Qualitative Organization of Collections of Shapes via Quartet Analysis

Shi-Sheng Huang[1]*, Ariel Shamir[3], Chao-Hui Shen[1], Hao Zhang[4], Alla Sheffer[5], Shi-Min Hu[1], Daniel Cohen-Or[2]
[1]TNList, Tsinghua University, Beijing, [2]Tel-Aviv University, [3]The Interdisciplinary Center
[4]Simon Fraser University, [5]University of British Columbia

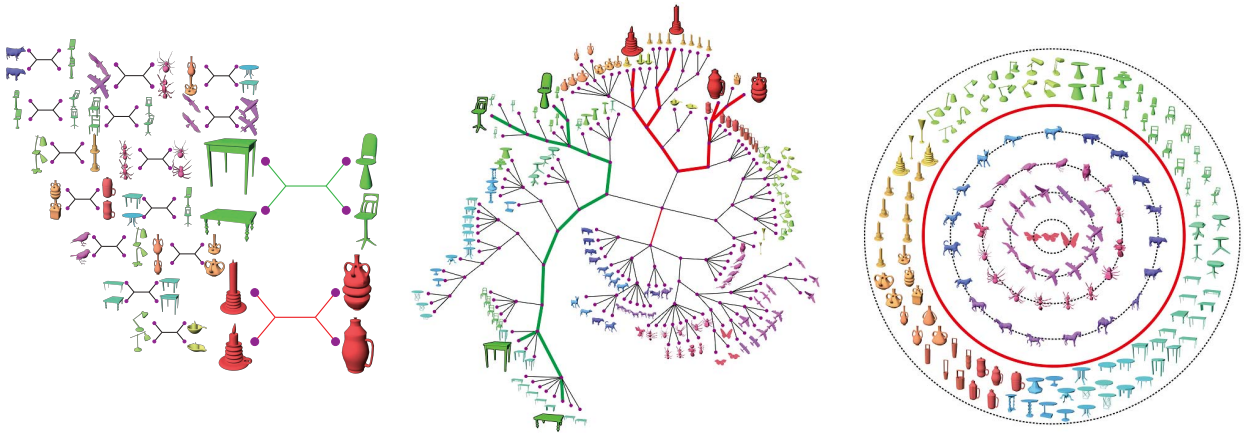


Figure 1: *A heterogeneous collection of shapes is organized in a hierarchical categorization tree (middle) via quartet-based qualitative analysis. Every quartet (left) defines a topological relation between two pairs of shapes that must be maintained by the categorization tree: note the embedding of the relations of the red and green quartet examples in the tree. Based on this organization, the collection can be dynamically reordered around a given shape by their Degree of Separation (right). Far shapes (beyond the red circle) are the ones whose paths to the given shape pass through the top level of the categorization tree.*

Abstract

We present a method for organizing a heterogeneous collection of 3D shapes for overview and exploration. Instead of relying on quantitative distances, which may become unreliable between dissimilar shapes, we introduce a *qualitative* analysis which utilizes multiple distance measures but only in cases where the measures can be reliably compared. Our analysis is based on the notion of *quartets*, each defined by two pairs of shapes, where the shapes in each pair are close to each other, but far apart from the shapes in the other pair. Combining the information from many quartets computed across a shape collection using several distance measures, we create a hierarchical structure we call *categorization tree* of the shape collection. This tree satisfies the topological (qualitative) constraints imposed by the quartets creating an effective organization of the shapes. We present categorization trees computed on various collections of shapes and compare them to ground truth data from human categorization. We further introduce the concept of *degree of separation* chart for every shape in the collection and show the effectiveness of using it for interactive shapes exploration.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: shape collections, clustering, organization

*This work was performed while the first author was a visiting scholar at Tel-Aviv University

Links: DL PDF WEB

1 Introduction

An ever growing number of digital 3D models are produced and stored in on-line shape collections. The rapid growth demands novel ways to organize large collections of shapes so as to facilitate search, summarization, and exploration of these collections so as to understand the overall categorization and hierarchical grouping of large and diverse collections. Any such organization must be built on a comparison mechanism between the individual shapes. The success of a comparative analysis is highly dependent on choosing the right "distance" between shapes.

A variety of distance measures have been developed to quantify the similarity or dissimilarity between 3D shapes [Shilane et al. 2004; Tangelder and Veltkamp 2008]. When the shapes in a given collection are sufficiently similar, it is often possible to find a proper quantitative distance that reflects well the shape semantics and allows analyzing them in a common framework by clustering or embedding into a metric space. However, shape distances are not always metrics (e.g., failing the triangle inequality), making them ineffective when the compared shapes are highly dissimilar. For instance, a numerical distance between a chair and a bicycle is most likely less informative than a distance between two chairs or between two

bicycles. In a large shape collection possessing rich variations, it can be extremely difficult, if not impossible, to properly quantify all pairwise shape similarities using a single distance measure that will allow a meaningful global analysis.

In this paper, we introduce a *qualitative* analysis method for organizing a heterogeneous collection of shapes. Rather than attempting to embed the whole collection into some common metric space based on what is likely an unreliable *quantitative* distance measure, our method combines more reliable *topological* information derived from multiple distance measures. We denote any set of four shapes in our collection as a "quadruplet". We perform a series of tests on quadruplets to find a subset of *quartets*. The quartets produce a set of topological partial orders which together allow a global organization of the whole collection of shapes (Figure 1). This technique has proven to be effective in bioinformatics [Strimmer et al. 1997; Willson 1998; Ranwez and Gascuel 2001].

Specifically, each quartet only needs to identify, with reasonable confidence, four objects where the objects are divided into two pairs. Each pair contains two similar objects that are clearly separated from the other pair (see Figure 1, left). We aggregate this topological information from several distance measures into a global organization of the shape collection. We use an *un-rooted* tree, which we call a *categorization tree*, or *C-tree*, to organize the collection. All the shapes in the collection reside at the leaves of the tree, with the number of edge "hops" between them reflecting their *degree of separation* within the given collection. The algorithm converting the set of quartets to the C-tree ensures that for most quartets, the topological pair-to-pair relation among the four associated objects is maintained in the tree.

Other possibilities for organizing the shapes using state-of-the-art clustering methods perform poorly when the distances are unreliable. In particular, organizing large collections requires some hierarchical structure, but using hierarchical clustering or other clustering means must also compare dissimilar shapes, where the reliability and effectiveness of shape distances diminish.

In contrast, our organization of a collection is performed by a global analysis, optimizing many soft constraints that combine several distance measures using the QMC algorithm [Snir and Rao 2010]. Each constraint provides a topological relation among four shapes (a quartet) that is expected to be reflected in the C-tree. By considering only reliable topological relations that are qualitative, the global analysis bypasses the direct deployment of inaccurate quantitative distances. In other words, while common hierarchical methods aim to satisfy pairs of quantitative distances, our optimization aims to satisfy a large set of topological and qualitative constraints.

For any given shape $\mathcal{S}$ in the collection, we can use the C-tree to partition all other shapes into layers, organized by their degree of separation around $\mathcal{S}$. This can be done effectively and efficiently without committing to an accurate distance measure. We call this type of organization – the Degree of Separation (DoS) chart.

Our contributions are: (i) A qualitative shape analysis technique, inspired by studies of evolutionary trees, that does not directly depend on numerical metrics, and (ii) A hierarchical organization of heterogeneous shape collections. We demonstrate the performance of our analysis approach in organizing large and heterogeneous collections like the one displayed in Figure 1. We also present an interactive interface based on DoS charts, allowing the exploration of the whole collection of shapes by imposing a local structure on the space of shapes, as illustrated in Figure 11.

2 Related Work

Shape retrieval. Perhaps the most prominent need for organizing large shape collections comes from shape retrieval [Shilane et al. 2004; Tangelder and Veltkamp 2008; Bronstein et al. 2011]. Shapes are clustered based on some distance measure to facilitate fast querying. The queries are typically answered by a nearest neighbor classifier, hence a higher-level organization of the shapes is not needed. The distances measures are often defined based on global shape descriptors [Kazhdan et al. 2003a; Osada et al. 2002; Chen et al. 2003; Gal et al. 2007; Bronstein et al. 2011]. In our work, we adopt some of these distances only for local analysis within the quartets. The global organization of the shapes is entirely based on topological information extracted from the reliable quartets.

Co-analysis. Several recent works concentrate on unsupervised co-analysis of sets of shapes [Golovinskiy and Funkhouser 2009; Xu et al. 2010; Sidi et al. 2011; Wang et al. 2012]. These methods aim at shape analysis at the part level, e.g., computing a co-segmentation, whereas we perform unsupervised analysis at the shape level. Moreover, existing works on co-analysis all assume that the set of shapes belong to the same family, while our work focuses on analyzing a *heterogeneous* shape collection.

Shape exploration. A few works propose effective means to explore large shape collections. Ovsjanikov et al. [2011] present a correspondence-free method that uses a template-based deformation model. The user can directly manipulate the model to bring up shapes from the collection. Kim et al [2012] present an interactive exploration technique based on fuzzy correspondence among partial shape regions. An important distinction is that these exploration techniques follow a shape retrieval paradigm, where the user manipulates a model to find similar shapes, and cannot gain an overview of the collection. These methods again assume that the shapes in the set being explored belong to the same family. Our method organizes a heterogeneous collection of shapes hierarchically, allowing both categorical and coarse-to-fine exploration.

Clustering. One of the best practiced means of organizing a set of elements is clustering analysis [Everitt et al. 2011]. Clustering is typically carried out via quantitative analysis relying critically on the distance measure chosen. In this classic setting, via affinity propagation, improved clustering results have been obtained by replacing metric affinities by non-metric ones, e.g., for image categorization [Dueck and Frey 2007]. In contrast, our work is not affinity-based clustering, but relies on a qualitative analysis based on topological constraints to derive a categorization tree. Furthermore, we are not aware of works that hierarchically cluster a set of 3D shapes belonging to a heterogenous collection.

In the field of document analysis and information retrieval, there have been works on hierarchically organizing and exploring text documents [Weigend et al. 1999] or web content [Dumais and Chen 2000]. Another field with relevant work is evolutionary biology [Semple and Steel 2003].

Ordination. In exploratory data analysis, *ordination* orders objects that are characterized by values on multiple variables so that similar objects are near each other and dissimilar objects are farther apart. Examples of such techniques include multi-dimensional scaling (MDS) [Mardia et al. 1980] and self-organizing maps [Kohonen 1982]. However, similar to clustering, these methods rely heavily on quantitative distance measures and usually do not produce hierarchical organizations.

Quartet processing. The construction of phylogenetic (evolutionary) trees is a classical subject in biology. Broadly speaking,

phylogenetic reconstruction methods are either sequence-based or topology-based. Quartet-based analysis belongs to the latter. A quartet tree is the smallest possible informative un-rooted tree. This, along with the fact that small trees representing partial orders are easier and more reliable to obtain than larger ones, makes quartet-based methods attractive for large-scale phylogenetic reconstruction. Most quartet-based methods [Willson 1998; Ranwez and Gascuel 2001] first infer quartet trees from aligned sequences and then use a combinatorial algorithm to solve the maximum quartet consistency problem. Snir and Rao [2010] describe an algorithm for constructing a tree from a set of input quartet trees even amid a significant fraction of errors. The algorithm is based on a divide and conquer algorithm where the division step uses a semidefinite programming (SDP) formulation of MaxCut. This algorithm, which has been referred to as the Quartet MaxCut (QMC) algorithm, is the one we adopt in this paper.

3 Categorization Tree

Given a collection of shapes, we would like to organize them into one hierarchical structure, a categorization tree or C-tree. This tree is defined by satisfying a large number of topological constraints specified by quartets, using an optimization method. A quartet is a sub-tree that consists of four leaves. Such a sub-tree expresses a minimal meaningful topological relation among a set of nodes that can be used as a constraint for construction the C-tree. A quartet relation is said to be satisfied by a given C-tree, if an equivalent quartet structure is expressed in the tree, ignoring internal nodes. The key idea in using quartets is that the topological structure expressing the relations among the four nodes in the quartet should be manifested among the same four nodes in the tree (see red and green quartets embedding in Figure 1).

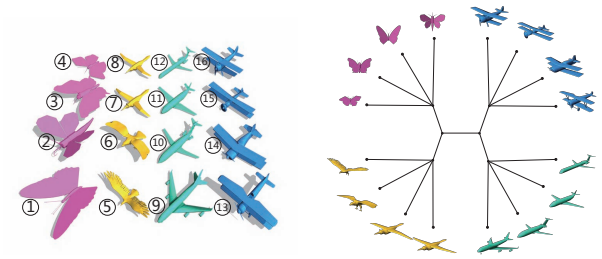
Note that we use the topological structure defined by four nodes since the topological structure of a sub-tree of two or three nodes is always manifested in any tree: two nodes have a single path between them and not structure, and the paths between three nodes will always link together in a single internal node.

Although previous works have shown that a few representative quartets suffice to uniquely define a tree [Erdos et al. 1997], the shape of the tree can be influenced to satisfy as many quartets as possible. In our premise, the construction of the C-tree uses a large number of quartets created by considering multiple distance measures. This enables us to combine reliable qualitative information from several distance measures. Thus, given a set of shapes, our approach is to construct a large number of reliable quartets, and define a C-tree by an optimization technique whose objective function aims to satisfy as many of the given quartets as possible.

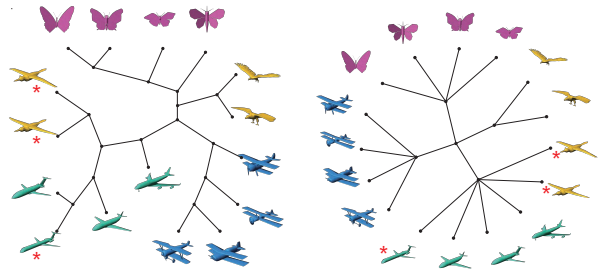
An alternative approach would be to directly cluster the collection of shapes using some distance measure, which is less effective for at least two reasons. First, any single distance measure among the shapes will not be sufficiently accurate to induce a reliable metric, especially for a heterogenous shape collection, and second, clustering methods are typically greedy in nature. Our approach is more global, where the optimization accounts for reliable topological constraints rather than directly satisfying the less reliable quantitative distances. The following example illustrates the difference between the two alternatives.

3.1 A Motivating Example

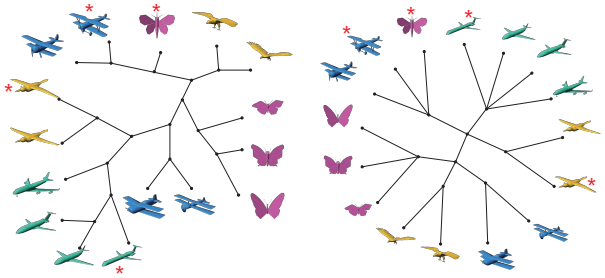
Figure 2(a) shows sixteen objects which are colored according to four clusters, with butterflies, birds and two types of airplanes. These objects are selected from the PSB dataset [Shilane et al. 2004]. We use two popular 3D shape distance measures to build



(a) A set of shapes. (b) Our categorization tree.



(c) LFD-based categorization trees obtained using NeighborJoining (left) and ApClustering (right), respectively.



(d) SDF-based categorization trees obtained using NeighborJoining (left) and ApClustering (right), respectively.

Figure 2: *A motivating example: using quantitative shape distance measures and clustering can lead to mixed categorization of shapes (c-d), while using our qualitative approach the categorization captures the correct clusters (b). Objects that lead to miscategorizations are marked with red stars.*

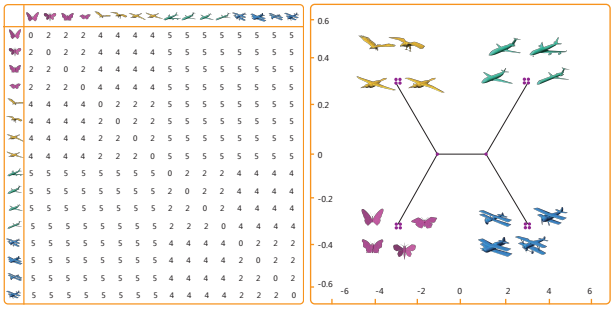


Figure 3: *The DoS-distance table and the C-tree overlaid on the projection of the points by MDS using this table. This clearly shows how the DoS-distances capture the grouping of these shapes.*

distance matrices among the sixteen objects. The first measure uses the Light Field Descriptor (LFD) [Chen et al. 2003] with inner-distance [Ling and Jacobs 2007], and the second uses the Shape Diameter Function (SDF) [Shapira et al. 2008]. While LFD and SDF distances are effective in comparing similar objects, they are inaccurate in reflecting distances between highly dissimilar shapes.

Next, we use two popular clustering methods with the affinity matrices of LFD and SDF (see the supplementary material for these matrices), to yield a total of four trees seen in Figures 2(c-d). We use simple neighbor joining and APclustering [Frey and Dueck 2007], representing a straightforward method and a more sophisticated one, respectively. As can be observed, the small LFD-distances between the pairs of shapes $\{7, 12\}$ and $\{8, 12\}$, and the small SDF-distances between pairs of shape $\{7, 12\}$ and $\{2, 13\}$ lead to mis-categorizations via clustering.

In contrast, our quartet-based method combining the LFD measure with the SDF measure, successfully recovers the correct C-tree representing the four clusters; see Figure 2(b). The key advantage of our algorithm is its ability to combine information from several distance measures together and only when the measures can be reliably compared, as described in Section 5. In Figure 3, we use the generated C-tree to represent the degree of separation (DoS) among the shapes, and build a DoS-distance table. An MDS layout of the shapes, based on the DoS-distance table, reveals the inherent clusters in the input set.

This minimal example merely provides an intuition for the power of quartet-based analysis. In the following, we describe the method in detail, demonstrate its performance on larger shape collections, and evaluate the quality of the generated C-trees.

4 From Quartets to C-Tree

The problem of reconstructing a tree from a set of quartets has received a lot of attention in various domains. In particular, in bioinformatics, it is used as a means to reconstruct Phylogenetic trees. This problem can be formulated as a MaxCut on a graph whose edges are constructed based on the quartets definitions. Since the solution is NP-Complete, we adopt the Quartets MaxCut algorithm (QMC) of [Snir and Rao 2010] to construct a C-tree of 3D shapes from a given set of quartets. For now, we assume that the set of quartets is given, and would like the C-tree to satisfy as many of them as possible. Later, in Section 5, we elaborate on how we collect and define the quartets based on multiple shape distances.

The QMC algorithm adopts a two stage approximation approach: first embedding and then partitioning. In the first stage, it embeds points representing the shapes into a metric space based on the topological information of the quartets. In the second stage, it uses the distance metric between points in this space to recursively partition the set of points in a top-down manner, creating a hierarchical tree structure.

4.1 Embedding

Techniques for embedding objects into a metric space such as MDS optimize the difference between a given quantitative measure to the final metric. In our case, the actual value of the distance measure between shapes is of less importance than the relations between the shapes represented by the quartets. Hence, the embedding optimization does not measure actual distances between the points representing shapes, but invests more effort to keep "close" shapes nearby and "far" shapes apart.

Given m shapes in the collection, we represent them as points in the Euclidean space $\mathbb{R}^n$, restricted to the unit sphere $\mathbb{S}^n$. Each shape a is represented by a point v_a that lies on the sphere. We optimize the positions of these points by imposing the topological relations defined by the quartets instead of using the actual distance measure between the shapes that the points represent. Given a quartet $(ab|cd)$, we want to keep shapes a and b close, c and d close, and all other pair combinations far apart. Since v_a, v_b, v_c, v_d all lie on the sphere, we use the angle between them $\widehat{v_i, v_j}$ as an indicator for their distance. We define two terms representing the average of the "close" pairs and "far" pairs:

$$\begin{aligned} C(a,b,c,d) &= (\widehat{v_a, v_b} + \widehat{v_c, v_d})/2, \\ F(a,b,c,d) &= (\widehat{v_a, v_c} + \widehat{v_a, v_d} + \widehat{v_b, v_c} + \widehat{v_b, v_d})/4. \end{aligned} \tag{1}$$

Given a list of k quartets $(a_j b_j | c_j d_j)$, the embedding of the representative points is defined as the solution of a Semi-definite program (SDP), that minimizes the following expression:

$$\sum_{1 \le j \le k} C(a_j, b_j, c_j, d_j) - \sum_{1 \le j \le k} F(a_j, b_j, c_j, d_j), \tag{2}$$

subject to $\langle v_i, v_i \rangle = 1$ $(1 \le i \le m)$.

The result of minimizing this expression is that point v_{a_j} will be close to point v_{b_j}, point v_{c_j} will be close to point v_{d_j}, and the two pairs (v_{a_j}, v_{b_j}), and (v_{c_j}, v_{d_j}) are separated from each other as much as possible, respecting the given quartet $(a_j b_j | c_j d_j)$. Satisfying these constraints accurately is usually done by setting $n = m$. However, as shown in [Snir and Rao 2010] embedding the shapes into a three-dimensional sphere in $\mathbb{R}^3$ yields satisfactory results, and we set $n = 3$ in all our experiments.

4.2 Partitioning

Once all representative points are embedded onto $\mathbb{S}^n$, we can simply use the Euclidean distances between them as a metric $d(\cdot, \cdot)$. The partitioning stage builds a hierarchical tree by top-down recursive partitioning of the set of embedded points. Yet again, information from the quartets is used to guide the partitioning. For any quartet $(a_j b_j | c_j d_j)$, the pairs (a_j,b_j) and (c_j,d_j) are expected to be in a separate group after the partitioning. To achieve this, we define a small graph on each quartet containing six edge; see Figures 4(b-c). We denote the edges between a_j, b_j and between c_j, d_j as "bad" edges, and all other four edges as "good" edges. As can be seen in Figures 4(b-c), a good partition is one that cuts through as many good edges as possible and as few bad edges as possible. Thus, the optimized partition is obtained by maximizing the following expression:

$$\sum_{e_{ij} \in G} d(i,j) - \alpha * \sum_{e_{ij} \in B} d(i,j), \tag{3}$$

where the set G (B) is the set of good (bad) edges from all quartets, $d(i, j)$ is the Euclidean distance between the embedded points v_i and v_j representing shapes i and j, and $\alpha > 0$ is a scalar weight. This optimization can be efficiently solved using the MaxCut algorithm. The partition operates recursively until no more partitions are obtained. The resulting hierarchy of sub-partitions is defined as the C-Tree. To guarantee that no empty partitions are obtained, we adopt the dynamic ratio parameter as suggested by [Snir and Rao 2010]. If $\alpha \ge |G|$, then we always obtain empty partitions when optimizing expression 3. Thus, we use a binary search to find the largest α in the interval $[2, |G|]$, which yields a non-empty partition.

5 Defining the Quartets

Although the QMC optimization is robust against a reasonable number of outlier quartets, defining reliable quartets is critical to

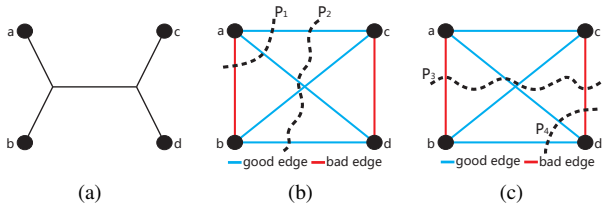


Figure 4: *A quartet $(ab|cd)$ in (a) induces six edges: two bad (red) and four good (cyan). Four cuts $P_1, \ldots, P_4$ are illustrated in (b) and (c) by dashed lines. The cut P_2 is better than all other cuts with respect to the quartet in (a), as no bad edges are included in it.*

the success of building a reliable C-tree. The key questions are: how to define a reliable quartet? and later, how to extract them from a large collection of shapes?

Quartets are used to separate two pairs of shapes. The more clear the topological structure of this separation is, based on a given distance measure, the more reliable the quartet is. This structure induces two pairs of shapes that are close to each other among themselves, but far apart between them. In total, there are six pairwise distances between four shapes. Therefore, we need to examine whether we can find only two *small* distances among the six, that are between two relatively *distant* pairs of shapes.

Using different thresholds to define how large (or small) these distances can be, will result in defining more (or less) quartets as reliable. The challenge is on one hand to be conservative enough to include only reliable quartets, but on the other hand, not to be too conservative so as to have enough observations to generate a reliable C-tree. By combining several distance measures, we can achieve both. We are conservative while accepting quartets based on each single distance, but still have enough observations as we combine quartets obtained using several distances. This way, we combine only the strengths of these distances and not their flaws, as we use distance information *only when* it is reliable.

The full graph among any four shapes (a, b, c, d) (a quadruplet) includes six edges, as shown in Figure 5(a), where each edge is associated with a distance based on any desired measure. Following the above observations, a necessary condition to form a reliable quartet is that its four vertices remain connected after removing three edges corresponding to the largest distances. Any quadruplet that does not pass this test is discarded; see Figures 5(e). Next, the three remaining edges are sorted according to their distances: let $d_1 \leq d_2 \leq d_3$ be the three smallest distances. If the edge of the largest (remaining) distance d_3 is not a *bridge*, that is, if its removal does not separate the four nodes into two pairs, then the quadruplet is, again, discarded; see Figure 5(d). The remaining set of quadruplets have a potential of being a reliable quartet. We examine the ratio between the values of the edges and use a threshold R: if $d_3/d_1 > R$ and $d_3/d_2 > R$, then we define the quartet as reliable.

In general, for m shapes, the total number of quadruplets is C_m^4, which is in the order of $O(m^4)$. To accelerate the search for reliable quartets, we first construct pairs of shapes that are close enough, and build quartets from them. This increases the chance of finding reliable quartets using sampling. For each distance measure we use, we build m groups (bins) of the k nearest neighbors of each shape — one bin per shape. Using these bins we create candidate quadruplets for filtering by picking all combinations of two pairs from two separate bins. Using this approach we found that typically around 30% of the examined quartets are reliable. Thus, the total number of samples reduces to $C_k^2 \cdot C_m^2 = O(m^2)$, which is also the order

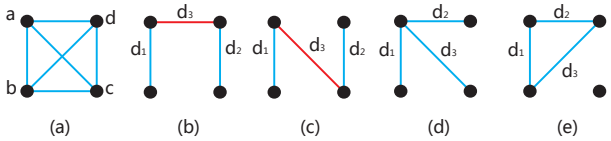


Figure 5: *Four shapes a, b, c, d and six associated edges (a). After removing three edges, there are several possible configurations. For instance, (d) and (e) are discarded as the edge d_3 is not a bridge that separates the four into two pairs, while (b) and (c) can be further examined to define a reliable quartet.*

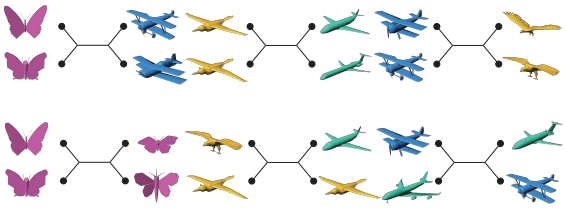


Figure 6: *Six quartet samples of the small set in Figure 2. Upper row shows reliable quartets and the bottom shows discarded ones.*

of the number of quartets needed to achieve a high accuracy (see discussion in Section 6). Larger k values allow more accuracy at the cost of speed. We set k to be between 20 and 50 for a collection of around $1,000$ shapes, arriving at around $10^6 \sim 10^7$ quartets.

In our experiments, we employ four shape distance measures based on different shape descriptors: LFD with Inner-Distance, SDF, Spherical Harmonic Descriptor (SHD) [Kazhdan et al. 2003b], and the HKS-BoF Descriptor [Bronstein et al. 2010]. The threshold R is defined with respect to the scale of the different measures: $R_1 = 1.2$ for LFD, and $R_2 = 1.5$ for SHD, SDF and HKS-BoF (see Section 6 for an analysis of the effect of varying these parameters). We create a set of reliable quartets using each of the distance measures separately, and combine them to form a C-tree. Figure 6 illustrates some examples of quartets deemed reliable and some that are filtered out during generation of the C-tree in Figure 2.

6 Evaluation

Sample Shape Collections. To demonstrate the effectiveness of the method, we used a given categorization of objects from the Princeton Shape Benchmark [Shilane et al. 2004] to build a ground truth categorization tree. We also used four more collections of shapes and categorized them in a similar manner as PSB using manual organization by multiple users. We used an iterative process with 10 users where each one receives the previous user's categorization tree and can move objects from one category to another until the tree converges. It should be noted that although categorization is subjective, the objects in the collections we used have a rather simple and well-accepted human categorization as can be seen in Figure 13 and in the supplementary material.

Tree Distance. To evaluate the performance of the different methods which produce categorization trees, we need to measure the distance between a given tree T and the ground truth tree GT. Conventional tree distances mostly measure the similarity between tree structures. In our case, the main reason for creating the tree is to order shapes, and the tree structure is of less importance.

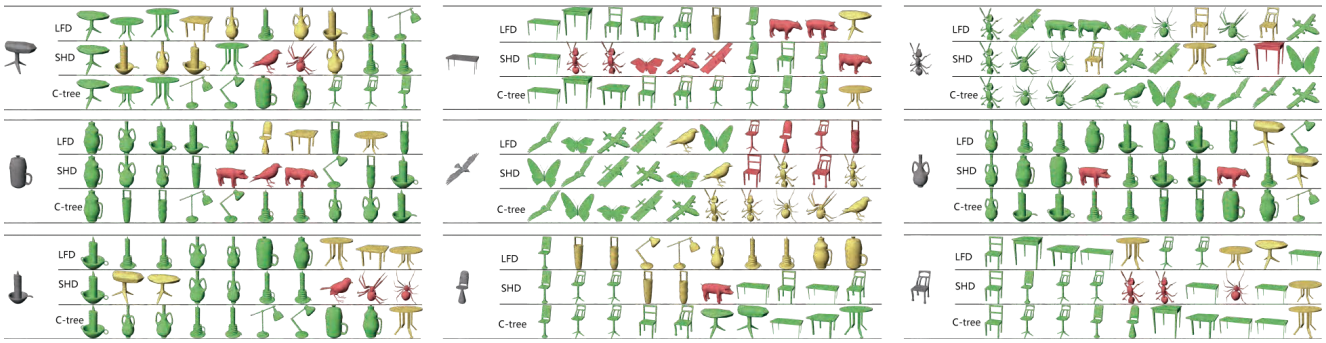
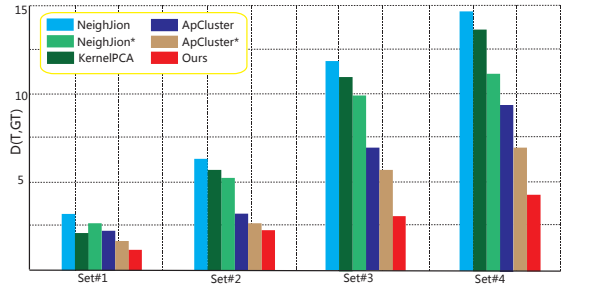
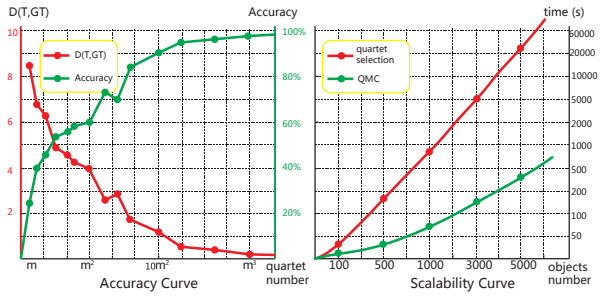


Figure 7: *9 models are selected from set#1 (Figure 13(a)), and the 10 nearest models are listed in each row using LFD similarity, SHD similarity, and our C-tree DoS-distance. Every model is colored by its DoS on the ground truth tree: green for near shapes, yellow for moderately close shapes, red for far shapes.*



(a)



(b)

Figure 8: *Quality and performance evaluation. (a) comparing the tree distance $D(T, GT)$ for our method against a few other clustering methods; our results are closer to the ground truth on all four shape collections. (b)-left: the quality (red) and accuracy (green) plots, measured against the ground truths, as the number of quartets increases. (b)-right: plot of running time for quarter selection and QMC as the number of shapes increases.*

Given a pair of shapes p_i, q_i, we measure the difference between their degree of separation in the two trees:

$$\Delta(p_i, q_i) = |DoS_T(p_i, q_i) - DoS_{GT}(p_i, q_i)|_1. \quad (4)$$

The distance between the two trees T and GT is defined as the average difference between all pairs of shapes in the two trees. Since calculating this distance is quadratic in time, we use random sampling of a large number k of pairs to estimate this distance:

$$D(T, GT) = \frac{1}{k} \sum_{i=1}^{k} \Delta(p_i, q_i). \quad (5)$$

Using this measure we can evaluate the performance of different tree generation methods for arrangements of shapes: we compare the distances from the different trees generated to the ground truth tree over four collections (including two from the PSB) using the $D(T, GT)$ measure. We compare the performance of our method to two well known algorithms for clustering: (i) Neighbor Joining, and (ii) ApClustering. In each algorithm, we used four quantitative distance measures: LFD, SDF, SHD and HKS-BoS. The average tree distance of these four is denoted as NeighJoin and ApCluster in Figure 8(a).

We also combined the four measures by first normalizing their affinity matrices and then creating a combined matrix in which the distance between two shapes is the minimum of the four (normalized) measures and used the combined measure for clustering (both NeighJoin$*$ and ApCluster$*$). Lastly, we used Gaussian Kernel PCA on each of the affinity matrices of the four distance measures to reduce the dimensions by a factor of 5 (to 20%) and built a tree using Neighbor Joining in the lower-dimensional space. The average tree distance over the four methods in the reduced space is also given in Figure 8. A comparison of the tree distance using all different methods is given in Figure 12. As can be seen, the distance is much smaller for the trees built by our method compared to all alternatives.

K-Nearest Neighbor (KNN) Test. We randomly selected 9 objects from shape set #1 (Figure 13 shows an overview of all the shape collections) to test the performance of an KNN algorithm based on various distances. In Figure 7, for each of the 9 objects, we show in three rows the 10 most similar shapes retrieved from the set by (i) LDF, (ii) SHD, and (iii) our qualitative method using DoS distance. Next, we measure the distance between the extracted shapes and the query shape on the ground truth tree and color the retrieved shape in three colors: red for far shapes, yellow for moderately close shapes, and green for near shapes. As can be seen, the (bottom) rows using our method contain mostly green (near) shapes, while the other rows contain a mixture of red and yellow.

Accuracy Test. The quality of a categorization tree obviously depends on the number of quartets used. To measure this, we built a ground truth tree GT out of m shapes. Based on this tree, we create a set of quartets by choosing pairs of shapes from separate sub-trees; see [Snir and Rao 2010]. Using a random subset of these quartets, we build a categorization tree T and measure its accuracy compared to GT. We define an accuracy measure as $accuracy = 1.0 - D(T, GT)/DoS_{GT}$, where DoS_{GT} is the average DoS distance between pairs of objects in the GT tree. Since $D(T, GT) < DoS_{GT}$, the accuracy will always be smaller than 1, and will reach 1 when $D(T, GT) = 0$. In Figure 8(b)-left, we plot

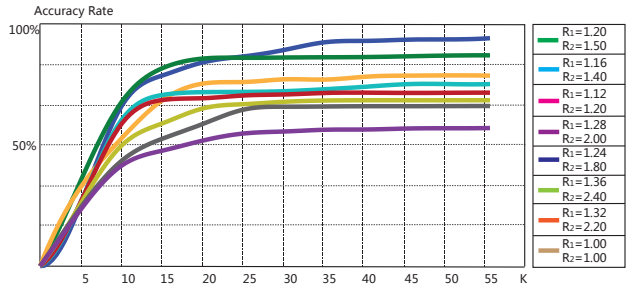


Figure 9: *Varying quartet definition parameters: the accuracy as a function of k, the number of nearest neighbors for different values of the ratio R.*

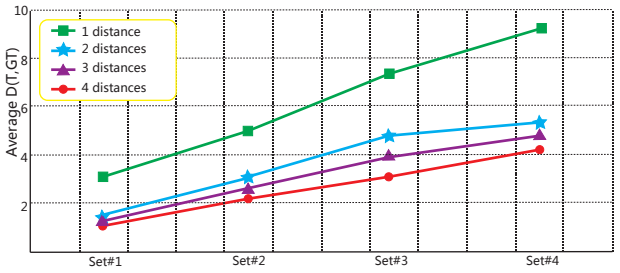


Figure 10: *The average distance of the categorization trees from the ground truth merging one to four distances to create the quartets. The more distances are combined the lower the distance, but the gain drops as more distances are added.*

the accuracy of the categorization tree against the number of quartets used. As can be seen, the accuracy increases with an increase in the number of quartets, but we find that $O(m^2)$ quartets can lead to a C-tree with 90% accuracy.

Varying Parameters for Quartet Definition. In Section 5, we defined two main parameters for choosing quartets: k is the number of nearest neighbors taken for each shape while searching for quartet candidates, and R is the ratio between the value of inner to inter pair distances in a quartet used to define a reliable quartet. The larger the k the more candidates are created at the expense of higher running times. When R is too small the quartets are no longer reliable, resulting in low accuracy. However, when R is too large, the reliable quartets are too conservative resulting in a small number of quartets, which can again reduce accuracy. Figure 9 shows a plot of the accuracy of our algorithm (as defined above) while varying k for different settings of R_1 used for LFD and R_2 used for SHD, SDF and HKS-BoF. Note that our setting of $R_1 = 1.2$ and $R_2 = 1.5$ seems to strike the best balance for values of k above 20.

Behavior of Merging Distances. One of the strengths of our method is the combination of several distance measures. We chose four well-known shape distance measures but others could be used as well. To illustrate the strength of combining multiple distances, we compare the quality of the resulting C-trees when only a subset of the measures are used.

Denote by Q_j, $j = 1, \ldots, 4$, the set of quartets created using only the j-th distance measure; the four distances used were listed in Section 5. Let I_k be the set of all combined quartet sets by merging k of the Q_j's, $k = 1, \ldots, 4$, sets Q_j. For example, $I_1 = \{Q_1, Q_2, Q_3, Q_4\}$ and $I_2 = \{Q_1 \cup Q_2, Q_1 \cup Q_3, Q_1 \cup Q_4, Q_2 \cup Q_3, Q_2 \cup Q_4, Q_3 \cup Q_4\}$, etc. For each I_k, we build the set of C-trees corresponding to the quartet sets in I_k (e.g., there are six C-trees built for I_2) and calculate the average tree distance from the created trees to the GT tree. The results of these average distances on all four collections shown in Figure 13 can be found in Figure 10. As can be seen, the average tree distance using two kinds of distances to build the quartet set is significantly smaller than using a single distance. Similarly, using a combination of three or four distances also increase the accuracy but less significantly. Hence, we chose to merge all four distances in all our experiments.

Complexity and Scalability. The QMC algorithm worst case complexity is $O(m^2)$, where m is the number of shapes in an input collection, as it uses Goemans-Williamson Algorithm to compute MaxCut approximation. This is a random iterative algorithm and there is no guarantee on convergence, hence the number of iterations could be large. The dominant part of our method, timing wise, is the quartet selection step which is quadratic in the number of shapes, as explained in Section 5. Figure 8(b)-right shows a plot of the running times of these two components of our method. Both show a roughly quadratic growth with m, since they depend linearly on the number of quartets, and we choose $O(m^2)$ quartets to build the C-tree. The largest collection we handled contains around $5,000$ shapes.

Limitations. Quartet-based analysis has two main limitations stemming from the heuristic nature of quartet selection. First, the sampling approach of selection implies that the quality of the results depends on the number of reliable quartets. This means that it may not be suitable for small collections. Second, when quartets are defined based on multiple distances, conflicting reliable quartets may be introduced. For example, a quadruplet (a, b, c, d) may define the quartet $(ab|cd)$ by distance A, and $(ad|bc)$ by distance B. Once the relative number of such conflicts is large, the optimization method can lead to unreliable results. This implies that the success of the method still depends on the quality of the geometric descriptors and the similarity distances employed.

7 Interactive Exploration of Shape Collection

The organization of a collection of shapes into a C-tree allows fast and intuitive exploration. Rather than browsing and exploring the collection by a sequence of retrieval-type queries that display *linear* series of retrieved nearest neighbors, the C-tree allows displaying *two-dimensional* neighbor maps. These 2D maps facilitate the exploration of the collection due to the 2D nature of human vision and conventional displays. Specifically, the 2D display space is better utilized. As well, the wider field of view provided by the 2D maps provides more instant access to a larger number of models, leading to more effective summarized view of the shape collection. We have developed a user interface that demonstrates such an exploration; see Figure 11 for an illustration and the accompanying video for interactive demonstration.

To explore a collection of shapes, the constructed C-tree is first presented to the user in a display area. The user can then freely select any leaf node of the tree, which represents one shape from the collection and serves as the starting point of the exploration.

Once a shape is selected, the rest of the shapes are automatically repositioned to form a DoS chart around the selected one. The DoS chart is formed in such a way that shapes closer to the selected shape on the C-tree will be located at the inner circles of the DoS chart. This organization provides the user with intuitive understanding of how other shapes in the collection compare with the currently selected one. The user can also rotate the different circles to better examine the shapes.

Exploration can continue to other shapes by selecting any shape in

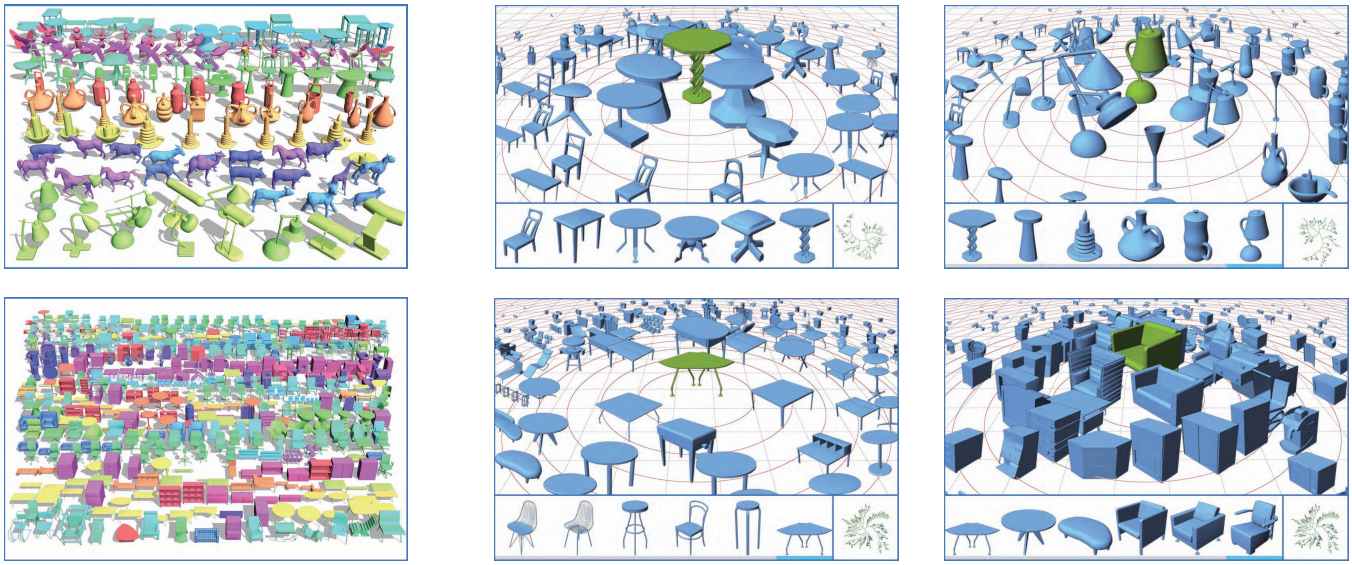


Figure 11: *Two examples (for two collections of shapes shown on the left) of using the shape exploration tool based on local "maps" created by DoS-charts. The main window of the tool shows an interactive depiction of the DoS chart around the current selected shape (green), the lower right panel shows the categorization tree and the lower left panel shows the history of selected shapes (see also accompanying video).*

the current DoS chart. A new DoS chart around the newly selected shape will then be displayed with a smooth transition between the previous one and the new one. The system maintains the exploration history of the collection in a separate window, allowing simple understanding of the exploration path and quick backtracking. Lastly, at any time the user can switch back to the C-tree display to choose a different shape as a starting point for a new exploration.

8 Conclusion and Future Work

As the size of 3D shape repositories grows, so does the need to organize them to enable intuitive and efficient browsing and exploration. While previous approaches to the problem focused on finding good distance measures between objects and used those in standard organizational frameworks, we demonstrate the effectiveness of a topological, qualitative approach to organizing 3D shape collections. As demonstrated by results throughout the paper, our approach utilizes a number of common distance measures, but fuses them in a most effectively manner to alleviate the drawbacks of conventional use of these distances. The key idea is to gather *reliable* qualitative information from multiple distances to enable a coherent organization of heterogeneous collections of objects.

We believe that such qualitative approach for sorting and organization is new to computer graphics, and we believe that it can be used in a larger scope. For example, in organizing photo collections, where finding general and effective distance measures for photos is particularly challenging. Another possible extension to our work would be to handle very large collections of objects. This can be done in conjunction with conventional clustering method. Since common distances measures perform well on similar instances of objects (e.g., shapes or photos), we can use a two-stage approach. Extremely large collections can first be clustered into groups of similar instances, and then we can apply a qualitative approach only to organize the clusters using representative shapes. Such shapes would be heterogonous, hence benefiting from our approach.

It would also be interesting to use the categorization-tree to create better abstractions of a family of shapes. Finally, the degree of separation of each shape in the collection can be a valuable feature to analyze the diversity or density of the collection, which can lead to applications such as collection-level analysis and control.

Acknowledgements

This work was supported by the Israel Science Foundation (Number 324/11), NSERC Grant (Number 611370), the National Basic Research Project of China (Project Number 2011CB302202), the Natural Science Foundation of China (Project Number 61120106007), the National High Technology Research and Development Program of China (Project Number 2012AA011801) and Tsinghua University Initiative Scientific Research Program.

References

BRONSTEIN, A. M., BRONSTEIN, M. M., KIMMEL, R., MAHMOUDI, M., AND SAPIRO, G. 2010. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *Int. J. Comput. Vision 89*, 2-3, 266–286.

BRONSTEIN, A. M., BRONSTEIN, M. M., GUIBAS, L. J., AND OVSJANIKOV, M. 2011. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30*, 1, 1:1–1:20.

CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3D model retrieval. *Comput. Graph. Forum 22*, 3, 223–232.

DUECK, D., AND FREY, B. J. 2007. Non-metric affinity propagation for unsupervised image categorization. In *IEEE 11th International Conference on Computer Vision*, IEEE, 1–8.

DUMAIS, S., AND CHEN, H. 2000. Hierarchical classification of web content. In *Proc. of ACM SIGIR conference on Research and development in information retrieval*, 256–263.

ERDOS, P. L., STEEL, M. A., SZEKELY, L. A., AND WARNOW, T. J. 1997. A few logs suffice to build (almost) all trees (ii). Tech. rep.

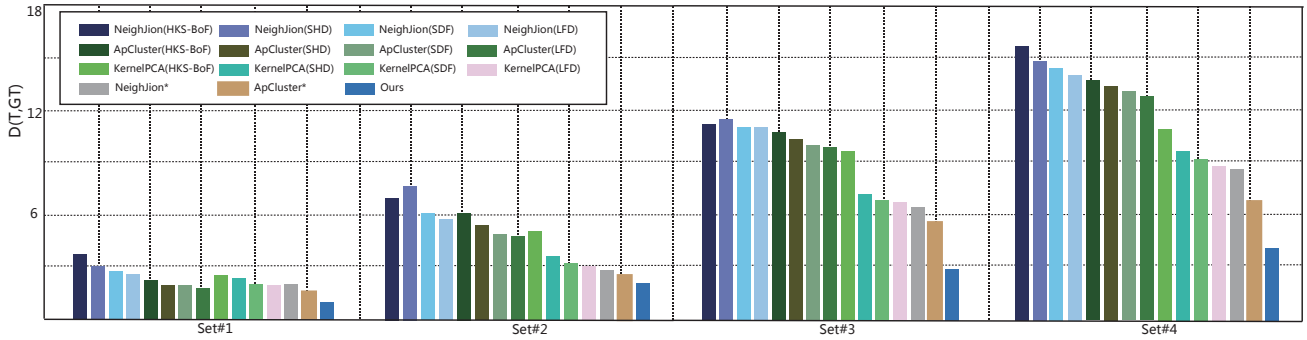


Figure 12: *Comparing the tree distance $D(T, GT)$ between the ground truth tree GT and the tree T constructed using various clustering methods; our method results in a tree closer to the ground truth on all four shape collections.*

EVERITT, B. S., LANDAU, S., LEESE, M., AND STAHL, D. 2011. *Cluster analysis, 5th edition*. Wiley Series in Probability and Statistics.

FREY, B. J., AND DUECK, D. 2007. Clustering by passing messages between data points. *Science 315*, 972–976.

GAL, R., SHAMIR, A., AND COHEN-OR, D. 2007. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics 13*, 2, 261–271.

GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. Consistent segmentation of 3D models. *Comput. Graph. 33*, 3, 262–269.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP'03*, 156–164.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '03, 156–164.

KIM, V. G., LI, W., MITRA, N. J., DIVERDI, S., AND FUNKHOUSER, T. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Trans. Graph. 31*, 4, 54:1–54:11.

KOHONEN, T. 1982. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*, 59–69.

LING, H., AND JACOBS, D. W. 2007. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell. 29*, 2, 286–299.

MARDIA, K. V., KENT, J. T., AND BIBBY, J. M. 1980. *Multivariate Analysis (Probability And Mathematical Statistics) Author: , Publisher*. Academic Press.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graph. 21*, 4, 807–832.

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. Graph. 30*, 4, 33:1–33:10.

RANWEZ, V., AND GASCUEL, O. 2001. Quartet-based phylogenetic inference: Improvements and limits. *Molecular Biology and Evolution 18*, 6, 1103–1116.

SEMPLE, C., AND STEEL, M. 2003. *Phylogenetics*. Oxford Univerity Press.

SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput. 24*, 4, 249–259.

SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The princeton shape benchmark. In *SMI'04*, 167–178.

SIDI, O., VAN KAICK, O., KLEIMAN, Y., ZHANG, H., AND COHEN-OR, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph. 30*, 6, 126:1–126:10.

SNIR, S., AND RAO, S. 2010. Quartets maxcut: A divide and conquer quartets algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinformatics 7*, 4, 704–718.

STRIMMER, K., GOLDMAN, N., AND VON HAESELER, A. 1997. Bayesian probabilites and quartet puzzling. *Molecular Biology and Evolution. 14*, 2, 210–211.

TANGELDER, J. W., AND VELTKAMP, R. C. 2008. A survey of content based 3D shape retrieval methods. *Multimedia Tools Appl. 39*, 3, 441–471.

WANG, Y., ASAFI, S., VAN KAICK, O., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Active co-analysis of a set of shapes. *ACM Trans. Graph. 31*, 6 (Nov.), 165:1–165:10.

WEIGEND, A. S., WIENER, E. D., AND PEDERSEN, J. O. 1999. Exploiting hierarchy in text categorization. *Information Retrieval 1*, 3 (Oct.), 193–216.

WILLSON, S. J. 1998. Building phylogenetic trees from quartets by using local inconsistency measures. *Molecular Biology and Evolution 16*, 5, 685–693.

XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z.-Q. 2010. Style-content separation by anisotropic part scales. *ACM Trans. Graph. 29*, 6 (Dec.), 184:1–184:10.

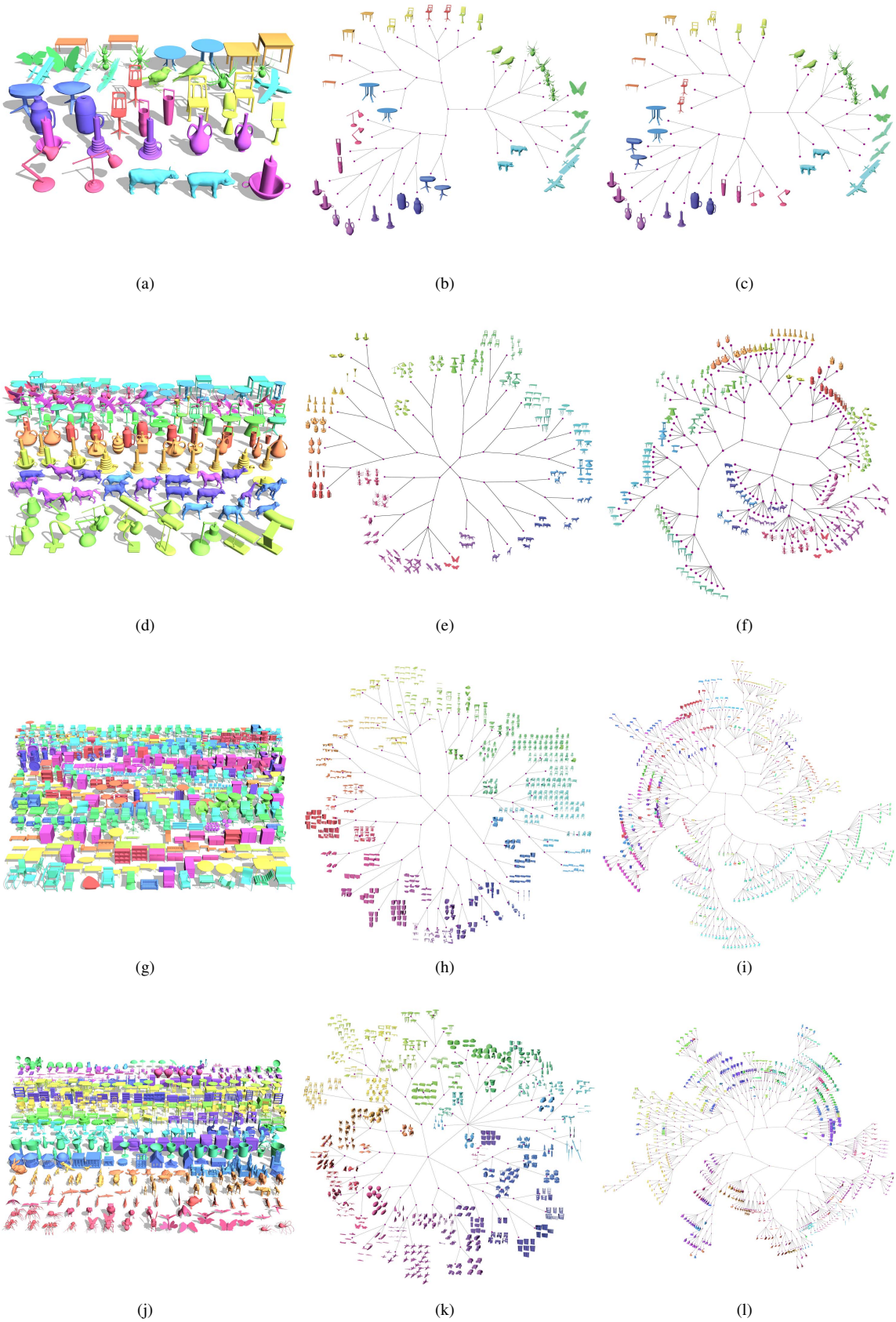


Figure 13: *Results on large sets of shapes. Left: a view of the heterogenous shape collections. Middle: ground truth tree. Right: C-trees computed by our method. Shape collection* $1-3$ *are from a mixture dateset and shape collection* 4 *is totally from the PSB. Please zoom in the electronic version to see details.*