capability of Voronoi diagrams. Furthermore, it can be easily modified for the MapReduce framework, so we use it as one of the retrieval tools in our work.

§3 Data driven composite shape descriptor design

In this section, a data driven method is developed to select a pair of descriptors to form a composite shape descriptor (CSD). Specifically, after calculating the scale-based shape descriptors (for example, the shape descriptors designated in Eqs. (1)-(7)) for each shape model in a given shape database, we calculate the Pearson correlation coefficient [15] between each pair of scale-based shape descriptors. The retrieval precisions of the shape descriptors are also measured. Based on the resulting Pearson correlation coefficients and retrieval precisions, the most desirable pair of shape descriptors is chosen by a probability model, forming the CSD. In the following section, we will elaborate on the data driven CSD construction method.

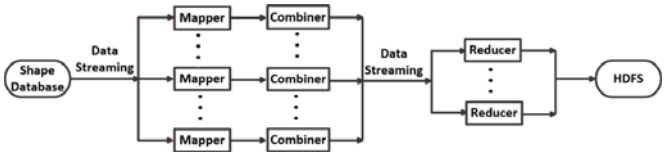


Figure 1: The MapReduce framework for calculating shape descriptors.

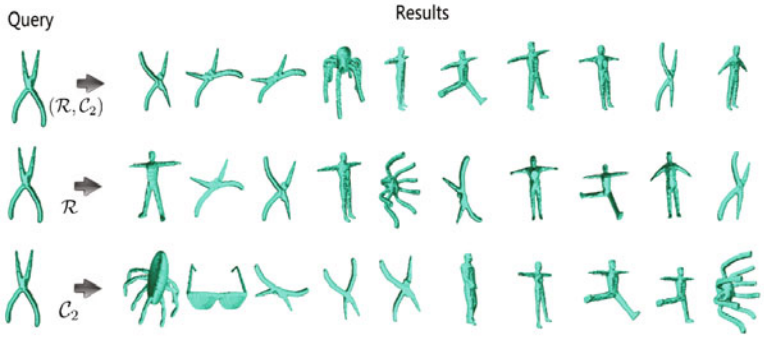


Figure 2: Because the two shape descriptors $\mathcal{R}$ and $\mathcal{C}_2$ have a high positive correlation, the retrieval results from $\mathcal{R}$, $\mathcal{C}_2$, and $(\mathcal{R},\mathcal{C}_2)$ are all highly similar. The retrieval results are all comprised of bodies, pincers, and octopuses, except that there is an instance of glasses in the retrieval results of $\mathcal{C}_2$.

Suppose there are m types of scale-based shape descriptors $\mathcal{D}_i, i = 1, 2, \cdots, m$ (such as

those presented in Eqs. (1)-(7)). Given a shape database, the m types of scale-based shape descriptors should be calculated for every shape model. In order to accommodate computations for large scale shape databases, we employ the MapReduce framework to calculate the scale-based shape descriptors (Fig. 1). When using MapReduce, 3D shape models are first serialized and transformed into a data stream. The MapReduce framework includes two procedures: Map and Reduce. In the Map procedure, all of the 3D shape models in a data stream are distributed into n *Mappers* running in parallel. In each Mapper, each scale-based shape descriptor is calculated by a different program. The Mapper inputs the shape data from the data stream, calculates a scale-based shape descriptor, and outputs it back to the data stream in the format $< shape\_id, des\_id, des\_val >$, where $shape\_id, des\_id$ and des_val are the ID of the shape, ID of the scale-based shape descriptor, and value of the shape descriptor, respectively. Next, a *Combiner* program captures these results from the data stream and assembles them into records, each of which reads as:

$$< shape\_id, d_1, d_2, \cdots, d_m >,$$

where d_i is the value of the i^{th} shape descriptor $\mathcal{D}_i, i = 1, 2, \cdots, m$. Finally, the *Reducer* combines the records into a data file and writes it back to HDFS.

(a) (b) (c)

Figure 3: The two shape descriptors $\mathcal{R}$ and $\mathcal{C}_2$ are positively correlated. (a) $\mathcal{R} = 0.1161, \mathcal{C}_2 = 0.03250$. (b) $\mathcal{R} = 0.1174, \mathcal{C}_2 = 0.03440$. (c) $\mathcal{R} = 0.1207, \mathcal{C}_2 = 0.03508$.

Evidently, scale-based shape descriptors are not totally independent. Refer to Fig. 3, when the shape model in Fig. 3(a) deforms into the shapes contained in Figs. 3(b) and 3(c), the two shape descriptors $\mathcal{R}$ (rectilinearity) and $\mathcal{C}_2$ (convexity) display the same increasing trend. Thus, they are positively correlated. It can be extrapolated that, if two scale-based shape descriptors have high positive correlation or high negative correlation, not only are the retrieval results of the two descriptors highly similar, but the retrieval results of the combination of the two descriptors are also highly similar to those of each of the two descriptors individually. For example, refer to Fig. 2, because the two shape descriptors $\mathcal{R}$ and $\mathcal{C}_2$ have a high positive correlation, the retrieval results from $\mathcal{R}$, $\mathcal{C}_2$, and $(\mathcal{R}, \mathcal{C}_2)$ are all highly similar. Actually, the retrieval results are all comprised of bodies, pincers, and octopuses, except that there is an instance of glasses in the retrieval results of $\mathcal{C}_2$. In other words, the combination of two descriptors with a highly positive or highly negative correlation cannot significantly improve the retrieval results of either of the individual descriptors. Therefore, in order to choose the best pair of scale-based descriptors, the correlation between each pair of descriptors must be measured.

We employ the Pearson's correlation coefficient [15] to measure the correlation between each

pair of scale-based shape descriptors. Suppose there are n shapes in a given shape database, and m types of scale-based descriptors for all of the shapes have been calculated. The same types of descriptors for all of the shapes are then extracted, forming a shape descriptor vector:

$$\boldsymbol{D}_i = (d_{i,1}, d_{i,2}, \cdots, d_{i,n}),\ i = 1, 2, \cdots, m, \tag{8}$$

where $d_{i,k}$ is the i^{th} scale-based shape descriptor for the k^{th} shape ($k = 1, 2, \cdots, n$). We denote $\bar{d}_i = \frac{\sum_{j=1}^{n} d_{i,j}}{n}$ and construct an average shape descriptor vector $\bar{\boldsymbol{D}}_i$ as:

$$\bar{\boldsymbol{D}}_i = (\bar{d}_i, \bar{d}_i, \cdots, \bar{d}_i)_{1\times n}.$$

The Pearson's correlation coefficients for the m types of scale-based descriptors are defined as [15]:

$$r_{ij} = \frac{(\boldsymbol{D}_i - \bar{\boldsymbol{D}}_i)\cdot(\boldsymbol{D}_j - \bar{\boldsymbol{D}}_j)}{\sqrt{(\boldsymbol{D}_i - \bar{\boldsymbol{D}}_i)^2}\sqrt{(\boldsymbol{D}_j - \bar{\boldsymbol{D}}_j)^2}}. \tag{9}$$

Additionally, in order to measure the retrieval precision of a pair of scale-based descriptors, the retrieval precision of each individual scale-based descriptor should be calculated in advance. In our implementation, we employ four scores [25] to measure the retrieval precision of scale-based descriptors: Nearest Neighbor (NN), First-tier (1-tier), Second-tier (2-tier), and Discounted Cumulative Gain (DCG). For the meanings and computations of the four scores, please refer to Ref. [25].

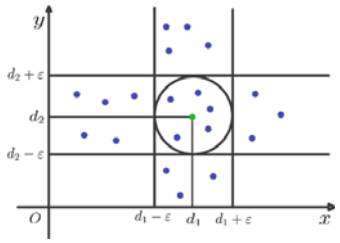


Figure 4: Probability model.

We are now in a position to seek the most desirable pair of scale-based shape descriptors to create the CSD based on the Pearson's correlation coefficients (9) and a score of the measurement for retrieval precision. The most desirable pair of scale-based descriptors is the pair of descriptors with the highest retrieval precision. In this section, we develop a *probability model* to calculate a score for measuring the retrieval precision of a pair of scale-based descriptors. In our developed probability model, for each scale-based shape descriptor $\mathcal{D}_k, k = 1, 2, \cdots, m$, an *event* is defined as:

$$E_k = \{\text{retrieved shape in the range } |d - d_k| < \varepsilon \text{ is similar to the query shape }\},$$

where d_k is the value of shape descriptor $\mathcal{D}_k, k = 1, 2, \cdots, m$ for a query shape. The retrieval precision of the descriptor $\mathcal{D}_k$ is considered to be the probability of E_k, i.e., $P(E_k)$, $k = 1, 2, \cdots, m$. For a CSD $(\mathcal{D}_i, \mathcal{D}_j)$, an *event* is defined as:

$E_{ij} = \{\text{retrieved shape in the range } (x - d_i)^2 + (y - d_j)^2 < \varepsilon^2 \text{ is similar to the query shape }\}$,

where d_i and d_j are the values of the two scale-based shape descriptors $\mathcal{D}_i$ and $\mathcal{D}_j$, $i \neq j, i, j =$

$1, 2, \cdots, m$, for a query shape. The probability of E_{ij} (i.e., $P(E_{ij})$) is the retrieval precision of the CSD $(\mathcal{D}_i, \mathcal{D}_j)$.

Refer to Fig. 4, the $x-$ and $y-$axis represent the values of the two scale-based descriptors $\mathcal{D}_i$ and $\mathcal{D}_j$, respectively. Suppose we are given a query shape model whose values of descriptor $\mathcal{D}_i$ and $\mathcal{D}_j$ are d_i and d_j, respectively. In the $O-xy$ plane, the shape lies on the green point. Additionally, suppose that the shapes with their (d_i, d_j) lying in the circle $\sqrt{(x-d_i)^2+(y-d_j)^2} < \varepsilon$ are similar shapes to the query. In Fig. 4, there are 5 points (in blue) in the circle. However, in the range $|x-d_i| < \varepsilon$, there are 12 points (in blue), and in the range $|y-d_j| < \varepsilon$, there are 13 points (in blue). So, the probability of event E_{ij} is 1 ($P(E_{ij}) = 1$), the probability of event E_i is $\frac{5}{12}$ ($P(E_i) = \frac{5}{12}$), and the probability of event E_j is $\frac{5}{13}$ ($P(E_j) = \frac{5}{13}$). Actually, in the case illustrated in Fig. 4,

- if the event E_i occurs, E_{ij} occurs, i.e., $E_i \subseteq E_{ij}$;
- if the event E_j occurs, E_{ij} also occurs, meaning $E_j \subseteq E_{ij}$.

Therefore, we use $E_i \cup E_j$ to approximate E_{ij}.

Note that,

$$P(E_i \cup E_j) = P(E_i) + P(E_j) - P(E_i \cap E_j), \tag{10}$$

where the probabilities $P(E_i)$ and $P(E_j)$ are taken as retrieval precisions of the scale-based descriptors $\mathcal{D}_i$ and $\mathcal{D}_j$, respectively, measured by scores NN, 1-tier, 2-tier, and DCG, as stated above. Now, what is left unknown in Eq. (10) is $P(E_i \cap E_j)$. On one hand, if the events E_i and E_j are totally independent, corresponding to $r_{ij} = 0$ (9), we have,

$$P(E_i \cap E_j) = P(E_i)P(E_j).$$

On the other hand, if the events E_i and E_j are totally positively correlated ($r_{ij} = 1$ (9)), or totally negatively correlated ($r_{ij} = -1$ (9)), the retrieval results of $\mathcal{D}_i$ and $\mathcal{D}_j$ are the same, meaning that $P(E_i) = P(E_j)$. Therefore, when $r_{ij} = 1$ or $r_{ij} = -1$, it holds that:

$$P(E_i \cap E_j) = P(E_i) = P(E_j) = \frac{P(E_i) + P(E_j)}{2}.$$

We then formulate the computation of $P(E_i \cap E_j)$ as the model:

$$P(E_i \cap E_j) = (1 - |r_{ij}|^{\alpha})P(E_i)P(E_j) + |r_{ij}|^{\beta} \frac{P(E_i) + P(E_j)}{2}, \tag{11}$$

where, $0 \le \alpha \le 1$ and $0 \le \beta \le 1$ are the two parameters.

In total, the retrieval precision of the pair of descriptors $(\mathcal{D}_i, \mathcal{D}_j)$ is measured by:

$$P(E_i \cup E_j) = P(E_i) + P(E_j) - (1 - |r_{ij}|^{\alpha})P(E_i)P(E_j) - |r_{ij}|^{\beta} \frac{P(E_i) + P(E_j)}{2}, \tag{12}$$

where $P(E_i)$ and $P(E_j)$ are taken as the retrieval precisions of descriptors $\mathcal{D}_i$ and $\mathcal{D}_j$, measured by the scores NN, 1-tier, 2-tier, and DCG, respectively, and r_{ij} is the Pearson correlation coefficient (9) of $\mathcal{D}_i$ and $\mathcal{D}_j$.

The next task is to determine the parameters α and β in model (12). For this purpose, the retrieval precisions, i.e., the scores NN, 1-tier, 2-tier, and DCG, for each pair of scaled-based shape descriptors $(\mathcal{D}_i, \mathcal{D}_j)$ are first calculated on a training set, denoted $C_{ij}, i, j = 1, 2, \cdots, m$. With the data C_{ij}, the two parameters α and β can be determined by solving a constrained

minimization problem:

$$\begin{aligned} &\min_{\alpha,\beta} \sum_{i,j} (P_i + P_j - (1 - |r_{ij}|^{\alpha}) P_i P_j - |r_{ij}|^{\beta} \frac{P_i + P_j}{2} - C_{ij})^2 \\ &s.t.\ 0 \le \alpha \le 1,\ 0 \le \beta \le 1, \end{aligned} \tag{13}$$

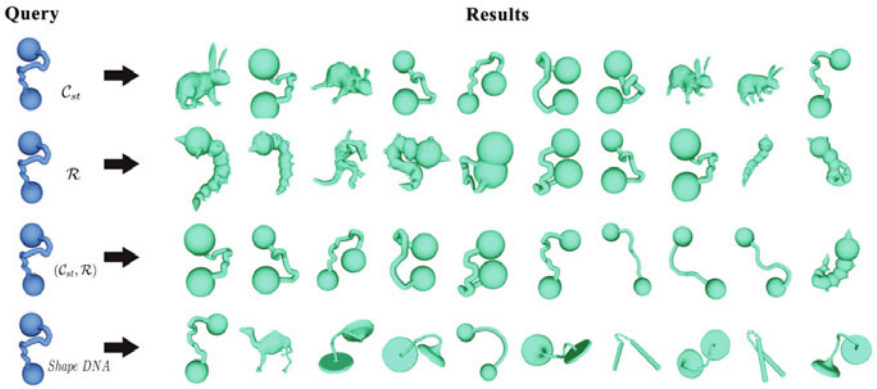
where $P_i = P(E_i), P_j = P(E_j)$ (refer to Eq. (12)). In our implementation, we use the Matlab function *lsqnonlin* to solve the constrained minimization problem (13).

In conclusion, for the m scale-based shape descriptors $\mathcal{D}_k, k = 1, 2, \cdots, m$, the retrieval precision of each pair of descriptors $(\mathcal{D}_i, \mathcal{D}_j)$ is calculated by the model (12), and the pair of shape descriptors with highest retrieval precision is used as the CSD.

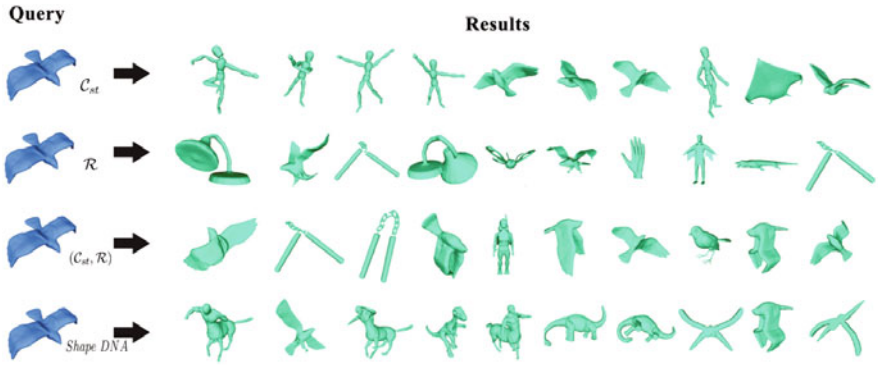
§4 Shape retrieval by VoR-Tree

Given a shape database, we computed the CSD for each shape model in the database. The computed CSDs constitute a planar point set G. In order to perform shape queries on the planar point set G, we user the VoR-Tree data structure proposed in Ref. [24]. A VoR-Tree is a combination of an R-tree and Voronoi diagram. The R-tree [11] is an indexing structure commonly used in spatial query processing. An R-tree encloses subsets of the point set G with minimum bounding rectangles. Leaf nodes of an R-tree store the points of a subset of the point set G and each intermediate node contains the minimum bounding rectangle for its child nodes. On the other hand, a Voronoi diagram [19] of a planar data point set with n points segments the plane into n regions, each of which corresponds to a data point. Each region contains all the points closest to the data point which it corresponds to. In order to construct the VoR-Tree for the planar point set G, the Voronoi diagram and R-tree of G are first formed independently. Then, for each leaf node of the R-tree, which stores a subset of the point set G, two records are appended to each point in the leaf node. One record contains pointers to locations of Voronoi neighbors of the point, and the other collects the vertices of the Voronoi cell for the point. In this manner, the entire VoR-Tree for the point set G is constructed.

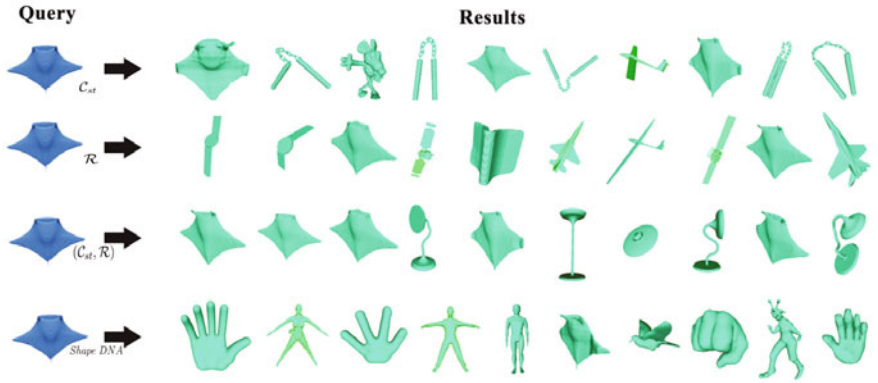
Given a query point q, we want to find the k Nearest Neighbors (kNN) to q with the help of the VoR-Tree. For this purpose, a Best-First Search (BFS) algorithm is first employed to find the Nearest Neighbor to q. With BFS, nodes are visited in the order from the root R of the VoR-Tree to its leaves. These nodes are sorted in a *minheap* H according to their distances from the query point q, and the node with minimum distance is put at the top of the minheap H. Specifically, BFS first visits the root R and puts it into the minheap H. Then, at each step of BFS, The node at the top of the heap H is removed and its entries are added into H. The nearest neighbor p is found if it is an entry in a leaf node and its corresponding Voronoi cell $V(p)$ contains the query point q. Then, the nearest neighbor p is pushed into a new minheap H' with its distance to the query q. At each following step, the top of the minheap H' is deleted and output as one of the kNN to the query q. The neighbors of the top of minheap H' in the Voronoi diagram are then added into H' with their distances to the query q. After k steps, the kNN to the query q are outputted. For additional details, please refer to Ref. [24].



(a) The first example.



(b) The second example.



(c) The third example.

Figure 5: Three retrieval examples by $\mathcal{C}_{st}$, R, $(\mathcal{C}_{st}, \mathcal{R})$, and ShapeDNA, respectively.

§5 Results and discussions

In our implementation, we employ the McGill Articulated 3D Shape Benchmark (MASB), which contains 255 shape models, as a training set. The test set is a blending of several

articulated shape sets, including Shrec15 (1200 shapes) and four subclasses of the Princeton Shape Benchmark(PSB) [25]: winged_vehicle class (135 shapes), animal class (155 shapes), hand class (17 shapes), and lamp class (8 shapes). In total, the test set contains 1515 shape models. The MapReduce frameworks for shape descriptor calculation, VoR-Tree construction, and shape retrieval operations are run on a Hadoop platform (version: 2.6.0) comprised of four PCs, each with a 2.0GHz dual-core CPU, 2GB RAM, and CentOS 7.0.

Table 1: Average time for computing seven descriptors for a shape model.

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
Time	8ms	20.4s	1632ms	46.78s	3ms	560ms	160ms

Table 2: The Pearson correlation coefficients (9) of seven shape descriptors.

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	1.00	-0.31	-0.12	-0.34	-0.48	0.30	-0.20
$\mathcal{R}$	-0.31	1.00	0.18	0.84	0.47	-0.16	0.53
$\mathcal{C}_1$	-0.12	0.18	1.00	0.21	-0.00	0.61	0.16
$\mathcal{C}_2$	-0.34	0.84	0.21	1.00	0.52	-0.29	0.44
$\mathcal{C}_{st}$	-0.48	0.47	-0.00	0.52	1.00	-0.35	0.19
$\mathcal{C}_d$	0.30	-0.16	0.61	-0.29	-0.35	1.00	0.01
$\mathcal{F}$	-0.20	0.53	0.16	0.44	0.19	0.01	1.00

For each shape in the training set (MASB), we first calculate the seven scale-based shape descriptors (Eqs. (1)-(7)) presented in Section 2.1 by using the MapReduce-based method developed in Section 3. The average time for computing the seven descriptors for a shape model is listed in Table 1. It can be seen that the cost time ranges from 3ms to 46.78s. The Pearson correlation coefficients r_{ij} (9) of the seven shape descriptors are then calculated. They are listed in Table 2. We then use the four scores [25], *NN, 1-tier, 2-tier*, and *DCG*, to measure the retrieval precisions of the seven scale-based descriptors (Eqs. (1)-(7)). The four scores of the seven descriptors are presented in Table 3. In the developed probability model (12), the scores are used as the probabilities.

Table 3: Retrieval precisions of seven shape descriptors on the training set.

	NN	*1-Tier*	*2-Tier*	*DCG*
Eccentricity $\mathcal{E}$	26.3%	21.9%	38.3%	57.2%
Rectilinearity $\mathcal{R}$	20.8%	18.9%	34.5%	54.6%
convexity $\mathcal{C}_1$	20.0%	16.8%	32.2%	53.1%
convexity $\mathcal{C}_2$	26.7%	21.4%	39.4%	56.7%
compactness $\mathcal{C}_{st}$	21.2%	18.0%	34.1%	54.0%
compactness $\mathcal{C}_d$	32.2%	28.7%	44.2%	60.8%
Fractal Dim $\mathcal{F}$	21.6%	18.8%	36.0%	54.0%

Next, we compute C_{ij} (13), the true retrieval precision of a pair of scale-based shape descriptors $(\mathcal{D}_i, \mathcal{D}_j)$, measured by the four scores, on the training set. The data for C_{ij} are listed in Tables 10 and 11 in the Appendix. Because the matrix for the retrieval precision of each

pair of scale-based descriptors is symmetric, we combine the two matrices resulting from two scores into a single matrix. For example, in Table 10, the upper triangular matrix stores the scores measured by NN and the lower triangular matrix stores the scores measured by 1-Tier. Additionally, the structures of the matrices in Tables 5, 6, 11, 12, and 13 are the same.

With the data r_{ij} (Table 2), $P(E_i)$ (Table 3), and C_{ij} (Tables 10 and 11), we can learn the parameters α and β for the probability model (12) by solving the constrained minimization problem (13). The learned parameters α and β corresponding to the four scores *NN, 1-Tier, 2-Tier*, and *DCG* are listed in Table 4.

Table 4: The learned parameters α and β (12) corresponding to four scores.

	NN	*1-Tier*	*2-Tier*	*DCG*
α	0.6281	0.1032	0.0533	0.0447
β	0.7865	0.1323	0.0699	0.0540

Table 5: The computed retrieval precision for each pair of descriptors on the test set measured by NN(upper triangle) and $1-Tier$(lower triangle).

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	N/A	0.20	0.13	0.12	0.30	0.12	0.12
$\mathcal{R}$	0.11	N/A	0.24	0.21	0.31	0.21	0.20
$\mathcal{C}_1$	0.10	0.14	N/A	0.12	0.27	0.10	0.14
$\mathcal{C}_2$	0.09	0.11	0.09	N/A	0.25	0.14	0.10
$\mathcal{C}_{st}$	0.20	0.21	0.19	0.18	N/A	0.26	0.25
$\mathcal{C}_d$	0.09	0.12	0.09	0.10	0.19	N/A	0.13
$\mathcal{F}$	0.08	0.11	0.09	0.07	0.18	0.08	N/A

Table 6: The computed retrieval precision for each pair of descriptors on the test set measured by $2-Tier$(upper triangle) and DCG(lower triangle).

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	N/A	0.18	0.17	0.14	0.31	0.15	0.13
$\mathcal{R}$	0.44	N/A	0.22	0.18	0.34	0.20	0.18
$\mathcal{C}_1$	0.45	0.49	N/A	0.15	0.32	0.16	0.16
$\mathcal{C}_2$	0.44	0.44	0.44	N/A	0.29	0.15	0.12
$\mathcal{C}_{st}$	0.52	0.53	0.53	0.50	N/A	0.31	0.30
$\mathcal{C}_d$	0.44	0.46	0.44	0.45	0.52	N/A	0.15
$\mathcal{F}$	0.42	0.44	0.44	0.40	0.50	0.43	N/A

We are now ready to test the developed probability model (12) on the test set with the parameters listed in Table 4. Similar to operations on the training set, the seven scale-based shape descriptors (Eqs. (1)-(7)) are first calculated along with their Pearson correlation coefficients r_{ij} (9). The retrieval precisions of the seven shape descriptors for the test set are then measured by the four scores *NN, 1-Tier, 2-Tier*, and *DCG*. With this data, the retrieval precision for each pair of shape descriptors can be calculated by using the probability model (12) with the parameters α and β listed in Table 4. The computed precisions are listed in Tables 5 and 6. It

can be seen that, in all of the precisions measured by the four scores, the highest precision is that of the pair of scale-based descriptors $(\mathcal{C}_{st}, \mathcal{R})$. For comparison, we also calculate the true retrieval precision of each pair of scale-based descriptors measured by the four scores and list them in Tables 12 and 13 in the Appendix. In the true precisions, the highest value is also the one for the pair of $(\mathcal{C}_{st}, \mathcal{R})$.

Moreover, to check the correlation between the computed precisions and true precisions, a significance test for the correlation coefficient between the true and the computed precisions is performed. To each of the four scores, the true and the computed precisions of the 21 shape descriptor pairs form two vectors of length 21, respectively, and the correlation coefficient r between them are calculated. Suppose ρ is the *true* correlation coefficient between the computed precisions and true precisions. The significance test contains the following steps:

1. Give the null hypothesis $H_0 : \rho = 0$ and the alternative hypothesis $H_1 : \rho \neq 0$, respectively.

2. Test the significance of r using the Student's t-test, where the t-value of r is calculated as:
$$t = \frac{r - 0}{\sqrt{\frac{1-r^2}{n-2}}} \tag{14}$$

3. Search the Student's t-distribution table [14] to determine the corresponding two-side p-value. If the p-value is below the threshold α chosen for statistical significance, the null hypothesis is rejected, and the alternative hypothesis is accepted.

Denote n as the size of sample, and d as the degree of freedom. In our test, $n = 21$ and $d = n - 2 = 19$. The correlation coefficients r and their corresponding t-values are listed in Table 7. It can be seen that under each of the four scores, $r > 0.9$ and the absolute value of t-value $\left|t_{\frac{p}{2},d}\right| > t_{\frac{0.01}{2},d} = 2.861$. Then we have $p < \alpha = 0.01$, meaning that with $\alpha = 0.01$, H_0 is rejected and H_1 is accepted. This result shows that the precisions calculated by the computation model (12) are highly significant correlated with the true precisions.

Therefore, with respect to the selection of the best pair of descriptors with the highest retrieval precision, the selection made by the probability model perfectly conforms to the true data in the test set.

Table 7: The correlation coefficients r between the true and computed precisions and the t-values used in the significance test.

	NN	*1-Tier*	*2-Tier*	*DCG*
correlation coefficient r	0.9128	0.9511	0.9328	0.9407
t value	9.7445	13.4194	11.28	12.0871

As stated above, by using the probability model, the pair of descriptors $(\mathcal{C}_{st}, \mathcal{R})$ are chosen for being the CSD. We will now evaluate its performance for shape retrieval. The CSDs of the shapes in the test set constitute a planar point set. A VoR-Tree of the point set is constructed [24] and taken as the data structure for shape retrieval. In Fig. 5, three retrieval examples

Table 8: Retrieval precisions of shape descriptors on the test set.

	NN	*1-Tier*	*2-Tier*	*DCG*
$\mathcal{E}$	6.4%	6.7%	12.6%	39.6%
$\mathcal{R}$	19.1%	12.5%	21.9%	45.8%
$\mathcal{C}_1$	7.6%	8.5%	16.3%	43.6%
$\mathcal{C}_2$	6.5%	6.1%	10.8%	39.0%
$\mathcal{C}_{st}$	30.0%	27.0%	45.3%	59.2%
$\mathcal{C}_d$	8.0%	8.1%	15.2%	42.6%
$\mathcal{F}$	8.4%	6.6%	11.8%	39.4%
$(\mathcal{C}_{st}, \mathcal{R})$	50.5%	35.6%	52.1%	67.1%

are illustrated. In these examples, we compare the retrieval results of the CSD $(\mathcal{C}_{st}, \mathcal{R})$ with those of the two indidivual shape descriptors $\mathcal{C}_{st}$, $\mathcal{R}$, and ShapeDNA [22]. For fairness, the first two smallest eigenvalues of ShapeDNA are used for shape retrieval, also based on the VoR-Tree. In the first example (Fig. 5(a)), while there are 9 correct results in the top 10 results retrieved by the CSD $(\mathcal{C}_{st}, \mathcal{R})$, the numbers of correct results in the top 10 retrieval results by $\mathcal{C}_{st}$, $\mathcal{R}$, and ShapeDNA, are $6, 4$, and 2, respectively. In the second example (Fig. 5(b)), the numbers of correct results in the top 10 retrieval results by $\mathcal{C}_{st}$, $\mathcal{R}$, $(\mathcal{C}_{st}, \mathcal{R})$, and ShapeDNA, are, $4, 3, 6$, and 2. Note that, in this example, only the nearest neighbor to the query, retrieved by the CSD $(\mathcal{C}_{st}, \mathcal{R})$, is correct. In the third example (Fig. 5(b)), the numbers of correct results in the top 10 retrieval results by $\mathcal{C}_{st}$, $\mathcal{R}$, $(\mathcal{C}_{st}, \mathcal{R})$, and ShapeDNA, are, $3, 2, 5$, and 1. In all three examples, the performance of the CSD $(\mathcal{C}_{st}, \mathcal{R})$ is the best, while that of ShapeDNA with the two smallest eigenvalues is the worst, even worse than that of the two individual scale-based shape descriptors. Additionally, we calculate the retrieval precisions of the seven scale-based descriptors and the composite descriptor $(\mathcal{C}_{st}, \mathcal{R})$, measured by the four scores (Table 8). Refer to Table 8, the retrieval precisions of the CSD $(\mathcal{C}_{st}, \mathcal{R})$ show improvements of $20.5\%, 8.6\%, 6.8\%$, and 7.9% compared with the highest precisions of the individual scale-based descriptors under the four scores, *NN, 1-Tier, 2-Tier*, and *DCG*, respectively. We also illustrate the precision-recall curves for the seven scale-based descriptors (Eqs. (1)-(7)), the CSD $(\mathcal{C}_{st}, \mathcal{R})$, and ShapeDNA in Fig. 6. It can be seen that the performance of the precision-recall curve for the CSD $(\mathcal{C}_{st}, \mathcal{R})$ is the best.

Furthermore, the average times for a kNN query with the shape descriptors $\mathcal{C}_{st}$, $\mathcal{R}$, $(\mathcal{C}_{st}, \mathcal{R})$, and ShapeDNA are listed in Table 9 (where $k = 10$). For retrieval with $(\mathcal{C}_{st}, \mathcal{R})$ and ShapeDNA, a VoR-Tree is used as the data structure. For retrieval with the scale-based descriptors $\mathcal{C}_{st}$ and $\mathcal{R}$, we employ a one-dimensional array and a dual-pivot quicksort algorithm [27]. It can be seen that, while the average query time for scale-based shape descriptors $\mathcal{C}_{st}$ and $\mathcal{R}$ is 1 second or so, the average query time for the CSD $(\mathcal{C}_{st}, \mathcal{R})$ and ShapeDNA needs 2.58 seconds.

Table 9: Average time for kNN query on the test set.

	$\mathcal{C}_{st}$	$\mathcal{R}$	$(\mathcal{C}_{st}, \mathcal{R})$	ShapeDNA
Time	1.00s	1.02s	2.58s	2.58s

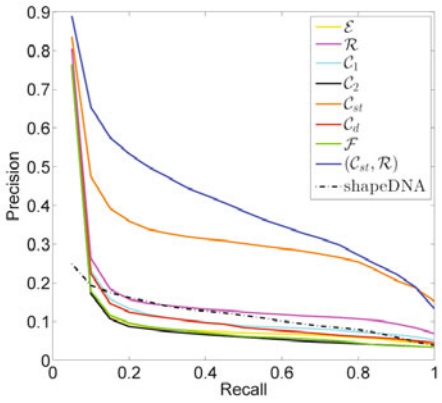


Figure 6: Precision-recall curves for random queries on the test set, using the seven scale-based shape descriptors (Eqs. (1)-(7)), the CSD $(\mathcal{C}_{st}, \mathcal{R})$, and ShapeDNA.

§6 Conclusion

In this paper, a probability model is developed to select a pair of scale-based shape descriptors to form a CSD. The probability model concerns the retrieval precisions of individual scale-based shape descriptors, and the Pearson correlation coefficient of each pair of descriptors. To construct the probability model, we define the retrieval of similar shapes to a query as an event, and the probability model is taken as the probability of the union of two events. In addition, the parameters in the probability model are learned by solving a constrained minimization problem. Finally, the pair of shape descriptors with highest probability (i.e., retrieval precision) are taken as the components of the CSD. The CSDs of shapes in a shape set constitute a planar point set. With the aid of VoR-Tree, the retrieval operation by CSD is efficient, and the retrieval precision is clearly improved, compared with the scale-based shape descriptors.

Appendix

In this appendix, the true retrieval precision for each pair of descriptors on the training set and test set is listed, measured by four scores *NN, 1-Tier, 2-Tier*, and *DCG*. Because the matrix of retrieval precisions measured by a score for pairs of descriptors is symmetric, we integrate two matrices measured by two scores into a matrix. For example, in Table 10, the upper triangular matrix stores the precisions measured by *NN*, and the lower triangular matrix is the precisions measured by *1-Tier*.

Table 10: The true retrieval precision for each pair of descriptors on the training set measured by NN(upper triangle) and $1-Tier$(lower triangle).

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	N/A	0.36	0.29	0.39	0.30	0.47	0.38
$\mathcal{R}$	0.24	N/A	0.33	0.31	0.31	0.42	0.23
$\mathcal{C}_1$	0.21	0.22	N/A	0.32	0.26	0.37	0.28
$\mathcal{C}_2$	0.23	0.23	0.18	N/A	0.35	0.50	0.35
$\mathcal{C}_{st}$	0.21	0.24	0.21	0.20	N/A	0.39	0.29
$\mathcal{C}_d$	0.31	0.33	0.26	0.36	0.29	N/A	0.40
$\mathcal{F}$	0.21	0.22	0.24	0.22	0.25	0.29	N/A

Table 11: The true retrieval precision for each pair of descriptors on the training set measured by $2-Tier$(upper triangle) and DCG(lower triangle).

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	N/A	0.39	0.37	0.39	0.38	0.45	0.37
$\mathcal{R}$	0.60	N/A	0.38	0.38	0.40	0.50	0.39
$\mathcal{C}_1$	0.57	0.57	N/A	0.34	0.38	0.42	0.40
$\mathcal{C}_2$	0.60	0.59	0.56	N/A	0.35	0.50	0.39
$\mathcal{C}_{st}$	0.57	0.59	0.56	0.57	N/A	0.43	0.42
$\mathcal{C}_d$	0.64	0.65	0.61	0.66	0.63	N/A	0.46
$\mathcal{F}$	0.58	0.56	0.58	0.58	0.59	0.62	N/A

Table 12: The true retrieval precision for each pair of descriptors on the test set measured by NN(upper triangle) and $1-Tier$(lower triangle).

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	N/A	0.22	0.23	0.16	0.42	0.20	0.14
$\mathcal{R}$	0.17	N/A	0.33	0.23	0.51	0.26	0.20
$\mathcal{C}_1$	0.16	0.24	N/A	0.21	0.49	0.20	0.22
$\mathcal{C}_2$	0.12	0.17	0.16	N/A	0.40	0.19	0.14
$\mathcal{C}_{st}$	0.24	0.36	0.29	0.26	N/A	0.43	0.44
$\mathcal{C}_d$	0.11	0.18	0.12	0.11	0.32	N/A	0.18
$\mathcal{F}$	0.10	0.16	0.14	0.10	0.32	0.13	N/A

Table 13: The true retrieval precision for each pair of descriptors on the test set measured by $2-Tier$(upper triangle) and DCG(lower triangle).

	$\mathcal{E}$	$\mathcal{R}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_{st}$	$\mathcal{C}_d$	$\mathcal{F}$
$\mathcal{E}$	N/A	0.26	0.26	0.20	0.35	0.18	0.16
$\mathcal{R}$	0.49	N/A	0.38	0.27	0.52	0.28	0.26
$\mathcal{C}_1$	0.50	0.57	N/A	0.26	0.41	0.19	0.23
$\mathcal{C}_2$	0.45	0.50	0.50	N/A	0.39	0.17	0.16
$\mathcal{C}_{st}$	0.57	0.67	0.63	0.60	N/A	0.49	0.49
$\mathcal{C}_d$	0.44	0.51	0.47	0.44	0.64	N/A	0.22
$\mathcal{F}$	0.43	0.49	0.49	0.43	0.64	0.47	N/A

References

[1] A Akdogan, U Demiryurek, F Banaei-Kashani, C Shahabi. *Voronoi-based geospatial query processing with mapreduce*, In: *IEEE Second International Conference on Cloud Computing Technology and Science*, 2010, 9-16.

[2] C B Akgül, B Sankur, Y Yemez, F Schmitt. *Similarity score fusion by ranking risk minimization for 3D object retrieval*, In: *Proceedings of the 1st Eurographics Conference on 3D Object Retrieval*, 2008, 41-48.

[3] E Bribiesca. *An easy measure of compactness for 2D and 3D shapes*, Pattern Recogn, 2008, 41(2): 543-554.

[4] M Chahooki, N Charkari. *Shape retrieval based on manifold learning by fusion of dissimilarity measures*, IET Image Process, 2012, 6(4): 327-336.

[5] J Dean, S Ghemawat. *MapReduce: simplified data processing on large clusters*, Commun ACM, 2008, 51(1): 107-113.

[6] V Dhar. *Data science and prediction*, Commun ACM, 2013, 56(12): 64-73.

[7] L Dong, Y Wu, S Zhou. *Constructing the voronoi diagram of planar point set in parallel*, In: *International Conference on Computational Intelligence and Software Engineering*, 2009, 1-5.

[8] Y Fang, J Xie, G Dai, M Wang, F Zhu, T Xu, E Wong. *3D deep shape descriptor*, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, 2319-2328.

[9] S Fortune. *A sweepline algorithm for Voronoi diagrams*, In: *Proceedings of the Second Annual Symposium on Computational Geometry*, ACM, 1986, 313-322.

[10] L J Guibas, D E Knuth, M Sharir. *Randomized incremental construction of Delaunay and Voronoi diagrams*, Algorithmica, 1992, 7(1): 381-413.

[11] A Guttman. *R-trees: a dynamic index structure for spatial searching*, In: *Proceedings of ACM Management of Data (SIGMOD)*, 1984, 47-57.

[12] J Jiang, W Zhu, F Shi, Y Zhang, L Lin, T Jiang. *A robust and accurate algorithm for estimating the complexity of the cortical surface*, J Neurosci Meth, 2008, 172(1): 122-130.

[13] M Kolahdouzan, C Shahabi. *Voronoi-based k nearest neighbor search for spatial network databases*, In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, VLDB Endowment, 2004, 840-851.

[14] R J Larsen, M L Marx. *An Introduction to Mathematical Statistics and Its Applications*, 4th ed, Prentice Hall, 2006.

[15] J L Rodgers, W A Nicewander. *Thirteen ways to look at the correlation coefficient*, Amer Statist, 1988, 42(1): 59-66.

[16] Z Lian, A Godil, P L Rosin, X Sun. *A new convexity measurement for 3d meshes*, In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, 119-126.

[17] Z Lian, P L Rosin, X Sun. *Rectilinearity of 3D meshes*, Int J Comput Vision, 2010, 89(2-3): 130-151.

[18] B B Mandelbrot. *The Fractal Geometry of Nature*, Macmillan, 1983.

[19] J O'Rourke. *Computational Geometry in C*, Cambridge University Press, 1998.

[20] P Papadakis, I Pratikakis, S Perantonis, T Theoharis. *Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation*, Pattern Recogn, 2007, 40(9): 2437-2452.

[21] E Rahtu, M Salo, J Heikkila. *A new convexity measure based on a probabilistic interpretation of images*, IEEE Trans Pattern Anal, 2006, 28(9): 1501-1512.

[22] M Reuter, FE Wolter, N Peinecke. *Laplace-Beltrami spectra as "Shape-DNA" of surfaces and solids*, Comput Aided Design, 2006, 38(4): 342-366.

[23] P L Rosin. *Measuring shape: ellipticity, rectangularity, and triangularity*, Mach Vision Appl, 2003, 14(3): 172-184.

[24] M Sharifzadeh, C Shahabi. *Vor-tree: R-trees with voronoi diagrams for efficient processing of spatial nearest neighbor queries*, Proc VLDB Endow, 2010, 3(1-2): 1231-1242.

[25] P Shilane, P Min, M Kazhdan, T Funkhouser. *The princeton shape benchmark*, In: *International Conference on Shape Modeling and Applications*, 2004, 167-178.

[26] J W Tangelder, R C Veltkamp. *A survey of content based 3D shape retrieval methods*, Multimed Tools Appl, 2008, 39(3): 441-471.

[27] S Wild. *Java 7's Dual Pivot Quicksort*, Master Thesis, Technische Universität Kaiserslautern, 2013.

[28] J Xu, B Zheng, WC Lee, D L Lee. *The D-tree: An index structure for planar point queries in location-based wireless services*, IEEE Trans Knowl Data Eng, 2004, 16(12): 1526-1542.

[29] S Zhang, M Yang, T Cour, K Yu, D N Metaxas. *Query specific fusion for image retrieval*, In: *Computer Vision-ECCV 2012*, Springer, 660-673.

[30] J Zunic, P L Rosin. *A new convexity measure for polygons*, IEEE Trans Pattern Anal, 2004, 26(7): 923-934.

[31] J Zunic, P L Rosin. *Rectilinearity measurements for polygons*, IEEE Trans Pattern Anal, 2003, 25(9): 1193-1200.

[32] J Zunic, P L Rosin, L Kopanja. *On the orientability of shapes*, IEEE Trans Image Process, 2006, 15(11): 3478-3487.

School of Mathematical Science, State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, China. Email: hwlin@zju.edu.cn