EUROGRAPHICS 2010 / T. Akenine-Möller and M. Zwicker
(Guest Editors)

Volume 29 (2010), Number 2

Electors Voting for Fast Automatic Shape Correspondence

Oscar Kin-Chung Au[3] Chiew-Lan Tai[1] Daniel Cohen-Or[2] Youyi Zheng[1] Hongbo Fu[3]

[1]Hong Kong University of Science & Technology [2]Tel Aviv University [3]City University of Hong Kong

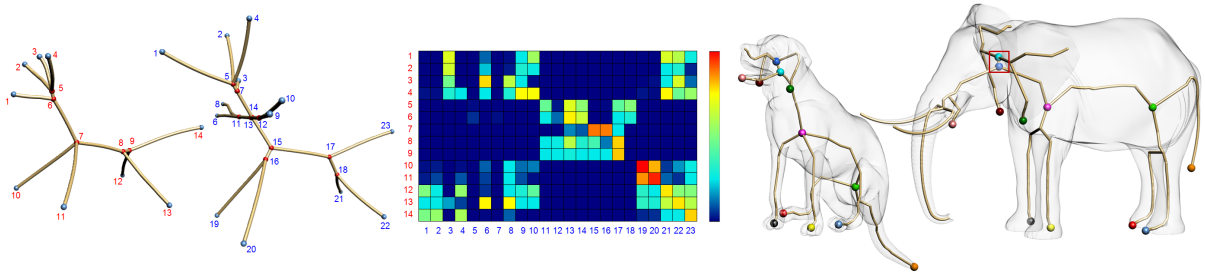


Figure 1: Our electors voting algorithm automatically and instantaneously computes a semantic correspondence between the sitting dog and the standing elephant by considering their skeletons. A large set of selected electors vote for the skeletal feature pairs (middle) to synthesize the final correspondence between the two shapes (right). Corresponding nodes that are matched by our algorithm are rendered in the same colors; two incorrect matchings are marked with a red square.

Abstract

This paper challenges the difficult problem of automatic semantic correspondence between two given shapes which are semantically similar but possibly geometrically very different (e.g., a dog and an elephant). We argue that the challenging part is the establishment of a sparse correspondence and show that it can be efficiently solved by considering the underlying skeletons augmented with intrinsic surface information. To avoid potentially costly direct search for the best combinatorial match between two sets of skeletal feature nodes, we introduce a statistical correspondence algorithm based on a novel voting scheme, which we call electors voting. The electors are a rather large set of correspondences which then vote to synthesize the final correspondence. The electors are selected via a combinatorial search with pruning tests designed to quickly filter out a vast majority of bad correspondence. This voting scheme is both efficient and insensitive to parameter and threshold settings. The effectiveness of the method is validated by precision-recall statistics with respect to manually defined ground truth. We show that high quality correspondences can be instantaneously established for a wide variety of model pairs, which may have different poses, surface details, and only partial semantic correspondence.

1. Introduction

Shape correspondence is a difficult problem. Humans are competent in such a task for recognizable shapes since we are familiar with the semantics of shape components. However, inferring semantics is still extremely difficult, and thus correspondence algorithms have to rely solely on matching geometric properties. The correspondence problem becomes particularly difficult when the shapes exhibit large variation in poses and surface details, since the similarity measures of local geometric properties become irrelevant.

Many applications like cross parametrization and deformation transfer require semantically meaningful correspondence, which, however, is typically found with user assistance. A notable automatic correspondence solution is the work of Zhang et al. [ZSCO*08]. Their method automatically searches for the best correspondence between shapes that significantly differ in poses and local geometric details. However, it measures the quality of every possible correspondence by evaluating the deformation distortion of local surface details, which is indirect and expensive, taking an order of dozens of minutes to compute. Recently, Lipman and Funkhouse [LF09] limit their input to nearly-isometric surfaces and propose a faster automatic correspondence algorithm based on Möbius voting, but the running time is still of an order of minutes. A fast automatic correspondence is needed to allow processing a large number of models, dealing with model database and applying learning methods.

The curve-skeleton is an effective abstraction of an object's geometry and structure. Curve-skeletons naturally incorporate the notion of parts, which are likely used by the human recognition system to identify semantic correspon-

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.
Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

dences [Heb49]. Often, skeletons are augmented with the local radius, providing additional valuable geometric information. Matching two shapes through their skeletons is intuitively a promising strategy since skeletons provide structural information which is crucial for establishing correspondence. However, matching 3D skeletons directly [HSKK01, TS04, SSGD03, BMSF06] is not a straightforward task due to the connectivity variation which leads to high computational complexity (finding a common subgraph is an NP-hard problem). The challenge in designing a general shape correspondence algorithm stems from the fact that the combinatorial search space is exponential. Thus, a simple evaluation of each possible combination of feature pairs, even for two small feature sets, is clearly computationally prohibitive.

We observed that although there is no rigorous definition on what a good correspondence is, it is rather easy to identify very poor correspondences. This motivated us to design a statistical method to synthesize a high quality correspondence rather than a direct search for the best correspondence using score function. The key idea is to quickly form a set of reliable *electors* which then vote on individual feature-to-feature matching. This large set of electors passed a cascade of pruning tests designed to detect different types of bad correspondences. Note that the aim of the fast pruning process is not to perform very accurate filtering; in fact it is very likely that some wrong correspondences pass the tests and a few correct ones are pruned. However, most electors have high probabilities of containing a significant subset of the best set of feature-to-feature correspondences. Thus an aggregation of the feature-to-feature votes of all the electors ultimately generates a high-quality correspondence (Figure 1).

The presented algorithm is fast, fully automatic and easy to implement. We show experimentally that our statistical approach successfully generates good quality correspondences for a wide variety of model pairs with similar semantic structures. It is robust to some degree of fuzziness in the definition of the electors set, thus not requiring careful setting of thresholds and parameters in the pruning tests. The algorithm supports (i) partial matching where the two shapes possibly have components that do not correspond, (ii) different poses, and (iii) shapes with large variation in their surface details (see examples in Figures 5 and 6). We validate the effectiveness of the method by precision-recall statistics with respect to the manually defined ground truth. We show that, typically, correspondences between two shapes, such as those shown in Figure 5, are computed in only 2-4 seconds. Finally, we demonstrate the effectiveness of our correspondence algorithm in two applications: animation transfer and tagging of shape components.

2. Related work

We classify extensive previous work related to shape matching and feature correspondence according to the problem they attempt to solve: shape retrieval, registration and correspondence. We also review existing voting methods in the context of digital geometry processing.

Shape retrieval. Retrieving shapes that are similar to a given query shape from a database involves shape matching. However, determining the similarity between two given shapes does not necessarily require finding an exact correspondence between their shape components. In fact, there exist successful retrieval systems based only on statistical shape descriptors [SF07, GSCO07] or spectral decomposition [JZ07, DvLV08], without finding correspondence. There are other retrieval systems which provide correspondence between the query shape and the retrieved shapes; we classify them below under the shape correspondence category.

Shape registration. The registration process matches corresponding regions to align multiple scans of the same object taken from different viewpoints. Most existing methods detect salient features using local surface signatures and find feature correspondence based on the assumption that *local* surfaces are rigidly transformed. This rigidity assumption leads to simpler salient feature definition, low-dimensional feature representation and smaller search space. Various search schemes have been proposed, such as iterated closest point [CR03] and combinatorial search [GMGP05]. More advanced methods have also been proposed to work under non-rigid warp due to device nonlinearities [BR07] or approximate piecewise rigid transformations [CZ08, CZ09, LSP08, HAWG08]. However, all these methods are not applicable to finding correspondence between shapes with large variation in surface details.

Shape correspondence. There has been relatively less previous work on finding correspondence between two objects with different surface details. Since establishing low-level geometric feature correspondence is unreliable and meaningless here, semantic correspondence needs sought, demanding global and structural shape descriptors. A variety of methods have been proposed to construct a graph (e.g., through skeleton extraction [SSGD03, CDS*05], Reeb graph construction [HSKK01, BMSF06, TVD09]) which provides a high-level structural description of a shape. Each graph node corresponds to a semantic part of the shape and is assigned topological and/or geometric signatures for graph matching, forming attributed graphs.

Previous work typically focused on 1-1 correspondences between graph nodes. To enable partial matching, a common approach is finding a maximum subgraph isomorphism. However, finding an exact subgraph isomorphism is an NP-complete problem. To avoid exponential complexity, different matching heuristics have been proposed. Many previous methods [HSKK01, TS04, SSGD03, BMSF06] match the common subgraphs by searching along graph connectivity. Such an approach is either sensitive to topological differences or requires complicated strategies to edit the topological structures. Noticing that visually similar attributed graphs may have completely different topological structures,

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

Bai and Latecki [BL08] proposed to ignore all internal nodes as well as all graph edges, and match only graph endpoints by comparing their geodesic paths using sequential matching. Our method also seeks a 1-1 correspondence between subsets of graph nodes without considering the edge correspondences. However, apart from supporting partial matching, our method finds correspondences for structurally important internal nodes, which are essential for applications like animation transfer. Note that all these previous methods do not explicitly consider the spatial configuration of symmetric components and thus often suffer from the symmetry switching problem [ZSCO*08].

Like ours, there are other methods that do not directly search the graphs, but perform a combinatorial search. Funkhouser and Shilane [FS06] describe a priority-driven search that allows 3D shape retrieval from a database by ranking any subset of k feature pairs on the query and target objects. However, their method is mainly designed under local rigidity assumption. Recently, Zhang et al. [ZSCO*08] introduce a deformation-driven combinatorial search algorithm for finding a 1-1 correspondence between shape extremities, without considering internal structure correspondences. Each possible correspondence is evaluated by deforming one of the models to align the corresponding feature pairs and measuring the resulting distortion. Their technique is robust to large shape variance and solves the symmetry switching problem. However, computing deformation is an overly costly process, thus interactive performance is not achievable.

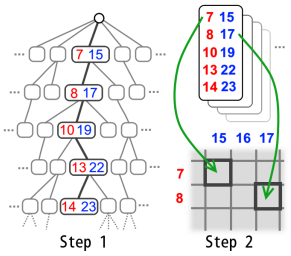
Voting Methods. Voting has been extensively used in estimating parametric shapes or transformations. It is typically based on the assumption that a shape or transformation representation is low-dimensional and can be characterized analytically. Based on the rigidity assumption, RANSAC [FB81] repeatedly chooses a random subset of sample feature pairs on the query and target shapes and then evaluates the matching error of the two shapes to vote for the transformation induced by the chosen samples. Assuming a known and low-dimensional transformation space, geometric hashing [LW88] pre-computes all possible alignments of the target model with respect to every subset of its features and stores them in a hash grid, allowing fast computation of best transformations between the query and target models by voting. Chang and Zwicker [CZ08] propose to vote for a set of rigid transformations in the transformation space to register articulated shapes. Such voting-based methods have proved to be very robust to significant noise and naturally support partial matching [LG05, GCO06, AMCO08].

The recent work of Lipman and Funkhouse [LF09] is probably the most closely related to ours. Observing that the space of isometries is of low dimensionality, they devise a voting-based correspondence algorithm for nearly-isometric surfaces. Unlike the above voting methods which mostly rely on local shape descriptors and vote for transformations, their work leverages the invariants of global conformal structures of the meshes and votes for individual feature-to-feature correspondences. Our work resembles theirs in voting for feature-to-feature correspondences but we aim to find intrinsic correspondences between surfaces with semantically similar features, which may be far from isometric and thus the search space is no longer of low dimension.

3. Electors Voting Correspondence

We give an overview of our method in this section. We pose the problem of shape correspondence as the problem of establishing a semantic correspondence between the curve skeletons of two models. Specifically, our objective is to find a 1-1 semantic correspondence between two sets of *feature nodes* of the curve skeletons, comprising *terminal nodes* and *junction nodes* (see Figure 1). The *terminal nodes* represent the extremities of the shape while the *junction nodes* (with at least three incident bones) capture the shape topology.

Our automatic correspondence algorithm operates in two steps. First, it performs a combinatorial search using fast pruning tests to eliminate exponentially many bad correspondences, leaving behind still a large set (possibly thousands) of more reliable correspondences, which we call *electors* (Section 4). Second, the electors cast votes on individual candidate feature pairs to establish the final correspondence (Section 5).



The first step involves a branch-and-bound search on a combinatorial tree (see the right figure). Every tree node represents a possible feature-to-feature correspondence, except for the root which is an auxiliary node and represents an empty set. A path from the root to any tree node (e.g., highlighted path in figure above) represents a possible correspondence solution comprising all the feature-to-feature correspondences along the path. The tree is expanded while searching from the root. A node is added only if the new correspondence set including the new node passes a cascade of pruning tests, otherwise the subtree rooted at that node is pruned. Each of the tests efficiently filters away *different types* of incorrect correspondences by considering different geometry or topology information. The tests are ordered by their time complexities to achieve the best efficiency.

On completion of the search, every tree node† represents an elector, which contains all the feature-to-feature correspondences along the path from the root to that node. In the

† More specifically, nodes at level 4 or above. See Section 4 for more details.

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

second step, every elector casts one vote for each of its constituent feature pairs (see inset figure and examples of voting results in Figures 1 and 4). The feature pair with the highest votes is then iteratively picked subject to 1-1 and topological conditions to incrementally form the best correspondence. We will show that the electors voting step is robust to inexact pruning in the electors selection step.

4. Electors Selection

In this section, we describe the search process which applies pruning tests to each tree node and eventually identify the electors.

4.1. Combinatorial Search

We use a combinatorial tree search similar to Zhang et al. [ZSCO*08]. However, unlike [ZSCO*08] which performs a best-first search based on an explicit score function to evaluate the quality of a correspondence set, we expand the tree in a breadth-first-search manner.

In the first level (i.e., the child level of the root node), we consider only every possible junction feature pair instead of every possible feature pair. This leads to a smaller search space based on the premise that a good correspondence must match at least one junction pair. In every subsequent level, we expand a tree node (representing a feature pair) by considering every other feature pair that has both its feature nodes not in any feature pair along the path from the root. This condition enforces 1-1 feature correspondences. In addition, to avoid redundant computation, we ensure that each possible correspondence is uniquely represented as a path in the search tree. This is achieved as follows: when expanding a tree node, we consider adding only those feature pairs that succeed the current tree node according to a predefined ordering that has all junction feature nodes preceding non-junction nodes.

A new tree node representing a feature pair is added only if the newly formed set of feature-to-feature correspondences passes *all* the pruning tests, denoted by T1 to T4.

4.2. Pruning Tests

We design the set of pruning tests based on the observation that incorrect correspondences often lead to easily measurable deviations of intrinsic information associated with individual features (T1), pairs of features (T2), and three or more features (T3 and T4). The tests are applied in a cascading manner, with the faster tests first, such that the most computationally expensive test is only applied to those correspondences that are not eliminated by the preceding tests.

We denote the sets of skeletal feature nodes of two input models as $\hat{P}$ and $\hat{Q}$, between which we seek a 1-1 semantic correspondence. For simplicity of notation, in this section we let (p_k, q_k), $p_k \in \hat{P}$ and $q_k \in \hat{Q}$, denote the feature pair being considered for addition to the k-th level of the search tree. The set $\Omega = \{(p_1, q_1), \ldots, (p_k, q_k)\}$ denotes the new correspondence set containing all the feature pairs on the path from the root to (p_k, q_k).

T1: Node-Centricity. We define the centricity c_p of feature node p as the average geodesic distance from p to all other skeletal nodes[‡]: $c_p = \frac{1}{|P|}\sum_{p_i \in P} geo(p, p_i)$, where $geo(p, p_i)$ is the geodesic distance between p and p_i along skeletal paths. Larger values of c_p means p is further away from the center of the shape. We normalize c_p to $[0,1]$ by dividing it by the maximum centricity value of individual models, which makes the pruning test independent of the scales of individual models. We reject the feature pair (p_k, q_k) if their centricity difference is larger than a threshold: $\frac{|c_{p_k} - c_{q_k}|}{(c_{p_k} + c_{q_k})/2} > \varepsilon_C$. We use a *relative* difference here since longer paths usually bear larger absolute difference.

T2: Path-Length and Path-Radius. When considering adding (p_k, q_k) to the tree, we test if all the corresponding new *skeletal* paths have similar lengths and average path radii. We reject (p_k, q_k) if either one of the following is true:

$$\frac{|geo(p_k, p_j) - geo(q_k, q_j)|}{(geo(p_k, p_j) + geo(q_k, q_j))/2} > \varepsilon_C, \ \forall (p_j, q_j) \in \Omega, \quad (1)$$

$$\frac{|rad(p_k, p_j) - rad(q_k, q_j)|}{(rad(p_k, p_j) + rad(q_k, q_j))/2} > \varepsilon_C, \ \forall (p_j, q_j) \in \Omega, \quad (2)$$

where $geo(\cdot,\cdot)$ and $rad(\cdot,\cdot)$ are the geodesic distance and the average path radius of two feature nodes along skeletal paths. The local radius at a skeletal node is computed as the closest distance of the node to the input polygonal mesh. Again, we normalize the values of path-length and path-radius to the range $[0,1]$. These two quantities provide shape information in both local and global scales and are invariant to local bending (i.e., pose invariant). Note that we could use a unified threshold ε_C for both T1 and T2, since both are in terms of spatial distances (geodesic distance and local radius).

T3: Topology Consistency. Applying a strict topology consistency test on a (partial) correspondence set is infeasible since two skeletons generally have different topological structures (see an example in Figure 1). Instead, we apply a local topology consistency test so that the subgraphs formed have similar local topological structures (see Figure 2). Let p_i and q_j denote the closest (in terms of geodesic distance) junction feature nodes of p_k and q_k, respectively, among the feature nodes already inserted into Ω ($i < k$ and $j < k$). If both such junction nodes exist in Ω but p_i is not matched with q_j (i.e., $i \neq j$), we reject (p_k, q_k).

T4: Spatial Configuration. A correspondence that passes the above three tests may still be a bad correspondence. Without considering any spatial information of the skeletons, T1 to T3 regard any ordering or switching of similar

[‡] Including both feature nodes and non-feature nodes (i.e., skeletal nodes with two incident bones).

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

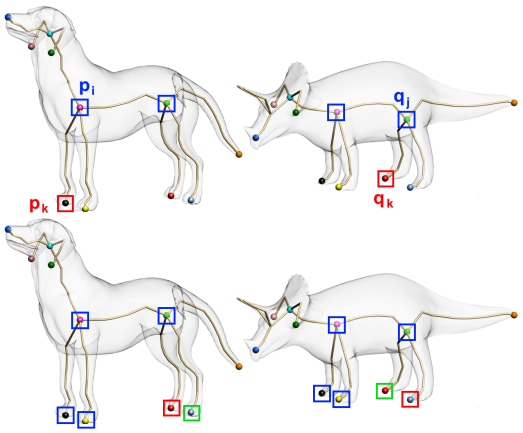


Figure 2: Topology consistency test (T3) and Spatial configuration test (T4). Blue squares are nodes that are already correctly matched. ***Top:*** *T3 rejects feature pair (p_k, q_k) since their nearest matched junction nodes (i.e., p_i and p_j) do not form a matched pair.* ***Bottom:*** *T4 detects large rotation distortion caused by flipping of similar unmatched components (e.g., hind legs).*

components at a branch node as the same, easily causing wrong matching of similar components. For example, they cannot identify local flips of hind legs of the models in Figure 2. Therefore, to avoid the symmetry-switching problem and further prune away incorrect correspondences, we consider the spatial relationship between the feature pair (p_k, q_k) and the previously matched feature pairs in Ω.

Consider the ideal case of matching an object to itself (possibly rotated). An incorrect correspondence that contains wrong matches (such as a switched matching of legs in two animal models) would involve a certain transformation which is significantly different from a pure rotation. Therefore, we measure the difference between that transformation and a pure rotation to evaluate the similarity between the spatial configuration of a pair of node sets.

Articulated models like animals and human bodies often have different poses, which also cause the transformation between matched nodes to deviate from a pure rotation. An ideal solution is to first normalize two different poses to be the same, which, however, is a challenging problem on its own due to the unknown correspondences between the poses. Thus, we chose to alleviate the influence of different poses by performing a spatial embedding using a variant of the least-squares multidimensional scaling (MDS) [EK03] which spans out the branches of the skeletons (Figure 3).

MDS maps measurements of similarity among pairs of geometric entities into an Euclidean space. Elad and Kimmel [EK03] use a least-squares MDS to normalize the poses of meshes; here we use it to normalize skeletons. We replace the similarity measurement used in [EK03] (i.e., geodesic distance along surface) with the geodesic distance between skeletal nodes along skeletal paths. Similar to [EK03], we then use an iterative majorization method to embed individual skeletons to the Euclidean space while retaining the similarity measurement in a least-squares manner. Note that the iterative least-squares MDS works directly in the original 3D domain (see accompanying video) and keeps the original spatial ordering of the skeleton nodes, which is crucial in addressing the common symmetry-switching problem. Figures 1 and 3 show some pose-normalization results.

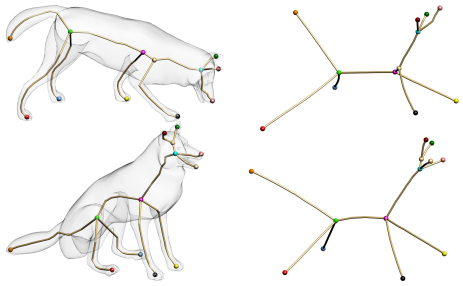


Figure 3: The skeletons are pose-normalized by performing a spatial embedding that avoids switching of symmetry components. The matching colored nodes indicate the result of our algorithm.

Our spatial configuration test is then performed on the pose-normalized skeletons. We use a linear transformation, i.e., a 3×3 matrix denoted as $\mathbf{A}_{pq}$, to represent the non-translational transformation between the two matching node sets containing p_k and q_k, respectively. Specifically, we estimate the best-fit linear transformation $\mathbf{A}_{pq}$ that maps the vectors $\{p'_{k-d} - p'_k, \ldots, p'_{k-1} - p'_k\}$ to $\{q'_{k-d} - q'_k, \ldots, q'_{k-1} - q'_k\}$, where p'_i and q'_i are the pose-normalized positions of the feature nodes. For efficient computation, we use only a small fixed number of matched nodes to determine $\mathbf{A}_{pq}$ ($d = 3$ in our case).

To determine the difference between the linear transformation $\mathbf{A}_{pq}$ and a pure rotation, we measure the difference between $\mathbf{A}_{pq}$ and its rotation component $\mathbf{R}_{pq}$ found by polar decomposition. The determinant of $\mathbf{R}_{pq}$ is -1 when the transformation involves a reflection, leading to an undesirable correspondence (i.e., a global flipping). To detect such undesirable global flipping effect, we multiply $\mathbf{R}_{pq}$ with -1 when $\det(\mathbf{R}_{pq}) = -1$. We define the rotation distortion $RD_{pq} = \|\mathbf{A}_{pq} - \mathbf{R}_{pq}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. Figure 2 shows a case with large rotation distortion. To make the distortion measure symmetric, we also compute RD_{qp} of the transformation $\mathbf{A}_{qp}$ which maps the positions $\{q'_{k-d}, \ldots, q'_k\}$ to $\{p'_{k-d}, \ldots, p'_k\}$ and let $RD_\Omega = \max\{RD_{pq}, RD_{qp}\}$. We reject a correspondence if the rotation distortion are larger than the given threshold: $RD_\Omega > \varepsilon_{RD}$.

Thresholds and Implementation Details. To achieve faster performance, we apply pose normalization to the input skeletons before starting the search process (i.e., before performing any pruning test), since it needs done only once and does not influence tests T1-T3. Our method is robust to threshold settings in the pruning tests, since the search and prune process only aims at eliminating a vast majority of bad feature correspondences rather than finding the best one. In

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

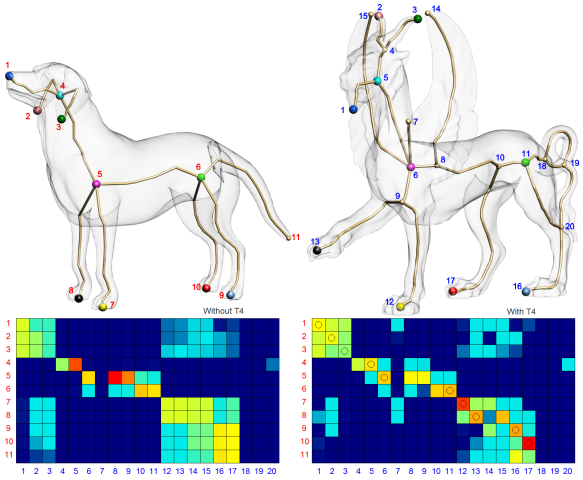


Figure 4: A visualization of votes for matching dog and feline before and after applying T4, demonstrating that T4 has good discriminative power. The winning feature pairs picked by our algorithm (circles) coincide with the manually defined ground truth.

our implementation, we simply set $\varepsilon_C = 0.5$ and $\varepsilon_{RD} = 1$. We apply T2 only to nodes at level $k \geq 2$ since at least two nodes are needed to define a skeletal path. Similarly, T4 is applied only to level $k \geq 4$, since at least four feature pairs are needed to define an affine transformation. We let the correspondence sets associated with nodes at level $k \geq 4$ be the electors. Denote the set of all electors as $C = \{\Omega_i\}$.

5. Voting Process

The set of electors may be in the order of thousands, making a detailed evaluation of each of them to select the best one still potentially costly. Moreover, it remains an extremely difficult problem to design an effective evaluation procedure that reliably yields the best correspondence. Instead, we design an efficient electors voting method. We consider all electors as equally weighted (i.e., independent of the correspondence size and number of nodes at a particular tree level). In this section, we let (p_i, q_j) denote an arbitrary feature pair, where the subscript i (j) is now the pre-defined node index of p_i in $\hat{P}$ (q_j in $\hat{Q}$). Each matching feature pair (p_i, q_j) in an elector Ω_k contributes one vote. Specifically, we construct a 2D score table $\mathbf{S}$, where the element s_{ij} is the total occurrences of the feature pair (p_i, q_j) in all electors:

$$s_{ij} = \sum_k \Phi_{i,j,k}, \quad \Phi_{i,j,k} = \begin{cases} 1 & \text{if } (p_i, q_j) \in \Omega_k, \\ 0 & \text{if } (p_i, q_j) \notin \Omega_k. \end{cases} \tag{3}$$

Clearly, a feature pair (p_i, q_j) with a higher score has a higher probability of being a correct matched feature pair. Figure 4 visualizes the votes for matching the dog and feline models. Observe that the feature pairs with high scores indeed correspond to the manually predefined ones. Moreover, the visualization shows that while the spatial configuration test T4 further prunes away wrong correspondences and increases the correctness percentage of the remaining electors, it better discriminates the correct feature pairs from the rest.

Once we have the scores for the individual feature pairs, the next task is to form the final correspondence Ω^*. Traditional algorithms for finding a maximum matching in a bipartite graph weighted by our scores are expected not to work well, since they are unable to enforce a global ordering and are confused by object symmetries [BL08]. A straightforward solution is to sum the scores of all feature pairs for each elector in C and identify the elector with the largest summed score as Ω^*. However, this gives sub-optimal results, as discussed in Section 6. Instead, we apply a greedy algorithm to construct Ω^*, All the feature pairs are first sorted by their scores, and the pair (p_i, q_j) with the highest score is iteratively added to Ω^* subject to satisfying the following conditions:

- ◇ **1-1 mapping**: Neither p_i nor q_j is already in Ω^*;
- ◇ **Topology consistency**: Same as the T3 pruning test, i.e., the closest junction nodes of p_i and q_j that are already in Ω^* must match.

Note that the objective of the topology consistency check is not to ensure that the selected feature sets form a common subgraph of the original skeletons. Due to connectivity variation in the skeletons, such a strict topology constraint would cause some features to have no match. For example, the junction nodes at the body of the dog and elephant models in Figure 1 would have no match due to their very different local connectivity. On the other hand, we note that not enforcing a strict topology checking could lead to mismatches like the junction nodes at the neck of these models.

The iteration stops when all the features of one model are matched, or if all the remaining feature pairs violate the 1-1 and topology consistency tests. This means the size of Ω^* could be larger than the maximum size of all Ω_k, implying that the voting process can in fact recover a pruned good correspondence. Since the sorting of integer scores can be done in linear time (i.e., $O(|\hat{P}||\hat{Q}|)$), the whole voting process takes only $O(|\hat{P}||\hat{Q}| + |C|(|\hat{P}| + |\hat{Q}|))$, where $|C|$ is the number of electors, and $|\hat{P}|$ and $|\hat{Q}|$ are the numbers of feature nodes in the two models. Therefore, the voting process is efficient, typically taking less than a second with 10000 electors.

6. Results and Validation

For simplicity and efficiency, we use the skeleton extraction algorithm based on mesh contraction proposed by Au et al. [ATC*08]. In any case, our algorithm is not restricted by a specific skeleton extraction method.

Timing. We assume that the skeletons and their pose-normalized versions are precomputed and they form part of the object's representation. Their computation time depends on the model size, taking about 7s for models with 10,000 vertices [ATC*08]. All the data were recorded on an Intel Core 2 Dual T5750 machine with 2GB memory, using a single thread implementation. In addition, we also precompute the radius and centricity of each skeletal node and the geodesic distances between every pair of feature

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

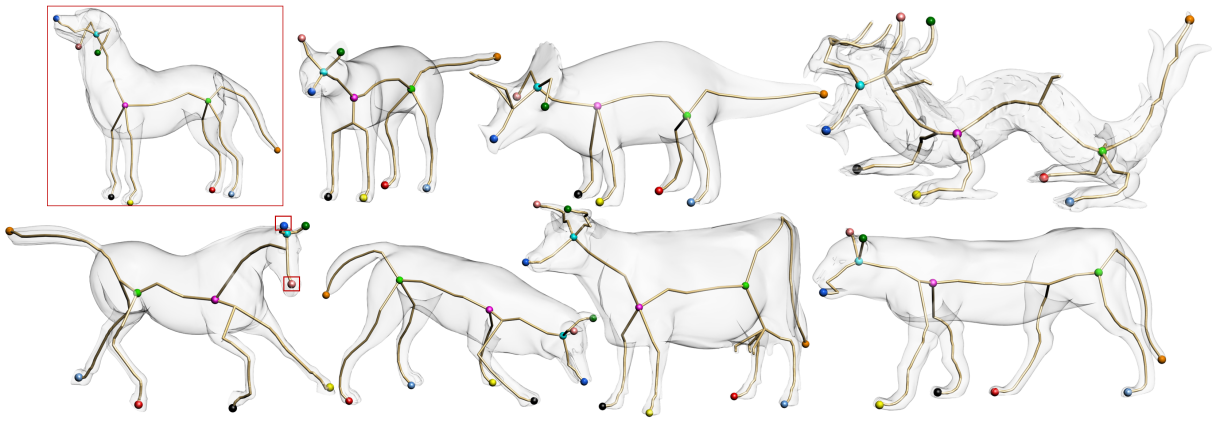


Figure 5: Matching a dog with different four-legged animals with different surface details: cat, triceratop, Asian dragon, horse, wolf, cow and tiger. Incorrect feature pairs are marked with red squares. Unmatched features nodes are not drawn.

nodes. This takes negligible time due to the small skeleton size. In terms of memory usage and processing time, the correspondence algorithm is dominated by the searching process, which is dependent on the number of features. For matching the dog with various animals, the searching process takes: 1.05s (horse), 0.74s (cow), 4.20s (feline) and 4.68s (elephant). In contrast, achieving comparable results, the technique of [ZSCO*08] reported taking dozens of minutes for shapes with few thousands vertices and the approach of [LF09] takes around 6 minutes. For matching objects with many feature nodes, interactive response may be unattainable. For example, the two elephant pair in Figure 6 (22 and 23 features) took 32s, thus some simplification of the skeletons might be necessary.

Results. We first demonstrate the effectiveness of our method on four-legged animals since they vary greatly in shapes and have clear semantic correspondences. Figure 5 shows our correspondence results for the dog and seven other four-legged animals. Our method produces good quality correspondences for all these pairs in that prominent features (e.g., legs and tail) are all correctly matched, including pairs with large geometry difference like dog with Asian dragon, and dog with feline. Small features such as ears and nose may be mismatched in some cases, due to small nearby features having similar skeletal information (e.g., node positions, centricity, path lengths) and thus difficult for the pruning tests to distinguish. We argue that the challenge in the general correspondence problem is to find a correct coarse mapping, which is typically manually defined. Once known, existing techniques can be applied to refine a coarse correspondence (e.g., [SP04]).

#L	T1 only		T1 to T2		T1 to T3		T1 to T4	
1	30	17%	30	17%	30	17%	30	17%
2	3390	8%	1488	10%	1488	10%	1488	10%
3	159840	8%	20491	13%	20148	13%	20148	13%
4	n/a	n/a	115896	16%	111252	16%	13550	20%
5	n/a	n/a	316059	19%	290863	19%	15314	25%
6	n/a	n/a	469487	23%	405760	23%	11992	27%
7	n/a	n/a	417734	29%	334098	30%	4724	38%
8	n/a	n/a	234298	33%	173424	34%	1214	51%
9	n/a	n/a	76372	37%	51672	38%	147	88%
10	n/a	n/a	11448	40%	6544	40%	–	–

Table 1: Pruning data for dog-elephant. Each pruning test successively reduces the search space and increases the correctness percentage. Electors are at level $L \geq 4$. n/a mean out-of-memory error.

Our correspondence algorithm is general, allowing matching a wide variety of models as shown in Figure 6 and the supplemental material. These examples also demonstrate that our algorithm is not restricted to symmetric or loop-free skeletons, and can match fairly complex shapes such as the motorcycles.

Pruning Performance. Table 1 lists the number of electors at each tree level for the dog and elephant models in Figure 1, with 11 and 23 feature nodes, respectively. Observe that using both T1 and T2 greatly reduces the search space which is no longer exponential. T3 and T4 successively further prune away more correspondences. T4 has good discriminative power, but requires relatively costly computation, hence it is applied last. In this example, applying T4 before T3 (for nodes at level 4 and above) would increase the searching time from 3.9s to 4.2s.

Thresholds. Our method is insensitive to thresholds thanks to the robustness of the voting process against certain missing data. The thresholds are also insensitive to the model type since the tests are not used to select a best correspondence. For example, experimentally we did not find any noticeable difference in the quality of the correspondences with ε_C set within $[0.4, 0.7]$ and ε_{RD} within $[0.5, 1.5]$. In general, the thresholds setting has to tolerate local shape differences between the two input models while distinguishing global structural differences in bad correspondences. We found that the current setting gives a good balance between the two roles. It is noteworthy that the pruning tests may leave behind some bad correspondences as well as prematurely prune away some subtrees containing good correspondences. However, the subsequent voting process is tolerant to such fuzziness in the electors set due to its high degree of correctness. This tolerance is illustrated in Table 2. In this

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

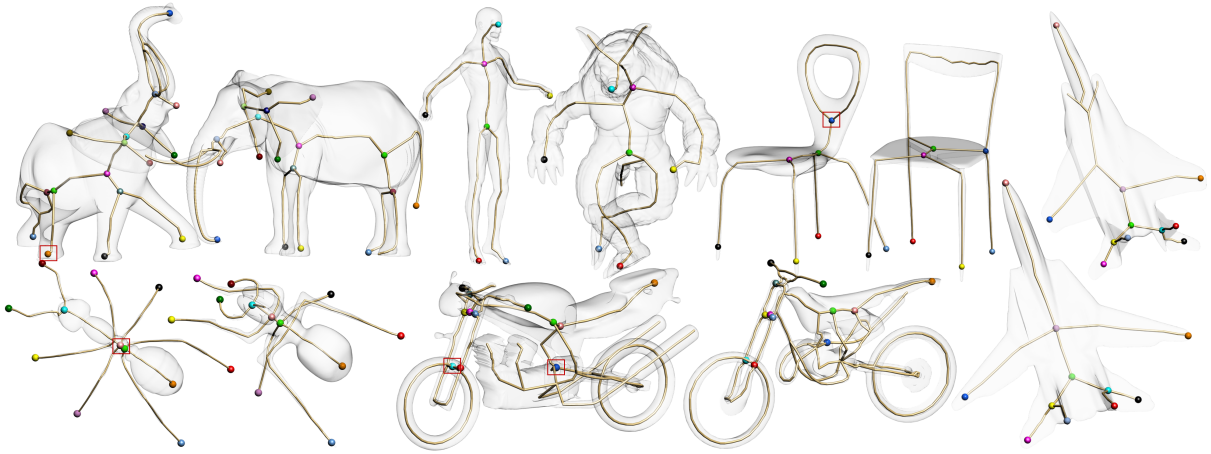


Figure 6: Our correspondence algorithm is general, allowing matching of a wide variety of model pairs, which may have different poses, surface details, partially matching semantic components, and small topological difference (e.g., tail of elephant and engine of motorcycles).

Level	dog-cat		dog-cow		dog-horse		dog-triceratop	
1	12	25%	12	25%	17	18%	12	25%
2	330	21%	456	20%	582	18%	423	23%
3	2365	23%	4305	20%	5532	19%	4117	25%
4	1883	29%	2008	27%	3991	23%	3110	35%
5	1692	37%	1926	34%	3869	29%	3875	44%
6	826	45%	1474	44%	2232	41%	3401	55%
7	151	55%	572	60%	594	53%	1953	68%
8	12	88%	72	69%	99	68%	651	77%
9	–	–	–	–	9	85%	113	84%
10	–	–	–	–	–	–	7	91%

Table 2: Comparing the correspondence sets at each tree level with manually tagged ones. Note the increasing correctness percentage at higher i

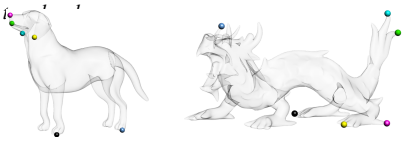


*Figure 7: The method of Zhang et al [ZSCO*08] fails to establish a reasonable correspondence for models with very different geometry (bottom).*

example the largest electors have 9 feature pairs, but the correspondence Ω^* found by our algorithm has 11 matching feature pairs (Figure 5), demonstrating that simply selecting the final correspondence from the set of electors may give sub-optimal results.

Validation. Table 2 shows the percentages of the correct feature pairs in the correspondence sets at each tree level compared to the manually predefined ground truth. Observe that the percentages increase with the tree level, showing that the pruning tests successfully filter away bad correspondences. Since we use only the correspondence sets at level 4 and above as the electors, we avoided those with relatively more incorrect matching pairs at the first three levels. Table 3 shows the high precision and recall scores of our algorithm when cross-matching the eight animal models in Figure 5. Precision is defined as a/b and recall as a/c, where a is the number of correct feature pairs returned by our algorithm, b is the number of feature pairs output by our algorithm, and c is the number of manually tagged feature pairs.

	#1	#2	#3	#4	#5	#6	#7	#8
#1		11/11	11/11	11/11	11/11	11/11	11/11	9/11
#2	11/11		8/9	11/11	10/12	12/12	11/12	10/12
#3	11/11	8/11		11/11	12/12	11/11	7/9	12/12
#4	11/11	11/11	11/11		10/11	11/11	14/14	7/8
#5	11/11	10/12	12/12	10/11		12/12	11/13	10/11
#6	11/11	12/12	11/11	11/11	12/12		10/12	9/12
#7	11/11	11/12	7/11	14/14	11/13	10/12		10/14
#8	9/11	10/12	12/12	7/11	10/13	9/12	10/14	

Table 3: Precision (upper triangle) and recall (lower triangle) scores of matching the 4-legged animals in Figure 5 using our algorithm. #1:Dog, #2:Cat, #3:Cow, #4:Triceratops, #5:Tiger, #6:Wolf, #7:Asia Dragon, #8:Horse.

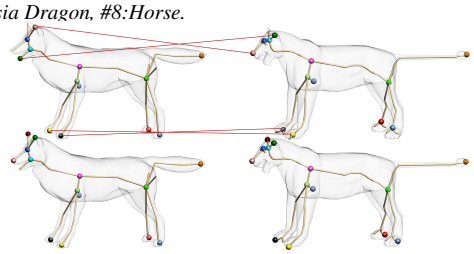


Figure 8: Comparing our result (bottom) with Biasotti et al. [BMSF06] (top), which has symmetry switching problem and incorrect matches (highlighted by red lines) due to construction of a common subgraph.

Comparisons. We compare our method with the method of Zhang et al. [ZSCO*08] in terms of correspondence results. Since directly running their algorithm on high-resolution models is computationally prohibitive, we first decimate the models down to a few thousand vertices before running both algorithms. For models exhibiting moderate variation of geometric details, their method is able to find good correspondences comparable to ours. However their method fails to produce reasonable results for models with very different geometry, as shown in Figure 7. We have also compared our method with [BMSF06] which searches for a subgraph isomorphism between two DAGs created by simplifying Reeb graphs defined on the object surface. For fairness of comparison, we use their Reeb graphs as input to our algorithm. Like other skeleton-based matching

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

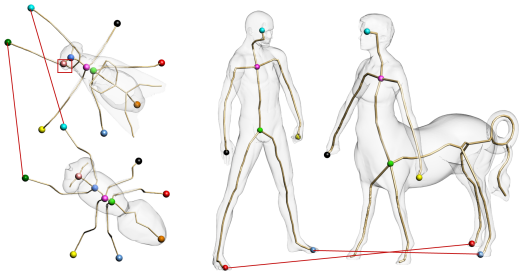


Figure 9: Our algorithm may produce semantically incorrect correspondences when (left) components have different semantics but similar geometry or (right) shapes have large structural difference.

methods, their method cannot handle symmetry components (see switched fore legs in Figure 8). Besides, there are mismatched features at the heads due to the strict construction of a common subgraph between the matching nodes.

Limitations. Generally speaking, semantic correspondence is an ill-posed problem. Our automatic algorithm is purely geometric-based, oblivious to the semantics of parts. Therefore it may mismatch components with different semantics but similar geometry (e.g., insects pair in Figure 9). This may lead to large errors (e.g., mismatch long tail and neck of two dinosaurs or back-and-front of human bodies) or small local errors (e.g., small components in Figure 5). We speculate that high-level shape signatures (e.g., face detection, symmetry detection) are needed to further improve matching quality. However, this likely leads to slower response and, more importantly, loses the generality of the method in matching a wide variety of shapes. Another solution is to allow limited user interaction when the fully automatic solution fails. The user can specify one or more feature pairs that must appear in all electors.

Finally, our algorithm may fail to produce desirable correspondences between models whose curve-skeletons do not represent well the object shapes (e.g., man-made models such as some chairs) or models with large extra components which imply large structural difference (e.g., centaur and human in Figure 9).

7. Applications

We demonstrate the use of our automatic correspondence algorithm in two applications.

Animation Transfer. Making animation transfer more accessible to non-expert users is an application of our automatic correspondence solution. Typically, character animation is transferred through compatible skeletons of different characters. The compatible skeletons are usually specified by users through a time-consuming process, limiting the use of animation transfer systems to only professional users. A notable exception is the work of Baran and Popović [BP07], which automatically embeds a template skeleton to models of the same class but possibly with very different geometric details. Unlike their work, we do not rely on any

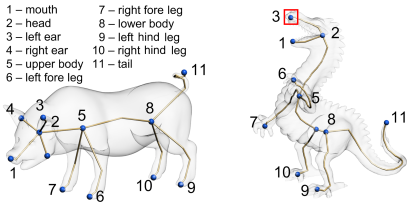


Figure 10: Automatic tagging of shape components using the models in Figure 5 as pre-tagged training set. One node is incorrectly tagged.

existing template skeleton, but construct compatible skeletons directly from shape correspondence. Since our correspondence algorithm already provides a 1-1 node correspondence, we need only form compatible paths connecting the nodes. For simplicity, we assume that compatible skeletons are tree structures. This allows us to keep the (unique) shortest geodesic path between every pair of geodesically adjacent nodes that have corresponding nodes on the other skeleton as compatible paths. Possible overlapping sub-paths are then split to ensure a 1-1 path correspondence. In the accompanying video, we demonstrate the use of our compatible skeletons in producing visually pleasing animation transfer between characters with different geometric details.

Tagging Shape Components. Providing semantic information as prior knowledge to an automatic shape correspondence solution can increase its accuracy. The fast speed of our correspondence algorithm allows matching of many models within a reasonable time. Assuming the availability of a training set containing models of similar semantic structures and the model parts are tagged as left/right fore/hind legs, upper/lower body, tail, head, mouth, left/right ears, etc. Then, given a new input model, we apply our shape correspondence algorithm to match the new model with each of the training models. By summarizing the tags of the matching features of the training models, we tag the parts of the new model. We only tag a feature if the matching features of the training models

have consistent tags. Specifically, if the majority tag is less than half of the size of the training set, we mark the feature as uncertain and do not assign a tag. As an example of this application, we automatically tag the raptor and pig in Figure 10 using the 4-legged animals in Figure 5 as our training set.

8. Conclusion

We have presented a fast and fully automatic correspondence algorithm that allows matching of a wide variety of shapes with semantically similar structures but different geometric details. We avoided the potentially slow approach of direct searching to find the best correspondence. Instead, we designed an electors voting scheme whereby the electors are quickly selected through a combinatorial search. This large set of reliable electors then enables the use of a fast statistical voting process to construct a high quality correspondence.

Our algorithm represents a step towards solving the chal-

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.

lenging problem of automatic shape correspondence. There are many possible ways to extend our algorithm, e.g., extending it to handle shapes with larger structural and topological differences, and incorporating limited semantic analysis to improve the accuracy of automatic correspondence. Finally, besides shape correspondence, we believe that our framework can offer efficient solutions to other problems that have no well-defined optimal criteria.

Acknowledgement

We would like to thank the anonymous reviewers for their constructive comments, and Dror Aiger and Hao (Richard) Zhang for valuable feedback. We also thank Richard for the comparison example in Figure 7 and Michael Brown for video narration. This work was supported in part by grants from the Hong Kong Research Grant Council (Project No. GRF619908), the City University of Hong Kong (Project No. 7200148) and the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities.

References

[AMCO08] AIGER D., MITRA N. J., COHEN-OR D.: 4-points congruent sets for robust surface registration. *ACM TOG 27*, 3 (2008), 85. 3

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. *ACM TOG 27*, 3 (2008), 44. 6

[BL08] BAI X., LATECKI L.: Path similarity skeleton graph matching. *IEEE TPAMI 30*, 7 (2008), 1282–1292. 3, 6

[BMSF06] BIASOTTI S., MARINI S., SPAGNUOLO M., FALCIDIENO B.: Sub-part correspondence by structural descriptors of 3D shapes. *CAD 38*, 9 (2006), 1002–1019. 2, 8

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3D characters. *ACM TOG 26*, 3 (2007), 72. 9

[BR07] BROWN B., RUSINKIEWICZ S.: Global non-rigid alignment of 3-D scans. *ACM TOG 26*, 3 (2007), 21. 2

[CDS*05] CORNEA N. D., DEMIRCI M. F., SILVER D., SHOKOUFANDEH A., DICKINSON S. J., KANTOR P. B.: 3D object retrieval using many-to-many matching of curve skeletons. In *SMI* (2005), pp. 368–373. 2

[CR03] CHUI H., RANGARAJAN A.: A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding 89*, 2-3 (2003), 114–141. 2

[CZ08] CHANG W., ZWICKER M.: Automatic registration for articulated shapes. *SGP '08 27*, 5 (2008). 2, 3

[CZ09] CHANG W., ZWICKER M.: Range scan registration using reduced deformable models. *EG 28*, 2 (2009), 447–456. 2

[DvLV08] DEMIRCI M. F., VAN LEUKEN R. H., VELTKAMP R. C.: Indexing through laplacian spectra. *Comput. Vis. Image Underst. 110*, 3 (2008), 312–325. 2

[EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE TPAMI 25* (2003), 1285–1295. 5

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395. 3

[FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3D shapes with priority-driven search. In *SGP '06* (2006). 3

[GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM TOG 25*, 1 (2006), 130–150. 3

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. *SGP* (2005). 2

[GSCO07] GAL R., SHAMIR A., COHEN-OR D.: Pose-oblivious shape signature. *IEEE TVCG 13*, 2 (2007), 261–271. 2

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. *Comput. Graph. Forum 27*, 5 (2008), 1449–1457. 2

[Heb49] HEBB D. O.: *The Organization of Behavior*. JohnWiley, 1949. 2

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. *SIGGRAPH* (2001), 203–212. 2

[JZ07] JAIN V., ZHANG H.: A spectral approach to shape-based retrieval of articulated 3D models. *CAD 39* (2007), 398–407. 2

[LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM TOG 28*, 3 (2009), 72. 1, 3, 7

[LG05] LI X., GUSKOV I.: Multi-scale features for approximate alignment of point-based surfaces. In *SGP '05* (2005), p. 217. 3

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. *Comput. Graph. Forum 27*, 5 (2008), 1421–1430. 2

[LW88] LAMDAN Y., WOLFSON H. J.: Geometric hashing: A general and efficient model-based recognition scheme. In *ICCV '88* (December 5–8, 1988), pp. 238–249. 3

[SF07] SHILANE P., FUNKHOUSER T.: Distinctive regions of 3D surfaces. *ACM TOG 26*, 2 (2007), 7. 2

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (2004), 399–405. 7

[SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S. J.: Skeleton based shape matching and retrieval. In *SMI '03* (2003), pp. 130–139. 2

[TS04] TUNG T., SCHMITT F.: Augmented reeb graphs for content-based retrieval of 3d mesh models. In *SMI '04* (2004), pp. 157–166. 2

[TVD09] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Partial 3D shape retrieval by reeb pattern unfolding. *Computer Graphics Forum 28*, 1 (2009), 41–55. 2

[ZSCO*08] ZHANG H., SHEFFER A., COHEN-OR D., ZHOU Q., VAN KAICK O., TAGLIASACCHI A.: Deformation-driven shape correspondence. *EG 27*, 5 (2008). 1, 3, 4, 7, 8

© 2009 The Author(s)
Journal compilation © 2009 The Eurographics Association and Blackwell Publishing Ltd.