Sketch-Based Articulated 3D Shape Retrieval

Yusuf Sahillioğlu ■ *Middle East Technical University*

Metin Sezgin ■ *Koç University*

Sketch-based queries are a suitable and superior alternative to traditional text- and example-based queries for 3D shape retrieval. The authors developed an articulated 3D shape retrieval method that uses easy-to-obtain 2D sketches. It does not require 3D example models to initiate queries but achieves accuracy comparable to a state-of-the-art example-based 3D shape retrieval method.

Three-dimensional shape retrieval is an active area of research with many practical applications. Sketches serve as natural and exceptionally powerful means for expressing visual information. This property makes sketch-based queries a suitable and superior alternative to traditional text-based queries for 3D shape retrieval. Articulated 3D shape retrieval specifically facilitates the efficient exploration of large repositories that accommodate a growing number and variety of 3D models.[1] For this reason, we developed a method that uses 2D sketches to retrieve 3D articulated shapes. Because articulated shapes are a superset of nonarticulated rigid shapes, our method can be compared with the view-based methods that handle only rigid transformation.

The most basic retrieval systems use text-based queries, which require the tedious task of annotating each database model with metadata. More recently, retrieval systems have evolved to support query-by-example (QbE) schemes, in which the query object is of the same type as the database models. Although QbE is much easier to use than the text-based queries, there is a more intuitive and even simpler hybrid approach that facilitates 2D sketch queries for 3D shape retrieval.

In this article, we focus on the query-by-sketch approach for 3D shape retrieval from a novel perspective. We enable articulated 3D shape retrieval from sketches, which to the best of our knowledge, has yet to be investigated.

Attempting to query articulated shapes using 2D sketches gives rise to certain challenges as well as opportunities that are specific to this problem. Unlike existing approaches to sketch-based retrieval that typically adopt an articulation-variant view-based setup, we cast the problem in a purely geometry-based framework that computes articulation-invariant pairwise distances over 2D sketches and 3D models. To this effect, we use the well-defined geodesic distances on the surfaces of 3D models. For 2D sketches, on the other hand, we apply a good continuation rule[2] to construct a query graph that enables geodesic distance computations with respect to depth. We demonstrate the benefit of employing a good continuation rule by comparing our method with a baseline method that lacks such a rule. We also show superior performance over the state of the art in sketch-based retrieval of shapes with no articulation. It is easy to test our algorithm with the public offline sketch repositories because we do not require any information other than 2D sketch point locations, which are readily available in bitmap images. This feature also shows the potential of our algorithm for future interesting applications, such as composing a 3D scene from a single 2D bitmap image.

The source code, executables, and query sketches for the method we present here are available at www.ceng.metu.edu.tr/~ys/pubs.

Background in Shape Retrieval

Shape retrieval problem can be investigated under three categories based on the query specification

 Published by the IEEE Computer Society

method: keyword-based retrieval, example-based retrieval, and sketch-based retrieval. Each category can then be further analyzed based on the similarity measure considered while matching the query with the database models—namely, rigid matching and articulated matching.

Keyword-based retrieval is trivial, but it requires tedious annotation work to facilitate the comparison of strings describing the query and the database models. Example-based retrieval methods, which are much more intriguing, have received great deal of attention in the graphics and vision communities. These methods count on the availability of full models to serve as queries, however. This requirement renders these methods relatively impractical because a good example is rarely available.

To overcome this problem, shape retrieval research is shifting toward sketch-based applications, starting with the early works that used sketches in combination with keywords, going all the way to the most recent purely sketch-based method.[3] In this article, we propose another purely sketch-based method, but unlike the previous approach[3] that supports only rigid matching, we allow articulated matching, which is a superset handling rigid transformations and bending deformation. To the best of our knowledge, our work is the first attempt to address articulated shape matching using sketch queries.

In rigid shape retrieval, the query and the database models are typically represented with global shape descriptors that are invariant to rigid transformations—that is, rotation, translation, and uniform scaling.[4] The retrieval task then reduces to the efficient comparison of low-dimensional descriptors, such as statistical moments, spherical harmonics, wavelets, shape contexts, spin images, shape distributions, and lightfield descriptor. Note that some researchers[5] do allow sketch queries as part of their example-based retrieval engines for rigid shapes.

As for articulated shape retrieval, there are two popular approaches explored thus far. The first is the descriptor-based approach, which works in the same way as the rigid retrieval case, except the descriptors are invariant to both rigid transformations and articulations. For instance, one approach uses heat kernel signatures as descriptors.[6] Other choices include eigenvalues of the Laplace-Beltrami differential operator, a histogram of gradients, and multiresolution wavelet-based shape signatures.

The second approach to articulated shape retrieval is the use of geodesic distances that are invariant to articulations, or equivalently isometric deformations. Recall that if two shapes are perfectly isometric—that is, one is the articulated version of the other—then the geodesic distance between any two points on one shape is exactly the same as the geodesic distance between the corresponding points on the other. One approach exploited this fact by creating canonical forms based on the preservation of geodesic distances under articulation (bending).[7] These canonical forms, representing the intrinsic geometry of shapes in a low-dimensional Euclidean space, can then be aligned efficiently with iterative closest point like algorithms for comparison purposes. Zhouhui Lian and his colleagues improved the retrieval accuracy by computing and comparing detail-preserving canonical forms in a framework where near-rigid mesh segments are deformed toward the corresponding components on the distorted least-squares multidimensional scaling (MDS) canonical pose.[8] Reeb graphs defined by geodesic distances are also suited for the retrieval of the articulated objects.[7] One common problem for all geodesic-based methods is their sensitivity to slight topological shape changes, such as connecting a human's toes with one edge, may alter most of the pairwise distances drastically. To alleviate this problem, Juan Zhang and his colleagues handled shape articulation via skeletal graphs based on medial erosions,[9] whereas Alexander Bronstein and his colleagues used diffusion distances.[6]

Our work is the first attempt to address articulated shape matching using sketch queries.

Sketching is the most intuitive and convenient querying scheme for novice users of shape retrieval applications.[3] The challenges in sketch-based shape retrieval applications are to infer the intent of the user who has limited drawing skills and to establish a connection between the 2D sketch data and the 3D model data to facilitate similarity comparisons. The former relates to the sketch-recognition problem, which has been studied extensively.[2,10] These recognizers differ from our sketch interpretation scheme in that we do not use any time or stroke information during the process, which makes our algorithm workable with offline binary sketch images, such as the ones in the benchmark developed by Mathias Eitz and his colleagues.[3] The latter challenge, on the other hand, is thus far addressed with the view-based

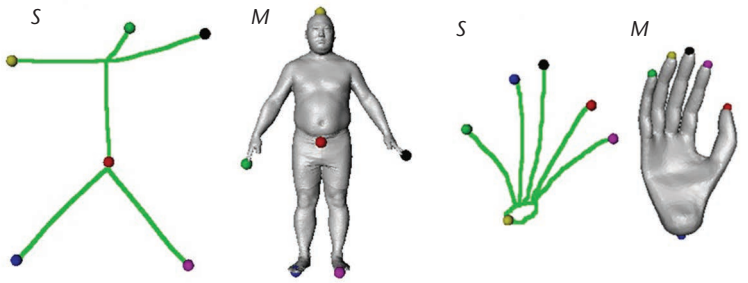

Figure 1. Sampling of 2D and 3D shape extremities (spheres) using the same farthest-point sampling framework. S and M represent the tips and center of shapes that can be computed efficiently via farthest-point sampling based on geodesic distances.

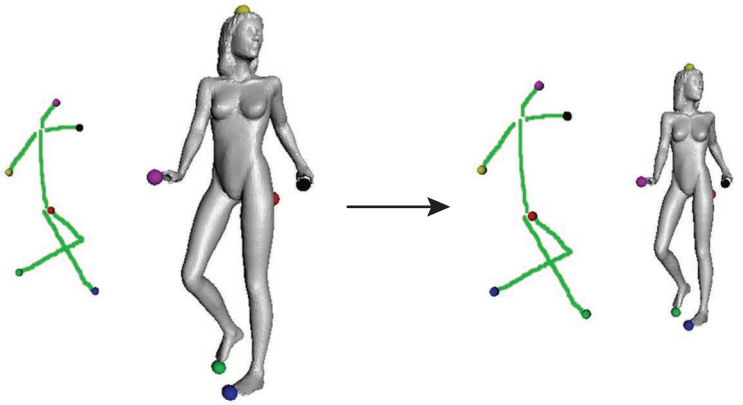

Figure 2. 2D query sketch and 3D database model brought to the same scale. We achieve scale invariance between sets by normalizing their maximum geodesic distances to the same value of 1.

similarity methods,[3,5,11] all of which compare nonphotorealistically rendered multiple views of 3D database models with the 2D sketch data using image matching techniques such as bag of features.[3]

Our method takes the opposite approach by implicitly lifting the 2D sketches to 2.5D instead of lowering the 3D models to 2D. In other words, we perform 3D matching by inferring 3D depth information from possibly self-intersecting sketches using a rule that is based on Gestalt's principle of good continuation.[2] This approach not only avoids information loss or distortion in projections but also allows articulated retrieval using geodesic-driven 3D isometric shape correspondence methods.[12–14]

Once a robust sketch-based shape retrieval system is built, an intriguing application would be the composition or modification of 3D scenes from 2D sketches or images. Existing techniques[15] rely on the fact that each scene object is already available as a separate sketch, opting out of the possibility of using offline 2D images as input. We believe that our offline sketch-based retrieval system, coupled with edge detection and sketch recognition algorithms, may realize this difficult, yet equivalently interesting scenario.

Overview

To perform a pose-independent 3D model retrieval using an offline 2D sketch query, we consider two metric spaces: (S, g) for the sketch and (M, g) for the database model, where S and M are sets and g is a notion of the distance between the elements in each set for which we use a geodesic distance that is invariant to articulated (isometric) shapes.

We discretize the sets to the points at the shape extremities—that is, S and M represent the tips and center of shapes that can be computed efficiently via farthest-point sampling based on geodesic distances g (Figure 1). We then launch our matching algorithm, which consists of the following steps:

1. Achieve scale invariance between S and M by normalizing their maximum geodesic distances to the same value of 1 (Figure 2). Update g such that it stores these normalized distances.
2. Search for an optimal mapping $\phi^* : S \rightarrow M$ with minimum distortion. To this end, minimize the isometric distortion function D_{iso} via combinatorial search over all possible mappings. We use the following isometric distortion function, which can be seen as a variant of the various distortion functions from the 3D shape correspondence literature[12,13]:

$$D_{\text{iso}}(\phi) = \frac{1}{|\phi|} \sum_{(s_i, m_j) \in \phi} d_{\text{iso}}(s_i, m_j), \tag{1}$$

where ϕ denotes the set of corresponding point pairs (s_i, m_j) between S and M, and

$$d_{\text{iso}}(s_i, m_j) = \frac{1}{|\phi| - 1} \sum_{\substack{(s_i, m_j) \in \phi \\ (s_l, m_n) \neq (s_i, m_j)}} \left| g(s_i, s_l) - g(m_j, m_n) \right|. \tag{2}$$

Using Equation 1, we can measure and quantify the deviation of any given mapping from isometry.
3. Quantify the similarity between S and M with $D_{\text{iso}}(\phi^*)$ as the sort key and decide whether M should be retrieved from the database.

Geodesic Distances with Good Continuation

The key to our algorithm's success—that is, sampling (Figure 1) and matching (Equation 1) of the shape extremities—is the accurate computation of pairwise geodesic distances g on the query S and model M. Thanks to the available 3D geometry, g for M is computed robustly using the standard Dijkstra's shortest-paths algorithm. The real chal-

lenge is to obtain g for a hand-drawn S, for which we construct a sparse graph from scratch and follow a good continuation rule on it.

Sparse Graph Generation

Given a set of dense unorganized 2D points forming our offline sketch input, we first resample evenly spaced points S' for both efficiency and accuracy reasons. The spacing s here is one-fourth of the diagonal of the bounding box, as suggested in earlier work.[16] Our resampling algorithm arbitrarily selects an unseen point from the initial dense set, adds it to S', marks all points that are at a distance less than s, and repeats the process until no available vertex is left (see Figure 3a).

We construct our sparse graph G = (S', E) on vertices S' with the closest-point connections E in the next-best manner as follows. For each sample s'_i, we create an edge between s'_i and the closest point s'_j subject to two constraints on s'_j: edge (s'_i, s'_j) in E does not already exist, and $\|s'_i - s'_j\| < 2s$. The first constraint makes s'_j the next-best candidate to be connected with s'_i. The first closest point may already have been connected to s'_i in a previous iteration, in which case s'_i gets an additional neighbor in the other direction like a chain (Figure 3b). The second condition simply prevents unrealistic pairings between distant vertices.

Stroke Generation

Given G, we want to extract rigid strokes with good continuations. That is, we want to cluster a set of connected vertices into a line as long as they respect the good continuation rule we employ. This rule is based on the Gestalt's principle of good continuation stating that we are less likely to group elements with sharp abrupt directional changes as being one object, or one stroke in our context.[2] These rigid strokes are important because they will later be used as the edges in the Dijkstra's shortest-paths algorithm.

For stroke generation, we initially unmark all samples. We select a sample s'_i of degree 1 as the starting point of stroke R_a. By constructing G, s'_i is guaranteed to land at the tip of an object extremity, which makes it a good starting point. We then add the first neighbor s'_j of s'_i to R_a and run our main loop that extends $R_a = \{s'_i, s'_j\}$ with good continuation as follows. We select the next unmarked neighbor of s'_j (say, s'_k) and check the linearity ratio

$$\rho(s'_i, s'_j, s'_k) = \frac{\|s'_i - s'_j\| + \|s'_j - s'_k\|}{\|s'_i - s'_k\|}, \quad (3)$$

which is close to 1 if the path from s'_i to s'_k is close to a line, or in other words, there is no bending as we are move from s'_i to s'_k. With that, we add s'_k to R_a if ratio $p < t$. Then we repeat the main loop with the following shift: $s'_i = s'_j$ and $s'_j = s'_k$. We empirically set threshold t = 1.05 and stop the extension of R_a when there is no unmarked neighbor of the new s'_j. Note that a similar window search process is used in the polyline simplification algorithm in earlier work.[16] The main advantage of our algorithm is that we do not require any polyline setup in the input.

There are some details that complete our stroke generation mechanism. First, we start the next stroke R_b with the next unmarked s'_i of degree

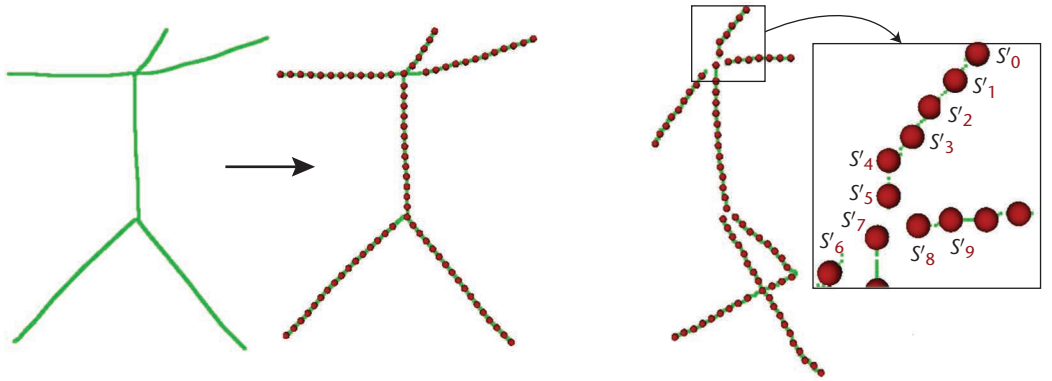

Figure 3. Sparse graph generation. (a) Initial dense sketch points (green) are resampled (red spheres). (b) The resampling process is illustrated in the zoomed-in region within the box. For s'_0, s'_1 is selected as the neighbor. For s'_1, because the closest point s'_0 is already paired, the next-best s'_2 is selected as the second neighbor, which in turn grows the edge list like a chain. Later on, s'_4 is processed, creating the edge (s'_5, s'_4). When s'_5 is processed next, (s'_5, s'_8) is created, and then when s'_7 is processed (s'_5, s'_7) is created. In the end, the degree of s'_5 becomes 3.

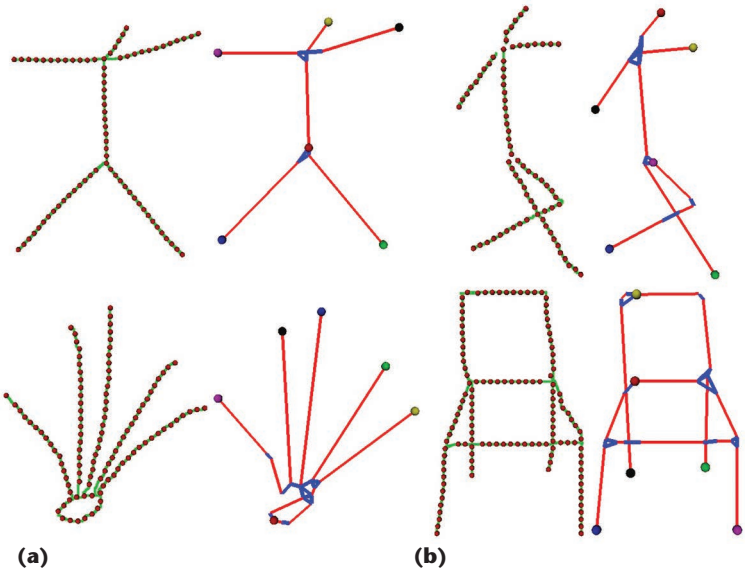

Figure 4. Abstraction of sketches with a set of lines, namely strokes (red) and virtual strokes (blue). Geodesic distances based on these lines lead to successful farthest-point sampling (big spheres). There is no bending on the red lines despite the self-intersections on the input sketches: (a) human and (b) chair.

1. When we run out of unmarked samples of degree 1, we start the new stroke with an unmarked sample of degree d, with the assumption that high-degree samples fall on the sketch's joint locations, such as the beginning of a torso that connects the head and arms. We generously set $d = 10$ and decrease it by 1 until there are no more unmarked samples left to be added to a new stroke. Note that the strokes generated with this incremental algorithm are stored as polylines.

As the second extension, we apply the Douglas-Peucker polyline simplification algorithm[17] to segment our strokes further, if necessary. The purpose is to benefit from that algorithm's global perspective compared with our relatively local triplet-based decider in Equation 3, which might miss some difficult bends. We store the final strokes in $E' = \{R_a\}$, where each stroke is represented as an edge with the two endpoints of the polyline—that is, $R_a = (s'_i, s'_j)$. The endpoints of these disjoint strokes constitute the vertex set V' (subset of S') of our reduced graph G' that abstracts the input sketch, where $|V'| = 2|E'|$.

The red lines in Figure 4 illustrate the set of strokes E′ produced by the algorithm in this section. All these strokes respect the good continuation rule we employ. That is, there is no bending on the red lines despite the self-intersections on the input sketches (human and chair).

Stroke Connections

We connect strokes with endpoints that are close to each other in order to facilitate further geodesic distance computations on G'. To this end, we conservatively initialize our closeness threshold $k = s$ (spacing) and add a virtual edge (s'_i, s'_k) to our virtual stroke set E'' if $||s'_i, s'_k|| < k$, where s'_i, is an endpoint of a stroke R_a in E' and s'_k is one of the endpoints of another stroke R_b in E'. We then test if the set of all edges E' and E'' are sufficient to connect all vertices in V'.

To realize this, we launch Dijkstra's shortest-paths algorithm for each s'_i in V' and check whether this traversal reaches all other points $\{s'_i\}$ in V'. If the result is affirmative, then we are satisfied with our current virtual stroke set E'' with the conservative value. If not, then we recompute E'' from scratch using a larger threshold of $k = k + 0.5s$.

The important threshold is computed as automatically as possible with this connection algorithm. It is also selected conservatively in order to minimize the risk of connecting irrelevant strokes. The blue lines in Figure 4 illustrates the set of virtual strokes E'' produced by the algorithm in this section.

Distance Computation

The graph $G' = (V', E' \text{ union } E'')$ describes the reduced graph abstracting the sketch with a few prominent vertices and sparse continuation-friendly edges. The computation of geodesic distances on the sketch boils down to the standard shortest-paths problem on G' with edges that are weighted by the Euclidean distances between their endpoints. Namely, we run Dijkstra's shortest-paths algorithm from each vertex $\{s'_i\}$ in V', which sets the desired pairwise geodesic distances g to be used in farthest-point sampling and matching (Equation 1).

Results

We conducted sketch-based articulated 3D shape retrieval experiments on three articulated, one nonarticulated, and one hybrid 3D model database—namely, the SHREC (3D Shape Retrieval Contest) Watertight database,[18] McGill articulated 3D shape benchmark,[19] TOSCA (Tools for Nonrigid Shape Comparison and Analysis) dataset,[12] Princeton Shape Benchmark (PSB),[4] and our Global database. The SHREC Watertight database consists of 3D human, cup, glasses, airplane, ant, chair, octopus, table, teddy, hand, pliers, fish, bird, spring, armadillo, bust, mechanics, bearing, vase, and four-legged classes, each containing 20 instances. We used all the instances from each class to create a database of 400 models. We also used all 255 models in 10 different categories from the McGill set. This set enables a fair comparison.[8] The number of vertices in the Watertight and McGill database models range from 1,000 to 25,000.

We also used the full TOSCA database consisting of 35 four-legged animals, 39 humans, and six centaurs, for a total of 80 objects, each with approximately 50,000 vertices. For comparison purposes, part of the PSB was used.[3] Many of the models in the PSB are not connected, which makes them unsuitable for our geodesic distance computations. We therefore manually connected and used 100 models that we selected uniformly over the categories of the whole collection.

Finally, we put all 835 models from these four databases into our Global database to test our algorithm's retrieval performance on a large 3D shape set.

Hand-Drawn Sketch Database

For each database, the retrieval system requires a hand-drawn sketch of an articulated/isometric pose of the target object that we wish to retrieve. To evaluate the retrieval system, we compiled a database of hand-drawn sketches. These sketches were not constrained to a particular pose because we would like to retrieve any and all poses representing the isometrically deformed versions of the target.

To collect the sketches, we gathered a group of 15 undergraduate and graduate students who had recently taken the Digital Geometry Processing course at the Middle East Technical University. The students, who had already been exposed to the articulation concept during the course, were nevertheless shown a representative set of 3D shapes from our databases to offer them a glimpse of what we expected.

The students were then provided with a 10.1-inch 1,280 × 800 Windows tablet PC along with a Wacom Stylus digital pen to create their sketch drawings. Our canvas software recorded the stroke information in real time into an XML file. The same software also produced the bitmap image of the final drawing. No representative 3D shapes were shown to the students in an effort to minimize any biases in the drawings.

We collected a total of 132 sketches from the group. Only two of the students were left-handed. The hand-drawn sketches are considered more representative than mouse-painted sketches. We provide some of the hand-drawn sketches throughout the figures and all of them in our public sketch database, which we cast as a side contribution of this research.

Quantitative Results

Tables 1 through 4 report quantitative results summarizing our algorithm's retrieval performance. We also visually evaluate our retrieval quality in

Table 1. Quantitative evaluation of our shape retrieval method compared with the baseline method.*

	Correctness of the top-ranked item (%)									
	Watertight database									
Method	A	B	C	D	E	F	G	H	I	J
Ours	70	91	65	66	87	86	75	64	69	70
Base	46	75	45	38	60	37	51	44	46	42
	McGill database									
	K	L	M	N	O	P	R	S	T	U
Ours	71	70	83	77	70	75	98	95	68	70
Base	48	43	68	46	44	50	80	79	41	45
	TOSCA database									
	Four legged			Human			Centaur			
Ours	89			92			96			
Base	66			66			72			
	Global database									
	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
Ours	80	72	81	84	65	79	71	70	64	61
Base	59	44	54	60	42	68	46	49	40	39

**Sketch queries from A to J are human, glasses, airplane, ant, chair, table, fish, bird, armadillo, and four legged. Queries K to U are ant, crab, hand, human, octopus, pliers, snake, spectacles, spider, and teddy. Queries G1 to G10 are centaur, human, chair, glasses, ant, table, four legged, hand, pliers, and teddy. We provide, in percentages, the correctness of the top-ranked item when the database is queried 10 times with the isometrically deformed sketches.*

Table 2. Quantitative comparison of our shape retrieval method with a state-of-the-art method.[8]

Method	Nearest neighbor (NN)	First tier	Second tier
Ours	86.1	78.4	89.1
State-of-the-art query-by-example method	99.6	86.6	95.3

Table 3. Quantitative comparison of our shape retrieval method with the Mahoney-Fromherz nonarticulated method.[3]

Database	Our method			Mahoney-Fromherz method		
	NN	First tier	Second tier	NN	First tier	Second tier
Watertight	80.2	71.7	82.8	51.4	49.4	55.1
McGill	86.1	78.4	89.1	56.7	52.3	59.9
TOSCA	96.0	89.5	97.2	63.6	60.3	67.9
Global	75.9	70.1	79.8	49.1	45.2	52.3

Table 4. Quantitative evaluation of our shape retrieval method for different human query types on the McGill database.

Query Type	NN	One tier	Two tier	Time (sec)*
With face	85.1	73.5	85.2	(3.9, 9.1)
With head shape	85.2	73.7	85.5	(3.4, 9)
Skeleton-like	85.4	74	85.9	(2.5, 3.7)
Contour	30.7	23.1	31.8	(8.2, 13.1)

**The last column depicts the average time required for our participants to draw their human and centaur sketches.*

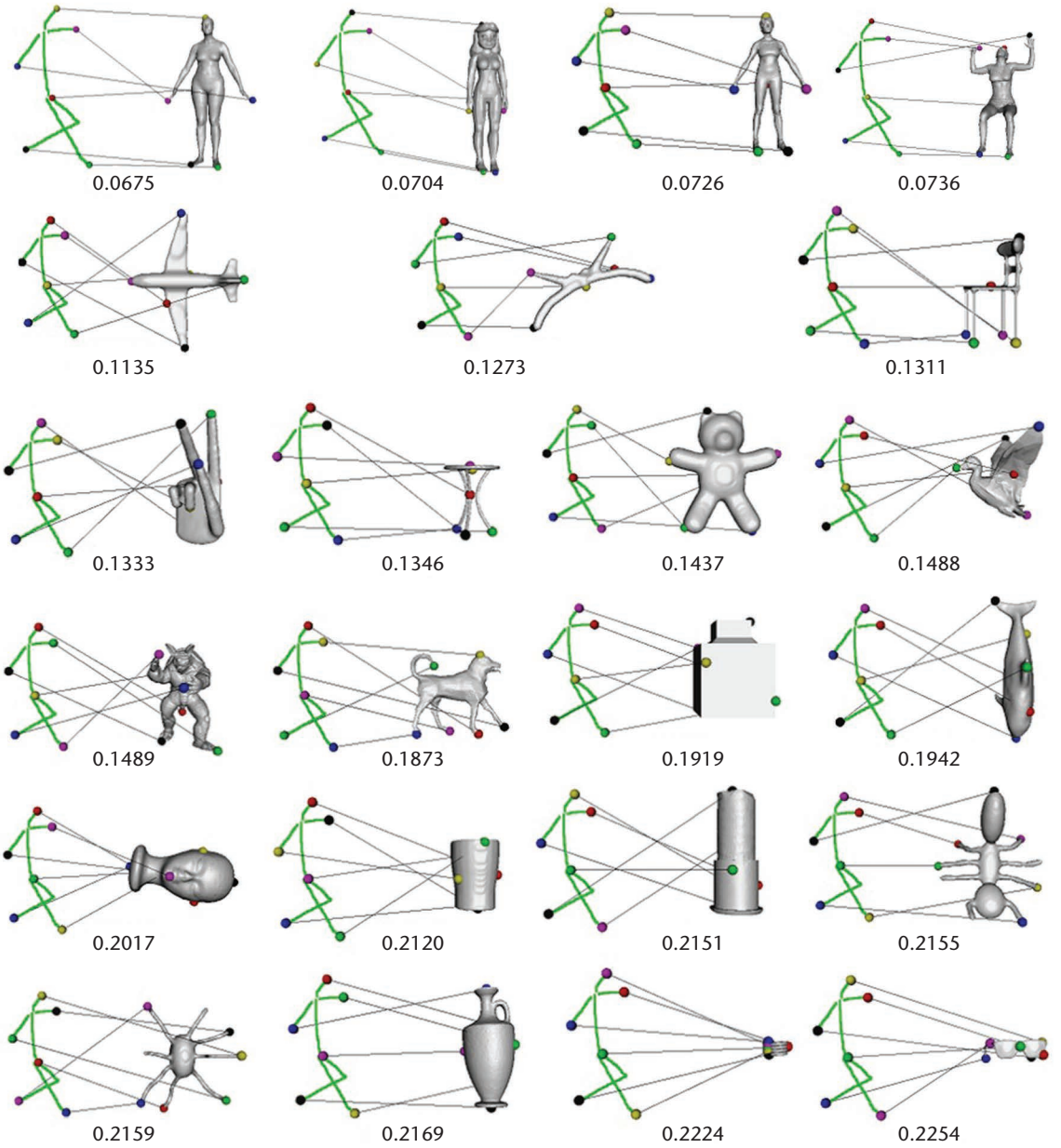

Figure 5. Top four retrieval results for the query human sketch on the Watertight database of 400 models (first row). The remaining rows show interaction of the same sketch with the other 19 classes of the database, in a sorted order with respect to the matching scores. These scores are given below each pair. A small matching score indicates high similarity

Figures 5 through 9, where we display the best match between the query and the database model (with lines and spheres) as well as the matching scores (text) that are used to sort the retrieval result set. Note that the best mappings may deviate from the ground truth due to the symmetric flip problem inherent to coarse isometric matching (for example, see Figure 10a) and inconsistent sampling (see the chair in Figure 6). This is because close isometric distortions (matching scores) among top mappings may be confused easily, as Figure 10 illustrates. Although this confusion might be remedied by adopting ideas from the 3D shape correspondence domain,[20] it is not a problem for a shape retrieval application, as our result sets verify. Our advantage of being a nonview-based method is made apparent with, for instance, the chair examples in Figure 6, where the extra lines on the back side of the query chair may easily confuse a view-based method, yet our results are robust.

The execution time for our algorithm on a 2.53-GHz PC with 8-Gbyte RAM was 101 seconds, without preprocessing on the Watertight database. If samples, along with the geodesic distances in between, are known a priori on the database models, the whole search takes only 0.05 second. Similar timing was achieved on the McGill database. The TOSCA dataset, which is one-quarter of the size of the Watertight, was queried in 30 seconds without any preprocessing and in 0.01 second with precomputed samples and geodesic distances.

For the high-resolution models in the TOSCA database, our algorithm remained quite fast because it exploits geodesic distances between a small

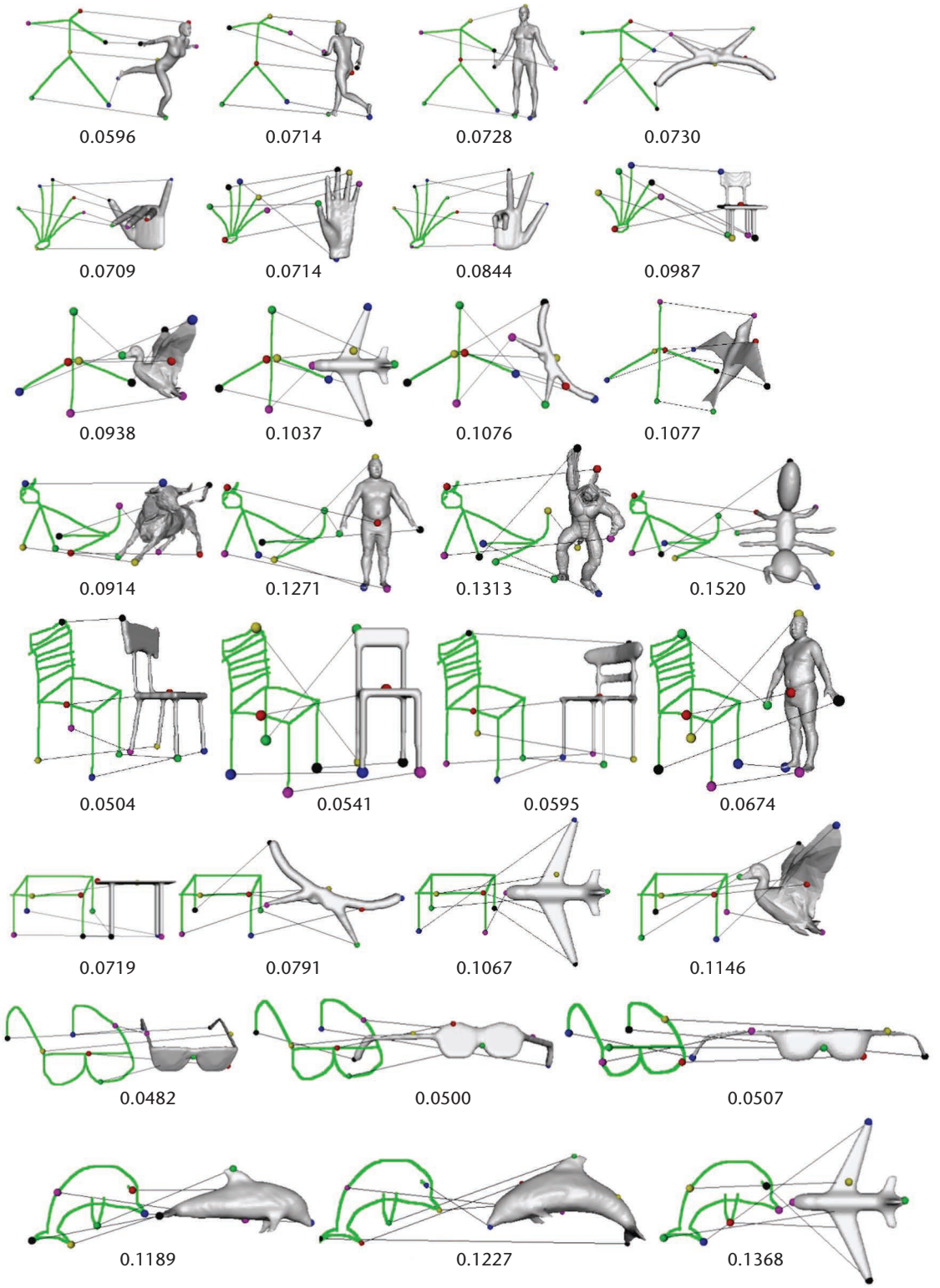

Figure 6. Top three or four retrieval results for various query sketches on the Watertight database. Matching scores used as sort keys are given below each pair.

set of samples. All the models used six extremity samples, except for the centaur model, which used eight samples. Finally, one query on the largest Global database of size 835 took approximately 206 seconds, which was reduces to merely 1.03 second when the geodesics were computed a priori.

We also note our abstraction performance by reporting that the initial human sketch of 1,639 unconnected points in Figure 9 is reduced to our abstract graph consisting of 18 vertices and 15 edges, six of which are virtual strokes. Similarly, 1,173 points for the hand sketch in Figure 6 is abstracted by 20 vertices and 24 strokes.

Comparison with a Baseline Method

Although articulated 3D shape retrieval based on

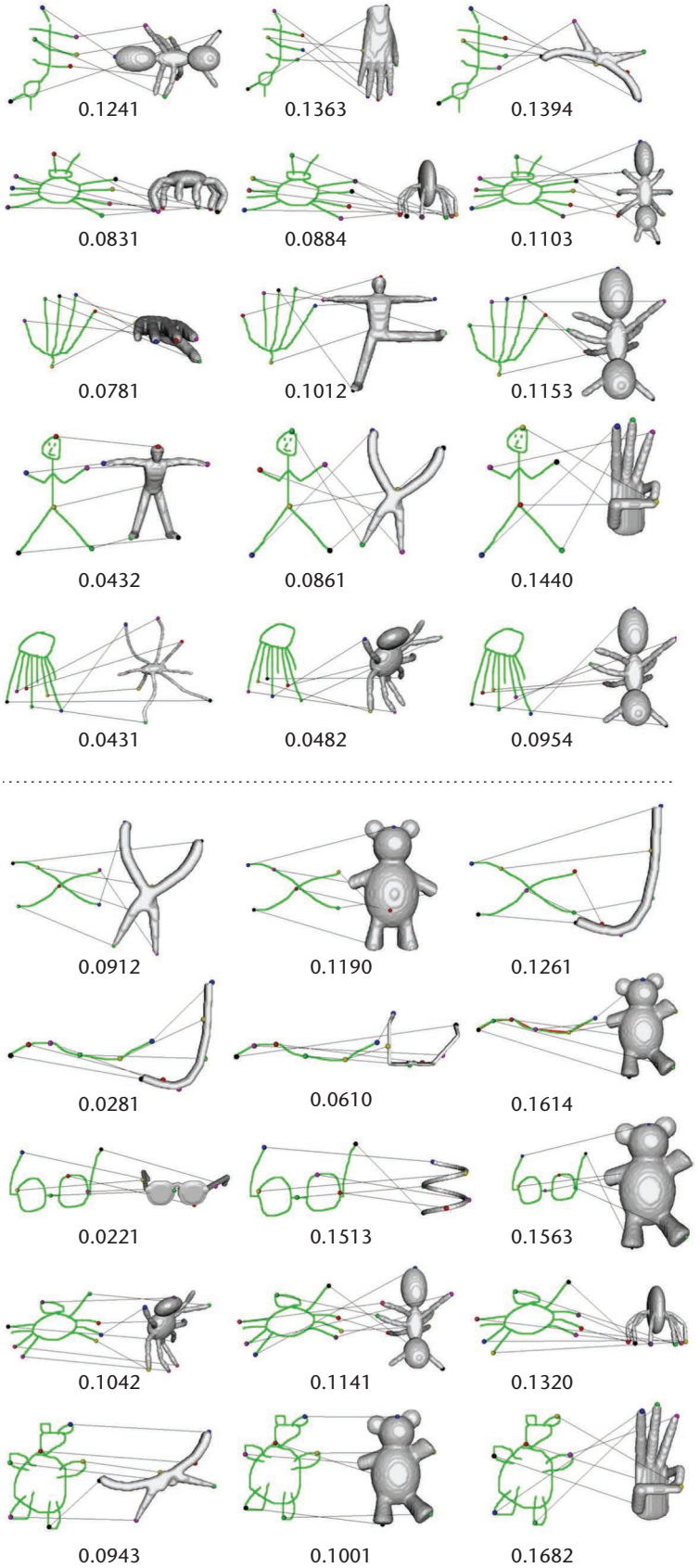

Figure 7. Top three retrieval results for various query sketches on the McGill articulated 3D shape benchmark. Matching scores used as sort keys are given below each pair.

3D model queries has received a great deal of attention in the past few years,[6–8] to the best of our knowledge, no hybrid approach exists that uses 2D sketch queries for this purpose. Note that existing sketch-based 3D retrieval applications do not address articulated queries.[3,11]

We therefore designed an intuitive baseline method to compare our results. In this baseline method, we disable the good continuation rule in the geodesic distance computations by discarding Equation 3. Specifically, we add the next arbitrary sample s'_k to the growing stroke R_a. The retrieval results in Table 1 show the benefit of applying the good continuation rule, as our original algorithm that makes use of this rule outperforms the baseline method.

Comparison with an Articulated Method

In addition to the baseline method that performs query-by-sketch, we also compared our method with the current state of the art in articulated 3D shape retrieval that is based on the query-by-example paradigm.[8] This method outperforms the existing work on the example-based articulated 3D shape retrieval because they distinguish the shapes better by maintaining details on articulation-invariant canonical representations of the models. Canonical-based retrieval methods are more promising than descriptor-based counterparts because the canonical form can easily be integrated into a simpler and well-studied rigid shape retrieval process. The disadvantage of this method, however, is the need to supply an example 3D query model for the search. Our simple and natural sketch queries remove this difficulty and render our method valuable as we achieve only slightly worse performance, with less complex input than the state-of-the-art method (see Table 2).

We used the same database (McGill) and the same performance metrics as the state-of-the-art method[8] to make the comparison fair. The metrics are the nearest neighbor (NN), which is the percentage of the first-closest matches that belong to the query class, and the first-tier metric is the ratio of the relevant matches to the size of the query class C when the number of retrieved models (top K matches) is $|C|$. We relaxed $K = 2|C|$ to obtain the second-tier metric. Our high NN value indicates the potential of our algorithm in a classification application.

Other than increasing the reliability of these performance metrics, the large McGill database of 255 models is also useful for the computation of precision-recall curves (see Figure 11). Precision is the ratio of the relevant matches to the num-

ber of retrieved models, whereas recall is the ratio of relevant matches to the size of the query class. Ideally, this curve should be a horizontal line at unit precision. In addition to the plot associated with our method, we provide plots for the other canonical-based retrieval methods: the detail-preserving method[8] and its detail-distorting version (least-squares MDS).[7] Figure 11 also shows the results for a descriptor-based method, which directly compares the low-dimensional heat-kernel signatures of the query and database models, as done in earlier work.[6] The lower performance of the descriptor-based approach supports our claim that favors canonical-based methods over descriptor-based counterparts. Note that all plots except ours in Figure 11 and the second row in Table 2 were copied directly from earlier work.[8] Figure 11 also includes the precision-recall results of our method on the larger Global database of size 835.

The performance of the state-of-the-art method[8] is better because its input is more complicated, harder-to-obtain 3D models, whereas we use simple, easier-to-obtain sketch queries. Thanks to this simplicity, our execution time is much faster. And despite this simplicity, we do not perform much worse than the state-of-the-art method.[8]

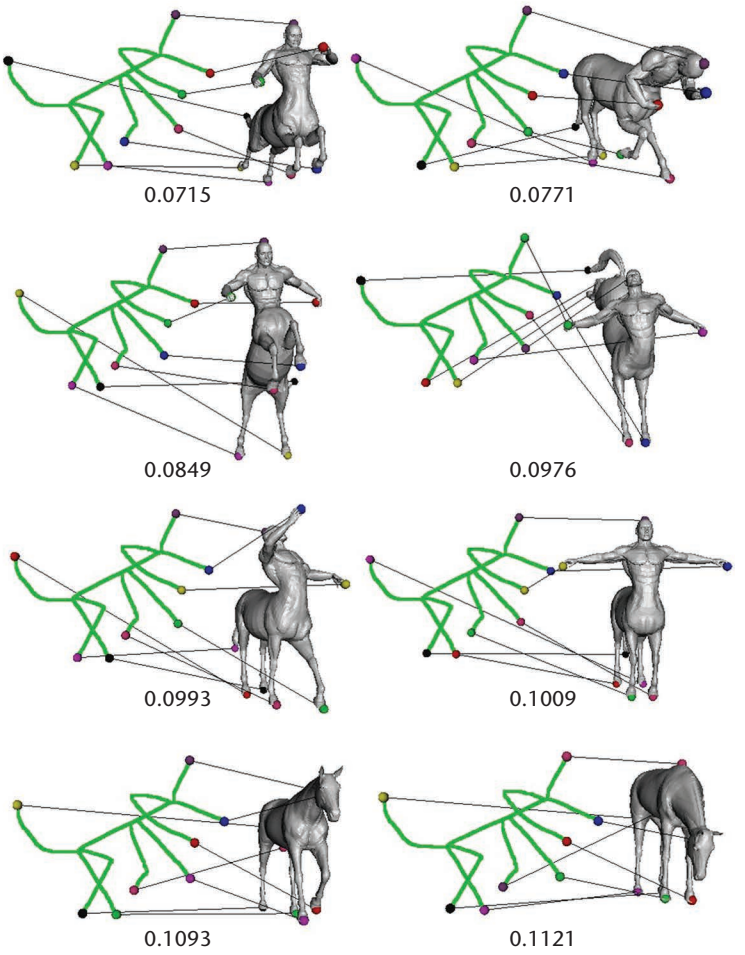

Figure 8. Top eight retrieval results for the centaur query on the TOSCA dataset (from left to right, top to bottom). Matching scores used as sort keys are given below each pair.

Comparison with a Nonarticulated Method

One recent robust sketch-based 3D shape retrieval method was designed to support merely nonarticulated/rigid transformations,[3] meaning that the query is expected to represent a rigidly transformed, (globally rotated and translated) version of the target model. Our method supports isometric transformations, a superset of the rigid transformations, and thus can be applied to the same database used in the Mahoney-Fromherz study[3] (PSB). In fact, Figure 12 shows that our method retrieves more accurate models up to articulations. Our method treats sketches as pure bitmap objects (sets of 2D coordinates without any additional time and stroke information), so it directly uses the offline sketches copied from the author's website. In other words, we ran our algorithm with the query sketches available from the earlier study[3] and still obtained a better performance on the common categories such as humans, glasses, and fishes (see Figure 12). Note also that we had

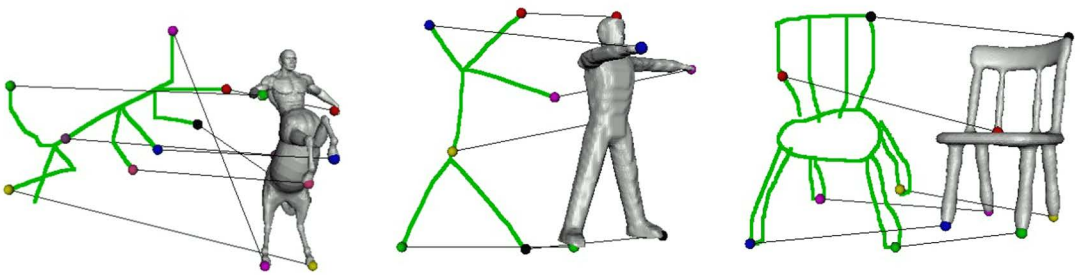

Figure 9. Top retrievals for three different sketch queries from our Global database. The initial human sketch of 1,639 unconnected points is reduced to our abstract graph consisting of 18 vertices and 15 edges, six of which are virtual strokes.

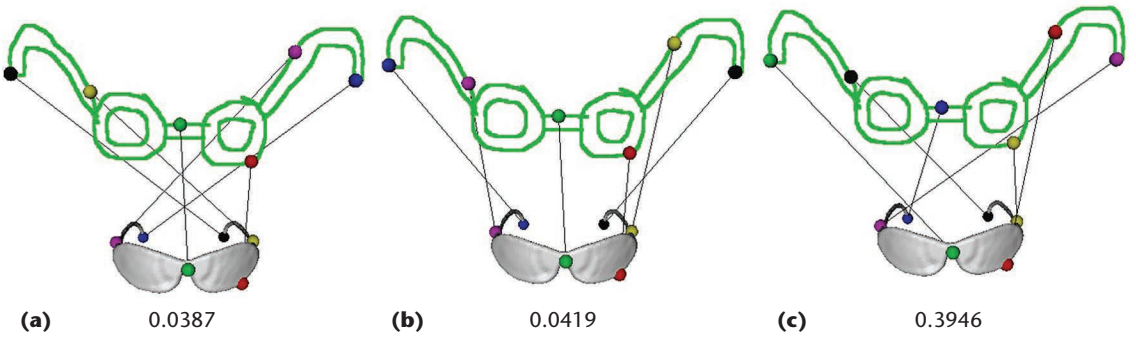

Figure 10. Matching scores of (a) symmetrically flipped, (b) ground-truth, and (c) highly distorted maps. Note the similarity of the first two scores, which are almost one-tenth of the high score. Either of the low scores is good enough to represent the compatibility between the 2D query and the 3D model.

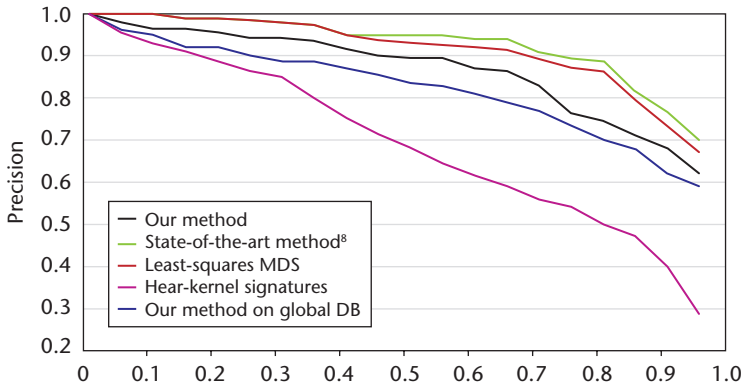

Figure 11. Precision-recall curves for each retrieval method. Precision is the ratio of the relevant matches to the number of retrieved models, whereas recall is the ratio of relevant matches to the size of the query class.

to manually repair the disconnected PSB meshes to operate, whereas the Mahoney-Fromherz view-based approach[3] performs the retrieval without such a preprocessing. Also, the Mahoney-Fromherz method supports partial matching, making online querying for an interactive system possible, whereas our geodesic-based method requires the complete sketch to be present. We finally note that our main focus is on the articulated shape retrieval.

To show the need for designing an articulated retrieval algorithm, we extended our comparisons with the rigid Mahoney-Fromherz method on articulated databases such as the Watertight, McGill, TOSCA, and Global databases. Because our work is the first method addressing articulated shape matching using sketch queries, we are restricted to comparing it to a nonarticulated state-of-the-art sketch-based method.[3] This comparison is still valuable, however, in that it emphasizes the benefits of developing an articulated method.

The results in Table 3 reveal that our articulated method is more capable of finding the correct query class than the nonarticulated method.[3] This is not surprising because the chances of an articulated query having the exact same shape as one of the examples in the articulated object database is practically zero. Subsequently, the Mahoney-Fromherz approach[3] could not find any instances of a human class for a crouched pose query, whereas our method could. Furthermore, even if a match occurred (for example, the database contained the exact crouched pose), the match would be virtually useless because the point is to retrieve the class of objects that have the same articulated structure, not an instance with the exact rigid shape as the query input.

Finally, if we consider the case where the user wants to retrieve the class of humans from a rigid database, such as the PSB, a rigid method would tediously have to query every simple pose and orientation that the database accommodates, whereas an articulated method such as ours would require only one pose from the human class.

Query Drawing Style

Our algorithm is robust to different drawing styles as long as the resulting sketch does not deviate significantly from the object's skeleton (see Figure 13). It is also robust to unnecessary details in the sketch that act as outliers or noise, such as when the face information on a human is not valuable and should be discarded as noise when matching humans to a database of various objects (see Figure 13a). Similarly, the shape of the head might also be irrelevant for the purposes of the match we are looking for (see Figure 13b). We consequently argue that although a skeleton-like query sketch is always safe with our system, the retrieval performance of a contour-like query sketch depends on the object to be retrieved. For example, this works for spectacles (Figure 10) or a table (Figure 6), but it fails for a human (Figure 13d).

In addition to the results in Figure 13, this argument is further supported by the close values of the performance metrics in Table 4, which are

obtained by querying the McGill articulated 3D shape database using human sketches similar to those in Figure 13a (with face), Figure 13b (with head shape), and Figure 13c (skeleton-like). The last row in Table 4 gives the low values for the problematic contour sketch. Note that there is neither skill nor training required to draw a skeleton-like query sketch (Figure 8). All the queries of different styles were provided by our 15 novice users. The average timing values of this user study given in the last column of Table 4 show that the skeleton-like queries that we expect as input to our system are not only accurate but also fast to produce.

Limitations

Virtual strokes may sometimes inadvertently connect endpoints that should remain disconnected, such as the figure's kneecaps in Figure 14. This shortcoming is inevitable with our current automatic stroke connection algorithm because the closeness threshold that makes true connections, such as from leg to hip in Figure 14, may also incorrectly connect other endpoints. However, we believe that it is reasonable to expect the user to draw scenes susceptible to such ambiguities with particular emphasis to address this issue while querying the database. Our retrieval system is pose-independent, and there is always an articulated/isometric query alternative that will yield the desired result set from the database. We also auto-select as conservatively as possible, which effectively minimizes the risk of generating false connections, even in complex sketches.

A heuristic solution to this problem would be to use different a closeness threshold k for different regions, but this would complicate the algorithm. Another solution could be to get users involved using an interactive query-specification paradigm that lets them check our processing results and use stroke-over or overtracing gestures to emphasize good continuations in cases where they have been missed.

Another limitation also concerns our automatic stroke connection procedure, which can miss crucial connections on contour sketches. Figure 15 demonstrates such a case, where the conservative value connects the yellow sphere to the black one via a long counterclockwise path (see the top row in Figures 15a through 15d). Breaking this path (bottom row) solves this problem easily, as in this case k is automatically increased to connect these spheres. Another simple solution is to rapidly increase k—that is, ($k = k + 2.5s$). Using both solutions, we obtain a consistent sampling between the sketch and the 3D hand model, which in turn leads to a healthier matching. Note that this limitation arises for contour sketches only, but not for skeleton-like stick figure sketches. Contour sketches might also cause problems if

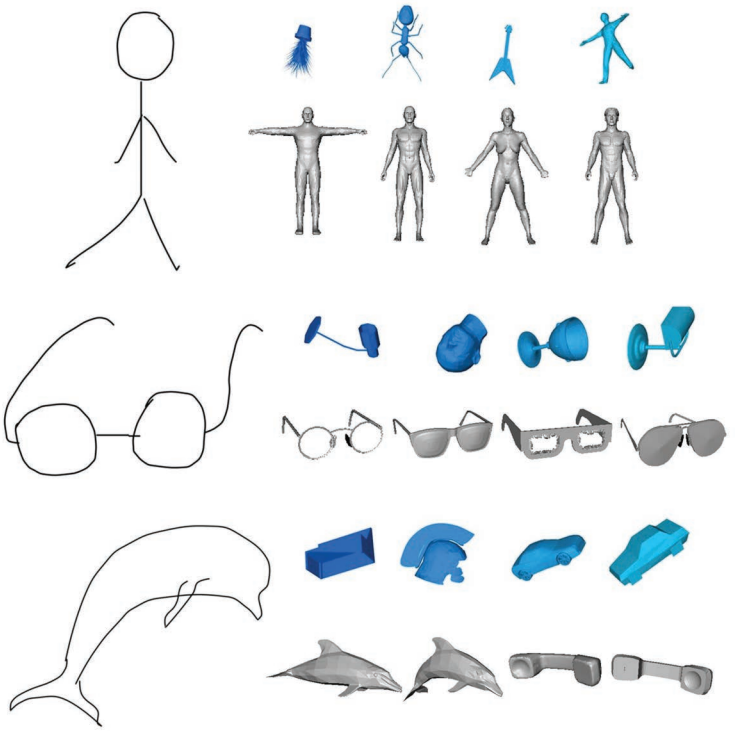

Figure 12. Three sketch queries and the top four retrieval results using the Mahoney-Fromherz method[3] (blue) and our method (gray). We copied the sketches and blue images from the author's public repository. The Mahoney-Fromherz method did not retrieve glasses in the top 20, and its first fish was the 15th object retrieved (not shown).

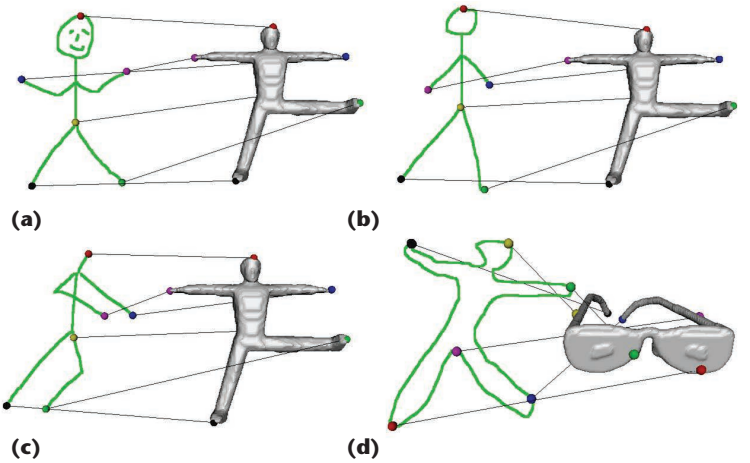

Figure 13. Top retrievals from the McGill articulated 3D shape database for query sketches drawn in various styles. (a, b) Thanks to our sampling process that represents a region by one of its most prominent features, such as its extremity, some details may be ruled out. (a, c) Different drawing styles can even lead to correct retrievals. (d) The last example retrieval is wrong because the path from the foot to the head traverses the arm, which in turn significantly increases the corresponding pairwise distance.

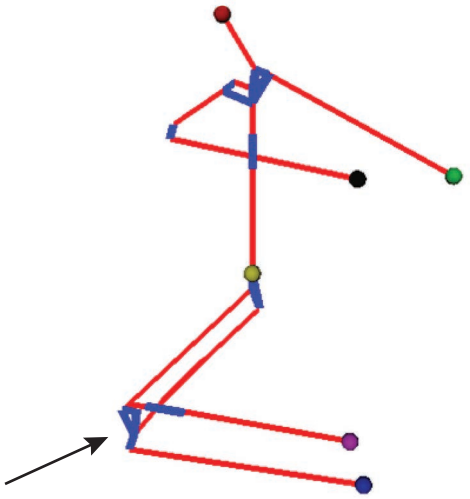

Figure 14. Inadvertently connected endpoints. One limitation of our algorithm is shortcut pairwise distances because of extra virtual strokes. For example, see the blue line between the stick figure's kneecaps.

they introduce significantly longer paths that could be avoided by traversing the inner skeleton (see Figure 13d).

Because we do not incorporate any local features in our algorithm, it fails to distinguish between two objects if the measured geodesic distances are all similar across the objects. This does not mean that topologically similar objects will be confused, however. For example, although a cat and a giraffe have the same topological structure, a longer neck will allow the user to guide the retrieval toward giraffes, and a shorter neck will favor cats. Our powerful representation even allows discrimination across breeds of dogs. For example, it can easily distinguish between dachshunds and greyhounds based on distance information, despite the similarities among dog breeds.

Although it has been well studied under the query-by-example setting, the shape retrieval problem is yet in its infancy in query-by-sketch mode. Our approach is, to our knowledge, the first to leverage and exploit the expressive power of free-hand drawings for sketch-based retrieval of articulated 3D objects. The offline feature of our sketch-based retrieval algorithm may later allow users to synthesize scenes from single images with edges that are extracted with robust detection algorithms. Once the extracted edges are segmented into offline sketches, our algorithm facilitates retrieval of the corresponding 3D shapes from the database to the scene. For more stability, we may also consider using the stroke information in query sketches for synthesis purposes. Another interesting area of future work is to extend the static shape retrieval we present here to time-varying motion retrieval based on sketches. Finally, in future work we expect to address the limitations we mentioned in the last section. For example, we could allow the user to be guided with real-time feedback during query specification in order to improve the input quality. This also applies to bringing partial matching support for an interactive system by updating our scale normalization procedure.

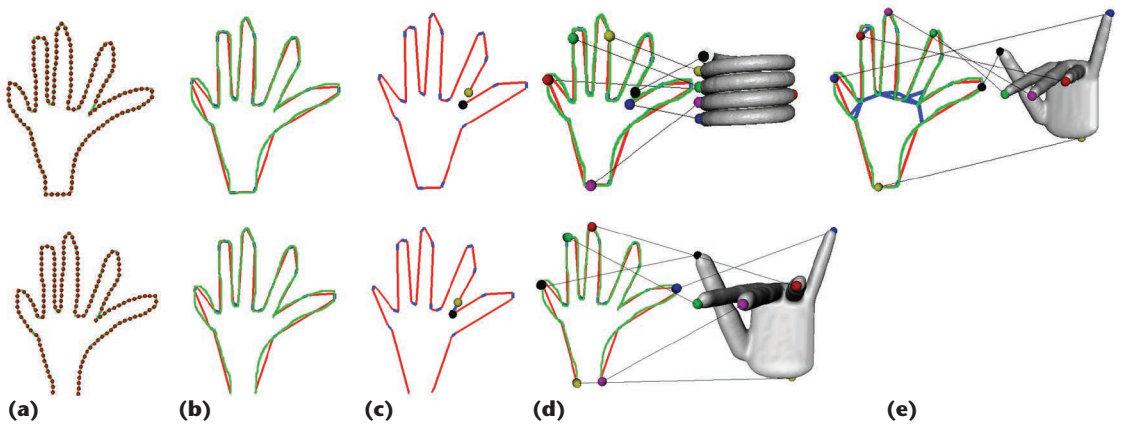

Figure 15. Automatic stroke connection. (a) Two contour sketches (green) are overlaid with an evenly spaced sample set S' and (b) strokes E' union E'. (c) We first shows the stroke separately along with two sphere endpoints for visual clarity and (d, e) then show the top-ranked retrievals. Note that lack of the virtual stroke between two spheres in the third example (top row, c) causes the hand sketch to unroll like a spring, yielding incorrect retrieval (top row, d). (e) A solution to this problem is to allow longer virtual strokes.

Acknowledgments

This work was supported by TUBITAK (The Scientific and Technological Research Council of Turkey) under the EEEAG-215E255 and EEEAG-113E325 projects.

References

1. D. Pickup et al., "Canonical Forms for Non-rigid 3D Shape Retrieval," *Proc. Eurographics Workshop on 3D Object Retrieval*, 2015, pp. 99–106.
2. J. Mahoney and M. Fromherz, "Three Main Concerns in Sketch Recognition and an Approach to Addressing Them," *Proc. AAAI Spring Symp. Sketch Understanding*, 2002, pp. 105–112.
3. M. Eitz et al., "Sketch-Based Shape Retrieval," *ACM Trans. Graphics*, vol. 31, no. 4, 2012, article 31.
4. P. Shilane et al., "The Princeton Shape Benchmark," *Proc. IEEE Int'l Conf. on Shape Modeling and Applications* (SMI), 2004., pp. 167–178
5. T. Funkhouser et al., "A Search Engine for 3D Models," *ACM Trans. Graphics*, vol. 22, no. 1, 2003, pp. 83–105.
6. A.M. Bronstein et al., "Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval," *ACM Trans. Graphics*, vol. 30, no. 1, 2011, article 1.
7. A. Elad and R. Kimmel, "On Bending Invariant Signatures for Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, 2003, pp. 1285–1295.
8. Z. Lian, A. Godil, and J. Xiao, "Feature-Preserved 3D Canonical Form," *Int'l J. Computer Vision*, vol. 102, no. 1-3, 2013, pp. 221–238.
9. J. Zhang et al., "Retrieving Articulated 3-D Models Using Medial Surfaces and Their Graph Spectra," *Proc. Int'l Workshop On Energy Minimization Methods in CVPR*, vol. 1, 2005, pp. 285–300.
10. T.M. Sezgin, T. Stahovich, and R. Davis, "Sketch Based Interfaces: Early Processing for Sketch Understanding," *Siggraph 2006 Courses*, 2006, article 22.
11. T. Shao et al., "Discriminative Sketch-Based 3D Model Retrieval via Robust Shape Matching," *Computer Graphics Forum*, vol. 30, no. 7, 2011, pp. 2011–2020.
12. A.M. Bronstein, M.M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*, Springer, 2008.
13. Y. Sahillioğlu and Y. Yemez, "Minimum-Distortion Isometric Shape Correspondence Using EM Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, 2012, pp. 2203–2215.
14. Y. Sahillioğlu and Y. Yemez, "A Dynamic Programming Based Approximation Method for Multiple Shape Correspondence," *Computer Graphics Forum*, vol. 33, no. 7, 2014, pp. 121–130.
15. K. Xu et al., "Sketch2Scene: Sketch-Based Co-retrieval and co-placement of 3D Models," *ACM Trans. Graphics*, vol. 32, no. 4, 2013, article 123.
16. A. Wolin, B. Eoff, and T. Hammond, "Shortstraw: A Simple and Effective Corner Finder for Polylines," *Proc. Eurographics 5th Ann. Workshop on Sketch-Based Interfaces and Modeling*, 2008, pp. 33–40.
17. D. Douglas and T. Peucker, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature," *Cartographica: Int'l J. Geographic Information and Geovisualization*, 1979, pp. 112–122.
18. D. Giorgi, S. Biasotti, and L. Paraboschi, "SHREC: Shape Retrieval Contest: Watertight Models Track," *SHREC Competition*, vol. 8, 2007.
19. K. Siddiqi et al., "Retrieving Articulated 3D Models Using Medial Surfaces," *Machine Vision and Applications*, vol. 19, no. 4, 2008, pp. 261–274.
20. Y. Sahillioğlu and Y. Yemez, "Coarse-to-Fine Isometric Shape Correspondence by Tracking Symmetric Flips," *Computer Graphics Forum*, vol. 32, no. 1, 2013, pp. 177–189.

Yusuf Sahillioğlu *is an associate professor in the Department of Computer Engineering at Middle East Technical University. His research interests include digital geometry processing and computer graphics. Sahillioğlu has a PhD in computer science from Koç University. Contact him at ys@ceng.metu.edu.tr.*

Metin Sezgin *is an associate professor in the Department of Computer Engineering at Koç University, where he leads the Intelligent User Interfaces Lab, and a visiting fellow at Yale University. His research interests focus on enabling people to interact with computers in a more natural fashion by combining techniques from machine learning, computer vision, computer graphics, and human-computer interaction. Sezgin has a PhD in computer science from the Massachusetts Institute of Technology. Contact him at mtsezgin@ku.edu.tr.*