

Article

Research on Similarity Measurements of 3D Models Based on Skeleton Trees

Xin Chen ¹, Jingbin Hao ^{1,2,*}, Hao Liu ¹, Zhengtong Han ¹ and Shengping Ye ²

¹ School of Mechatronic Engineering, China University of Mining and Technology, Daxue Road 1, Xuzhou 221116, China; pekah5385@cumt.edu.cn (X.C.); liuhao8556@163.com (H.L.); cumthanzt@126.com (Z.H.)

² State Key Laboratory of Materials Forming and Mould Technology, Huazhong University of Science and Technology, Luoyu Road 1037, Wuhan 430074, China; yes@hust.edu.cn

* Correspondence: jingbinhao@cumt.edu.cn; Tel.: +86-152-6201-6986

Academic Editor: Paolo Bellavista

Received: 8 March 2017; Accepted: 19 April 2017; Published: 22 April 2017

Abstract: There is a growing need to be able to accurately and efficiently recognize similar models from existing model sets, in particular, for 3D models. This paper proposes a method of similarity measurement of 3D models, in which the similarity between 3D models is easily, accurately and automatically calculated by means of skeleton trees constructed by a simple rule. The skeleton operates well as a key descriptor of a 3D model. Specifically, a skeleton tree represents node features (including connection and orientation) that can reflect the topology and branch features (including region and bending degree) of 3D models geometrically. Node feature distance is first computed by the dot product between node connection distance, which is defined by 2-norm, and node orientation distance, which is defined by tangent space distance. Then branch feature distances are computed by the weighted sum of the average regional distances, as defined by generalized Hausdorff distance, and the average bending degree distance as defined by curvature. Overall similarity is expressed as the weighted sum of topology and geometry similarity. The similarity calculation is efficient and accurate because it is not necessary to perform other operations such as rotation or translation and it considers more topological and geometric information. The experiment demonstrates the feasibility and accuracy of the proposed method.

Keywords: skeleton tree; similarity measurement; model recognition; topology feature; geometry feature

1. Introduction

With rapid developments in computer hardware and computer technology, the construction of 3D models has become much easier. This has contributed to an increasing accumulation of 3D models. In the last 20 years, model recognition has become one of the most popular fields of computer science. It has wide application in the fields of Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM) [1,2], integrated circuit design [3], digital city planning [4], biomedical engineering [5], military applications [6], mesh decomposition [7], virtual reality [8], education [9] and animation [10]. Making full use of the existing resources of 3D model data can greatly reduce the workload of designing new models and promote the flow of 3D data and its application in various fields [11,12].

The core content of model recognition is similarity measurement between models. At present, there are a lot of similarity measurement methods:

- (1) **Statistic characteristic-based methods.** A classical algorithm is the shape distribution histogram formed by using the sampling function as the shape descriptor. The geometric similarity between shapes can be measured by the histogram [13]. To calculate the distance histogram between any two points on a shape, the enhanced shape function is used to obtain better experimental results [14]. The statistic characteristic-based method has a good outcome for global matching of models, but it does not have good performance for local matching.
- (2) **Geometry-based methods.** This method is based on various frequency domain features of a model. By using a weighted point set to describe a 3D polyhedral model, the similarity between two shapes can be computed by employing the Earth Mover's Distance to compare their weighted point sets [15]. The global properties of a 3D shape can be represented by the reflective value where all planes pass through the shape's center of mass [16].
- (3) **Projection-based methods.** This method mainly does the projection transform processing of a 3D model in different directions, which can obtain a series of 2D projection images of 3D models for model retrieval. A comparison method has been proposed by Min et al., based on 2D contouring of 3D models [17]. However, this method can only describe the brightness distribution of models and cannot effectively reflect their topological features.
- (4) **Topology-based methods.** Most prior work has focused on skeleton graph or skeleton tree-based methods. The basic idea is as follows: first transform the skeleton or shape axis into an attribute (or relation) graph or a tree structure, called the skeleton tree. Then a graph or tree matching algorithm is used to measure the similarity between models. A detailed review of the skeleton-based method is summarized in the next section.

In this paper, we propose a method for measuring the similarity of 3D models. A skeleton tree constructed by a simple rule is used as a descriptor of a 3D model, which can completely retain its topological features. Based on the skeleton tree, we add topological and geometric information deriving from the model to node and branch features, and their respective feature distances are reasonably defined. The final overall similarity is defined by the weighted sum of topologic and geometric similarity, reflected by similarities in the node and branch features. Compared with related existing methods, our method considers the information on topology and geometry more comprehensively because of taking the node connection and orientation features and geometric feature of the skeleton's points and branches into account, which contributes to high accuracy and good results. Our research work is a significant development in 2/3D model matching, recognition and retrieval.

The remainder of the paper is organized as follows. The next section contains a summary of related work. Section 3 gives an overview of our proposed method. Section 4 develops the skeleton tree construction. Section 5 describes the details in node feature similarity. Section 6 presents the details in branch feature similarity. Section 7 offers overall similarity measurement. Section 8 involves experiment and discussion. Finally, Section 9 concludes and describes future research directions.

2. Related Work

The vast majority of methods in model recognition have concentrated on skeleton-based methods, which are usually based on graph or tree representations of skeletons. These have been well studied by many researchers. Below, we focus on research areas related to the efforts in this paper. For a broad introduction to model recognition method, please refer to any of References [18–22].

In early work, a large number of skeleton graph-based recognition methods were proposed and have achieved good performance on object recognition. Blum [23] transformed the skeleton or medial axis into attribute relation graphs (ARG). The similarity between two objects can be measured by matching their ARGs. Zhu and Yuille [24] used a branch bounding that is confined to animate objects to match the skeleton graphs of objects. Siddiqi et al. [25–29] proposed a kind of ARG, the shock graph, based on shock grammar. The similarity between two 2D objects can be measured by matching their shock graphs. Sundar et al. [30] first transformed skeletons into skeleton graphs by using a minimum

spanning tree (MST) algorithm, then matching it to a skeleton graph. Sebastian et al. [31,32] performed the recognition of shapes by editing shock graphs, defining the cost of the least action path deforming one shape to another as the distance between two shapes. Ruberto [33] took medial axis characteristic points as an attributed skeletal graph (ASG) to model a shape. The matching process for ASGs is based on a revised graduated assignment algorithm. This method can deal with the occlusion problem, but it cannot obtain an optimal matching result due to the heuristic rule. Torsello and Hancock [34,35] measured the similarity of 2D shapes with the help of a shock tree by using the rate of change of boundary length with distance along the skeleton to define this measure. Shokoufandeh et al. [36] described a topological index successfully developed from a shock graph in a large database. The eigenvalues of the adjacency matrices of their subgraphs are used to calculate the similarity between them. Aslan and Tari [37] developed an unconventional matching scheme for shape recognition using skeletons with disconnected branches in the course level. The presented matching algorithms can find the correct results of correspondences and generate a similarity value. Bai and Latecki [38] presented skeleton graph matching based on the similarity of the shortest paths between each pair of endpoints of the pruned skeletons. Xu et al. [39] matched skeleton graphs by comparing the geodesic paths between critical points (junction points and end points). Most of these skeleton graph-based recognition methods are time consuming because of the complexity of the shock grammar, graph matching algorithms, and calculation of eigenvalues. Moreover, these methods do not perform well for 3D object recognition.

More recent work has developed a method for shape recognition that is relatively simple and efficient compared to the skeleton tree-based method. This method transforms the skeleton into a tree structure, called a skeleton tree, according to the construction rule. Hilaga et al. [40] constructed a multi-resolution Reeb graph (MRG) representing the skeletal and topological structure of a 3D model based on geodesic distance. The overall similarity calculation between different 3D models is processed using a graph matching algorithm. This method can cope well with loop structures and generates intuitive results. Nevertheless, this method merely depends on topological features when recognizing different shapes, which may fail in distinguishing different shapes with similar topologies. Pelillo et al. [41] developed a different framework for matching unrooted trees by constructing an association graph with maximal cliques. Liu et al. [42] constructed a free tree structure and used a tree matching scheme to calculate the similarity between two 2D shapes; their method can deal with articulations, stretching, and occlusions. This method does not require any editing of the skeleton graph, but merge, cut, and merge-and-cut operations are essential before matching the free trees. Liu et al. [43] proposed a similarity measurement framework by using the skeleton tree represented by tree descriptor. The similarity between two branches is defined as the weighted sum of the average curvature difference (ACD) and the average area difference (AAD). This approach has the time complexity of $O(n^3)$. As this approach only uses a branch to represent geometric features, it may not include all the geometry information of an object, though this limitation can be improved by taking inherent geometry properties into account. Demirci et al. [44] proposed an accurate matching algorithm by constructing a metric tree representation of the two weighted graphs, which can establish many-to-many correspondences between the nodes of two noisy objects. However, the transformation from graphs to trees has to go through the heuristic rule. In addition, how to choose an optimal root node needs to be considered as this has a great influence on matching results. Xiao [45] recognized microscopic images of diatom cells by using skeleton tree matching, defining topological and geometric differences to establish a similarity mode for microscopic images of *Chaetoceros*, but this method is only suitable for diatom cells recognition. Jiang et al. [46] presented a skeleton graph matching algorithm, namely an order-preserving assignment algorithm, based on a novel tree shape which considers both the positive curvature maxima and the negative curvature minima of the boundary. It has low computational complexity and good performance, but the shape tree does not consider topological structures. Garro and Giachetti [47] introduced a novel framework for non-rigid and textured 3D shape retrieval and classification with the help of TreeSha-based shape representation, which can

offer better similarity recognition and better retrieval results than existing methods on textured and non-textured shape retrieval benchmarks and give effective shape descriptors and graph kernels.

There are other new methods in shape recognition. Chen and Ming [48] proposed a 3D model retrieval system based on the Reeb graph, linked with preprocessing that can accelerate the graph-matching step. Biasotti et al. [49] presented an efficient method for partial shape-matching based on Reeb graphs. Goh [50] described some useful strategies for 2D shape retrieval. These strategies include dynamic part decomposition, local and global measurement, and weighting skeletal segments. The incorporation of these strategies significantly improves shape database retrieval accuracy. Biasotti et al. [51] devised an original framework for 3D model retrieval and classification. Similarity between shapes is measured by attractive features of size functions computed from skeletal signatures. Experimental results demonstrate that this method is efficient and effective. Tierny et al. [52] used a Reeb graph to represent shapes and developed a fast and efficient framework for partial shape retrieval, where partial similarity between two shapes is evaluated by computing their maximum common sub-graph. Zhang et al. [53] achieved 3D non-rigid object retrieval by utilizing integral geodesic distance. Their proposed coarse-to-fine process can reduce the large computational cost of matching. Barra and Biasotti [54] developed a new unsupervised method for 3D shape retrieval based on the extended Reeb graphs. Using kernels as descriptions to measure the similarity between pairs of extended Reeb graphs, their method has been tested on three databases to verify its good performance. Usai et al. [55] presented a novel method for extracting the quad layout of a triangle mesh guided by its accurate curve skeleton; this quad layout is able to reflect the intrinsic characteristics of the shape. Also, this method has applications to semiregular quad meshing and UV mapping, which may provide good shape representation for 2/3D shape matching. Guler et al. [56] presented a SIFT-based image matching framework for 2D planar shape retrieval. Their shape similarity measurement is based on the number of matching internal regions. Yang et al. [57] proposed a novel 2D object matching method based on a hierarchical skeleton capturing the object's topology and geometry, where determining similarity considers both single skeletons and skeleton pairs. Yasseen et al. [58] developed a 2D shape matching method, which can perform a part-to-part matching analysis between two objects' visual protruding parts to measure the distance between them. Yang et al. [59] mentioned a new shape matching method based on the interesting point detector and high-order graph matching. It can consider geometrical relations and reduce computational costs for point matching. Shakeri et al. [60] devised a groupwise shape analysis framework for subcortical surfaces based on spectral marching theory. This spectral matching process can build reliable correspondences between different surface meshes and is likely to help to investigate groupwise structural differences between two study groups. Yang et al. [61] proposed a novel invariant multi-scale descriptor that can capture both local and global information simultaneously for shape representation, matching, and retrieval by adopting the dynamic programming algorithm to conduct shape matching. Since most of these new methods are applied to 2D object recognition, they do not have good applicability and performance on 3D object recognition.

To summarize: many previous methods either have operational complexity or cannot be well used for 3D models because they develop complicated rules for graphing (or tree definition) or pay more attention to 2D modeling. The motivation behind our work is to present a simple method of similarity measurement with high accuracy for 3D model matching, recognition, and retrieval.

3. Overview of Method

Our method easily and efficiently measures the similarity of 3D models. Compared with the skeleton graph or tree-based methods, we can construct an open and linear skeleton tree by a simpler rule. With the help of topologic and geometric information included in a skeleton tree, and by considering extra information on these two aspects, similarity measurements of 3D models can be successfully achieved for 2/3D model matching, recognition, and retrieval.

Initially, a simple rule of skeleton tree construction is proposed with the help of a skeleton, which completely retains the topologic information of the model. Next, the node feature is described

from two sides, the node connection feature and the node orientation feature. The former directly reflects how to connect with the sub-parts in the model. The latter has an ability to distinguish a model with similar topology but a different shape, which is usually overlooked by existing methods. Node connection distance and orientation distance are defined by 2-norm distance and tangent space distance, respectively. The final node feature distance is expressed by the dot product between them. Then the branch feature is used to depict the geometric features of the model, which mainly takes an average regional feature that can reflect the contours of the model and an average bending degree feature that can take bending into account. In calculating average regional distance, we first define three geometric properties for skeleton points, including relative support angle, relative density, and relative anchor point distance. Then the generalized Hausdorff is used to compute the distance between two skeleton point sets. We take this distance as the average regional distance. Average bending degree distance is defined by the curvature of a skeleton branch. Final branch feature distance is expressed by the weighted sum of these two distances. Finally, the overall similarity of skeleton trees is defined as the weighted sum of topologic and geometric similarity, reflected by node and branch feature similarity. The weight of topology and geometry can be adjusted according to different models. If the difference in topology between two models is larger, we give topology a larger weight. If the difference in geometry is larger, we give geometry a larger weight. If topology and geometry have an equal effect on the model, we give them same weight (0.5). A flow chart of our method is shown in Figure 1.

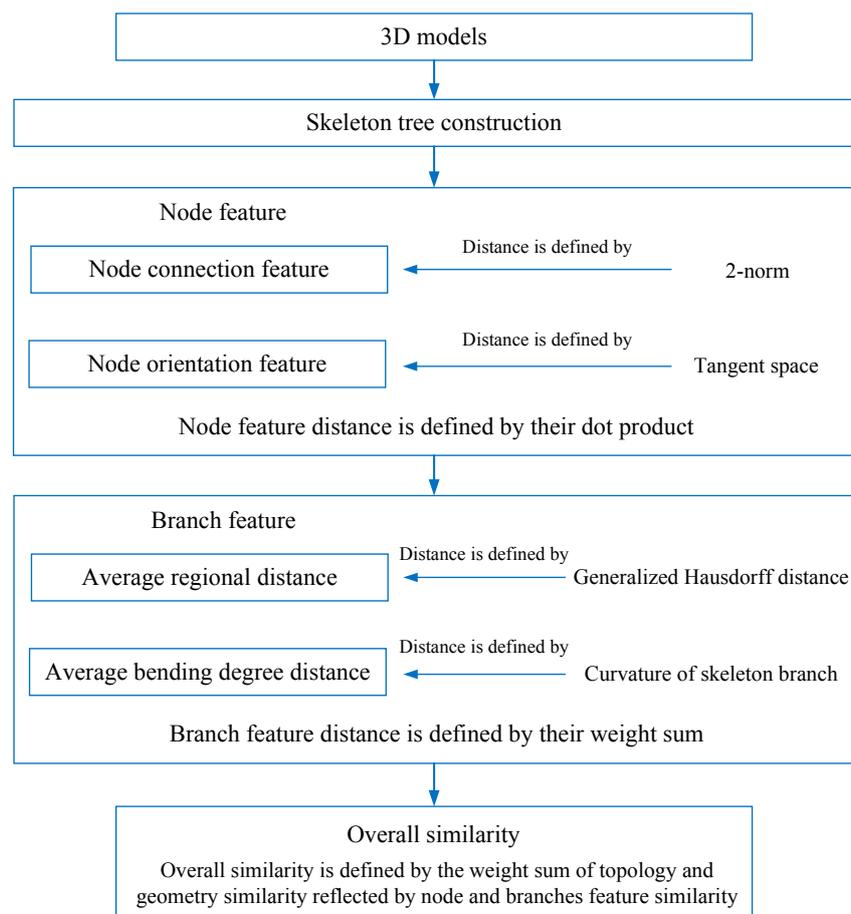


Figure 1. The whole flow of our method.

4. Skeleton Tree Construction

The skeleton is an important foundation for skeleton tree construction. An extracted skeleton should retain the key topological information of model. Here, we use the mesh contraction method proposed by Au et al. [47] to effectively extract a smooth curve-skeleton with correct connectivity and topology. This method is simple to perform and insensitive to noise. The extraction of 3D model skeletons in both experiments uses this method. Next, we map the skeleton to a tree structure named the skeleton tree.

Definition 1. If a skeleton point is only adjacent to one point on the skeleton, it is considered as an endpoint (EP); if a skeleton exists two or more adjacent points on the skeleton, it is considered as a junction point (JP).

Definition 2. Linking any two connected skeleton points to form the sequence constitutes a skeleton branch.

A simple and intuitive method of constructing the skeleton tree follows:

The endpoint and the junction point are selected as the nodes of the skeleton tree. The skeleton branch is a branch of the skeleton tree. We select a skeleton point located as close as possible to the center of model as the top node of skeleton tree, because there is usually important location and shape information in the center of a model. For example, in Figure 2, according to Definition 1, skeleton points (JP_1, JP_2) are both junction points, and skeleton points ($EP_{11}, EP_{12}, EP_{21}$ and EP_{22}) are all endpoints. Selecting JP_1 as the top node of skeleton tree and according to the skeleton's topology, the corresponding skeleton tree can be constructed.

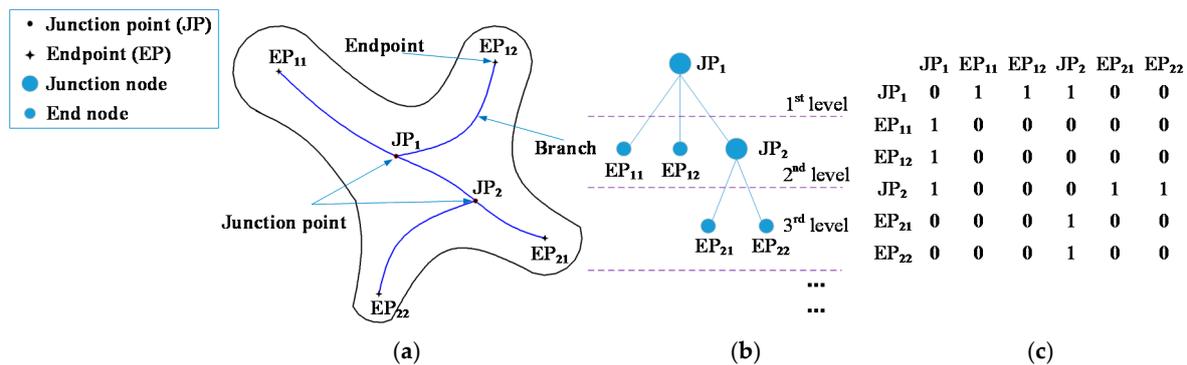


Figure 2. Mapping skeleton to skeleton tree and corresponding adjacent matrix: (a) skeleton and classification of skeleton point; (b) skeleton tree; (c) adjacent matrix.

Before constructing the skeleton tree, we mark the endpoint and junction point by sign and number. The junction point is marked first, by a solid circle and is numbered JP_i ($i = 1, 2, \dots, n$, n is the total number of junction points). Then, the endpoint is marked by the star shape and is numbered by EP_{ij} ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, k$, k is the total number of endpoints connected with i -th junction point). In the skeleton tree, the number of nodes are the same as for the skeleton. The large solid circle represents the junction node and the small circle represents an end node.

Definition 3. In a skeleton tree, the endpoint is considered an end node and the junction point is considered the junction node.

Definition 4. In a skeleton tree, the junction node is considered a root node and an end node is considered a leaf node. From the top node, the level of the top node is 1, its next level is 2, and so on. The nodes belonging to the same layer have the same level number. The upper node is considered the root node of the lower node.

In Figure 2b, JP_1 is the 1st level root node, JP_2 is the 2nd level root node, EP_{11} and EP_{22} are both 2nd level leaf nodes, and EP_{21} and EP_{22} are both 3rd level leaf nodes.

Definition 5. A skeleton tree is described by $ST = (N, B)$, where N is the node set of the tree and B is the branch set of the tree.

Ideally, skeleton trees are linear and open. However, according to the method described in Definition 2, a skeleton tree is likely to be closed. For a ring skeleton tree, at least one node exists for every two or more root nodes. Such a node needs special treatment: assuming that node P has n ($n > 1$) root nodes, copy P to be n duplicates (P_1, P_2, \dots, P_n), then connecting P_i with the i -th root node of P . Simultaneously, P_1 inherits all the leaf nodes of P and it and P_i are both set as leaf nodes. After treating all such nodes by this approach, one open and linear skeleton tree can be obtained.

5. Node Features

In this section, we perform similarity measurements of the node features of the model. Here, the node features include the node topology feature and node orientation feature.

5.1. Node Topology Feature

The Node topology feature directly reflects the connection between sub-parts in the model. From the skeleton shown in Figure 2a, the skeleton points diverge from the inside to the outside of model. The skeleton points like JP (JP_1 and JP_2) nearing the center of the model have a great influence on the topological divergence of the whole model, and skeleton points like EP (EP_{11} , EP_{12} , EP_{21} and EP_{22}), nearing the edge of model, are relatively small. According to this characteristic, we set different weights to the skeleton points located at different positions. The nodes located in the skeleton tree from top to bottom are set adaptive weights from large to small. The adaptive weights can reflect the difference in influence of the whole topology or the divergence of different nodes from the model. The adaptive weight ω_i of each junction node JP_i is set as follows:

$$\begin{aligned} \alpha_i &= L - i + 1 \\ \omega_i &= \alpha_i * \text{deg}(P_i) \end{aligned} \quad (1)$$

where L is the number of levels of the skeleton tree, α_i is the weight of i -th level, and $\text{deg}(P_i)$ is the number of in-degrees and out-degrees of JP_i . Here, the in-degree is the number of root nodes in the upper level of JP_i and the out-degree is the number of leaf nodes in the next level of JP_i . For example, in Figure 2, $\text{deg}(P_i)$ of JP_2 is 3 (in-degree is 1 and out-degree is 2).

The adaptive weight ω_{ij} of each end node EP_{ij} is set as follows:

$$\omega_{ij} = \frac{\omega_i}{k} \quad (2)$$

where k is the number of end nodes connecting with a junction node JP_i .

Using the above descriptions, the adaptive weight of each node in skeleton tree can be determined. At this time, the skeleton tree can be expressed by $ST = (N, B, T_f(\omega))$. T_f represents the node topology feature. Given two skeleton tree $ST_1 = (N1, B1, T_{f1}(\omega))$ and $ST_2 = (N2, B2, T_{f2}(\omega))$, the topology feature distance (TFD) between two nodes (q and p , $q \in ST_1, p \in ST_2$) is defined as follows:

$$\Pi(T_{f1}(q(\omega)), T_{f2}(p(\omega))) = \left(1 + \left| \gamma(T_{f1}(q(\omega))) - \gamma(T_{f2}(p(\omega))) \right| \right) \frac{\|\zeta(T_{f1}(q(\omega))) - \zeta(T_{f2}(p(\omega)))\|_2}{\|\zeta(T_{f1}(q(\omega)))\|_2 + \|\zeta(T_{f2}(p(\omega)))\|_2} \quad (3)$$

where $\gamma(T_f(u))$ ($u \in ST$) is the maximum node degree of skeleton tree, $\zeta(T_f(u))$ ($\zeta(T_f(u)) \in R^{\gamma(T_f(u)) - 1}$) is the topology characteristic vector (TCV) [27] of any

node, and $u(u \in ST)$. $\|\zeta(T_{f_1}(q)) - \zeta(T_{f_2}(p))\|_2$ is a 2-norm. In defining TCV, it should be pointed out that the adjacent matrix of the skeleton tree is the $n \times n$ symmetric matrix, about $\{0,1\}$. If $(i, j) \in B$, the (i, j) th value of the adjacent matrix is 1; otherwise it is 0, as shown in Figure 2c. The smaller the value of TFD is, the larger the topology similarity between two nodes (q and $p, q \in ST_1, P \in ST_2$) becomes.

5.2. Node Orientation Feature

Even if two skeleton trees have similar topologic features, their corresponding models are likely to be different. As shown in Figure 3, Models 1 and 2 have different structures but have the same skeleton trees. To distinguish them, we add orientation feature to node in the skeleton tree. This can be easily achieved by calculating the included angle between two vectors. One vector is formed by JP and an EP connecting with it, and another is formed by JP and another EP connecting with it. The direction of the vector is from JP to EP . In each calculation, setting JP as origin O and establishing the coordinate system $O - XYZ$. Supposing $EP_{11}(x_1, y_1, z_1), EP_{12}(x_2, y_2, z_2)$ and $JP_1(0,0,0)$, the included angle θ between $JP_1\vec{EP}_{11}$ and $JP_1\vec{EP}_{12}$ can be calculated by the following formula.

$$\theta = \arccos \langle JP_1\vec{EP}_{11}, JP_1\vec{EP}_{12} \rangle = \arccos \frac{JP_1\vec{EP}_{11} \times JP_1\vec{EP}_{12}}{|JP_1\vec{EP}_{11}| \times |JP_1\vec{EP}_{12}|} = \arccos \frac{x_1x_2 + y_1y_2 + z_1z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \times \sqrt{x_2^2 + y_2^2 + z_2^2}} \quad (4)$$

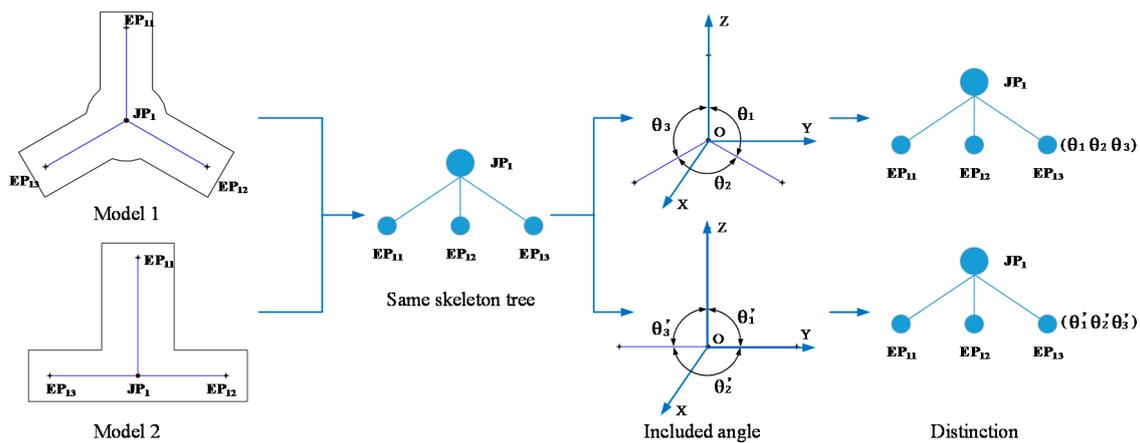


Figure 3. Using the included angle to distinguish the same skeleton tree.

The orientation feature will only appear in each level of skeleton tree. Therefore, we cannot use TFD formula between two nodes to calculate the distance between two levels. We adopt the tangent space method to figure out the distance between included angles. The basic idea is as follows:

Supposing that a level of a skeleton tree has included angles $(\theta_1, \theta_2, \theta_3)$. Starting from θ_1 and defining φ_1 as rotation angle between θ_1 and θ_2 , then $\theta_2 = \theta_1 + \varphi_1$; similarly, $\theta_i = \theta_{i-1} + \varphi_{i-1}$. We define the tangent space description of the included angle as $\vartheta(l)$. The horizontal axis represents the normalized skeleton length, and $l_k = \sum_{i=1}^k L_i / L$, L_i is the sum of two skeleton branch lengths forming one included angle θ . The vertical axis represents the acceleration of the rotation angles $\theta_k = 2\theta_{k-1} + \varphi_{k-1} (k = 2, \dots, n)$, as shown in Figure 4a,b. Through the normalized skeleton length, the domain of the tangent space of included angles is adjusted to 1, which means $\vartheta(l)$ is a function with a domain of $[0,1]$ in R space. $\vartheta(l)$ is a monotonic function. The starting point is a value v and the end point is a value $v + 2\pi$.

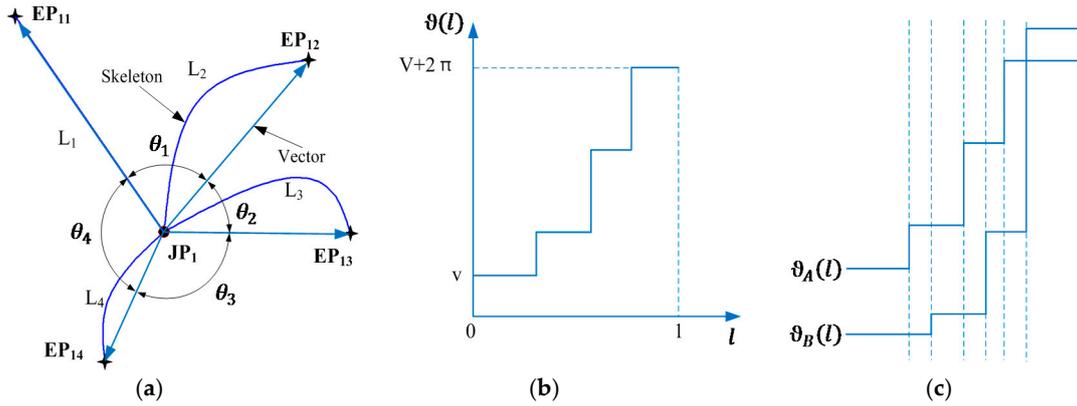


Figure 4. Tangent space description of included angles: (a) included angle; (b) tangent space; (c) shape distance.

After the included angles are described by the tangent space, we can use shape distance to calculate the distance between included angles. Let $A(\theta)$ and $B(\theta)$ be two matching levels. They can respectively be represented by $\vartheta_A(l)$ and $\vartheta_B(l)$ after the tangent space description, as shown in Figure 4c. The tangent space distance (TSD) between $\vartheta_A(l)$ and $\vartheta_B(l)$ is defined as follows:

$$D(\vartheta_A(l), \vartheta_B(l)) = \int_0^1 \left(1 - \left| \frac{\vartheta_A(l) - \vartheta_B(l)}{\max(\vartheta_A(l), \vartheta_B(l))} \right| \right) dl \quad (5)$$

From the definition of tangent space, we can see that tangent space $\vartheta(l)$ will be different if the starting point v is different. It is more meaningful for the tangent space to consider the included angle θ with minimal change. The smaller the value of $D(\vartheta_A(l), \vartheta_B(l))$ is, the greater the shape similarity of the models respectively represented by $A(\theta)$ and $B(\theta)$ becomes.

5.3. Node Feature Distance

In determining the TFD of the node, we should choose two nodes with minimum TFD, and setting this minimum TFD as TFD of the node. The basic idea is as follows: According to the root node priority principle described in the next subsection, we search for two nodes q and p ($q \in ST_1, p \in ST_2$) with minimum TFD and calculate $\prod_{\min} (T_{f1}(q), T_{f2}(p))$. The TFD of each level in the skeleton tree is the accumulation of the TFD of all nodes in this level.

$$\prod(T_{f1i}(q), T_{f2i}(p)) = \sum_{j=1}^k \left(\prod_{\min} (T_{f1i}(q_j), T_{f2i}(p_j)) + \prod(\emptyset) \right) (i = 1, 2, \dots, L) (q \in ST_1, p \in ST_2) \quad (6)$$

where k is the number of nodes in i -th level. If there are some nodes marked \emptyset in the i -th level, $\prod(\emptyset) \neq 0$; otherwise, $\prod(\emptyset) = 0$.

The node feature distance of each level in the skeleton tree is defined as the dot product between the TFD and TSD of each level. The specific formula is as follows:

$$Tdist(T_{f1i}, T_{f2i}) = \prod(T_{f1i}(q), T_{f2i}(p)) \times D(\vartheta_{T_{f1i}}(l), \vartheta_{T_{f2i}}(l)) \quad (7)$$

The node feature distance of the skeleton tree should be the accumulation of that of each level. Given two skeleton trees ST_1 and ST_2 , the steps of calculating $Tdist(T_{f1}, T_{f2})$ are as follows:

1. Initialize $Tdist(T_{f1}, T_{f2}) = 0$;

2. Calculate the value of TFD $\prod(T_{f1}(q), T_{f2}(p))$ between the 1st level of root node q in ST_1 and that p in ST_2 , $Tdist(T_{f1}, T_{f2}) = Tdist(T_{f1}, T_{f2}) + \prod(T_{f1}(q), T_{f2}(p))$;
3. From top to bottom along the skeleton tree, calculate $Tdist(T_{f1}, T_{f2}) = Tdist(T_{f1}, T_{f2}) + Tdist(T_{f1i}, T_{f2i})$. Determine whether there are some nodes marked \emptyset in current level; if yes, $\prod(\emptyset) \neq 0$; if no, $\prod(\emptyset) = 0$.
4. Repeat the above steps until all levels in ST_1 and ST_2 are accessed.

5.4. Root Node Priority Principle

In the calculation of TFD, it is more meaningful for us to search for two nodes (q and p) with minimum TFD. The root node is actually a junction node. We give priority to junction nodes, and then end nodes connecting with them, which can reduce the overall search and calculation time. Before searching for two junction nodes with minimum TFD, all junction nodes are listed in descending order by weight except the first level of junction node. Through descending order, the junction nodes with important topologic features are put into the front, which is beneficial to search for two junction nodes with minimum TFD.

As shown in Figure 5, given two skeleton trees ST_1 and ST_2 , the adaptive weight of each node is calculated. All junction nodes are listed in descending order by weight except the first level of junction node, as shown in Figure 5c1. Then we calculate the TFD between every junction node in ST_1 and all junction nodes in ST_2 , and determine the two junction nodes with minimum TFD, as shown in Figure 5c2. We call such nodes matching nodes and call the branch formed by linking with them the matching branch. In addition, we call the level including these matching nodes the matching level. The rest of the junction nodes are marked by \emptyset , which means these junction nodes have no matching node. Next, we calculate TFD between end nodes that connect with the junction nodes with minimum TFD, as shown in Figure 5c3. The rest of the end nodes are marked by \emptyset as well. We call the branch formed by linking one node with the node marked by \emptyset an empty branch. If there are one-to-more or more-to-one or more-to-more situations, like (EP_{51} , EP_{52} and EP_{53} in ST_1 correspondent to EP_{51} in ST_2), we first calculate TFD between any two nodes, then calculate the average and regard it as the TFD of this situation.

Determining the Value of $\prod(\emptyset)$

If there are some nodes marked by \emptyset in the i -th level of skeleton tree, it means $\prod(\emptyset) \neq 0$, which will increase the TFD of the i -th level. In this subsection, we determine the value of $\prod(\emptyset)$. The node distance density N_ρ is defined as follows:

$$N_\rho = \frac{\prod(u_{top}, u_{bottom})}{N} \quad (8)$$

where $\prod(u_{top}, u_{bottom})$ ($u \in ST$) represents the TFD between the top root node and the bottom in the skeleton tree. N is the number of nodes in the skeleton tree.

The value of $\prod(\emptyset)$ is defined as follows:

$$\prod(\emptyset) = \begin{cases} N_\rho \times \frac{\omega_i}{\omega_{total}} \times n \\ N_\rho \times \frac{\omega_{ij}}{\omega_{total}} \times n \\ N_\rho \times \frac{\omega_{average}}{\omega_{total}} \times n \end{cases} \quad (9)$$

where n is the number of nodes marked by \emptyset in the i -th level, ω_i and ω_{ij} respectively are the weight of the junction node and end node marked by \emptyset in the i -th level, and ω_{total} is the total weight of all nodes

in the skeleton tree. If there are only some junction nodes marked by \emptyset , choose the first formula; if there are only some end nodes marked by \emptyset , choose the second formula. If there are both junction nodes and end nodes marked by \emptyset , choose the third formula. The average weight of junction nodes and end nodes is expressed as $\omega_{average}$:

$$\omega_{average} = \frac{\omega_i \times k + \omega_{ij} \times m}{k + m} \tag{10}$$

where k is the number of junction nodes marked by \emptyset in the i -th level and m is the number of end nodes marked by \emptyset in the i -th level.

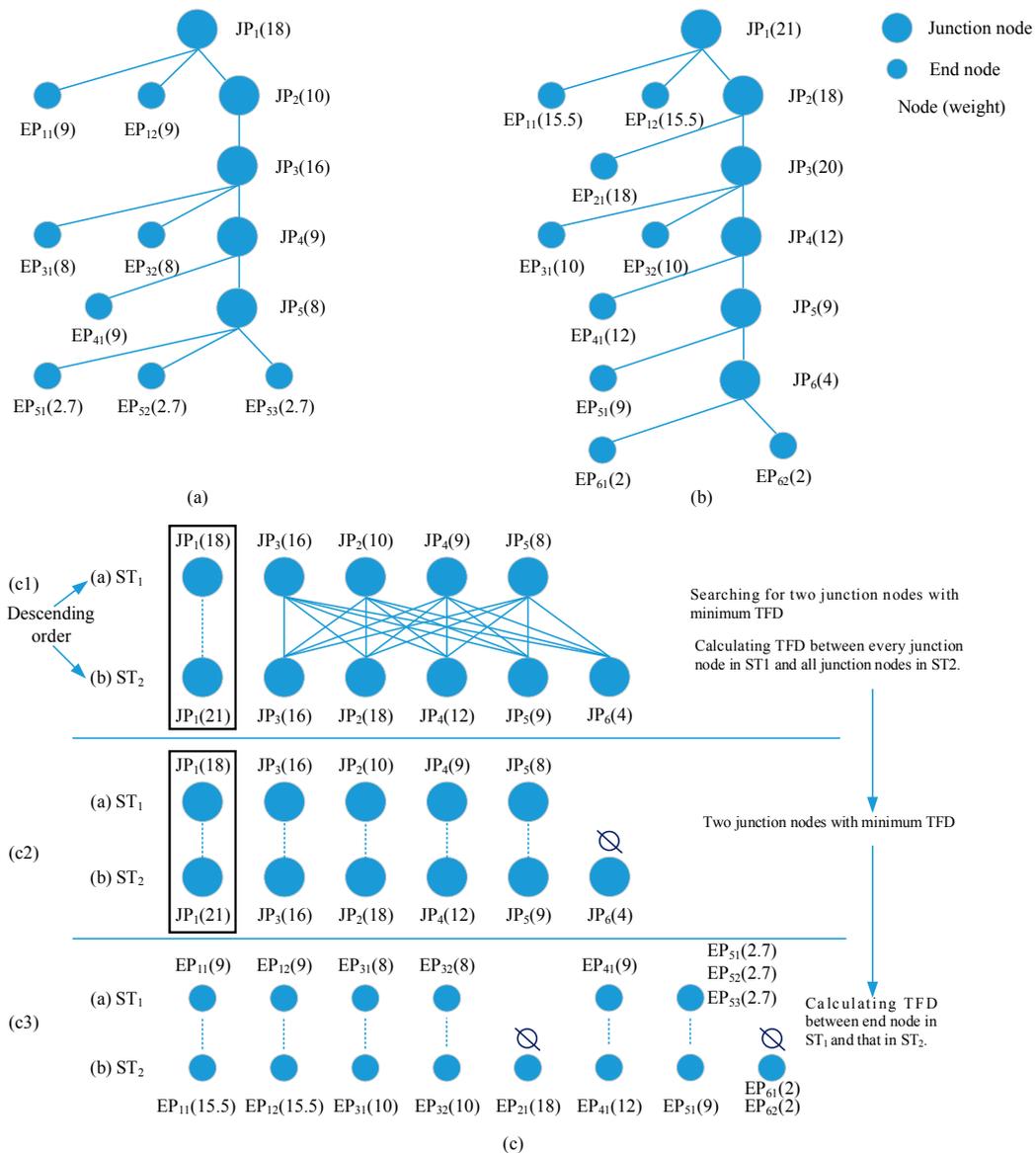


Figure 5. Root node Priority principle: (a) ST₁; (b) ST₂; (c) searching for two nodes with minimum TFD. (c1) descending order for junction nodes in ST₁ and ST₂, (c2) searching for two junction nodes with minimum TFD, (c3) calculating TFD between two end nodes corresponding to two junction nodes in (c2).

6. Branch Feature

Given a 2/3D model, we can extract its skeleton by using a relevant algorithm. Certainly, we can also figure out a model from the topological and geometrical information included in the skeleton, which means that the model and skeleton are inverses of each other. In other words, there is a one-to-one relationship between a skeleton branch and a sub-part in the model. The length and bending of a skeleton branch can express the geometry of a sub-part in the model. Therefore, similarity measurements of geometric features between models can be computed by comparing skeleton branches.

6.1. Geometry Feature of Skeleton Points and Branches

6.1.1. Geometry Features of Skeleton Points

Take a maximum inscribed sphere MS_x with any skeleton point x as the center of the ball and the boundary surface having at least two tangent points. These tangent points are regarded as anchor points of skeleton point x . As shown in Figure 6, A_x and A_x^* are the anchor points of skeleton point x .

- The support angle θ_x of skeleton point x is the minimum angle that takes x as the center and rotates from one anchor point of skeleton point x to the other. We define the relative support angle θ'_x of skeleton point x as $\theta'_x = \theta_x/\pi$ ($0 \leq \theta'_x \leq 1$). The larger the relative support angle is, the higher the symmetry of x becomes.
- We define the relative density r'_x of skeleton point x as $r'_x = r_x/r_{max}$ ($0 \leq r'_x \leq 1$), where r_x is the radius of MS_x and r_{max} is the radius of the maximum inscribed sphere in model.
- We define the relative anchor point distance of x as $d'_x = d_x/L$ ($0 \leq d'_x \leq 1$), where d_x is the geodesic distance between two anchor points and L is the minimum circumference of a box bounding the model. The larger the relative density and relative anchor point distances are, the higher the support range of skeleton point becomes.

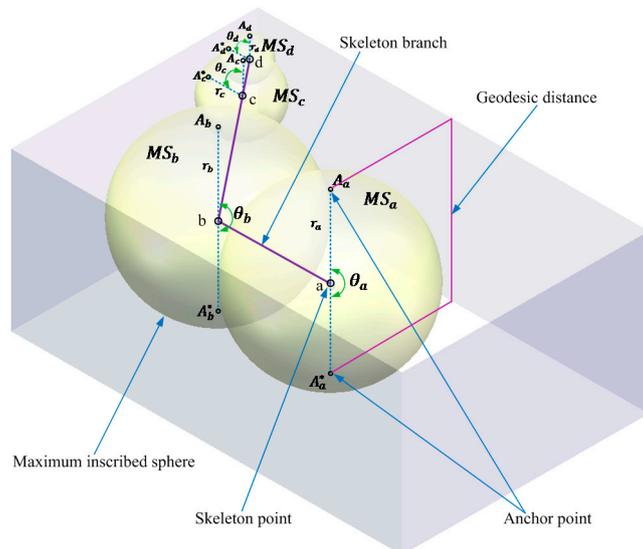


Figure 6. Maximum inscribed sphere and anchor points of a skeleton point.

Relative support angle, relative density, and relative anchor point distance constitute the elements that describe the geometric features of a skeleton point. They have characteristic independence with respect to the directions and scales of skeleton.

Given two skeleton branches Γ_1 and Γ_2 , the geometry feature difference $\varepsilon(\Gamma_1(q), \Gamma_2(p))$ between two skeleton points $q(q \in ST_1)$ and $p(p \in ST_2)$ can be calculated by Hausdorff [62]. The Hausdorff

distance is able to efficiently measure distance between two point sets O and E . We define $E = \{e_1, e_2, \dots, e_m\}$ as the set of skeleton points. The Hausdorff distance between O and E is represented by $H(O, E)$. To avoid the influence of a disturbance point, we use a generalized Hausdorff distance.

$$H(O, E) = \max\{h(O, E), h(E, O)\} \quad (11)$$

where

$$h(O, E) = \max_{o \in O} \min_{e \in E} \|o - e\| \quad (12)$$

$$h(E, O) = \max_{e \in E} \min_{o \in O} \|e - o\| \quad (13)$$

The geometry feature distance $D(\Gamma_1(q), \Gamma_2(p))$ between two skeleton points $q (q \in ST_1)$ and $p (p \in ST_2)$ can be calculated by the following formula:

$$D(\Gamma_1(q), \Gamma_2(p)) = \exp\left(\prod (T_{f_1}(q), T_{f_2}(p)) \varepsilon(\Gamma_1(q), \Gamma_2(p))\right) \quad (14)$$

6.1.2. Geometric Features of a Skeleton Branch

The geometric features of a skeleton branch consist of an average regional feature and an average bending degree feature. Their feature distances are defined as the average regional distance and average bending degree distance, respectively. The former can depict the difference in contour between models. The latter can depict the difference in curvature between sub-parts represented by skeleton branches.

Defining the average regional distance of a skeleton branch as the sum of the distance between skeleton points in that branch. As we only know two endpoints of a skeleton branch, we cannot directly calculate the sum. A good method is to distinguish skeleton branches through the equal-distance method. The average regional distance of skeleton branch $B_D(\Gamma_1(q), \Gamma_2(p))$ can be calculated by the following formula:

$$B_D(\Gamma_1(q), \Gamma_2(p)) = \frac{1}{L} \sum_{j=0}^m D(\Gamma_1(q_j), \Gamma_2(p_j)) \quad (15)$$

where $D(\Gamma_1(q_j), \Gamma_2(p_j))$ represents the geometric feature distance between two discrete skeleton points $q_j (q_j \in ST_1)$ and $p_j (p_j \in ST_2)$ in a certain skeleton branch and $L (L = (L_1 + L_2)/2)$ is the average length of two skeleton branches Γ_1 and Γ_2 .

The average bending degree distance can be calculated by the following formula:

$$B_curvature(\Gamma_1, \Gamma_2) = \frac{1}{L} \left(\int_0^{L_1} |\kappa_1(l)| dl - \int_0^{L_2} |\kappa_2(l)| dl \right) \quad (16)$$

where $\kappa_1(l)$ and $\kappa_2(l)$ are curvatures of skeleton branch Γ_1 and Γ_2 , respectively, and l is the arc-length parameterization of the skeleton branch.

6.2. Branch Feature Distance

We define the branch feature distance as the weighted sum of the average regional distance and the average bending degree distance. In fact, the branch feature distance is equal to the geometry feature distance (GFD).

$$B_Gdist(\Gamma_1, \Gamma_2) = \tau_1 B_D(\Gamma_1(q), \Gamma_2(p)) + \tau_2 B_curvature(\Gamma_1, \Gamma_2) \quad (17)$$

$$\tau_1 + \tau_2 = 1 (\tau_1, \tau_2 \in [0, 1])$$

where τ_1 and τ_2 indicate the importance of the average regional feature and the average bending degree feature.

The GFD of each level in a skeleton tree is the accumulation of that of all skeleton branches in this level.

$$B_Gdist(\Gamma_{1i}, \Gamma_{2i}) = \sum_{j=1}^k (B_Gdist(\Gamma_{1ij}, \Gamma_{2ij}) + B_Gdist(\emptyset)) \quad (18)$$

where $B_Gdist(\emptyset)$ represents the GFD between empty branches. The GFD of a skeleton tree should be the accumulation of that of each level. Given two skeleton trees ST_1 and ST_2 , the steps of calculating $B_Gdist(\Gamma_1, \Gamma_2)$ are as follows:

- Initialize $B_Gdist(\Gamma_1, \Gamma_2) = 0$;
- The GFD of the first level equals zero, $B_Gdist(\Gamma_1, \Gamma_2) = 0$;
- From the second level to the bottom along the skeleton tree, calculate $B_Gdist(\Gamma_1, \Gamma_2) = B_Gdist(\Gamma_1, \Gamma_2) + B_Gdist(\Gamma_{1i}, \Gamma_{2i})$. Determine whether there are some empty branches in the i -th level: if yes, $B_Gdist(\emptyset) \neq 0$; if no, $B_Gdist(\emptyset) = 0$.
- Repeat the above steps until all levels in ST_1 and ST_2 are accessed.

Determining the Value of $B_Gdist(\emptyset)$

If there are some empty branches in the i -th level of skeleton tree, it means $B_Gdist(\emptyset) \neq 0$, which will increase the GFD of the i -th level. In this subsection, we determine the value of $B_Gdist(\emptyset)$. The branch distance density B_ρ is defined as follows:

$$B_\rho = \frac{\prod(u_{top}, u_{bottom})}{M} \quad (19)$$

where $\prod(u_{top}, u_{bottom}) (u \in ST)$ represents the TFD between the top root node and the bottom in the skeleton tree and M is the number of branches in the skeleton tree.

The value of $B_Gdist(\emptyset)$ is defined as follows:

$$B_Gdist(\emptyset) = \begin{cases} B_\rho \times \frac{\omega_i}{\omega_{total}} \times m \\ B_\rho \times \frac{\omega_{ij}}{\omega_{total}} \times m \\ B_\rho \times \frac{\omega_{average}}{\omega_{total}} \times m \end{cases} \quad (20)$$

where m is the number of empty branches in the i -th level. Other parameters are the same, with descriptions in Section 5.4.

7. Overall Similarity Measurement

We define the overall similarity of skeleton trees as the weighted sum of topologic and geometric similarity reflected by node and branch feature similarity.

The topologic similarity of skeleton trees can be calculated by the following formula:

$$Tsim(ST_1, ST_2) = \frac{1}{Tdist(T_{f1}, T_{f2})} \quad (21)$$

The geometric similarity of skeleton trees can be calculated by the following formula:

$$Gsim(ST_1, ST_2) = \frac{1}{B_Gdist(\Gamma_1, \Gamma_2)} \quad (22)$$

The overall similarity of skeleton trees can be calculated by the following formula:

$$Osim(ST_1, ST_2) = \kappa_T Tdist(T_{f1}, T_{f2}) + \kappa_G Gsim(ST_1, ST_2) \quad (23)$$

$$\kappa_T + \kappa_G = 1 (\kappa_T, \kappa_G \in [0, 1])$$

where κ_T and κ_G are the adaptive weights of the topologic and geometric features. They can be adjusted according to different models. The larger the value of $Osim(ST_1, ST_2)$ is, the higher the similarity of the models becomes.

8. Experiments and Discussion

8.1. Experiment One

Table 1 shows 10 typical 3D models and their skeletons. Table 2 shows their similarity results. The results were obtained by using an Intel Pentium-M 3.0 GHz processor notebook PC with a 2.0 G memory and VC++6.0 and OpenGL software. The skeletons of the models were extracted by mesh contraction [63]. The data in Table 2 were treated by normalizing the similarity results of the two models named in the rows and columns. Normalization processing means the data in a row divides the maximum data in this row, which can make data in [0,1]. The larger the data is, the more similar a model becomes. Completely dissimilar is expressed as 0 and 1 means totally similar. The numbers in bold represent the similarity of a model very similar to a model in this row except itself. From Table 2, we can see that the models with similar topology and different geometry, like 1 Dog and 5 Man I, 7 Man II and 9 Horse, can be distinguished well by our proposed method.

Table 1. The models and their skeletons for experiment.

Number and Name	1 Dog	2 Dolphin	3 Table	4 Statue	5 Man I
Model and skeleton					
Number and Name	6 Pillar	7 Man II	8 Pipeline	9 Horse	10 Chair
Model and skeleton					

In this experiment, $\tau_1 = 0.6$ and $\tau_2 = 0.4$. The differences in contour are obvious, which increases the weight of the average regional features. The skeleton curve of the models is relatively straight, which makes the average weights of the bending degree small. In addition, we consider the topologic features and geometric features as having an equally important influence on the models, which means $\kappa_T = \kappa_G = 0.5$. We can flexibly adjust τ_1 , τ_2 , κ_T and κ_G according to different models.

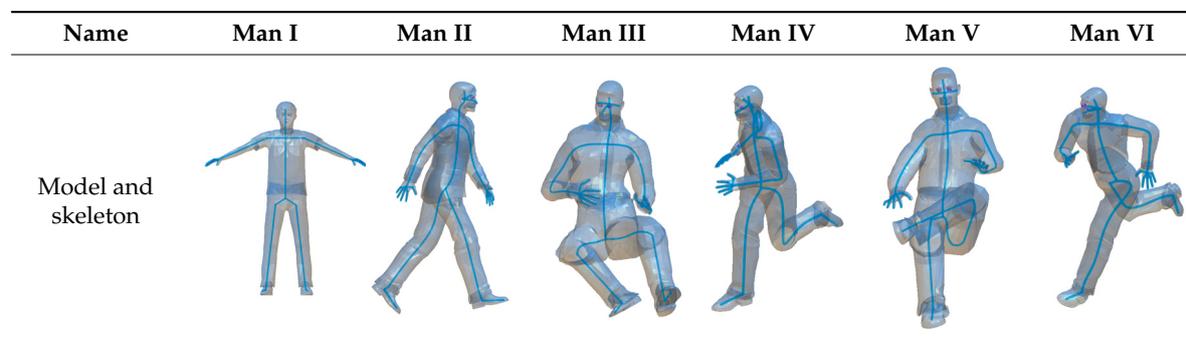
Table 2. The similarity results between different models ($\tau_1 = 0.6$, $\tau_2 = 0.4$ and $\kappa_T = \kappa_G = 0.5$).

Number and Name	1 Dog	2 Dolphin	3 Table	4 Statue	5 Man I	6 Pillar	7 Man II	8 Pipeline	9 Horse	10 Chair
1 Dog	1.000	0.246	0.325	0.083	0.442	0.064	0.462	0.228	0.852	0.284
2 Dolphin	0.308	1.000	0.062	0.249	0.314	0.103	0.288	0.886	0.316	0.076
3 Table	0.323	0.058	1.000	0.064	0.236	0.103	0.220	0.102	0.226	0.925
4 Statue	0.102	0.305	0.058	1.000	0.084	0.858	0.109	0.263	0.051	0.401
5 Man I	0.472	0.306	0.261	0.062	1.000	0.063	0.953	0.256	0.454	0.243
6 Pillar	0.072	0.082	0.072	0.831	0.060	1.000	0.072	0.064	0.063	0.102
7 Man II	0.466	0.294	0.208	0.085	0.968	0.054	1.000	0.252	0.434	0.214
8 Pipeline	0.283	0.837	0.107	0.227	0.238	0.078	0.248	1.000	0.241	0.092
9 Horse	0.894	0.323	0.234	0.038	0.483	0.074	0.413	0.207	1.000	0.186
10 Chair	0.307	0.057	0.906	0.486	0.227	0.087	0.187	0.103	0.204	1.000

Furthermore, we offer the experiment of similarity measurements of the same model with different postures, namely deformation, which is a key component of skinning animation [64]. It is important to note that we do not need to compute TSD because we are using the same model. This means the item TSD should be removed from Equation (7) and Equation (7) becomes as follows:

$$Tdist(T_{f1i}, T_{f2i}) = \prod (T_{f1i}(q), T_{f2i}(p)) \quad (24)$$

Table 3 shows five different postures of a man and their skeletons. Table 4 shows the similarity results. In this experiment, we need to reduce the difference in bending degree of skeleton branches, which can be done by giving the average bending degree feature a smaller weight ($\tau_1 = 0.8$ and $\tau_2 = 0.2$). From Table 4, we can find that the similarity value is almost more than 0.945, which satisfies our expectation.

Table 3. Five different postures of man and their skeletons.**Table 4.** The similarity results between different postures of man ($\tau_1 = 0.8$, $\tau_2 = 0.2$ and $\kappa_T = \kappa_G = 0.5$).

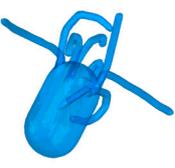
Name	Man I	Man II	Man III	Man IV	Man V	Man VI
Man I	1.000	0.953	0.947	0.950	0.962	0.956
Man II	0.968	1.000	0.971	0.963	0.957	0.959
Man III	0.955	0.964	1.000	0.945	0.974	0.968
Man IV	0.944	0.946	0.967	1.000	0.982	0.951
Man V	0.961	0.952	0.975	0.950	1.000	0.948
Man VI	0.954	0.967	0.948	0.966	0.975	1.000

8.2. Experiment Two

To further demonstrate the retrieval performance of the proposed method, a test dataset was constructed and used to carry out the matching and classification of the models. This dataset had the same number of elements of each class and consisted of regular 3D models with six classes of five

elements, as shown in Table 5. Most of the models represented articulated objects, and five elements in same class showed different complex poses, to make the matching work sufficiently complex. The original models of our dataset were collected from Shape Retrieval Contest 2014 (SHREC'14) [65]. Each model was uniformly scaled in a unit sphere, centered in the origin of the Cartesian coordinate, to make results independent of scale operations.

Table 5. Our testing dataset.

Name	Snake I	Snake II	Snake III	Snake IV	Snake V
3D Model					
Name	Puppet I	Puppet II	Puppet III	Puppet IV	Puppet V
3D Model					
Name	Animal I	Animal II	Animal III	Animal IV	Animal V
3D Model					
Name	Mickey I	Mickey II	Mickey III	Mickey IV	Mickey V
3D Model					
Name	Jellyfish I	Jellyfish II	Jellyfish III	Jellyfish IV	Jellyfish V
3D Model					
Name	Hand I	Hand II	Hand III	Hand IV	Hand V
3D Model					

Several retrieval evaluation measures have been used to model matching tasks including nearest neighbor (NN), the precision-recall plot, first tier (FT), and second tier (ST) [66]. The basic information of precision-recall is introduced as follows.

The precision-recall plot is one of the most popular evaluation criteria used to measure the performance of retrieval systems [67]. Its ordinate is accuracy or precision and abscissa is recall. Precision is the ratio between the number of correct models returned from a system and that of all models returned from that system, and recall is the ratio between the number of correct models returned from a system and that of the relevant models returned from that system. In general, both precision and recall are related to the number of models returned from system. Recall and the number of models returned from a system are in direct ratio, and precision and that are in inverse ratio. Assume that A is a collection of all relevant models and B is a collection of all models returned from system. Precision and recall can be expressed by the following formulas:

$$precision = \frac{A \cap B}{B} \quad (25)$$

$$recall = \frac{A \cap B}{A} \quad (26)$$

Precision and recall become higher and better. From overall consideration of these two criterions, the area surrounded by the plot and the axis is larger, which means that the retrieval performance is better.

For 3D model matching and classification, we ran the proposed method and the methods with the best performance in the SHREC'14 on our testing dataset and compared the results obtained by the former and the results obtained by the latter in terms of NN, FT and ST. The performance of the latter has been shown in Reference [47]. The comparative performance results between methods are shown in Table 6.

Table 6. The comparative performance results between methods.

Methods \ Evaluation Measures	NN	FT	ST
GG2	0.958	0.383	0.504
Gi2	0.909	0.430	0.559
Gi3	0.963	0.436	0.562
Ve1	0.918	0.398	0.499
The proposed method	0.908	0.448	0.572

The numbers in bold represent the method has the best retrieval performance in one certain evaluation measure compared to other methods.

Our method provides the best results in the scores of FT and ST and is greater than 0.9 in the score of NN, which shows a good ability to match and classify models.

Precision-recall plots for six selected classes in our testing dataset and averages over all models for SHREC'14 are shown in Figure 7. The classes snake, hand, and jellyfish have the best results, though the classes puppet, Mickey, and animal obtain good results as well, which verifies the good matching performance of the proposed method.

Next, we use our dataset to compare the retrieval performance of the proposed method with respect to the MRG-based method [40]. Table 7 summarizes the capability comparison. The precision and recall of the proposed method are both higher than those of the MRG-based method. So the retrieval accuracy of the former is higher under the same conditions.

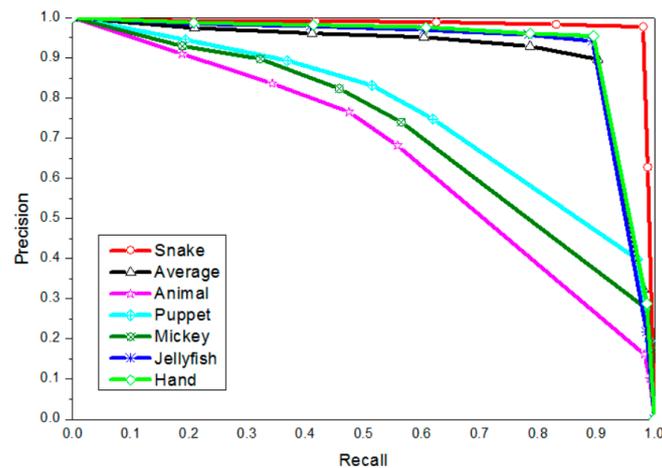


Figure 7. Precision-recall plots for six selected classes in our testing dataset and average over all models for SHREC'14.

Table 7. Capability comparison.

Retrieval Method	Average Recall (%)	Average Precision (%)
The MRG-based method	63.5	66.5
The proposed method	65.2	69.8

Figure 8a compares the average rank [68] for our testing dataset using the proposed method with the values obtained by the MRG-based method. The average rank was obtained by following two steps: first, every model in the testing dataset was performed as a query. Then the retrieval ranks of all elements in the class of the query were computed. The proposed method has the lowest value; note that the lower the average rank value is, the better the performance.

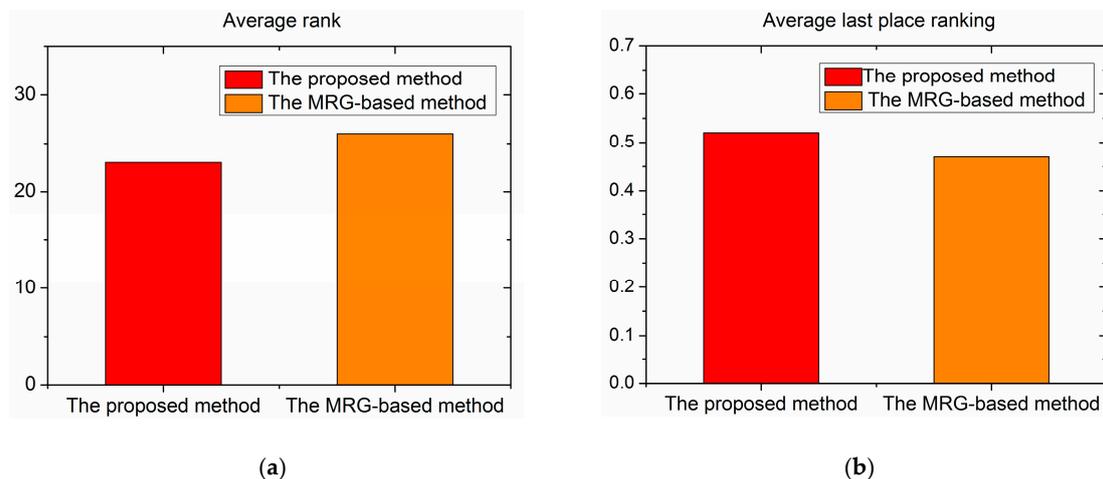


Figure 8. Comparison between the proposed method and the MRG-based method: (a) average rank; (b) average last place ranking.

Another measure, the average last place ranking [69], was also adopted to evaluate performance. It is defined as

$$L_n = 1 - \frac{Rank_l - n}{N - n} \quad (27)$$

where $Rank_l$ represents the rank at which the last relevant model is found, n is the number of relevant models, and N is the size of the overall dataset. Figure 8b shows the average last place ranking of

values obtained by two methods, respectively. This value represents the expectation that the user has retrieved all relevant models from the dataset. The higher this measure value in the range [0,1], the more the number of relevant models to find, indicating better results.

Finally, Figure 9 shows the precision-recall plots for four selected classes—snake, hand, animal and puppet—computed by the proposed method and the MRG-based method, respectively. It is worth remembering that curves moved upwards and to the right represent better retrieval performance. The curves obtained by the proposed method are higher than the curves obtained by the MRG-based method, which means the proposed method has a better retrieval performance and higher retrieval accuracy under the same conditions.

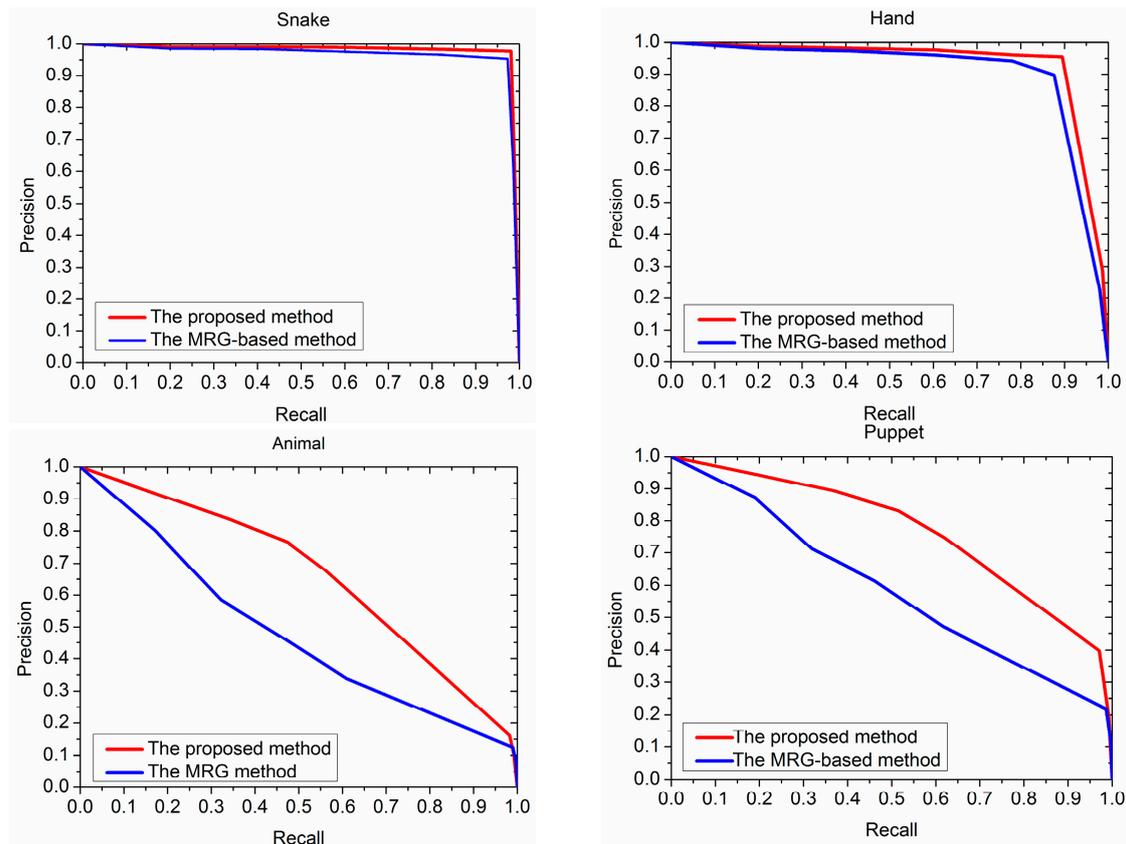


Figure 9. The precision-recall plots for four selected classes computed by the proposed method and the MRG-based method, respectively.

9. Conclusions and Further Work

The main contribution of this paper is that it proposes a simple method of similarity measurement of 3D models by using skeleton trees as descriptors of 3D models. The improvements of the proposed method are as follows:

- Using skeleton trees is simpler and more efficient than other methods of model expression such as attributed-relation graph, shock graphs, and so on. Compared to them, the skeleton tree construction rule is relatively simple, and it carries complete topological information of a model.
- The node feature contains both connection features, reflecting topology, and orientation features, distinguishing different modes with similar topology by their included angles. It uses 2-norm and tangent space to reasonably define TFD and TSD, respectively. The final node feature distance is expressed by the dot product between them.
- The branch feature can depict the geometric features of a model. It consists of the average regional feature and average bending degree feature. Their feature distances can reflect differences in

contour and bending, and are computed by generalized Hausdorff distance and curvature of branches, respectively. Final branch feature distance is expressed by their weighted sum. These two weights are adaptive.

- Overall similarity is defined by the weight sum of topologic and geometric similarity. These two weights can be adjusted according to different models. This method is able to produce good results for different models and for the same model with different postures, as proven by experiment.
- Several enhancements can be added to our algorithm:
- The skeleton tree-based descriptor of a 3D model can be optimized by using skeleton pruning algorithms and constructing multi-level skeleton trees.
- Geometric features can be more fully described by taking more geometric properties into account, such as minimum bounding box, circularity, eccentricity and so on.
- The efficiency of similarity measurements of whole skeleton trees can be improved by the maximal isomorphic subtree formation algorithm or level clustering algorithm.

We believe that this method will greatly expand the application of 2/3D model matching, recognition, and retrieval.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (51305443, 51475459), the Natural Science Foundation of Jiangsu Province (bk20130184, bk20160258), the State Key Laboratory of Materials Processing and Die & Mould Technology, Huazhong University of Science and Technology (P2016-18), and a project funded by Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

Author Contributions: Jingbin Hao and Hao Liu conceived and designed the experiments; Zhengtong Han and Shengping Ye contributed analysis tools; Xin Chen performed the experiment; Xin Chen and Jingbin Hao analyzed the data; and Xin Chen wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Casasent, D.P. 3D CAD-based object recognition for a flexible assembly cell. In Proceedings of the SPIE 1996, European, Denver, CO, USA, 4–9 October 1996; Volume 2904, pp. 167–177.
2. Majumdar, J.; Seethalakshmy, A.G. A CAD Model Based System for Object Recognition. *J. Intell. Robot. Syst.* **1997**, *18*, 351–365. [[CrossRef](#)]
3. Tsai, C.L.; Feng, J.H.; Lin, S.W.; Cheng, W.L.; Huang, W.C.; Liu, R.G. Pattern Recognition for Integrated Circuit Design. 2014. Available online: <http://www.freepatentsonline.com/8751976.html> (accessed on 1 February 2014).
4. Chang, Y.L. Spatial Cognition in Digital Cities. *Int. J. Archit. Comput.* **2003**, *1*, 471–488. [[CrossRef](#)]
5. Groves, P.M. Review of Signal analysis and Pattern recognition in biomedical engineering. *Psycritiques* **1976**, *21*, 897. [[CrossRef](#)]
6. Albano, S. Military Recognition of Family Concerns: Revolutionary War to 1993. *Armed Forces Soc.* **1994**, *20*, 283–302. [[CrossRef](#)]
7. Gadh, R.; Lu, Y.; Tautges, T.J. Volume Decomposition and Feature Recognition for Hexahedral Mesh Generation. In Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, CA, USA, 10–13 October 1999; pp. 269–280.
8. Aleotti, J.; Caselli, S. Grasp recognition in virtual reality for robot pregrasp planning by demonstration. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2801–2806.
9. Lee, J.H.; Jeon, H.J.; Kim, K.A.; Nam, H.W.; Woo, J.T.; Ahn, K.J. Diabetes Education Recognition Program. *J. Korean Diabetes* **2012**, *13*, 219–223. [[CrossRef](#)]
10. Aleksic, P.S.; Katsaggelos, A.K. Automatic Facial Expression Recognition Using Facial Animation Parameters and Multi-Stream Hmms. *IEEE Trans. Inf. For. Secur.* **2006**, *1*, 3–11. [[CrossRef](#)]

11. Godil, A. Applications of 3D shape analysis and retrieval. In Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 14–16 October 2009; pp. 1–4.
12. Iyer, N.; Kalyanaraman, Y.; Lou, K.; Jayanti, S.; Ramani, K. A Reconfigurable 3D Engineering Shape Search System: Part I—Shape Representation. In Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, USA, 2–6 September 2003; pp. 89–98.
13. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Shape distributions. *ACM Trans. Graph.* **2002**, *21*, 807–832. [[CrossRef](#)]
14. Ohbuchi, R.; Otagiri, T.; Ibato, M.; Takei, T. Shape-Similarity Search of Three-Dimensional Models Using Parameterized Statistics. In Proceedings of the IEEE 10th Pacific Conference on Computer Graphics and Applications, Beijing, China, 9–11 October 2002; pp. 265–274.
15. Tangelder, J.W.H.; Veltkamp, R.C. Polyhedral Model Retrieval Using Weighted Point Sets. *Int. J. Image Graph.* **2003**, *3*, 119. [[CrossRef](#)]
16. Kazhdan, M.; Chazelle, B.; Dobkin, D.; Funkhouser, T.; Rusinkiewicz, S. A Reflective Symmetry Descriptor for 3D Models. *Algorithmica* **2004**, *38*, 201–225. [[CrossRef](#)]
17. Min, P.; Chen, J.; Funkhouser, T. A 2D sketch interface for a 3D model search engine. In Proceedings of the ACM SIGGRAPH 2002 Conference Abstracts and Applications, San Antonio, TX, USA, 21–26 July 2002; p. 138.
18. Tangelder, J.W.H.; Veltkamp, R.C. A Survey of Content Based 3D Shape Retrieval Methods. *Multimed. Tools Appl.* **2008**, *39*, 441–471. [[CrossRef](#)]
19. Zhang, T.; Peng, Q. Shape Distribution-Based 3D Shape Retrieval Methods: Review and Evaluation. *CAD Appl.* **2013**, *6*, 721–735. [[CrossRef](#)]
20. Pandey, D.; Singh, P. A Review of Shape Recognition Techniques. *Int. J. Emerg. Res. Manag. Technol.* **2014**, *3*, 40–43.
21. Savelonas, M.A.; Pratikakis, I. An overview of partial 3D object retrieval methodologies. *Multimed. Tools Appl.* **2015**, *74*, 11783–11808. [[CrossRef](#)]
22. Arulmozhi, P.; Abirami, S. Shape Based Image Retrieval: A Review. *Int. J. Comput. Sci. Eng.* **2014**, *6*, 147–153.
23. Blum, H. Biological shape and visual science: Part I. *J. Theor. Biol.* **1973**, *38*, 205–287. [[CrossRef](#)]
24. Zhu, S.C.; Yuille, A.L. FORMS: A flexible object recognition and modeling system. In Proceedings of the International Conference on Computer Vision, Boston, MA, USA, 20–23 June 1995; Volume 20, pp. 187–212.
25. Siddiqi, K.; Kimia, B.B. Parts of Visual Form: Computational Aspects. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 239–251. [[CrossRef](#)]
26. Siddiqi, K.; Kimia, B.B. A shock grammar for recognition. In Proceedings of the IEEE Computer Society Conference Computer Vision and Pattern Recognition, San Francisco, CA, USA, 18–20 June 1996; pp. 507–513.
27. Siddiqi, K.; Shkoufandeh, A.; Dickinson, S.; Zucker, S. Shock Graphs and Shape Matching. In Proceedings of the IEEE Sixth International Conference on Computer Vision, Bombay, India, 4–7 January 1998; pp. 222–229.
28. Siddiqi, K.; Kimia, B.B.; Tannenbaum, A.; Zucker, S. Shocks, Shapes, and Wiggles. *Image Vis. Comput.* **1999**, *17*, 365–373. [[CrossRef](#)]
29. Siddiqi, K.; Shkoufandeh, A.; Dickinson, S.; Zucker, S. Shock Graphs and Shape Matching. *Int. J. Comput. Vis.* **1999**, *35*, 13–32. [[CrossRef](#)]
30. Sundar, H.; Silver, D.; Gagvani, N.; Dickinson, S. Skeleton Based Shape Matching and Retrieval. In Proceedings of the IEEE International Conference on Shape Modeling and Applications, Seoul, Korea, 12–15 May 2003; pp. 130–290.
31. Sebastian, T.B.; Klein, P.; Kimia, B.B. Recognition of Shapes by Editing Shock Graphs. In Proceedings of the International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; pp. 755–762.
32. Sebastian, T.B.; Klein, P.N.; Kimia, B.B. Recognition of Shapes by Editing Their Shock Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 550–571. [[CrossRef](#)] [[PubMed](#)]
33. Ruberto, C.D. Recognition of Shapes by Attributed Skeletal Graphs. *Pattern Recognit.* **2004**, *37*, 21–31. [[CrossRef](#)]
34. Torsello, A.; Hancock, E.R. A Skeletal Measure of 2D Shape Similarity. *Comput. Vis. Image Underst.* **2004**, *95*, 1–29. [[CrossRef](#)]
35. Torsello, A.; Hidovic-Rowe, D.; Pelillo, M. Polynomial-Time Metrics for Attributed Trees. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1087–1099. [[CrossRef](#)] [[PubMed](#)]

36. Shokoufandeh, A.; Macrini, D.; Dickinson, S.; Siddiqi, K.; Zucker, S.W. Indexing Hierarchical Structures Using Graph Spectra. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1125–1140. [[CrossRef](#)] [[PubMed](#)]
37. Aslan, C.; Tari, S. An Axis Based Representation for Recognition. In Proceedings of the Tenth IEEE International Conference on Computer Vision, Beijing, China, 17–20 October 2005; pp. 1339–1346.
38. Bai, X.; Latecki, L.J. Path similarity skeleton graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1282–1292. [[PubMed](#)]
39. Xu, Y.; Wang, B.; Liu, W.; Bai, X. Skeleton graph matching based on critical points using path similarity. In Proceedings of the 9th Asian Conference on Computer Vision, Xi'an, China, 23–27 September 2009; Volume 5996, pp. 456–465.
40. Hilaga, M.; Shinagawa, Y.; Kohmura, T.; Kunii, T. Topology matching for fully automatic similarity estimation of 3d shape. In Proceedings of the SIGGRAPH 2001, Los Angeles, CA, USA, 14–16 August 2001; pp. 203–212.
41. Pelillo, M. Matching Free Trees, Maximal Cliques, and Monotone Game Dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1535–1541. [[CrossRef](#)]
42. Geiger, D.; Liu, T.L.; Kohn, R.V. Representation and self-similarity of shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 86–99. [[CrossRef](#)]
43. Liu, W.Y.; Liu, J.T. Objects similarity measure based on skeleton tree descriptor matching. *J. Infrared Millim. Waves* **2005**, *24*, 432–436.
44. Demirci, F.; Shokoufandeh, A.; Keselman, Y.; Bretzner, L.; Dickinson, S. Object Recognition as Many-to-Many Feature Matching. *Int. J. Comput. Vis.* **2006**, *69*, 203–222. [[CrossRef](#)]
45. Qiao, X. Research for Skeleton Tree Matching of Microscopic Image of Diatom Cells. Available online: https://www.researchgate.net/publication/286843305_Research_for_skeleton_tree_matching_of_microscopic_image_of_diatom_cells (accessed on 22 April 2017).
46. Jiang, B.; Tang, J.; Luo, B.; Chen, Z. Skeleton graph matching based on a novel shape tree. In Proceedings of the ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM 2009), Sanya, China, 8–9 August 2009; Volume 4, pp. 636–639.
47. Garro, V.; Giachetti, A. Scale Space Graph Representation and Kernel Matching for Non Rigid and Textured 3D Shape Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1258–1271. [[CrossRef](#)] [[PubMed](#)]
48. Chen, D.Y.; Ming, O. A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph. *Comput. Graph. Workshop.* 2002. Available online: http://www.cmlab.csie.ntu.edu.tw/~dynamic/download/DYChen_CGW02.pdf (accessed on 1 November 2002).
49. Biasotti, S.; Marini, S.; Spagnuolo, M.; Falcidieno, B. Sub-part correspondence by structural descriptors of 3D shapes. *Comput. Aided Des.* **2006**, *38*, 1002–1019. [[CrossRef](#)]
50. Goh, W.B. Strategies for shape matching using skeletons. *Comput. Vis. Image Underst.* **2008**, *110*, 326–345. [[CrossRef](#)]
51. Biasotti, S.; Giorgi, D.; Spagnuolo, M.; Falcidieno, B. Size functions for comparing 3D models. *Pattern Recognit.* **2008**, *41*, 2855–2873. [[CrossRef](#)]
52. Tierny, J.; Vandeborre, J.; Daoudi, M. Partial 3D Shape Retrieval by Reeb Pattern Unfolding. *Comput. Graph. Forum* **2009**, *28*, 41–55. [[CrossRef](#)]
53. Zhang, Q.; Song, X.; Shao, X.; Shibasaki, R.; Zhao, H. Unsupervised skeleton extraction and motion capture from 3D deformable matching. *Neurocomputing* **2013**, *100*, 170–182. [[CrossRef](#)]
54. Barra, V.; Biasotti, S. 3D shape retrieval using kernels on extended Reeb Graphs. *Pattern Recognit.* **2013**, *46*, 2985–2999. [[CrossRef](#)]
55. Usai, F.; Livesu, M.; Puppo, E.; Tarini, M.; Scateni, R. Extraction of the Quad Layout of a Triangle Mesh Guided by Its Curve Skeleton. *ACM Trans. Graph.* **2015**, *35*. [[CrossRef](#)]
56. Guler, R.A.; Tari, S.; Unal, G. Landmarks inside the shape: Shape matching using image descriptors. *Pattern Recognit.* **2016**, *49*, 79–88. [[CrossRef](#)]
57. Yang, C.; Tiebe, O.; Shirahama, K.; Grzegorzec, M. Object matching with hierarchical skeletons. *Pattern Recognit.* **2016**, *55*, 183–197. [[CrossRef](#)]
58. Yasseen, Z.; Verroust-Blondet, A.; Nasri, A. Shape matching by part alignment using extended chordal axis transform. *Pattern Recognit.* **2016**, *57*, 115–135. [[CrossRef](#)]
59. Yang, C.; Feinen, C.; Tiebe, O.; Shirahama, K.; Grzegorzec, M. Shape-based object matching using interesting points and high-order graphs. *Pattern Recognit. Lett.* **2016**, *83*, 251–260. [[CrossRef](#)]

60. Shakeri, M.; Lombaert, H.; Datta, A.N.; Oser, N.; Létourneau-Guillon, L.; Vincent Lapointe, L.; Martin, F.; Malfait, D.; Tucholka, A.; Lippé, S.; et al. Statistical shape analysis of subcortical structures using spectral matching. *Comput. Med. Imaging Graph.* **2016**, *52*, 58–71. [[CrossRef](#)] [[PubMed](#)]
61. Yang, J.; Wang, H.; Yuan, J.; Li, Y.; Liu, J. Invariant multi-scale descriptor for shape representation, matching and retrieval. *Comput. Vis. Image Underst.* **2016**, *145*, 43–58. [[CrossRef](#)]
62. Huttenlocher, D.P. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 850–863. [[CrossRef](#)]
63. Au, K.C.; Tai, C.L.; Chu, H.K.; Cohen-Or, D.; Lee, T.Y. Skeleton extraction by mesh contraction. *ACM Trans. Graph.* **2008**, *27*, 1567–1573. [[CrossRef](#)]
64. He, K. Rapid 3D Human Body Modeling and Skinning Animation Based on Single Kinect. *J. Fiber Bioeng. Inform.* **2015**, *8*, 413–421. [[CrossRef](#)]
65. Shape Retrieval of Non-Rigid 3D Human Models. Available online: <http://www.cs.cf.ac.uk/shaperetrieval/shrec14/index.html> (accessed on 1 January 2014).
66. Leifman, G.; Katz, S.; Tal, A.; Meir, R. Signatures of 3D Models for Retrieval. In Proceedings of the 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics, Seoul, Korea, 25 May 2013; pp. 1–5. Available online: https://www.researchgate.net/publication/2899815_Signatures_of_3D_Models_for_Retrieval (accessed on 25 May 2013).
67. Salton, G.; McGill, M.J. *Introduction to Modern Information Retrieval*; McGrawpHill: New York, NY, USA, 1983; Volume 41, pp. 305–306.
68. Manjunath, B.S.; Ohm, J.R.; Vasudevan, V.V.; Yamada, A. Color and texture descriptors. *IEEE Trans. Circuits Syst. Video Technol.* **2001**, *11*, 703–715. [[CrossRef](#)]
69. Eakins, J.P.; Boardman, J.M.; Graham, M.E. Similarity retrieval of trademark images. *IEEE Multimed.* **1998**, *5*, 53–63. [[CrossRef](#)]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).