COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2011; **22**:295–305
Published online 12 April 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cav.396

SPECIAL ISSUE PAPER

A graph-based shape matching scheme for 3D articulated objects

Min-Wen Chao[1], Chao-Hung Lin[2], Chih-Chieh Chang[1] and Tong-Yee Lee[1*,†]

[1] CGVSLab CSIE - National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan, ROC
[2] DGLab Geomatics - National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan, ROC

ABSTRACT

In this paper, a novel graph-based shape matching scheme for three-dimensional articulated objects is introduced. The underlying graph structure of a given 3D model is composed of its topological skeleton and local geometric features. Matching two graph structures is generally an NP-hard combinatorial optimization problem. To reduce computation cost, two graphs are embedded on a high-dimensional space, and then matched based on an extension of Earth Mover's Distance (EMD). Furthermore, the symmetric components of an articulated object are determined by a voting algorithm with a self-matching strategy to refine the matching correspondences. Experimental results show that the proposed approach is robust, even when the models are under the surface disturbances of noise addition, smoothing, simplification, similarity transformation, and pose deformation. In addition, the proposed approach is capable of handling both global and partial shape matching. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS
shape matching; skeleton; geometric features; graph matching

Supporting information may be found in the online version of this paper.

***Correspondence**
Tong-Yee Lee, CGVSLab CSIE-National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, Taiwan, ROC.
E-mail: tonylee@mail.ncku.edu.tw

1. INTRODUCTION

Three-dimensional shape matching is a fundamental and important research topic in computer graphics and visualization. The main theme is to find a compact and accurate shape descriptor to efficiently and robustly match shapes. Most of previous researches focused on efficient shape matching. Their algorithms perform very well on the existing benchmarks for shape retrieval, but do not pay much attention to 3D objects with various surface disturbances such as noise addition, smoothing, simplification, similarity transformation, and even pose deformation. Handling 3D shape matching under these surface disturbances is challenging and interesting. In this paper, a graph-based shape matching scheme for this problem is introduced.

As compared with recent work on shape matching, the proposed scheme has three major contributions. First, a novel graph-based matching algorithm is proposed, which is useful for global and partial shape matching. Most previous works are designed solely either for global [1–4] or partial shape matching [5]. Second, the underlying graph structure is composed of a topological skeleton and local geometric features of 3D model. This graph structure is useful for robust shape matching. Specifically, the extracted topological skeleton and geometrical features are insensitive to various surface disturbances. Thus, our approach is robust against these surface disturbances. Third, the symmetry of a 3D object is extracted by a voting approach with a self-matching strategy, and then used to assist in shape matching. As a result, our approach can lead to more accurate matching.

In this paper, the local geometric feature is represented by a few SHs. Specifically, only a few low-frequency SHs are used to represent the local geometries. This makes the proposed approach insensitive to various surface disturbances. The use of SHs [3,6] or a model skeleton [7,8] in 3D shape matching is not a novel concept. However, there are substantial differences between our method and the previous methods. In the studies [3,6], the geometric characteristics of entire objects are represented

[†]Distinguished Professor.

as a set of SH coefficients, making them infeasible in cases of articulated object matching and partial matching. On the other hand, in the studies of Hilaga *et al*. [8] and Sundar *et al*. [7], shape matching may be incorrect because their skeleton-based representation lacks the geometric characteristics of 3D objects. In contrast, the proposed approach utilizes SH coefficients to represent local geometries of 3D objects and integrate these coefficients with the extracted skeleton for shape matching.

2. RELATED WORK

The approaches of 3D shape matching can be classified into two main categories, geometry-based and topology-based, depending on the type of shape features used in matching. In the geometry-based approaches, most of them are based on global geometric descriptors. The shape similarity is measured by various geometric shape descriptors such as geometric moments [9], spherical harmonics (SHs) [2,3,6], and shape distributions [10]. These approaches rely on scale normalization and orientation alignment, which is used to establish a rough correspondence between objects. The entire object are represented as a global feature vector. This will potentially cause misalignment in local surfaces with similar shapes, making them infeasible in articulated object matching and partial shape matching. Recently, some schemes based on local geometric descriptors have been proposed to solve this problem [5,11,12]. The idea is to describe the local geometries by salient geometric features [5,11] or spin images [12]. The local geometry descriptor allows them to perform articulated object matching and partial shape matching. However, the fidelity of salient geometric features mainly depends on the mesh quality and the curvature analysis, as mentioned in Reference [5], and similarly the spin images mainly depends on the vertex normals. This makes these approaches infeasible in handling objects with significant noise.

In the topology-based approaches [4,7,8,13], topology is usually represented as a skeleton. These approaches rely on the fact that the skeleton is a compact shape descriptor, and assume that similar shapes have similar skeletons. It allows a topology-based approach to facilitate efficient shape matching and even for partial matching [13]. However, this assumption is not always correct. As mentioned in Reference [14], similar skeletons may potentially have completely different shapes. The shapes may be mismatched because the skeleton-based representation lacks the substantial geometric information. Therefore, they propose a 2D shape matching approach by combining a graph representation with a geometric feature [14]. The shortest paths between every pair of skeleton endpoints are represented as sequences of radii of maximal disks at the corresponding skeleton points. The shape matching is based on the similarity of the shortest paths between each pair of endpoints of pruned skeletons. However, the accuracy of shape matching mainly depends on the skeleton pruning. Instead of pruning on graph space, we first embed the underlying graph on a high-dimensional space and then perform matching efficiently and accurately on a set of points.

3. ALGORITHM OVERVIEW

As schematically illustrated in Figure 1, the proposed scheme consists of four major steps: *skeleton extraction*, *geometry encoding*, *symmetry determination*, and *graph matching*. First, a skeleton extraction approach based on geometric contraction operation is adopted [15] (Figure 1(a)). The contraction process produces a mesh

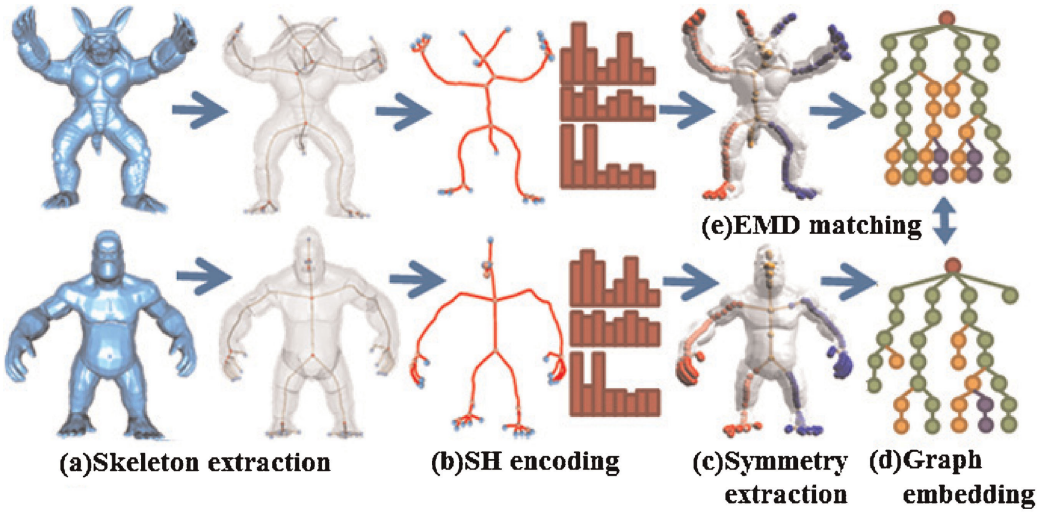


Figure 1. Overview of the proposed shape matching scheme.

DOI: 10.1002/cav

with thin skeleton shape, and the skeleton-vertex correspondences are recorded in each skeleton node. Second, instead of globally encoding the entire geometry of a given 3D object, the local surface geometry for each skeleton node is individually encoded by a small set of SHs (Figure 1(b)). Next, a voting approach with a self-matching strategy is adopted to extract the symmetric parts of articulated objects (Figure 1(c)). Combining this useful information with the skeleton graph can potentially improve shape matching. Finally, the skeleton containing nodes and edges are embedded on a high-dimensional space and becomes a set of points (Figure 1(d)). Therefore, the graph matching is reduced to the problem of point set matching. Here, a registration algorithm with the extended EMD metric is performed for point set matching (Figure 1(e)).

4. GRAPH CONSTRUCTION

The approach [15] is adopted to extract 3D model skeleton. This technique is pose-, sampling-, and noise-insensitive. Therefore, the extracted skeleton has good potential for use in shape matching. Its connectivity surgery step simultaneously records skeleton-vertex correspondence while collapsing edges. The information of skeleton-vertex mapping is used to find the local surface patches in the proposed shape matching approach. The extracted skeleton-curve is composed of a few junction nodes (i.e., degree >2), leaf nodes, and several degree $=2$ nodes (Figure 2). Generally, the topology of 3D shape can be concisely represented by only junction and leaf nodes, since they are the most significant ingredients of skeleton. Therefore, the local surface patches corresponding to either the junction or the leaf nodes are considered as the important geometric components in the proposed scheme. We utilize all junction and leaf nodes of the extracted skeleton and their corresponding surface patches to build a graph structure for shape matching. To compensate for the lack of degree $=2$ nodes, we consider their approximated geometric details to weight the edge of the graph structure. Given two adjacent nodes in a graph $n_a = (x_a, y_a, z_a, r_a)$ and $n_b = (x_b, y_b, z_b, r_b)$, where x, y, z represents the position of the node and r represents the average radius of a degree $=2$ nodes, we define the weight of edge connecting two adjacent nodes as

$$W(n_a, n_b) = \|n_a - n_1\| + \|n_1 - n_2\| + \cdots + \|n_h - n_b\| \quad (1)$$

where $n_1 \cdots n_h$ represent the successive degree $=2$ nodes between the nodes n_a and n_b.

5. GEOMETRY ENCODING

The surface patches obtained from skeleton-vertex mapping is encoded by SH, i.e., a geometry descriptor. We refer the readers to the related SHs papers [2,3] for a more comprehensive comparison on geometry descriptors. In this section, we describe the SH representation for surface patches. A set of SH functions constitute an

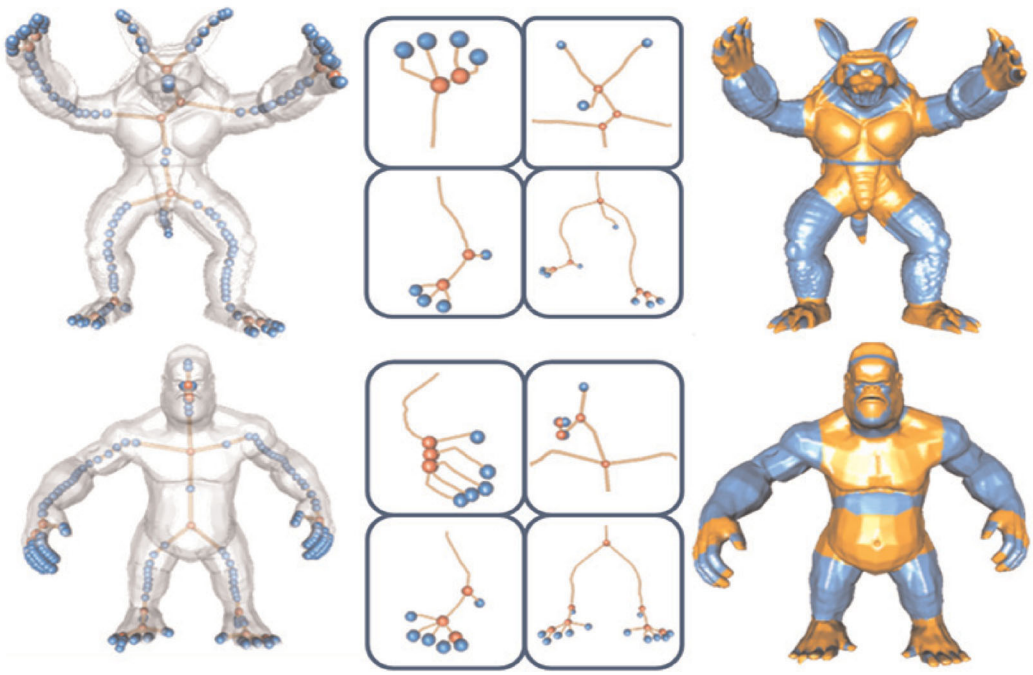


Figure 2. Skeleton extraction result. Left: Models and their skeletons. Middle: Close-up views of the junction and leaf nodes. Right: The corresponding surface patches visualized in gold color.

orthonormal system on a sphere. Any functions $f(\theta,\phi)$ on a sphere can be represented by a linear combination of these basis functions $Y_l^m(\theta,\phi)$ as follows:

$$f(\theta,\phi)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}a_l^m Y_l^m(\theta,\phi) \tag{2}$$

where a_l^m are the SH coefficients. Given a maximum degree $l_{\max}$, an orthonormal system expanded by the SHs involves $(l_{\max}+1)^2$ coefficients. For a function with a set of spherical samples (θ_i,ϕ_i) and their function values $f_i=f(\theta_i,\phi_i)$, $1\le i\le k$, the coefficients a_l^m can be obtained by solving a least-square fitting [16] as follows:

$$\begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,k}\\ y_{2,1} & y_{2,2} & \cdots & y_{2,k}\\ \vdots & \vdots & & \vdots\\ y_{n,1} & y_{n,2} & \cdots & y_{n,k}\end{pmatrix}\begin{pmatrix} b_1\\ b_2\\ \vdots\\ b_k\end{pmatrix}=\begin{pmatrix} f_1\\ f_2\\ \vdots\\ f_k\end{pmatrix} \tag{3}$$

where $y_{i,j}=Y_l^m(\theta_i,\phi_i)$, $b_j=a_l^m$, $j=l^2+l+m+1$, and $k=(l_{\max}+1)^2$. The coefficients calculated by Equation (3) are 3-tuple vectors $a_l^m=(a_{lx}^m,a_{ly}^m,a_{lz}^m)$. Since the L2-norm of the SH coefficients is rotation-invariant [16], a local surface patch is encoded as

$$\mathrm{SH}(f(\theta,\phi))=(\{|a_0^m|\}_0^0,\{|a_1^m|\}_{-1}^1,\cdots,\{|a_{l_{\max}}^m|\}_{-l_{\max}}^{l_{\max}}) \tag{4}$$

To make the geometric encoding insensitive to various surface disturbances, only a few low-frequency SHs are used to encode surface patches $l_{\max}$ is set to 5.

6. GRAPH MATCHING ALGORITHM

A graph G can be described as a pair (N, E), where N is a set of nodes and E is a set of edges connecting the nodes. In the general setting of graph matching, two graphs $G^s=(N^s,E^s)$ and $G^t=(N^t,E^t)$ are given. The goal is to establish correspondence among the nodes and edges of these two graphs. The proposed approach is based on Earth Mover's Distance (EMD) [17], which is a well-known similarity measurement in 2D image retrieval. EMD can be used to match graphs with various sizes in nodes and edges. However, the quality of graph matching relies not only on node matching but also on graph structure matching. To incorporate the graph structure in the node matching, we further extend a low-distortion embedding approach [18] to embed a graph containing both the edges and vertices to a set of points on a high-dimensional space (Section 6.2). Thus, a complicated graph matching is reduced to a point set matching. Generally, there is an affine transformation among these two point sets. Hence, a registration process is required to align point sets based on the EMD distance measurement (Section 6.3). To improve the matching accuracy, the local geometric features and the neighborhood information are also incorporated into the distance measurement (Section 6.4). In addition, the symmetric components of graphs are determined for better and consistent matching in the final result (Section 6.5).

6.1. Earth Mover's Distance (EMD)

A set of nodes with weights in a graph is denoted as $N=\{(n_i,w_i)\}_{i=1}^n$, where n_i and w_i represent the ith node and its weight, respectively, and n is the number of nodes. The weight w_i is set to the percentage of area of the local surface belonging to the node n_i over the total surface area of the input mesh. In this manner, a node corresponding to a large surface patch will be assigned a large weight. Given two node sets, N^s and N^t, a probability matrix (or called flow) between these two sets is defined as $F=[f_{ij}]\in R^{m\times n}$, where m and N^t, respectively, and f_{ij} is the probability of n_i^s matching to n_j^t. Denote all possible flows between N^s and N^t as $\Psi(N^s,N^t)$, and $\mathrm{EMD}(N^s,N^t)$ is defined as the minimum amount of matching cost in all possible flows, that is:

$$\mathrm{EMD}(N^s,N^t)=\frac{\min_{F\in\Psi(N^s,N^t)}\mathrm{Cost}(F,N^s,N^t)}{\min(\sum w^s,\sum w^t)} \tag{5}$$

where $\mathrm{Cost}(F,N^s,N^t)$ is the cost function for a possible flow F between N^s and N^t. The cost function is defined as:

$$\mathrm{Cost}(F,N^s,N^t)=\sum_{i=1}^{m}\sum_{j=1}^{n}f_{ij}d(n_i^s,n_j^t) \tag{6}$$

where $n_i^s\in N^s$, $n_j^t\in N^t$, and $d(n_i^s,n_j^t)$ represent the distance between nodes n_i^s and n_j^t.

6.2. Graph-space to Euclidean-space Embedding

The complexity of graph matching is usually much higher than point set matching. Graph matching requires appropriate pruning to speed up execution and to find the correct results. In contrast, matching in Euclidean space simply finds a transformation between two point sets. We consider the embedding $f_g: N\to P$, where N is a set of graph nodes with a set of distances D (i.e., the distances between all pairs of nodes), and P is a set of points in a high-dimensional Euclidean space. This embedding aims at transforming the matching problem defined over the complex graph space into the problem defined over the simple Euclidean space. This embedding is achieved as follows. First, a graph is converted into a tree structure and then a tree decomposition method [18] is applied to further embed a tree into a Euclidean space. Two input graphs may be embedded into two Euclidean spaces with different dimensionalities. Thus, a dimensionality expansion process is required in the second step to make them identical in dimensionality prior to matching.

Before embedding, the input graph is first converted to a tree structure. This requires that the shortest-path distances between any two nodes in the tree are similar to that in the graph. Thus, the minimal spanning tree is selected, and the node is selected as a root node if the summation of distances between it to all other nodes is minimal. The corresponding root node in another graph is determined according to the neighborhood similarity of node. Then the

DOI: 10.1002/cav

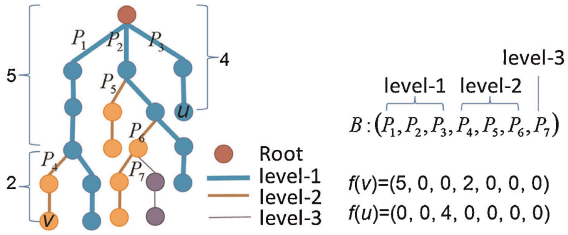


Figure 3. Euclidean space embedding by tree decomposition.

idea of tree decomposition [18] is adopted to embed a tree to a Euclidean space.

Tree Decomposition: As illustrated in Figure 3, the tree is first decomposed into several disjoint paths. Among these disjoint paths, the thick blue paths starting at the root node to the leaf nodes are called level-1 path, which are extracted by a modified depth-first-search (DFS) traversal approach. The tree traversal starts at the root node and then traverses the sub-trees of root node by DFS. In this example, the root node has three branches. Thus, there are three level-1 paths. After removing the level-1 paths from the tree, three trees remain. The same traverse strategy is used on the remaining trees. We can extract another three disjoint paths, called level-2 path (i.e., thin yellow thin paths). This process is repeated until all edges in the tree have been traversed and removed. In Figure 3, there are seven paths in total. Note that we randomly select branches in the process of DFS and thus, there is an affine transformation among the embedded point sets and a registration process is required to align the point sets.

Euclidean Space Embedding: In this step, a tree T is converted into a Euclidean space P, i.e., $f_t : T \rightarrow P$. The dimension of P is the number of paths, denoted as k. In other words, f_t is a k-tuple vector $(P_1, \cdots, P_k)$. The components $P_1, \cdots, P_k$ are associated with the decomposed paths in increasing order of tree decomposition levels. As first three components P_1, P_2, and P_3 correspond to the three level-1 paths. The components P_4, P_5, and P_6 correspond to the three level-2 paths, and the component P_7 corresponds to the level-3 path. The component coefficients are defined as the distances traversed in the corresponding paths. For example in Figure 3, the traversal path from root node to node v goes through level-1 path P_1, and the distance traversed in this path is 5. It then goes through the level-2 path P_4, and its traversal distance in this path is 2. Thus, $f(v)$ is represented as (5,0,0,2,0,0,0) in $\Re^7$. Note that the tree decomposition does not preserve any consistent traversal order between graphs. Thus, estimating the affine transformation between two embeddings is necessary before embedding matching.

6.3. Dimensionality Expansion and Point Sets Matching

The input graphs may be embedded into spaces of different dimensionalities on Euclidean space. Unifying the dimensionalities of their embeddings is required before matching. Generally, a dimensionality reduction process, transforming the embeddings of higher dimensionality to a new coordinate system of lower dimensionality, can be adopted here to reduce computation time. However, the dimension reduction will lose some characteristic features of the graph of higher dimensionality. This will lead to failed partial matching. For example in Figure 4, we expect that the red, blue, and yellow nodes in the embeddings are matched. However, after reducing the dimensionalities, not only the components with lower level are reduced but also the yellow component of the embedding shown in the top figure is reduced to the blue component. This makes the nodes marked by red and brown quadrangles mismatched. Some examples of partial matching by dimensionality reduction are shown in Fig. 5. To overcome this problem, we alternatively adopt a dimensionality expansion process that transforms the embedding with lower dimensionality to a new coordinate system of higher dimensionality by simply padding zeros in the corresponding empty components level by level. Adopting this lossless dimensionality expansion process gives a good chance to provide not only the global but also the local shape matching.

Once two embeddings have identical dimensionality, we find the affine transformation between these two embeddings based on the EMD distance measurement. We adopt an iterative process called optimal Flow and optimal Transformation (FT) that alternately find the best flow for a given transformation and find the best transformation for a given flow [17]. Given an initial transformation $T^{(0)}$, the iteration is formulated as

$$F^{(k)} = \text{argmin}_{F \in \Psi(E^s, E^t)} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d(n_i^s, T^{k-1}(n_j^t)) \quad (7)$$

$$T^{(k)} = \text{argmin}_{T \in \tau} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}^{(k-1)} d(n_i^s, T(n_j^t)) \quad (8)$$

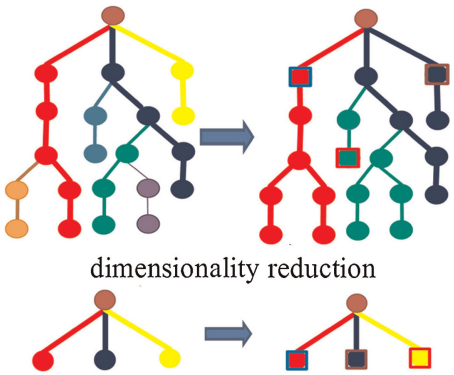


Figure 4. Point set matching by dimensionality reduction process. Left: The embeddings; Right: The results of dimensionality reduction (the dimensionalities are displayed by colors) and matching (the corresponding nodes are marked by quadrangles with the same color).

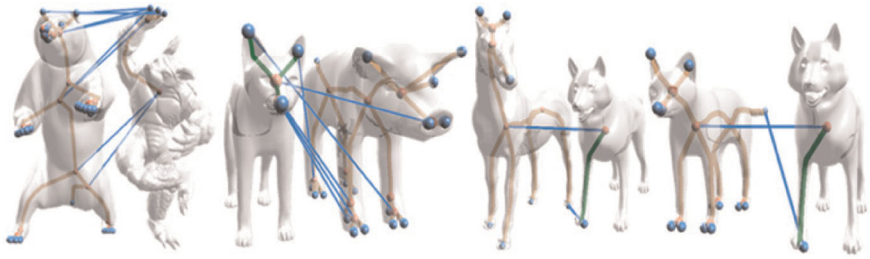


Figure 5. The failed partial matching results.

where E^s and E^t represent the embeddings of graphs G^s and G^t, respectively, in Euclidean space, and τ represents the set of allowable affine transformations. In each iteration, we calculate Equation (7) to obtain the correspondence of these two embeddings, and then estimate the new transformation between these two embeddings by Equation (8). When the iteration process converges, the flow matrix F contains the matching correspondences, and $\mathrm{EMD}(E^S, E^t)$ indicates the similarity value of these two graphs.

6.4. Distance Metric

We need to calculate the distance between nodes in matching. To define this distance, three distance terms are considered, that is (1) node distance, (2) geometric similarity, and (3) neighborhood similarity. The term of node distance is simply defined as the Euclidean distance between two embedded nodes. The term of geometric similarity is defined as the difference in SH coefficients between the nodes (i.e., sh() in Equation (9)). To define the neighborhood similarity (i.e., ns() in Equation (9)),a local similarity matrix $\mathrm{LS} = [ls_{ij}] \in R^{m \times n}$ between N^s and N^t is generated. The entity ls_{ij} represents the neighborhood similarity between the nodes n_i^s and n_j^t, in which node neighborhood is defined as the immediately adjacent nodes and the connecting edges. The similarity ls_{ij} is calculated by first embedding the neighborhood of n_i^s and n_j^t and then computing the similarity between two embeddings. Note that we also use Equation (9) to compute ls_{ij}, but do not include the term ns(). The computation cost of the local similarity matrix will not be time-consuming because the neighborhood of each node contains only a few nodes and edges. Finally, the distance used in EMD is reformulated as:

$$\begin{gathered}\mathrm{Cost}(F, E^s, E^t) = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} * (d(e_i^s, e_j^t) + \mathrm{sh}(n_i^s, n_j^t) + \mathrm{ns}(n_i^s, n_j^t)), \\ \mathrm{sh}(n_i^s, n_j^t) = \|\mathrm{SH}(\mathrm{surf}(n_i^s)) - \mathrm{SH}(\mathrm{surf}(n_j^t))\|, \\ \mathrm{ns}(n_i^s, n_j^t) = \mathrm{ls}_{ij}\end{gathered} \tag{9}$$

where e_i^s and e_j^t represent the embeddings of nodes n_i^s and n_j^t, respectively; surf(k) represents the corresponding local surface patch of a node k; and $d(e_i^s, e_j^t)$ represents the 2-norm Euclidean distance between the two embedded nodes e_i^s and e_j^t.

6.5. Symmetry Determination

Most shape matching approaches do not consider object symmetry, potentially resulting in inconsistent matching correspondence. Inspired by Podolak *et al.* [19], a fast global symmetry determination approach based on a voting strategy with a self-matching approach is introduced. To take efficiency into account, we only determine a global median axis for skeleton and skip local symmetric parts. The global median axis is generally sufficient to improve the shape matching. All nodes in the skeleton graph are separated into two categories: belonging to the symmetric parts and the median axis part. For each voting, a pair of leaf nodes denoted as (n_i, n_j) is selected, and the possibility of node n_k to be the median axis is formulated as:

$$S_{n_k}^{(n_i, n_j)} = L_1 / L_2 \tag{10}$$

where L_1 and L_2 represent the distances between n_k and the selected leaf nodes n_i and n_j. Let $L_2 > L_1$, the range of scope is [0,1]. For example in Figure 6, two corresponding pairs of leaf nodes in the fingers and toes are selected. The possibility of the remaining node n_k is visualized. Blue nodes near the selected leaf nodes have low possibility, and red nodes near the body parts have high possibility. In this manner, a vote can be casted on each n_k, and a possible median axis is determined by grouping nodes with higher scores. Once the possible median axis is obtained, the symmetric parts separated by this median axis can also be determined. Figure 7 shows some results of the possible symmetric parts extracted from different votes. For each case in this figure, blue and red sub-trees are extracted symmetric parts using the orange sub-tree as the median axis. For each voting, the symmetric parts can be extracted and the scores of all nodes can be obtained. For consistent symmetry determination, each voting is weighted by a self-matching strategy. A large weight is given to a vote if its extracted symmetric sub-trees have high similarity. Thus, the score S_{n_k} of node n_k is

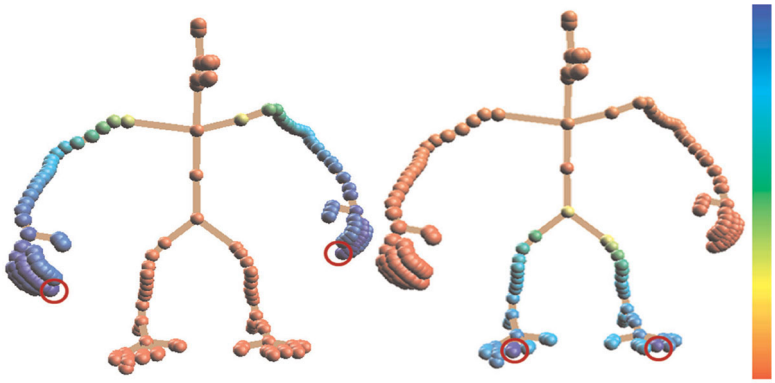


Figure 6. Voting example. The nodes in red circles are the selected leaf nodes. The voting scores are visualized by color ranging from 0.0 (blue) to 1.0 (red).

reformulated as:

$$S_{n_k} = \frac{\sum_{(n_i,n_j)\in\text{candidate set}} w_{ij} S_{n_k}^{(n_i,n_j)}}{\sum_{(n_i,n_j)\in\text{candidate set}} w_{ij}} \tag{11}$$

where w_{ij} is the weight of $S^{(n_i,n_j)}$, i.e., the similarity between two extracted sub-trees.

Once each node is computed by Equation (11), the nodes with high scores are defined as the median axis. Then, the connected parts separated by the median axis are extracted as the symmetric parts. Once the median axis is obtained, the correspondence of each sub-tree is determined by finding the sub-tree with the highest similarity. Figure 8 shows several results of symmetric parts and their symmetric axis. In this example, the median axis is composed of orange nodes, and the blue sub-tree corresponds to the red sub-tree.

7. EXPERIMENTAL RESULTS AND DISCUSSION

Experimental results are evaluated on a PC with a 2.13 GHz CPU and 2.0 GB memory. On average, the computation time for preprocessing is 12.37 seconds (10 seconds for skeleton extraction, 2.37 seconds for SHs encoding), and that for shape matching is 1.573 (0.032 seconds for graph embedding, 1.541 seconds for matching). To evaluate the robustness of our approach, the models with various surface distributions are tested. The generated matching correspondences shown in Figure 9 are almost identical. This demonstrates that our approach can potentially obtain correct matching even the models are substantially altered by these surface disturbances.

Most of the shape matching approaches are based either on geometric features [2,3,5,11] or topological features [7,8,12]. It is difficult to perform exact comparisons without obtaining their codes or re-implementing these methods. In order to make an objective conceptual comparison of these approaches regardless of the subtle details of each method, we proposed the following arrangement. The skeleton and SHs coefficients are selected as features in the topology-based and geometry-based shape matching, respectively. In other words, either the distance term $d()$ or the SH coefficient term $sh()$ is respectively used only in the cost evaluation function Equation (9) to represent methods using either topological or geometric attributes. The experimental results are shown in Figure 10. Expectedly, the matching results by using only topological or geometric features are worse than the results of combining both. This is because similar objects may have dissimilar skeletons, and similar skeletons may have visually different shapes. Our approach utilizing geometric features, topological features, a neighborhood

Figure 7. Result of extracted symmetric parts from different voting; red and blue sub-trees are symmetric parts, and the orange sub-tree is the corresponding medial axis part.

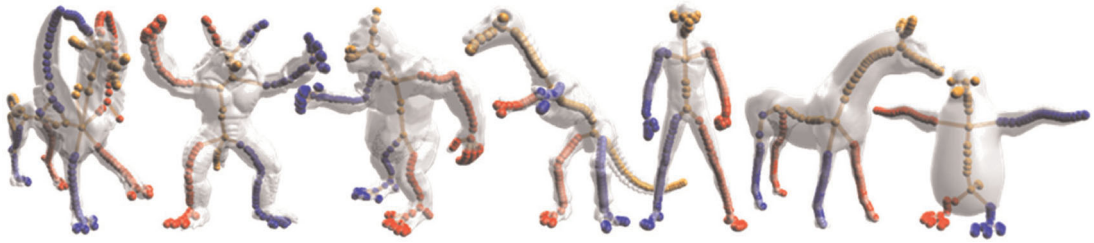


Figure 8. Result of symmetry. The group of red nodes and the group of blue nodes are symmetry.

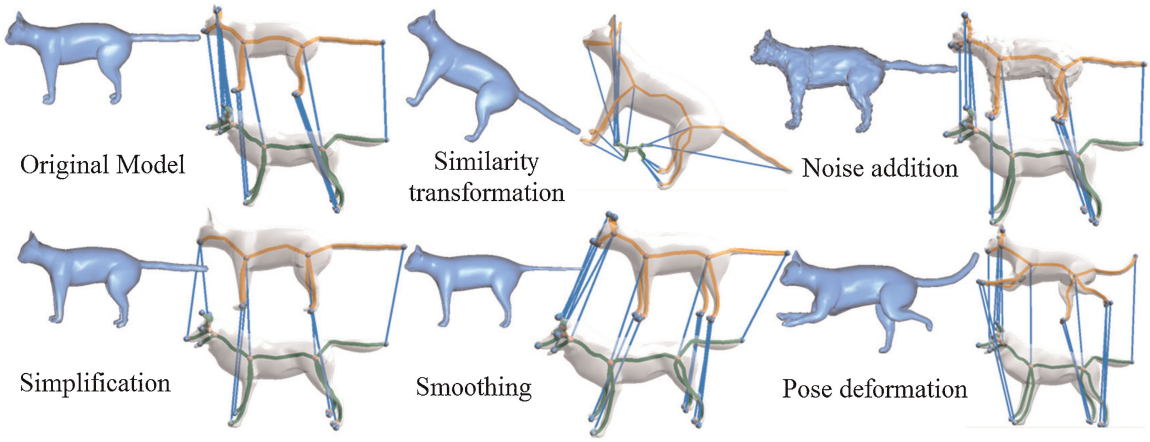


Figure 9. Shape matching under various disturbances.

similarity measurement, and object symmetry information in graph matching can potentially handle articulated objects well when geometric shapes or topological skeletons are very different (as shown in Figure 11). In addition, with the process of dimensionality expansion in graph matching, our approach can handle both partial and global shape matching (as shown in Figures 11 and 12) using the same algorithm.

Finally, we show a theoretical comparison of robustness between our approach and the related 3D shape matching

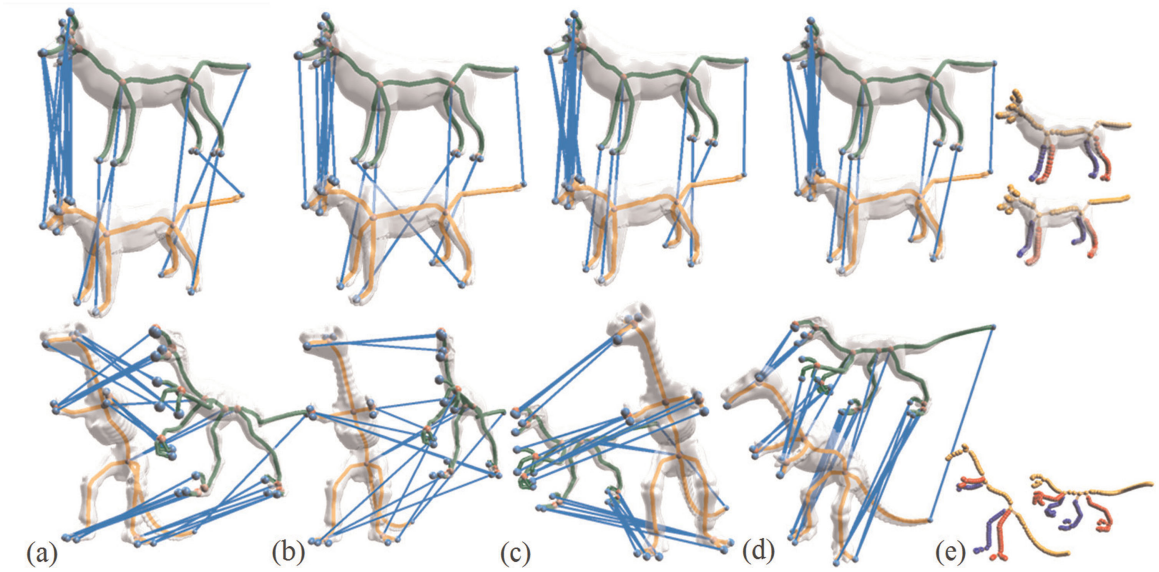


Figure 10. Comparison of shape matching. (a) Using skeletons; (b) using SH coefficients; (c) using both skeletons and SH coefficients; (d) using skeletons, SH coefficients, and symmetry information shown in (e).

Comp. Anim. Virtual Worlds 2011; **22**:295–305 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/cav

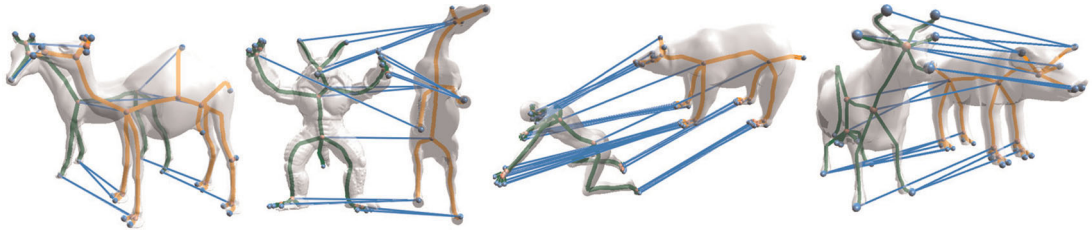


Figure 11. Result of global shape matching.

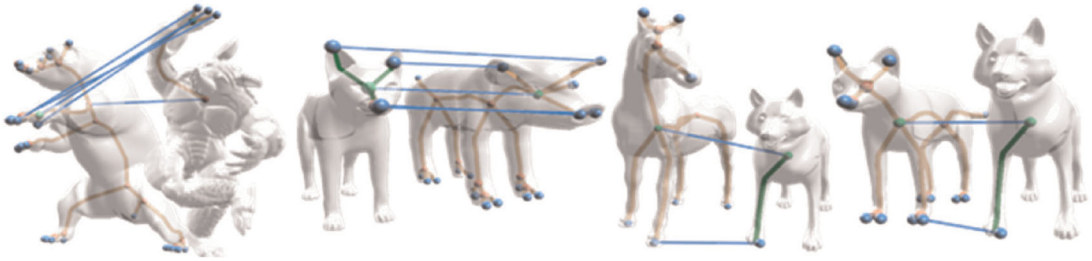


Figure 12. Result of partial shape matching.

approaches [1–3,5,8,13] in Table 1. The robustness of matching under various operations, and the capabilities of partial and global matching are considered. As mentioned previously, the approaches based on global geometric features [2,3] are sensitive to pose deformation since the geometric shape will be altered by pose deformation. The approach based on local salient geometric features [5] or matching in a spectral domain [1] is partially sensitive to noise addition and simplification because shape matching is mainly based on curvature analysis. As for the topology-based approaches [8,13], they are also slightly sensitive to noise addition and simplification because skeletons may be slightly different after the models are altered by these two kinds of surface disturbances. In our approach, we have a good chance to solve this problem by enhancing the graph nodes with local geometric features, adopting the process of dimensionality expansion and neighborhood similarity measurement in the graph matching, and utilizing object symmetries to assist in matching correspondences.

8. CONCLUSIONS AND FUTURE WORK

We presented a novel graph-based technique for 3D shape matching. Our approach can accurately obtain both partial and global matching correspondence between the 3D articulated shapes. The skeleton associated with local geometric features is constructed by noise-, connectivity-, and resolution-insensitive skeleton extraction and geometric representation approaches, thus making our approach insensitive to various surface distributions including similarity transformation, smoothing, noise addition, simplification, and pose deformation. The experimental results demonstrate that our approach is better in terms of matching accuracy than those using geometric or topological features only in shape matching. A limitation still exists in our approach. Our approach only works for closed mesh models with manifold connectivity since the geometry contraction in skeleton extraction requires a well-defined Laplace operator for

Table 1. Theoretical comparison between our approach and the related schemes [1,2, 3, 5, 8, 13].

	T. F.	G. F.	Matching	N. A.	Smoothing	P. D.	S.T.	Simp.
[2]	None	SHs	(1)	√	√	X	√	√
[3]	None	SHs	(1)	√	√	X	√	√
[1]	None	Dist.	(1)	△	○	√	√	△
[5]	None	Saliency	(2)	△	○	√	√	○
[8]	Reeb graph	None	(1), (2)	○	√	√	√	○
[13]	Skeleton curve	None	(1), (2)	○	√	√	√	○
Our approach	Skeleton graph	SHs	(1), (2)	√	√	√	√	√

The surface disturbances of noise addition (P. D.), smoothing, pose deformation (P. D.), similarity transformation (S. T.), and simplification (Simp.) are tested. The symbols 'X', '△', '○','√' indicate that this approach is sensitive, partially sensitive, slightly sensitive, and insensitive to the surface disturbance, respectively. The approaches based on what kind of topological feature (T. F.) and geometric feature (G. F.) are shown in columns 2 and 3. The abilities of global matching (1) and partially matching (2) are shown in column 4.

every vertex. In the near future, we plan to further extend this framework to solve the problem of automatic correspondence establishment, which is another interesting and challenging problem. Moreover, we also plan to apply our approach to polygon morphing [20] and to motion retargeting as suggested by Reference [21].

ACKNOWLEDGEMENTS

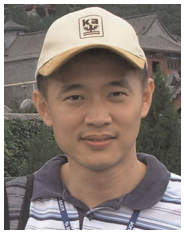
We would like to thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Science Council (contracts NSC-97-2628-E-006-125-MY3, NSC-98-2221-E-006-179 and NSC-99-2221-E-006-066-MY3), Taiwan.

REFERENCES

1. Jain V, Zhang H. A spectral approach to shape-based retrieval of articulated 3d models. *Computer-Aided Design* 2007; **39**: 398–407.
2. Funkhouser T, Mi P, Kazhdan M, *et al*. A search engine for 3d models. *ACM Transactions on Graphics* 2003; **22**(1): 83–105.
3. Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3D shape descriptors. *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, 2003; 156–164.
4. Au O K-C, Tai C-L, Cohen-Or D, Zheng Y, Fu H. Electors voting for fast automatic shape correspondence. *Computer Graphics Forum* 2010; **29**(2): 645–654.
5. Gal R, Cohen-Or D. Salient geometric features for partial shape matching and similarity. *ACM Transaction on Graphics* 2006; **25**(1): 130–150.
6. Saupe D, Vranić DV. 3D model retrieval with spherical harmonics and moments. *DAGM-Symposium on Pattern Recognition*, 2001; 392–397.
7. Sundar H, Silver D, Gagvani N, Dickinson S. Skeleton based shape matching and retrieval. *International Conference on Shape Modeling and Applications*, 2003; 130–139.
8. Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3d shapes. *SIGGRAPH '01;* 203–212.
9. Cybenko G, Bhasin A, Cohen KD. Pattern recognition of 3d cad objects: Towards an electronic yellow pages of mechanical parts. *International Journal of Intelligent Engineering Systems* 1996; **95**: 1–15.
10. Osada R, Funkhouser T, Chazelle B, Dobkin D. Matching 3d models with shape distributions. *International Conference on Shape Modeling and Applications*, 2001; 154–166.
11. Zou G, Hua J, Dong M, Qin H. Surface matching with salient keypoints in geodesic scale space. *Journal of Visualization and Computer Animation* 2008; **19**(3-4): 399–410.
12. Chang W, Zwicker M. Automatic registration for articulated shape. *Computer Graphics Forum* 2008; **27**(5): 1459–1468.
13. Cornea ND, Demirci MF, Silver D, Shokoufandeh A, Dickinson SJ, Kantor PB. 3d object retrieval using many-to-many matching of curve skeletons. *International Conference on Shape Modeling and Applications*, 2005; 368–373.
14. Bai X, Latecki LJ. Path similarity skeleton graph matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 2008; **30**: 1282–1292.
15. Au O K-C, Tai C-L, Chu H-K, Cohen-Or D, Lee T-Y. Skeleton extraction by mesh contraction. *ACM Transactions on Graphics* 2008; **27**(3): 1–10.
16. Shen L, Chung MK. Large-scale modeling of parametric surfaces using spherical harmonics. *International Symposium on 3D Data Processing, Visualization and Transmission*, 2006; 294–301.
17. Rubner Y, Tomasi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 2000; **40**(2): 99–121.
18. Matousek J. On embedding trees into uniformly convex banach spaces. *Israel Journal of Mathematics* 1999; **114**(1): 221–237.
19. Podolak J, Shilane P, Golovinskiy A, Rusinkiewicz S, Funkhouser T. A planar-reflective symmetry transform for 3d shapes. *ACM Transactions on Graphics* 2006; **25**: 549–559.
20. Lee T-Y, Yao Z-Y, Chu H-K, Tai M-J, Chen C-C. Generating genus-n-to-m mesh morphing using spherical parameterization. *Computer Animation and Virtual Worlds Journal* 2006; 433–443.
21. Poirier M, Paquette E. Rig retargeting for 3d animation. In *Graphics Interface 2009 Conference Proceedings*, 2009; 103–110.

AUTHORS' BIOGRAPHIES

Min-Wen Chao received the BS degree in Mathematics from the National Cheng-Kung University, Taiwan, in 2003 and the MS degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2005. She is currently working toward the PhD degree in the Department of Computer Science and Information Engineering, National Cheng-Kung University. Her research interests include computer graphics, motion retrieval, model matching, and data hiding.



Chao-Hung Lin received his MS and PhD degree in computer engineering from National Cheng-Kung University, Taiwan in 1998 and 2004, respectively. He is currently an associate professor in the department of geometrics at National Cheng-Kung University in Tainan, Taiwan. He leads the Digital Geometry Laboratory, National Cheng-Kung University. His research interests include digital geometry processing, digital map generation, information visualization, and remote sensing. He is a member of IEEE and ACM.

Chih-Chieh Chang received BS degree from Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2007 and the MS degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2009.

Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a distinguished professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He also serves on the editorial boards of the *IEEE Transactions on Information Technology in Biomedicine*, the *Visual Computer*, and the *Computers and Graphics Journal*. He served as a member of the international program committees of several conferences including the IEEE Visualization, the Pacific Graphics, the IEEE Pacific Visualization Symposium, the IEEE Virtual Reality, the IEEE-EMBS International Conference on Information Technology and Applications in Biomedicine, and the International Conference on Artificial Reality and Telexistence. He is a senior member of the IEEE and the member of the ACM.