



Autocorrelation Descriptor for Efficient Co-Alignment of 3D Shape Collections

Melinos Averkiou¹, Vladimir G. Kim^{2,3} and Niloy J. Mitra¹

¹Department of Computer Science, University College London, London, UK
m.averkiou@cs.ucl.ac.uk, n.mitra@ucl.ac.uk

²Department of Computer Science, Stanford University, Stanford, CA, USA
vk2@stanford.edu

³Adobe Research, Seattle, WA, USA

Abstract

Co-aligning a collection of shapes to a consistent pose is a common problem in shape analysis with applications in shape matching, retrieval and visualization. We observe that resolving among some orientations is easier than others, for example, a common mistake for bicycles is to align front-to-back, while even the simplest algorithm would not erroneously pick orthogonal alignment. The key idea of our work is to analyse rotational autocorrelations of shapes to facilitate shape co-alignment. In particular, we use such an autocorrelation measure of individual shapes to decide which shape pairs might have well-matching orientations; and, if so, which configurations are likely to produce better alignments. This significantly prunes the number of alignments to be examined, and leads to an efficient, scalable algorithm that performs comparably to state-of-the-art techniques on benchmark data sets, but requires significantly fewer computations, resulting in 2–16× speed improvement in our tests.

Keywords: digital geometry processing, modeling

ACM CCS: I.3.3 [Computer Graphics]: Computational Geometry and Object Modelling—Geometric algorithms

1. Introduction

As 3D shape collections continue to grow, various applications have emerged to take advantage of the available data. Typical examples include shape matching, model retrieval, analysis and visualization of model variations, model consolidation, etc. Almost all such applications assume that the input models are consistently aligned to facilitate part- or point-level correspondence establishment and subsequent transfer of information across the different shapes. The raw shape collections, however, rarely come consistently aligned. While for small collections such models can be manually pre-aligned, unsupervised automatic algorithms are essential for large to very large scale model collections.

A simple solution is to individually align each shape by mapping its principal axes to x -, y -, z -axis [DHS00]. Although this approach scales linearly with the number of models, it is unfortunately unstable, and can easily suffer from misaligned axes (see Figure 1 b). Another solution is to align each shape to an arbitrarily chosen reference shape by exhaustively searching for the best alignment in the space of pairwise relative orientations. This method, however, is heavily biased on the initial choice of model

and can degrade in the case of large shape variations across model collections (see Figure 1 c).

A better strategy is to co-align all the models simultaneously, without arbitrarily committing to a single reference model. However, directly comparing all the model pairs at all possible relative alignments is expensive and quickly becomes unattractive as the size of shape collections grows. An alternative is to select a subset of model pairs from the collection, uniformly sample their pairwise alignments and use consistency in the alignment of these pairs to co-align the models using a labelling formulation (e.g. [KLM*12, HSG13, ZCOM13]).

In this paper, we focus on the co-alignment problem in the context of large and diverse shape collections and propose a method that goes beyond uniform sampling of relative orientations, leading to efficient and accurate alignments. We first make two key observations: (i) A pair of similar shapes is easy to align even using simple alignment methods; while shapes with large geometric variations are difficult to align due to multiple candidate alignments with small intersurface distance. (ii) Comparing a shape to itself, i.e. the autocorrelation function of a shape reveals insights into the

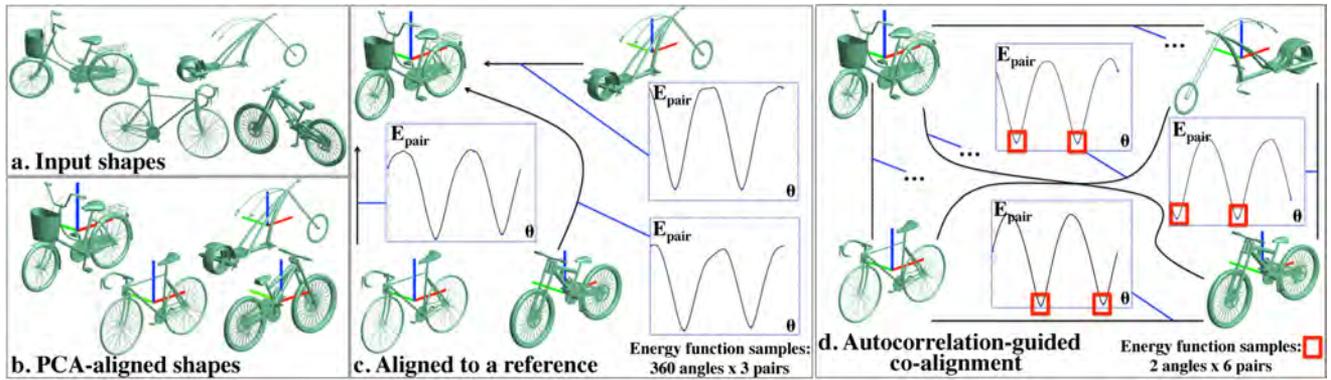


Figure 1: Shape collections typically come with inconsistent orientations (a). PCA-based alignment (b), or aligning to an arbitrarily chosen base model (c) is prone to error. The problem with pairwise alignments is attributed to several minima in alignment distances (E_{pair}), arising due to near-symmetries of shapes. We introduce an autocorrelation-guided algorithm to efficiently sample the minima (red boxes) and jointly co-align the input models (d).

possible sources of confusion arising out of self-similarity. Based on the first observation, we only align similar shape pairs and then diffuse the alignment information to other shapes through a shape graph. However, this requires first to identify which shape pairs are similar *without* explicitly comparing them. We exploit the second observation to not only determine which shape pairs to explicitly compare, but more importantly, to discover which relative alignments can potentially lead to ambiguity and hence should be further examined. For example, in Figure 1(d) our method efficiently

co-aligned a set of bikes, sampling only two low-energy alignments for each pair of shapes.

Models in shape collections (e.g. Trimble warehouse, TurboSquid) typically come with consistent up vectors. Therefore, co-aligning them effectively involves resolving a one-dimensional rotational ambiguity about the up vector. We propose a descriptor based on rotational autocorrelation of a shape, and an associated method that allows us to intelligently sample only a small number of candidate alignments for shapes. This results in an efficient algorithm that is also input sensitive, i.e. the running time depends on the extent of co-alignment ambiguity in the corresponding shape collection. Figure 2 illustrates the relationship between rotational autocorrelation and ambiguities in alignments, in the context of a chair-bench shape collection. The algorithm is robust as it only considers symmetrically similar shapes while deciding which shape pairs to compare.

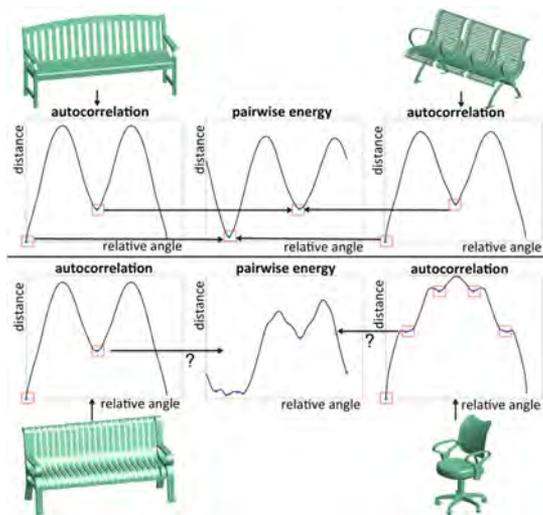


Figure 2: We use autocorrelation shape descriptors to predict model similarity without explicitly comparing them. If two shapes S_i and S_j are indeed similar, we use their autocorrelation descriptors E_i and E_j to predict potential relative alignment configurations between S_i, S_j that should be further investigated. In this example, the top half shows similar models whose autocorrelation functions can be used to predict their relative alignment; the bottom half shows dissimilar models whose autocorrelation functions provide confusing signals for their relative alignment.

Starting from an input set of shapes with similar semantics, for example, a set of bikes, we compute the autocorrelation descriptor per shape, and use it to cluster input shapes such that shapes with similar rotational symmetry end up in the same cluster. We then build an alignment graph per cluster of shapes, and align shapes inside the cluster by efficiently sampling their candidate alignments and minimizing a formulation of the sum of pairwise shape distances allowing multiple local minima, using the autocorrelation descriptor as a guide. Finally, we align shape clusters by a modified formulation of the sum of pairwise shape distances, this time from shapes between clusters.

We prepared 10 diverse benchmark data sets with ground truth alignments, and evaluated our approach against alternative alignment strategies. We report comparable alignment accuracy to state-of-the-art methods at only a fraction of the time. Specifically, we observe 2–16 \times speed improvement in our tests.

2. Related Work

Shape alignment is a common problem encountered in shape reconstruction [BM92], segmentation [GF09, SvKK*11, WAvK*12],

matching [HZG*12, KLM*13], exploration [FMK*03, KLM*12, AKZM14], automatic synthesis [FKS*04, KCKK12] and classification [HSG13]. Thus, a variety of techniques have been developed in recent years to address the problem, which we overview in this section.

Pairwise alignment. Typically, pairwise alignment algorithms fix one shape and search for a transformation for another one minimizing the distance between shapes. Horn [Hor87] demonstrates that optimal translation and scale can be found by aligning centroids of objects and uniformly normalizing the variance in the distance of points to their centre of mass. Thus, most techniques focus on searching for the optimal rotation that would align objects. To reduce the search space, some methods rely on Principal Component Analysis (PCA) and only choose among rotations that align principal components of shapes [FMK*03, Kaz04], while others combine PCA with a symmetry criterion [STP11, CVB09]. Kazhdan [Kaz07] proposes an efficient method that first converts shapes to spherical functions, which simplifies the search for the optimal rotation. Another way to reduce the search space is to detect upward orientation first [FCODS08], and then only look for rotations around the up axis [KLM*13, HSG13].

One can further refine the quality of pairwise alignments with the Iterative Closest Point (ICP) algorithm [BM92, CM92], which has several variants [RL01, BR07]. These methods are commonly used for locally refining the alignment of similar shapes, while our focus is on global co-alignment of shape collections, typically containing large in-class variations.

Co-alignment of shape collections. When aligning a collection of 3D models, considering all pairs independently is highly inefficient. To reduce the matching complexity, Huber [Hub04] introduced the idea of using a shape graph to align multiple scans of the same object in a common coordinate frame. However, when considering a heterogeneous collection of shapes (*cf.* [MWZ*13]), picking a single alignment reference becomes challenging. To address this challenge, Huang *et al.* [HSG13] and Zheng *et al.* [ZCOM13] pose the co-alignment problem as a labelling problem in the alignment space. The solution space is uniformly sampled to generate candidate labels, and the cumulative pairwise alignment error is evaluated against possible labelling to pick the best co-alignment solution. We observe that for most shapes this uniform sampling leads to a large number of unnecessary computations, since many alignments yield very high distance. Typically, only a few relative angles lead to local minima in the pairwise distances and need to be resolved by the joint matching. Our key observation is that these minima arise due to rotational near-symmetries of shapes, and thus can be estimated for each shape *independently*. The main technical contribution of this paper is an efficient algorithm that generates same quality of results with a significantly smaller computational overhead.

Symmetry and shape alignment. Our work belongs to the emerging research topic of symmetry-guided shape matching. Kazhdan *et al.* [KFR04] applied reflective and rotational symmetry descriptors for shape retrieval. Following their work, we use an autocorrelation descriptor to find similar shapes, however, our main contribution is a method to leverage autocorrelation to speed-up co-alignment of

shape collections by efficiently exploring energy minima in pairwise alignments. Podolak *et al.* [PSG*06] detect reflective symmetries of a shape to pick a canonical orientation for objects. They produce the orientation by iteratively selecting the strongest reflective symmetry planes that are orthogonal to previous planes. However, they treat each object independently, and thus their method does not aim at consistently aligning a collection of shapes.

Symmetry is also fruitful in relating non-rigid geometry. For example, Ovsjanikov *et al.* [OHG11] demonstrated that intrinsic symmetries are related to the complexity of shape matching methods, and further devised an algorithm to find multiple meaningful pairwise maps in the presence of intrinsic symmetries [OMPG13]. Liu *et al.* [LKF12] leveraged intrinsic reflective symmetries of shapes to improve efficiency and quality of pairwise shape matching. Unlike these methods, in this work we do not take advantage of perfect symmetries (in fact, our shapes are not rotationally symmetric), instead we search for near-symmetries that might cause ambiguities in shape matching. We then resolve such ambiguities with a co-alignment formulation.

3. Key Idea

Our system takes a collection of shapes $\mathcal{S} = \{S_i, i = 1, \dots, n\}$ as input, and produces a canonical transformation for every shape $T_i = R_i T_i^{\text{norm}}$, where normalization T_i^{norm} is performed for each shape independently, and the key focus of this paper is effectively estimating R_i that minimizes distances between all pairs of shapes. We assume that T_i^{norm} aligns the up vector of shape S_i to z -axis, and parametrize R_i by a rotation around up vector, $R_i = \text{Rot}_z(\theta_i)$. We formulate our objective function as

$$E := \sum_{i,j} E_{i,j}(\theta_i, \theta_j), \quad (1)$$

where $E_{i,j}$ estimates how well S_i and S_j mutually align if rotated by θ_i and θ_j , respectively. Sampling the energy function $E_{i,j}$ is the most expensive step of the algorithm since it requires computing distances between surfaces for all relative angles of the form $\Delta_{i,j} = \{\theta_i - \theta_j\}$.

The key observation behind our work is that ambiguity in shape alignments usually arises due to approximate rotational symmetries of shapes. For example, in Figure 2 the individual autocorrelation descriptors of the benches already provide valuable clues as to which relative angles between the two benches can be ambiguous, *even without* explicitly comparing the two benches. In other words, if shapes are expected to be similar, one does not need to evaluate pairwise energy to predict this potential ambiguity since the shapes would exhibit similar near-symmetries. Thus, we estimate the number of ambiguous alignments between pairs of objects, as well as the relative angle between the ambiguous alignments by analysing autocorrelation function

$$E_i(\theta) := E_{i,i}(0, \theta) \quad (2)$$

which captures how similar is a shape S_i to itself under a rotation θ . Figure 3 shows an example.

In particular, we expect self-symmetries of shape S_i to form an algebraic group of k_i elements: $G_i^{\text{symm}} = \{\theta_{i,l_i}^{\text{symm}} = 0, \dots, \theta_{i,k_i}^{\text{symm}}\}$, where S_i is self-similar under rotation $\theta_{i,l_i}^{\text{symm}}$. Thus, for each shape

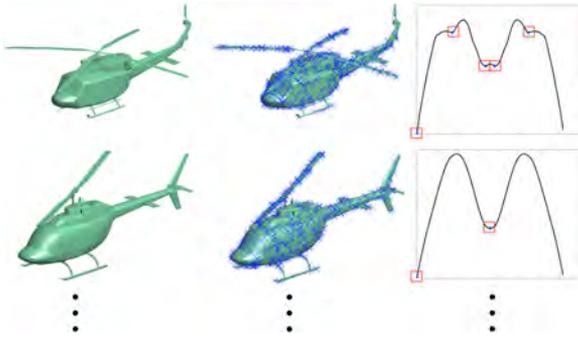


Figure 3: We normalize and sample points on each shape, compute its autocorrelation descriptor E_i and store the set of E_i 's local minima (highlighted by red boxes).

S_i we only need to consider k canonical alignments related by angles in the symmetry group. In order to evaluate $E_{i,j}$, however, these canonical alignments have to be consistent between S_i and S_j , thus groups have to be co-aligned by an offset alignment θ_i^{off} (see Figure 4). Note that finding offset alignment θ_i^{off} is much easier than finding canonical alignment θ_i , since we only need to find how to align one element of a group, without needing to resolve ambiguities. We estimate the offset by aligning each shape to an arbitrary reference S_r as: $\theta_i^{\text{off}} := \arg \min_{\theta'} E_{i,r}(\theta', 0)$.

Thus, for any $E_{i,j}$, we only need to consider

$$\Delta_{i,j} = \{(\theta_i^{\text{off}} + \theta_{i,l_i}^{\text{symm}}) - (\theta_j^{\text{off}} + \theta_{j,l_j}^{\text{symm}}), \quad (3)$$

$$l_i = 1, \dots, k_i, l_j = 1, \dots, k_j\}$$

alignments, which drastically reduces the number of pairwise distances we need to compute to find all interesting minima in $E_{i,j}$.

The above formulation assumes that all objects in the collection exhibit the same approximate rotational symmetries. This, however, may not be true for heterogeneous data. One possibility is to take a product of symmetry groups to take all canonical orientations into account, but this would increase the number of pairwise distance computations. Instead, we group objects based on their rotational self-symmetries using the similarities in their autocorrelation descriptors and optimize for alignments of objects within the same group. After all objects within the same group are co-aligned, we estimate intergroup alignment by solving a smaller optimization problem that only includes a few edges between shapes in different groups.

4. Method

Given a collection of shapes $\mathcal{S} := \{S_i\}$ our method produces canonical transformations $\{T_i\}$. The algorithm starts with per-shape analysis to find normalizing transformations T_i^{norm} and autocorrelation descriptors E_i for each shape. Next, we group shapes based on their descriptors, extracting clusters of shapes that share similar symmetries. The method co-aligns models in each cluster by leveraging autocorrelation descriptors to decide which pairs of models to align and which alignments to sample. After this intracluster alignment,

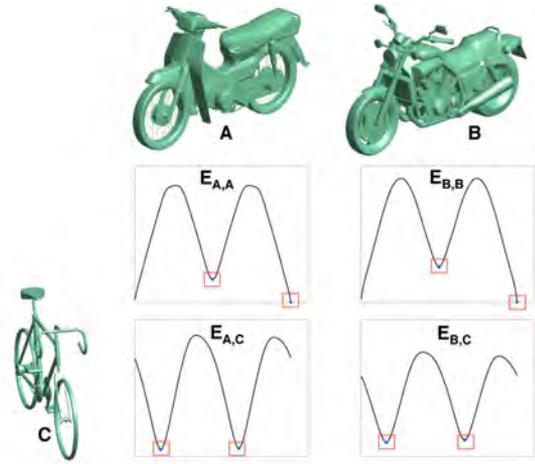


Figure 4: We show the autocorrelation functions ($E_{A,A}$, $E_{B,B}$) for two motorcycles (A, B), and their pairwise energy function ($E_{A,C}$, $E_{B,C}$) when comparing to a bike (C). A and B are aligned to the global axes while C is rotated 60° around z -axis. Note how this causes the two minima of $E_{A,A}$ and $E_{B,B}$ (at 180° and 360°) to shift by 60° in $E_{A,C}$ and $E_{B,C}$. We can therefore expect to find an alignment of the two motorcycles and the bike at 240° or 60° , so there is no need to sample other rotations.

our algorithm aligns the different clusters via another optimization. Our pipeline is summarized in Figure 5, and the rest of this section describes each step in detail.

4.1. Normalization

To estimate per-shape normalization T_i^{norm} , we scale the height to be one, and translate the centre of bounding box to $[0, 0, 0.5]$, such that the ground plane is $z = 0$. All shapes used in our experiments have consistent upward orientation. For other data sets, one can also use the method of Fu *et al.* [FCODS08] to consistently orient the shapes upwards. The goal then is to find a rotation around the up vector for each shape that would consistently co-align all shapes.

4.2. Autocorrelation descriptor

We leverage understanding of rotational near-symmetry of a shape to group shapes that have similar symmetries and efficiently sample good co-alignments. To represent the symmetry of each shape S_i , we compute the autocorrelation descriptor E_i [KFR04], which measures how much the shape correlates with itself, under a rotation:

$$E_i(\theta) = [D_S(S_i, Rot_z(\theta)S_i)], \quad \theta \in [0, 2\pi], \quad (4)$$

where $Rot_z(\theta)$ is a rotation around up vector by θ degrees, and $D_S : S_i \times S_j \rightarrow \mathbb{R}$ measures distance between surfaces S_i and S_j . In order to compute distances in our experiments, we uniformly sample 1000 points, P_i , on each surface, compute the mean

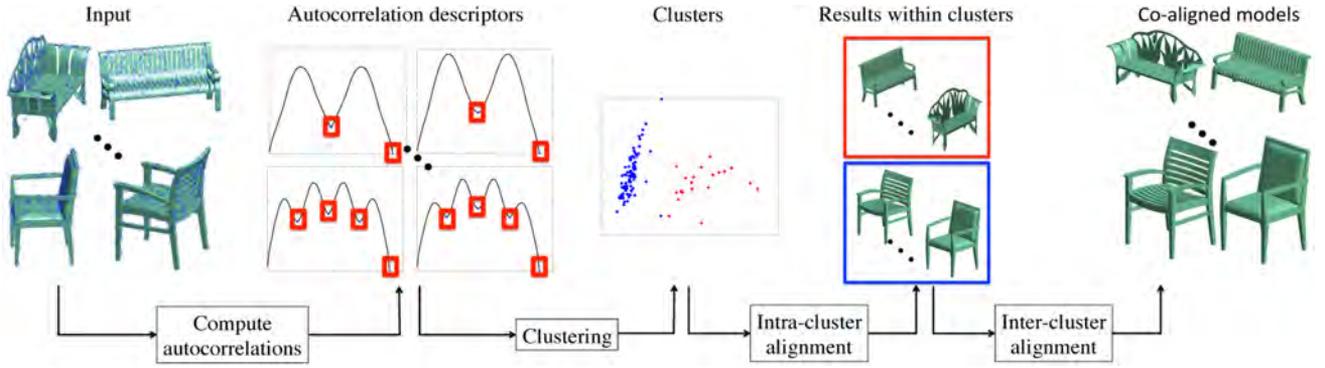


Figure 5: Algorithm overview. Starting from a set of shapes, we normalize and compute their autocorrelation descriptors to cluster the shapes. We then align the shapes first within and then across the clusters using a graph-based discrete formulation wherein we intelligently sample candidate alignments for each shape guided by their autocorrelation descriptors.

distance from all P_i to their nearest point in P_j and the mean distance from all P_j to their nearest point in P_i and take the maximum of the two. We uniformly sample E_i with 360 samples. Figure 3 shows an example for a set of helicopters. The autocorrelation descriptor is normalized by dividing each entry of $E_i(\theta)$ by $\sum_{\theta} E_i(\theta)$.

4.3. Shape clustering

We group shapes based on their symmetries to ensure that we can effectively sample pairwise alignment energy function and to find groups that can be co-aligned robustly. In particular, we group the input shapes into a set of clusters C_i using a graph-based technique. We take the k th largest pairwise distance in autocorrelation descriptor space as threshold t and connect all shapes with distance smaller than t with an edge. We then find connected components of the resulting graph to get the set of clusters C_i . In our experiments, we set k to 10% of the number of shape pairs which resulted in 6–21 clusters, depending on the data set. We use L1-norm distance between the autocorrelation descriptors to measure approximate

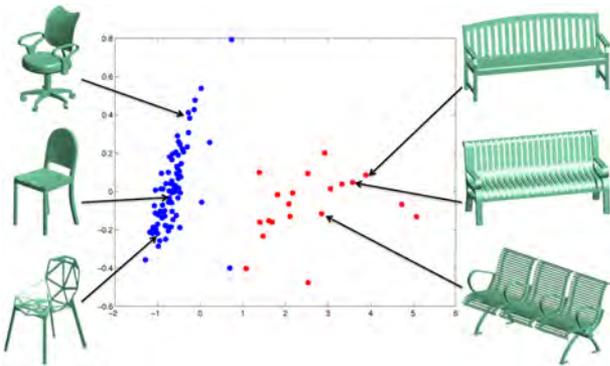


Figure 6: Embedding of autocorrelation descriptors for a data set of chairs in 2D, with points coloured according to the cluster they belong. Note how chairs are separated from long benches as they have different sets of self-symmetries.

earth-movers distance. Figure 6 demonstrates the clustering result on a chair-bench data set. Note how chairs and benches are separated in two clusters: benches typically have two near-symmetries with a potential confusion in a rotation by π , while chairs pose a bigger challenge with a larger number of near-symmetries.

4.4. Intracluster alignment

Next, we co-align all models within each cluster C_i . Note that the models in C_i share similar symmetries, and thus we can take advantage of E_i to efficiently sample pairwise energy $E_{i,j}$. First, we smooth E_i using moving average with a span of 0.03π and compute all of its local minima which gives us the group of symmetric rotations: G_i^{symm} , where each rotation $\theta_{i,l_i}^{\text{symm}} \in G_i^{\text{symm}}$ leads to an ambiguity that needs to be resolved by joint optimization. Note that the minima in G_i^{symm} are defined in shape S_i 's coordinate system, starting with 0 rotation, thus we need to bring all shapes to some canonical alignment space. This is, however, an easier problem than joint alignment, because it is sufficient to find one element of the group to estimate the offset alignment θ_i^{off} , which would define canonical alignments of shapes. We pick the shape S_r nearest to the cluster's centre in descriptor space and estimate offset angle θ_i^{off} as: $\theta_i^{\text{off}} := \arg \min_{\theta'} E_{i,r}(\theta', 0)$. With this offset, we can now effectively sample the pairwise energy function using the offsets described in Equation (3) as,

$$E_{i,j}(\theta) = [D_S(S_i, \text{Rot}_z(\theta)S_j)], \text{ where } \theta \in \Delta_{i,j}. \quad (5)$$

In practice, we use an Markov Random Field (MRF) labelling problem to minimize Equation (3), where possible canonical rotations for each shape S_i define the number of labels. In our case, these canonical rotations are $\theta_i^{\text{off}} + \theta_{i,l_i}^{\text{symm}}$ with $l_i = 1, \dots, |G_i^{\text{symm}}|$. In order to select the set of shape pairs (i, j) , we sparsely sample pairs of shapes by looking at $m = 20$ nearest neighbours in autocorrelation descriptor space. This parameter is set so that the resulting shape graph is kept sparse but remains connected. We use the method described in Leordeanu et al.[LHS09] to optimize Equation 1. The output of this step is an angle θ_i^{ic} for each shape in the cluster, chosen from the candidate canonical rotations.

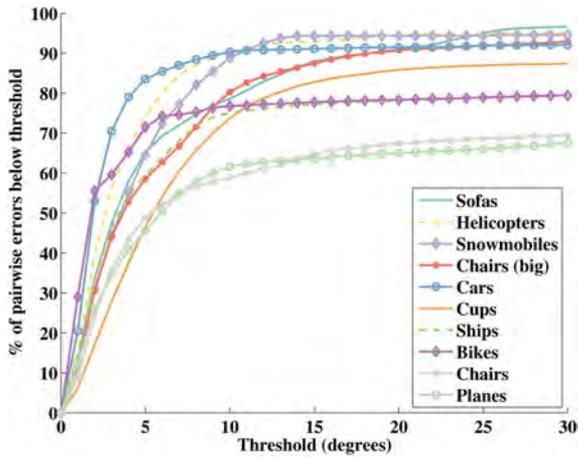


Figure 7: We plot the fraction of models (y-axis) aligned within a prescribed angle threshold (x-axis) for 10 data sets. All our results are substantially better than a baseline of random alignments, which would only give about 8% accuracy for a threshold of 15° . Similarly to all shape matching techniques our method suffers from near-symmetries of shapes.

4.5. Intercluster alignment

After co-aligning shapes in each cluster, our method estimates intercluster alignments. In particular, we construct another joint alignment MRF problem where we seek to select one rotation for every cluster out of a set of possible rotations (labels). First, we apply rotation $Rot_z(\theta_i^{ic})$ from intracluster alignment to each shape of every cluster. We keep the largest cluster fixed, and compute the energy for each pair of clusters by connecting pairs of shapes from the two clusters and summing their pairwise energy $E_{i,j}$. Since shapes in different clusters do not share similar symmetries, we densely sample the pairwise energy $E_{i,j}$ with 32 uniform samples. We choose $m_c = 20$ edges for each pair of clusters (same as in intracluster alignment), connecting shapes that have the most similar autocorrelation descriptors E_i from the two clusters. The output of this optimization is an angle for each cluster, which can then be added to the angle θ_i^{ic} from the intracluster alignment step to obtain the final canonical rotation θ_i for each shape.

5. Evaluation

In this section, we evaluate our method on 10 diverse data sets, ranging in size from 32 to 1000 shapes. We design experiments to evaluate performance of our method on different shape classes, quantify the efficiency improvement, effect of various choices we made when designing the algorithm and scalability.

5.1. Experimental setup

Data sets. Although many previous techniques rely on shape co-alignments, there is no standard benchmark for quantitatively evaluating these methods. Hence, in order to evaluate our approach we create a large and diverse benchmark with ground truth align-

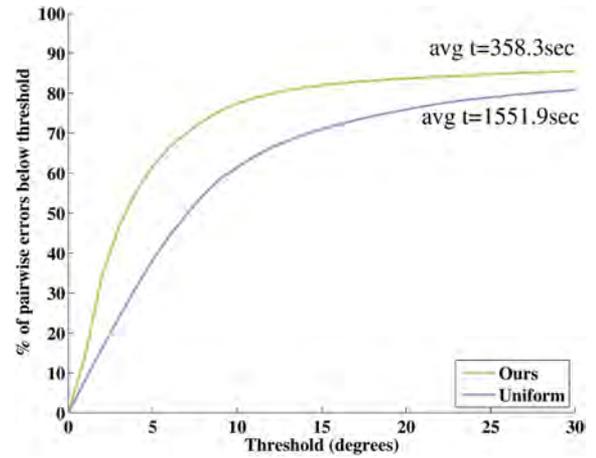


Figure 8: We compare our method (green) to our implementation of the UNIFORM method (blue), where the results are averaged over 10 data sets. Our method outperforms the UNIFORM method in accuracy. Most importantly, it has less computational complexity than UNIFORM, as it requires 2–16 times (depending on the data set) less samples of $E_{i,j}$ to align shapes within all clusters, compared to UNIFORM (see Table 1).

ments for 10 different data sets. First, we use the correspondence benchmark provided by Kim *et al.* [KLM*13] that includes a small number of consistently annotated feature points for 100 bikes (including bicycles and motorcycles), 100 chairs (including armchairs and benches), 100 helicopters and 104 aeroplanes. All models have consistent upward orientation aligned to the global z -axis. After normalizing the models as described in Section 4.1, we fix one shape in the collection, and align every other shape to it by finding the optimal rotation around the z -axis that minimizes the L2 distance between the ground truth correspondences. Secondly, we downloaded six additional data sets from Trimble 3D Warehouse including 32 snowmobiles, 100 cars, 100 cups, 100 ships, 100 sofas and a big data set of 1000 chairs. All models in each collection have a consistent upward orientation, and we curated the data sets to remove shapes that did not belong to their respective category. Finally, we manually prescribe ground truth rotation for each model. We believe that the benchmark can be valuable for evaluating future algorithms and is made freely available from the project webpage.

Evaluation metric. We evaluate the performance of our method based on how accurately it co-aligns shapes in a data set. For each shape S_i , let the ground truth rotation around the up vector be denoted by θ_i^{gt} and an algorithmically predicted rotation by θ_i . We measure the alignment error $a(i, j)$ of a method based on the distance between the true and predicted relative angles as

$$a(i, j) := |(\theta_i^{gt} - \theta_j^{gt}) - (\theta_i - \theta_j)|. \quad (6)$$

In the following comparisons, we plot the fraction of shape pairs (y-axis) whose alignment error $a(i, j)$ is smaller than a threshold (x-axis).

5.2. Per-class performance

Figure 7 shows the accuracy of our method for different data sets, and Figure 12 shows images of resulting alignments for a random selection of shapes (please, refer to Supporting Information for all alignments from all data sets). For a relatively strict threshold of 15° angle, our method correctly aligned 80% of models for most data sets. Note that in all cases the results are significantly better than a random rotation, which would only achieve about 8% for 15° threshold. Low accuracy in some data sets is caused by near-symmetries. For example, position of wings along the fuselage of an aeroplane can incorrectly favour nose-to-tail alignment. This problem is common for all shape matching algorithms, and our method is not designed to resolve the issue: it only speeds up the optimization time significantly by focusing on these challenging ambiguities.

Near-symmetries decrease accuracy both in intra- and intercluster alignments. For example, the two biggest clusters of aeroplanes data set get around 70% and 85% accuracy for 15° angle. In comparison, models in the biggest clusters of (small) chairs data set are aligned with about 90% accuracy within each cluster, which is then reduced to about 60% by the inaccurate intercluster alignment. We would like to point out that the user supervision can significantly improve accuracy at a very small cost in the latter case. Section 5.5 gives the details and Figure 11 shows how accuracy is affected by supervision; in particular we can get above 90% accuracy at 15° angle threshold for chairs with just six manual alignments (only one per cluster).

5.3. Efficiency improvement

Our main contribution is the method for efficient sampling of shape rotations based on potential ambiguities caused by near-symmetries. Hence, we compare our method to the state-of-the-art approach that uniformly samples the rotations as proposed by Huang *et al.* [HSG13]. In their approach, they assume that shapes are consistently aligned in the upward direction and take 32 uniformly sampled rotations around the up vector for each pair of shapes S_i and S_j . Their method is designed to compute fine-scale correspondences, and thus they further co-deform the shapes using affine transformations and free-form deformations. Since the goal of our work is rigid co-alignment of models we only keep the first step of their pipeline that optimizes for rotations. To ensure fair and consistent comparison, we modify our implementation to mimic the method of Huang *et al.* [HSG13] (titled as UNIFORM in all figures). In particular, we uniformly sample the rotation space, but we keep the same graph connectivity and intercluster alignment step for both methods.

Figure 8 demonstrates accuracy results averaged over all data sets (see Supporting Information for the full set of results and per-data set comparisons). Our method achieves better accuracy with significantly less computational overhead. We gain the most significant improvement over 0° – 15° thresholds since UNIFORM method is constrained to samples at 11° increments. Table 1 further provides detailed timing for each data set. We get up to 16 times speed-up (e.g. for ships) due to the symmetry-guided sampling of the energy function (the two rightmost columns compare the number of times

Table 1: The first two columns show the time, spent on solving the optimization from Section 4.4 for our method and UNIFORM. Intercluster alignment time is excluded since it is the same for both methods. The third and fourth columns show the number of samples taken from $E_{i,j}$ in the same optimization problem. Note that our method is faster than UNIFORM, which takes 32 samples for pairwise alignments. Our method becomes more computationally expensive for classes of shapes that exhibit more symmetries, such as cups, aeroplanes and helicopters.

	Ours (s)	UNIF. (s)	Ours (# $E_{i,j}$)	UNIF. (# $E_{i,j}$)
Sofas	18.2	345.1	1902	29 184
Helicopters	124.6	450.4	12 180	36 544
Snowmobiles	7.6	69.4	818	5824
Chairs (big)	3001	12149.5	291 273	930 528
Cars	23.8	435.3	2614	37 696
Cups	116.3	434.6	11 699	36 320
Ships	17.6	384.6	1900	30 400
Bikes	45.5	472.7	3560	32 224
Chairs	93.6	428.4	9046	34 688
Aeroplanes	135.2	349.5	14 300	30 944

the energy was computed). Table 1 also suggests that our method is input-sensitive, since data sets with more near-symmetries such as cups, helicopters and planes take longer to align. However, even for these data sets our method is two to three times faster than the UNIFORM method.

One can further improve the accuracy or speed of the UNIFORM method by changing the sampling frequency. Figure 9 shows a representative comparison on a bike data set of our method and the UNIFORM method with various numbers of samples. As the number of samples increase, both accuracy and complexity of UNIFORM method increase. In contrast, our method achieves accuracy that is comparable to the 64-sample UNIFORM method, at a fraction of the time.

5.4. Effect of clustering

We examine the effect of the clustering step in our alignment pipeline. This step is designed to handle heterogeneous collections where shapes do not have similar symmetries. In particular, we cluster shapes based on their symmetry descriptors and then optimize for per-cluster rotations. We compare the alignment accuracy of the results produced with the clustering step and by aligning all shapes jointly. The average alignment accuracy over all data sets can be seen in Figure 10 (and per-data set accuracy results can be found in the Supporting Information). The use of clustering and the two-step alignment procedure increases accuracy by around 15% for the 15° threshold.

5.5. Effect of human supervision

While all results presented in previous sections were created fully automatically, our method can also efficiently leverage human supervision during the intercluster alignment step. After aligning shapes within each cluster, the user can further align different clusters to

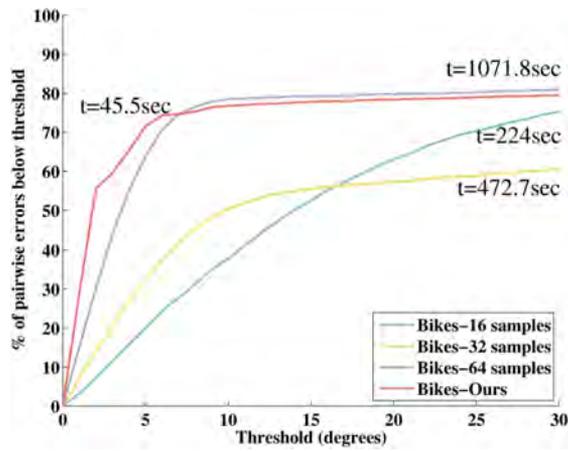


Figure 9: We plot the fraction of bikes aligned within a prescribed angle threshold, for our method and UNIFORM method with increasing number of samples. The accuracy of UNIFORM increases as the number of samples increases, at the cost of longer running time, while our method achieves higher accuracy at a fraction of the time.

avoid unreliable matching of dissimilar shapes. In particular, our method picks a representative shape from each cluster (i.e. the shape nearest to the cluster’s centre in descriptor space). Then the user is prompted to consistently rotate representative shapes for all clusters. This bears little overhead since the number of clusters is small in comparison to the size of the collections (6–21 clusters for the small data sets, 34 clusters for the 1000-chair data set) and the number of required manual alignments is equal to the number of clusters.

We simulate the human supervision using the ground truth angles, in particular, we pick a rotation that correctly aligns the representative shape of each cluster to the representative shape of the biggest cluster. Figure 11 shows the accuracy of our method with and without human supervision for the chairs and aeroplanes data sets (see Supporting Information for the results on all other data sets). These two data sets had the lowest performance in terms of accuracy for the unsupervised approach (see Figure 7). With the supervision, the accuracy increases to over 90% for chairs and 70% for aeroplanes. The improvement is prominent for chairs since most of the errors in that data set are due to incorrect intercluster alignment.

5.6. Scalability

To demonstrate the scalability of our method, we tested it on a large data set of 1000 chairs. We set the parameter k , which controls the clustering radius to $k = 2.5\%$ since the data set has higher diversity in comparison to the smaller collections. Since this data set is much larger than the other data sets, we also iteratively increase the number of nearest neighbours that are connected by an edge, starting from $m = 20$ (used for all other data sets) and increasing m in increments of 15 until all individual graphs are connected (terminating at $m = 50$). Table 1 demonstrates the timings for the 1000 chairs data set, where our method outperforms UNIFORM by four times.

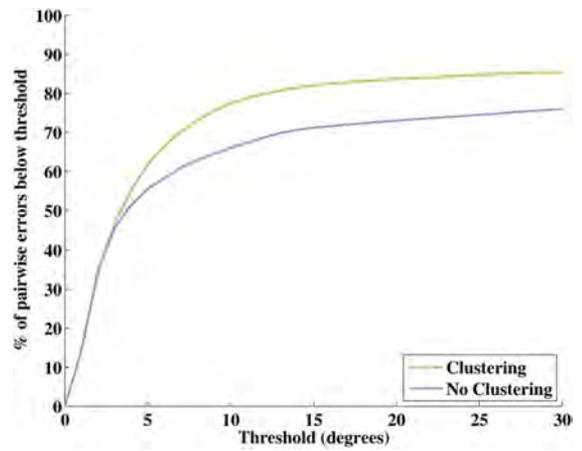


Figure 10: We evaluate the effect of the clustering part of our pipeline by comparing our method with clustering performed (green) to our method with clustering turned off, i.e. all shapes jointly aligned in one step (blue). This plot shows results averaged over 10 data sets. Clustering and aligning shapes in two steps increases accuracy on average, compared to jointly aligning all shapes.

5.7. Computational complexity

Since most of the computation time is spent on sampling the pairwise objective function $E_{i,j}$, we measure the computational complexity in terms of the number of samples that have to be computed. The intracluster alignment for a given cluster c requires $N_c m M_i$ samples, where N_c is the number of shapes in a cluster, m is the number of edges per shape (set to 20 in our experiments), and $M_i = |G_i^{\text{symm}}|$ is the number of candidate alignments (i.e. the number of approximate symmetries) per shape (2–12 in our experiments). The intercluster alignment requires $K m_c M_c$ samples, where K is the number of clusters (6–21 for all data sets), m_c is the number of edges we use to connect clusters (set to 20 in our experiments) and M_c is the number of candidate rotations per cluster pair (set to 32 uniform samples). The UNIFORM method has similar complexity, but the candidate per-shape alignments M_i is set to constant $M_{\text{uniform}} = 32$ rotations. Thus

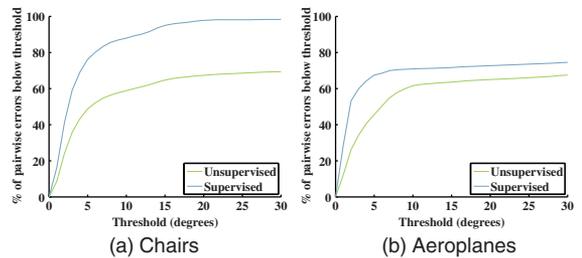


Figure 11: We evaluate the effect of human supervision by comparing the accuracy of our unsupervised alignment pipeline (green) to the accuracy achieved with human supervision (blue), for the chairs and the aeroplanes data sets. Supervision helps improve accuracy for these low-performing data sets, using just six manually prescribed rotations for chairs and 21 for aeroplanes.

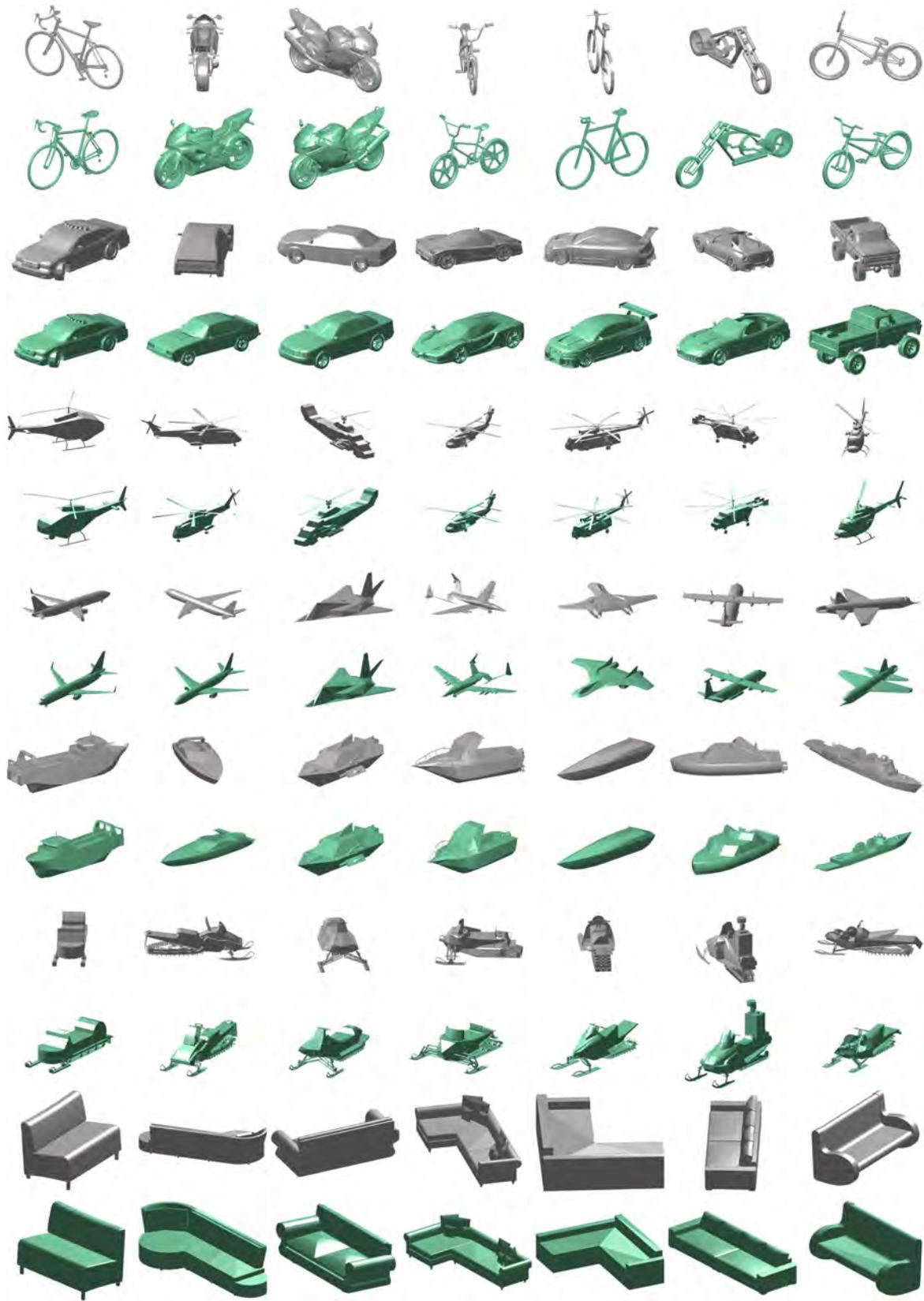


Figure 12: Randomly selected shapes from all our data sets, indicating their pose before (odd rows—in grey) and after (even rows—in green) alignment.

our method only speeds up the intraclustering step as long as $M_i = |G_i^{\text{symm}}| < M_{\text{uniform}}$ and $N_c > 1$.

5.8. Limitations

While our method makes no assumptions regarding symmetry of the input shapes, it is designed to improve efficiency only if shapes have a small number of near-symmetries, e.g., as we can see in Table 1, the improvement for bikes is more significant than for cups. Our technique also assumes that there is a meaningful alignment for each shape in the input collection, and thus it can be sensitive to outlier objects. To test this, we collected a dataset of 109 cars from Trimble 3D Warehouse, with around 10% of the cars having additional noisy geometry or wrong up vectors. Our method achieved 75% accuracy at 10 degrees threshold, compared to over 90% for our original clean 100 car dataset (see supporting information for both results). We also note that our method's efficiency degrades significantly if the clustering step produces a large number of small clusters, since we uniformly sample alignments in the intercluster alignment step. We have not encountered this in practice when dealing with collections that contain shapes from the same category. Lastly, our method only relies on geometry and ignores any additional cues such as texture, which often come with shapes retrieved from online collections. In the future, it would be interesting to study how such cues can affect alignment efficiency and accuracy.

6. Conclusions

We presented an efficient algorithm for co-aligning 3D models in a collection. The algorithm achieves robustness by solving for consistent joint alignment of the entire data set using several potentially erroneous pairwise alignments. Our method is also efficient since it produces semantic alignments with only a subset of pairs being aligned, and most importantly it effectively computes pairwise alignment energies only for relative angles that might cause ambiguity and need to be resolved with the joint matching. The key contribution is using the autocorrelation function of shapes to estimate potential sources of ambiguities, and only focusing on these to efficiently sample the pairwise energies. This results in superior performance with respect to state-of-the-art methods at only a fraction of computational cost.

Acknowledgements

We thank the reviewers for their comments and suggestions for improving the paper. This work was supported in part by ERC Starting Grant SmartGeometry (StG-2013-335373) and gifts from Adobe. Melinos Averkiou is grateful to the Rabin Ezra Scholarship Trust for the award of a bursary.

References

- [AKZM14] AVERKIOU M., KIM V. G., ZHENG Y., MITRA N. J.: ShapeSynth: Parameterizing model collections for coupled shape exploration and synthesis. *Computer Graphics Forum* 33, 2 (2014), 125–134.
- [BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.
- [BR07] BROWN B., RUSINKIEWICZ S.: Global non-rigid alignment of 3D scans. *ACM Transactions on Graphics* 26, 3 (2007).
- [CM92] CHEN Y., MEDIONI G.: Object modelling by registration of multiple range images. *Image and Vision Computing* 10, 3 (1992), 145–155.
- [CVB09] CHAOUCH M., VERRONST-BLONDET A.: Alignment of 3D models. *Graphical Models* 71, 2 (2009), 63–76.
- [DHS00] DUDA R. O., HART P. E., STORK D. G.: *Pattern Classification* (2nd Edition). Wiley-Interscience, New York, USA, 2000.
- [FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. *ACM Transactions on Graphics* 27, 3 (2008), 42:1–42:7.
- [FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Transactions on Graphics* 23, 3 (2004), 652–663.
- [FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A search engine for 3D models. *ACM Transactions on Graphics* 22, 1 (2003), 83–105.
- [GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3D models. *Computers & Graphics* 33, 3 (2009), 262–269.
- [Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of Optical Society of America A* 4, 4 (1987), 629–642.
- [HSG13] HUANG Q.-X., SU H., GUIBAS L.: Fine-grained semi-supervised labeling of large shape collections. *ACM Transactions on Graphics* 32, 6 (2013), 190:1–190:10.
- [Hub04] HUBER D.: *Automatic Three-dimensional Modeling from Reality*. PhD thesis, Robotics Institute, 2004.
- [HZG*12] HUANG Q.-X., ZHANG G.-X., GAO L., HU S.-M., BUTSCHER A., GUIBAS L.: An optimization approach for extracting and encoding consistent maps in a shape collection. *ACM Transactions on Graphics* 31, 6 (2012), 167:1–167:11.
- [Kaz04] KAZHDAN M.: *Shape Representations and Algorithms for 3D Model Retrieval*. PhD thesis, Princeton, 2004.
- [Kaz07] KAZHDAN M.: An approximate and efficient method for optimal rotation alignment of 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 7 (2007), 1221–1229.
- [KCKK12] KALOGERAKIS E., CHAUDHURI S., KOLLER D., KOLTUN V.: A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics* 31, 4 (2012), 55:1–55:11.

- [KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Symmetry descriptors and 3D shape matching. In *Proceedings of SGP* (Nice, France, 2004), pp. 115–123.
- [KLM*12] KIM V. G., LI W., MITRA N. J., DiVERDI S., FUNKHOUSER T.: Exploring collections of 3D models using fuzzy correspondences. *ACM Transactions on Graphics* 31, 4 (2012), 54:1–54:11.
- [KLM*13] KIM V. G., LI W., MITRA N. J., CHAUDHURI S., DiVERDI S., FUNKHOUSER T.: Learning part-based templates from large collections of 3D shapes. *ACM Transactions on Graphics* 32, 4 (2013), 70:1–70:12.
- [LHS09] LEORDEANU M., HEBERT M., SUKTHANKAR R.: An integer projected fixed point method for graph matching and MAP inference. In *Proceedings of NIPS 22* (Vancouver, Canada, 2009), pp. 1114–1122.
- [LKF12] LIU T., KIM V. G., FUNKHOUSER T.: Finding surface correspondences using symmetry axis curves. *Computer Graphics Forum* 31, 5 (2012), 1607–1616.
- [MWZ*13] MITRA N. J., WAND M., ZHANG H., COHEN-OR D., BOKELOH M.: Structure-aware shape processing. In *EUROGRAPHICS State-of-the-art Report* (2013).
- [OHG11] OVSJANIKOV M., HUANG Q.-X., GUIBAS L.: A Condition number for non-rigid shape matching. *Computer Graphics Forum* 30, 5 (2011), 1503–1512.
- [OMPG13] OVSJANIKOV M., MÉRIGOT Q., PĂTRĂUCEAN V., GUIBAS L.: Shape matching via quotient spaces. *Computer Graphics Forum* 32, 5 (2013), 1–11.
- [PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics* 25, 3 (2006), 549–559.
- [RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the ICP algorithm. In *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling* (Quebec City, Canada, 2001), pp. 145–152.
- [STP11] SFIKAS K., THEOHARIS T., PRATIKAKIS I.: ROSy+: 3D object pose normalization based on PCA and reflective object symmetry with application in 3D object retrieval. *International Journal of Computer Vision* 91, 3 (2011), 262–279.
- [SvKK*11] SIDI O., VAN KAICK O., KLEIMAN Y., ZHANG H., COHEN-OR D.: Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Transactions on Graphics* 30, 6 (2011), 126:1–126:10.
- [WAVK*12] WANG Y., ASAFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B.: Active co-analysis of a set of shapes. *ACM Transactions on Graphics* 31, 6 (2012), 165:1–165:10.
- [ZCOM13] ZHENG Y., COHEN-OR D., MITRA N. J.: Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum* 32, 2pt2 (2013), 195–204.

Supporting Information

Additional Supporting Information may be found in the online version of this article at the publisher’s web site:

Figure S1: Comparison of our method (green) and UNIFORM method (blue), for all datasets.

Figure S2: Comparison of our method with clustering (green) and our method without clustering (blue), for all datasets.

Figure S3: This figure illustrates the accuracy of our full unsupervised alignment pipeline (green) in comparison to the accuracy achieved if a human aligns shapes between clusters manually (blue).

Figure S4: Comparison of our unsupervised pipeline (green) and our supervised pipeline (blue), for all datasets.

Figure S5: Bikes dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S6: Cars dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S7: Noisy cars dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S8: Chairs dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S9: Chairs (big) dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S10: Cups dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S11: Helicopters dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S12: Planes dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S13: Ships dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S14: Snowmobiles dataset before (odd rows - in gray) and after (even rows - in green) alignment.

Figure S15: Sofas dataset before (odd rows - in gray) and after (even rows - in green) alignment.