

Multiple Shape Correspondence by Dynamic Programming

Y. Sahillioğlu¹ and Y. Yemez²

¹Computer Engineering Department, Middle East Technical University, Ankara, Turkey

²Computer Engineering Department, Koç University, Istanbul, Turkey

Abstract

We present a multiple shape correspondence method based on dynamic programming, that computes consistent bijective maps between all shape pairs in a given collection of initially unmatched shapes. As a fundamental distinction from previous work, our method aims to explicitly minimize the overall distortion, i.e., the average isometric distortion of the resulting maps over all shape pairs. We cast the problem as optimal path finding on a graph structure where vertices are maps between shape extremities. We exploit as much context information as possible using a dynamic programming based algorithm to approximate the optimal solution. Our method generates coarse multiple correspondences between shape extremities, as well as denser correspondences as by-product. We assess the performance on various mesh sequences of (nearly) isometric shapes. Our experiments show that, for isometric shape collections with non-uniform triangulation and noise, our method can compute relatively dense correspondences reasonably fast and outperform state of the art in terms of accuracy.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Computer Graphics—Computational Geometry and Object Modeling[Matching]

1. Introduction

Shape correspondence is an important problem in computer vision and graphics with numerous applications from morphing to retrieval [BBK08]. Being well studied for a pair of shapes in isolation, this problem is yet at its infancy for exploitation of context information when a collection of multiple shapes is given to be matched. In this paper, we address this new version of shape correspondence, that one may also refer to as *multiple shape correspondence*. More specifically, we aim to find consistent mappings between all shape pairs of a given collection while minimizing the overall distortion. A consistent correspondence within a shape collection is useful in creation of statistical models [BA03], group skeletonization [WH10], animation reconstruction [ATR*08], and many other correspondence-related tasks.

There are two major contributions of this paper. First, we explicitly show that, by using the idea of multiple shape correspondence, it is possible to obtain better correspondences (according to ground-truth) than one would obtain by pairwise matching, not only over the totality of a given collection of (nearly) isometric shapes but in some cases even over individual shape pairs. For this, we provide quantitative results to compare the resulting distortions in both scenarios, i.e., multiple correspondence vs. pairwise matching. We also

compare, for some computationally trivial cases, the distortions obtained by these two approaches with the distortions resulting from theoretically optimal correspondences. Second, we propose a multiple shape correspondence algorithm which explicitly minimizes the overall distortion while enforcing consistency. We define the overall distortion as the average isometric distortion over all shape pairs within the collection. To this end, we represent the input shape collection with a novel graph structure where vertices are maps between shapes, and then cast the problem as optimal path finding on this graph. We solve this optimization problem approximately based on dynamic programming. We assume that the shapes in the given collection are (nearly) isometric and that all possible maps between two given shapes can be enumerated based on some distortion measure which is in our case the deviation from isometry. Our method primarily seeks for coarse correspondences between shape extremities. However we compute these correspondences based on a dense matching framework, that is, we evaluate all possible coarse maps with denser sampling and matching, which in turn yields also denser correspondences as by-product. This matching strategy, which can be thought of as coarse-to-fine, improves resiliency of the method against noise, alleviates the symmetrical flip problem inherent to coarse matching, and results in a computationally very efficient algorithm.

2. Related Work

The straightforward solution to solve the multiple shape correspondence problem without leveraging the context information is to compute all pairwise correspondences independently, and any two-instance shape correspondence method available in the literature can be used for this purpose such as [BBK06, LF09, OMMG10, KLF11, SY12], among others. A detailed survey on shape correspondence methods can be found in [vKZHCO11]. A particular group of methods in this respect address the problem of generating dynamic mesh sequences from time-varying data, finding correspondences between time-consecutive shapes each constructed independently, which can hence incorporate Euclidean proximity clues to their algorithms as well as local shape descriptors [ATR*08, TM10]. A slight step towards taking collection information into account as a whole is taken by [DMW*07] which proposes to repeat pairwise matching between a template shape from the collection and each of the remaining ones, and then to recover all possible pairwise maps through map compositions over the template. A template-based solution is however prone to errors, especially if the collection exhibits a large amount of shape variations. Hence the method in [MTSW12], though in the context of 2D shape matching, proposes to pre-organize 2D shapes in a minimum spanning tree where edges identify two similar shapes to be matched with a pairwise algorithm in isolation from the collection. [BHKH13] extends this idea for 3D shape alignment.

Most of the works on multiple 3D shape correspondence strive to improve a given set of pairwise maps instead of computing them from scratch. One method from this category [Cha09] builds a complete graph connecting all shapes with edges weighted by the matching costs of the initially assumed pairwise maps. The method then computes the shortest path between each shape pair, which implies a map composition to replace the initial map with. Such an approach, however, considers only the pairwise distortions during compositions. An alternative that better exploits the collection information uses the same graph structure but with a different weighting scheme to force an approximate consistency rule such that all 3-cycles of consistent maps return to identity [NBCW*11]. The method weights an edge between two shapes by evaluating the deviations of all the 3-cycles containing the edge from identity, which in turn yields significant improvements, provided that a good seed set of initial maps and plenty of computational time are available. A more recent work [HZG*12], based on a sparse set of initial pairwise maps, employs similar cycle-consistency criteria as [NBCW*11] as well as additional constraints to enforce neighbor-preservation and alignment with the initial maps. The method first creates soft maps from a set of automatically extracted base shapes to all the shapes in the collection by diffusing the initial maps, and then computes a point-to-point map from each base to every shape using a global optimization procedure. These maps are used to create a compact graphical data structure (hub-and-spoke cor-

respondence network) from which globally optimal cycle-consistent maps are extracted using simple graph algorithms. Lastly and most recently, the method proposed in [HG13] formulates the cycle-consistency constraint as the solution to a semidefinite program, and casts the problem of estimating cycle-consistent maps to finding the closest positive semidefinite matrix to an input matrix that stores all the initial maps. Although the resulting method exhibits similar (only slightly better) performance on various shape benchmarks when compared to [HZG*12] and [NBCW*11], it provides theoretical guarantees for correctness of the generated correspondences, which lack in the other works. We note that all the three works mentioned above, [NBCW*11], [HZG*12] and [HG13], use the BIM (Blended Intrinsic Maps) method of [KLF11] to obtain the initial maps upon which the whole shape correspondence process then relies.

The process of multiple shape correspondence can help understanding shape variations in a given collection. Amongst several such efforts, the method in [KLM*12] enables region-based exploration of shapes by establishing fuzzy correspondences over the collection in spectral domain. To achieve spectral embedding, the method requires a sparse set of initial pairwise maps between sample shapes from the collection. The correspondence-less approach of [OLGM12] employs shape descriptors to retrieve the shape that best matches a given deformable template configuration up to rotation and scaling. Both methods employ navigation interfaces to browse shape collections based on similarities and differences. Instead of understanding shape variations, two recent works focus on creating them by generating new models for the population based on part correspondences. The first one [XZCOC12] performs shuffling and mutations whereas the other [KCKK12] learns a probabilistic model that represents the structural variability within the domain. Another group of recent methods address the problem of shape segmentation by incorporating context information from the collection based on consistent part-to-part correspondences [HKG11, WAvK*12].

The major difference of our multiple shape correspondence method from previous work, hence our primary contribution, is that we explicitly minimize the (isometric) distortion over all possible pairs of shapes in a collection while preserving consistency. Previous state of the art efforts [NBCW*11, HZG*12, KLM*12, XZCOC12, HG13] heavily rely on a given initial set of maps between all or some pairs of shapes, and instead of optimizing overall distortion, they rather enforce consistency on the initial set for constant-length cycles or via diffusion. They mostly rely on the BIM method to obtain initial maps, which is however computationally very demanding and sensitive to noise and non-uniform triangulation. On the other hand, our method does not require any initialization, and when compared to these methods, it is computationally much more efficient but more susceptible to symmetric flip errors since we primarily seek for correspondences at a coarse scale between shape extremities. We alleviate this symmetric flip problem by evaluating

coarse maps with denser sampling, hence our method generates also denser correspondences as by-product.

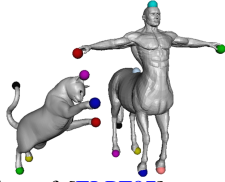
3. Problem Description

Given a shape collection $\mathcal{S} = \{S_1, S_2, \dots, S_M\}$, we aim to compute the optimal set of consistent correspondences, that minimizes the average distortion over all $\binom{M}{2}$ pairwise maps. Let $\psi_{k,l}$ be a bijective map between the points sampled from S_k and S_l , and $\Psi = \{\psi_{k,l} | k < l\}$ denote a set of such $\binom{M}{2}$ pairwise maps. A set Ψ of maps defined over a shape collection \mathcal{S} is said to be *consistent* if and only if all possible map compositions between any two shapes S_k and S_l yield the same map $\psi_{k,l}$. We then define the optimal Ψ^* as the set of consistent maps that minimizes the following overall distortion function:

$$\mathcal{D}(\Psi) = \frac{1}{|\Psi|} \sum_{k < l} D(\psi_{k,l}), \quad (1)$$

where $D(\psi_{k,l})$ is some distortion measure that evaluates the quality of a given correspondence.

We sample from the surface of each shape its geometrical extremities and use them as points to be matched. We find the most prominent N shape extremities by applying the farthest point sampling algorithm of [ELPZ97], starting from the most extreme point of the surface, that is the one which maximizes the average geodesic distance function [HSKK01]. From this point forward, we will use the notation $\psi_{k,l}$ to represent a coarse map between the extremities of two given shapes S_k and S_l .



4. Distortion Measure

We will define the distortion measure D in Eq. 1 based on isometric deviation in a dense evaluation framework.

4.1. Isometric Deviation

Let $D_{\text{iso}}(\psi_{k,l})$ represent the deviation of a map $\psi_{k,l}$ from isometry such that

$$D_{\text{iso}}(\psi_{k,l}) = \frac{1}{|\Psi_{k,l}|} \sum_{(a_k, b_l) \in \Psi_{k,l}} \frac{1}{|\Psi'_{k,l}|} \sum_{(c_k, d_l) \in \Psi'} |g(a_k, c_k) - g(b_l, d_l)| \quad (2)$$

where (a_k, c_k) and (b_l, d_l) are point pairs sampled from S_k and S_l , respectively, $g(\cdot, \cdot)$ is the geodesic distance between two points on a given surface, and $\Psi' = \psi_{k,l} - \{(a_k, b_l)\}$. Scale invariance is achieved by normalizing the function g with respect to the maximum geodesic distance over the surface. This isometric deviation function can also be seen as a variant of the distortion measures used in [BBK06, SY12].

4.2. Dense Evaluation

The isometric deviation measure D_{iso} given in Eq. 2 can be vulnerable to symmetric flips when evaluating the distortion of a coarse map, such as the ones running between shape extremities [SY13]. To increase robustness, we define the distortion measure D used in Eq. 1 in a dense matching framework as described in the sequel.

Given a coarse map $\psi_{k,l}$ between the extremities of two shapes S_k and S_l , we uniformly sample $\hat{N} \gg N$ points on the surface of each shape [ELPZ97], and then compute a dense map $\hat{\psi}_{k,l}$ via the following perfect matching based optimization procedure. We fill a cost matrix \mathbf{C} where each entry c_{ab} represents the cost of matching a dense sample \hat{a}_k from S_k to a dense sample \hat{b}_l from S_l . We build \mathbf{C} by computing each c_{ab} via the inner summation in Eq. 2 based on the given coarse map, hence by setting $\psi' = \psi_{k,l}$. We then perform minimum-weight perfect matching [Kol09] on \mathbf{C} , that reveals the desired dense map $\hat{\psi}_{k,l}$. We use the deviation of this dense map from isometry in the evaluation of the coarse map between the extremities. Hence we set the isometric distortion used in Eq. 1 as

$$D(\psi_{k,l}) = D_{\text{iso}}(\hat{\psi}_{k,l}) \quad (3)$$

This dense matching based evaluation allows us to consider the shapes at a finer scale while evaluating a coarse map, which in turn alleviates the symmetric flip problem that is actually inherent to all isometric shape correspondence methods, especially at coarse resolutions.

We also incorporate a simple strategy that improves the performance of our dense analysis above. Prior to perfect matching, we remove from ψ' possible outliers, i.e., potentially faulty matches. Such faulty matches may exist in ψ' due to possible inconsistencies in extremity sampling, which may eventually degrade the performance of the dense correspondence phase. We eliminate these outliers by excluding the correspondence pairs with high individual isometric distortion (computed via the inner summation term in Eq. 2). More specifically, we arrange the individual distortion values of the correspondence pairs in ψ' into a list with ascending order, detect the location of the first significant jump in distortion values, and then eliminate the matches with distortion larger than the value at this location. We assume that a significant jump occurs where the difference between two consecutive values becomes larger than the sum of the first two distortion differences, i.e., the sum of the difference between the first and the second values, and the difference between the second and the third values in the sorted list.

5. Optimization of the Overall Distortion

Optimization of the overall distortion function given in Eq. 1 is an NP-hard problem. An intuitive approximate solution is to compute a sequence of minimum-distortion maps between pairs of consecutive shapes, $\{\psi_{k,k+1}^*\}$, $k = 1, 2, \dots, M - 1$ (hence assuming an order in the collection \mathcal{S}), which can then be used to generate a consistent set of maps via map compositions. The resulting consistent map set, which is induced by these $M - 1$ individually optimal maps, is not however necessarily optimal on overall since this intuitive *sequential* approach does not leverage the context provided by the whole collection. In this section, we propose a method that also yields an approximate solution, but better exploits the information available in the collection as a whole, and

thus approximates the optimal solution more closely than the sequential approach, as explained in the sequel. We also note that while the optimal map set Ψ^* is independent of the order of the shapes in the collection \mathcal{S} , our method as well as the sequential approach generates order-dependent solutions.

5.1. Formulation - Optimal Path of Maps

For minimization of the overall distortion function, we represent the shape collection with a graph structure where vertices are maps between shapes, and then search for an optimal path on this graph. Let $\Phi_i = \{\phi_i^j\}$ be the set of all possible bijective maps between consecutive shapes S_i and S_{i+1} , such that $\phi_i^j = \psi_{i,i+1}^j$ with $i \in [1, M-1]$ and $j \in [1, N!]$, assuming N points are sampled from each surface. We then cast the problem as optimal path finding on the graph $G = (V, E)$, where $V = \bigcup_i \Phi_i$ and $E = \{(\phi_i^{j_1}, \phi_{i+1}^{j_2})\}$, where $j_1, j_2 \in [1, N!]$, as illustrated in Fig. 1. Hence each of $(M-1)N!$ vertices of this graph is a map ($\phi_i^j \in \Phi_i$) between two consecutive shapes, and its edges completely connect Φ_i and Φ_{i+1} for all i . Let $\pi(m) = \{\phi_1^{j_1}, \phi_2^{j_2}, \dots, \phi_m^{j_m}\}$ define a path of m maps on this graph. Our goal is to find the optimal path $\pi^*(M-1)$ which covers all the shapes in \mathcal{S} , minimizing $\mathcal{D}(\Psi(\pi(M-1)))$, where $\Psi(\pi(M-1))$ denotes the set of all $\binom{M}{2}$ pairwise maps induced by $\pi(M-1)$.

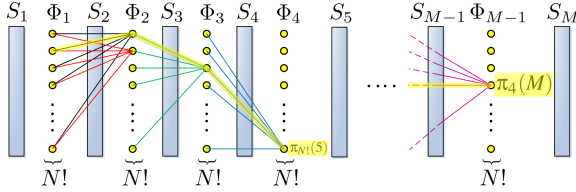


Figure 1: Each vertex $\phi_i^j \in \Phi_i$ (circles) in G is one of the $N!$ possible maps between two consecutive shapes S_i and S_{i+1} (rectangles), whereas edges completely connect Φ_i and Φ_{i+1} (some shown with lines). Two paths are highlighted in yellow where $\pi_{N!}(S)$ is a subpath of $\pi_4(M)$.

5.2. Pruning

Due to $N!$ complexity of the graph structure to be processed, we fix the number of samples as $N = 6$ or 9 , which is most often sufficient to represent a given shape. In order to further prune the set of $N!$ possible maps between N extremities of each consecutive shape pairs, we sort the maps in each Φ_i in ascending order with respect to their deviation from isometry, $\{D_{\text{iso}}(\phi_i^j)\}$ (see also Eq. 2), and discard all but the first R maps to be used in constructing the graph G . By choosing $R \ll N!$, we significantly shrink G with no expected loss of accuracy. In all our experiments we have used $R = 16$ for pruning and have not observed any performance drop compared to using larger values.

5.3. Dynamic Programming Approximation

We now describe our dynamic programming based algorithm that gives an approximate solution for the overall optimal path $\pi^*(M-1)$. Let $\pi_r^*(m)$, $m = 1, 2, \dots, M-1$, denote

the optimal path that minimizes the distortion over the partial shape collection S_1, S_2, \dots, S_{m+1} such that:

$$\pi_r^*(m) = \arg \min_{j_1, j_2, \dots, j_{m-1}} \mathcal{D}(\Psi(\phi_1^{j_1}, \phi_2^{j_2}, \dots, \phi_{m-1}^{j_{m-1}}, \phi_m^r)), \quad (4)$$

where $\Psi(\phi_1^{j_1}, \phi_2^{j_2}, \dots, \phi_{m-1}^{j_{m-1}}, \phi_m^r)$ denotes the set of all $\binom{m}{2}$ maps induced by a path of m maps terminating at ϕ_m^r . Hence $\pi_r^*(m)$ is the optimal path that terminates at ϕ_m^r , with distortion $\mathcal{D}_r^*(m)$. The optimal path $\pi_p^*(m+1)$ for $m < M-1$ can then be approximated for each node p by

$$\pi_p^*(m+1) \simeq \pi_q^*(m) \cup \phi_{m+1}^p \quad (5)$$

with

$$q = \arg \min_r \mathcal{D}_r^*(m) + \sum_{k=1}^{m+1} \mathcal{D}(\psi_{k,m+2}^{r,p}) \quad (6)$$

where $\{\psi_{k,m+2}^{r,p}\}$ are the new $m+1$ maps induced by inclusion of ϕ_{m+1}^p to $\pi_q^*(m)$ via Eq. 5, namely the bijections between extremities of S_k and S_{m+2} for $k = 1, 2, \dots, m+1$. Hence each time a new shape is added the process, the candidate optimal paths and their overall distortions are computed incrementally based on the previously computed optimal paths and distortions, and then those with minimum distortions are picked.

To implement the algorithm, we keep track of R (approximately optimal) paths, $\Pi(m) = \{\pi_1(m), \pi_2(m), \dots, \pi_R(m)\}$, from S_1 to S_{m+1} , which are expanded via Eq. 5 as the algorithm is iterated over m by adding each time a new shape $S_{m+1} \in \mathcal{S}$ to the process. The overall isometric distortions $\mathcal{D}_1(m), \mathcal{D}_2(m), \dots, \mathcal{D}_R(m)$ are also stored and accumulated based on Eq. 6. The set $\Pi(1)$ is initialized trivially with the set Φ_1 of maps between S_1 and S_2 , associated with the isometric distortions $\mathcal{D}(\phi_1^1), \mathcal{D}(\phi_1^2), \dots, \mathcal{D}(\phi_1^{N!})$. An approximately optimal final path is then given by $\pi_q(M-1)$ with $q = \arg \min_r \mathcal{D}_r(M-1)$. The creation of a path $\pi_p(m+1) \in \Pi(m+1)$ based on $\Pi(m)$ is illustrated in Fig. 2, whereas the overall multiple shape correspondence algorithm is given in pseudocode in Fig. 3.

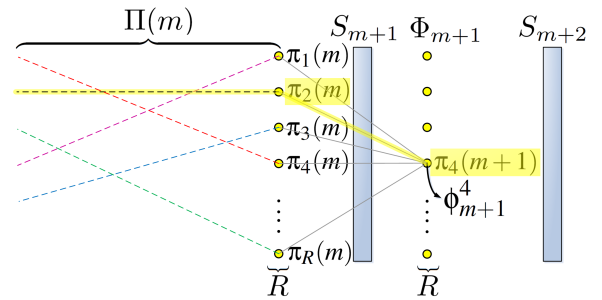


Figure 2: Amongst all R paths of $\Pi(m)$ (dashed lines), $\pi_2(m)$ (black dashed and highlighted) is selected to be the best fit for the current map ϕ_{m+1}^4 , hence creating $\pi_4(m+1) \in \Pi(m+1)$.

The success of our algorithm depends mainly on the validity of the approximation in Eq. 5. This approximation implicitly assumes that the optimal paths from S_1 to S_m given $\{S_1, S_2, \dots, S_{m+1}\}$ are equal or distortion-wise similar to the optimal paths from S_1 to S_m given $\{S_1, S_2, \dots, S_m\}$. Each time a new shape S_{m+1} is added to the process, the algorithm does its best to propagate the correspondence information accumulated at iteration $m - 1$ to iteration m by taking into account the distortions of the new map compositions induced by this addition, and expands the current set of maintained paths $\Pi(m - 1)$ to $\Pi(m)$.

Although it seems difficult to find an exact upper bound for the approximation error in Eq. 5, one can conjecture that it is closely related to how isometric the shapes in a given collection are. It is actually easy to see that, in the case of perfect isometry, the inclusion of a new shape S_{m+1} to the collection will by no means effect the optimal paths from S_1 to S_m . Hence while the approximation error tends to saturate for isometric shape collections as more shapes are added to the process, it can be arbitrarily large in the presence of severe non-isometries. Since we assume in this work that the collection consists of shapes which are (nearly) isometric, we expect our algorithm to perform reasonably well. We also note that, in practice, even for different poses of the same shape, it is usually impossible to have perfect isometry due to imperfections of the modeling process, sampling inconsistencies, and/or geometry discretization errors. As we will later verify by experiments in Section 7, our algorithm closely approximates the optimal solution for $M > 3$ on (nearly) isometric shapes. For $M = 2$, it simply chooses the optimal mapping with minimum distortion, whereas for $M = 3$ we guarantee to achieve the overall optimal $\pi^*(2)$ since the optimal subpath $\pi^*(1)$ is certainly available in $\Pi(1)$ which includes all available paths from S_1 to S_2 .

The naïve brute-force solution to find the exact optimal path that minimizes the overall distortion \mathcal{D} would require $\binom{M}{2}$ evaluations of the isometric distortion function in Eq. 2 for each of the R^M possible paths, and hence is intractable for $M > 8$ on a standard computer. Our method reduces the exponential search space of this brute-force solution by maintaining some good paths each equipped with an accumulated distortion information stored via dynamic programming. We note that the multiple shape correspondence output generated by our algorithm is dependent on the (assumed) order of the shapes in the collection, whereas the brute-force solution, that we will refer to as *exponential* algorithm, is not.

6. Computational Complexity

On a collection of M shapes each having W vertices, sampling of N shape extremities to be matched requires $O(MNW \log W)$ computation. We then construct V of G by listing all $N!$ possible maps between all pairs of consecutive shapes along with their isometric distortion values in

```

Input:  $\mathcal{S} = \{S_1, S_2, \dots, S_M\}$ 
Output: Approximately optimal path of maps over  $\mathcal{S}$ 

Sample  $N$  extremities to be matched on each shape  $S_m$ 

//Construct graph  $G$  by initializing  $\Phi_i$ : Set of all possible
//bijective maps between each consecutive shape pair
For each consecutive shape pair  $(S_i, S_{i+1}) \in \mathcal{S}$ 
  //Map enumeration
   $\Phi_i = \{\phi_i^j\}$  s.t.  $D_{\text{iso}}(\phi_i^1) \leq D_{\text{iso}}(\phi_i^2) \leq \dots \leq D_{\text{iso}}(\phi_i^{N!})$ 
  //Prune  $\Phi_i$  to keep top- $R$  minimum-distortion maps
   $\Phi_i = \{\phi_i^1, \phi_i^2, \dots, \phi_i^R\}$ 
Sample  $\hat{N}$  dense points on each shape  $S_m$  for upcoming
distortion computations

//Dynamic programming
//Create paths of interest on  $G$ 
//by filling  $\Pi(m)$  with paths  $\pi_r(m)$  from  $S_1$  to  $S_{m+1}$ 
//along with their overall distortions  $\mathcal{D}_r(m)$ 
//Initialization
 $\Pi(1) = \{\phi_1^1, \phi_1^2, \dots, \phi_1^R\} // = \Phi_1$ 
 $\{\mathcal{D}_r(1)\} = \{D(\phi_1^1), D(\phi_1^2), \dots, D(\phi_1^R)\}$  for  $r = 1, 2, \dots, R$ 
//Iterations over  $m$ 
For  $m = 1$  to  $M - 2$ 
   $\Pi(m + 1) = \emptyset$ 
  //Create path terminating at  $\phi_{m+1}^p$  for  $p = 1, 2, \dots, R$ 
  //by finding the best fit path for  $\phi_{m+1}^p$ 
   $q = \arg \min_r \mathcal{D}_r^*(m) + \sum_{k=1}^{m+1} D(\psi_{k,m+2}^{r,p}) // \text{Eq. (6)}$ 
  //and expanding the best fit with  $\phi_{m+1}^p$ 
   $\pi_q(m + 1) = \pi_q(m) \cup \phi_{m+1}^p // \text{Eq. (5)}$ 

   $\Pi(m + 1) = \Pi(m + 1) \cup \pi_q(m + 1)$ 
   $\mathcal{D}_p(m + 1) = \mathcal{D}_q^*(m) + \sum_{k=1}^{m+1} D(\psi_{k,m+2}^{q,p})$ 
Return from  $\Pi(M - 1)$  the path with minimum distortion

```

Figure 3: Our multiple shape correspondence algorithm.

$O(MN!N^2)$ time since the isometric distortion measure D is quadratic in the size of the input map. Then we prune each set of $N!$ maps between shape pairs to R in $O(MN! \log N!)$ time due to sorting. As the last step in building G , we densely sample the shapes with \hat{N} points in $O(M\hat{N}W \log W)$ time, followed by their perfect matching in $O(MR\hat{N}^2 \log \hat{N})$ time. Distortions of these dense maps are computed in $O(MR\hat{N}^2)$ time.

For dynamic path computations over G , we append each of R vertices to the best fit amongst all R paths by evaluating isometric distortion D for at most $M - 1$ new map compositions, in $O(M\hat{N}^2)$ time, plus adding the distortion accumulated on the path, for another $O(1)$. These appending operations are performed throughout all M shapes to achieve the output path of length M , costing $O(M^2\hat{N}^2R^2)$ time in total, where $R \ll N!$. The complexity of our coarse multiple correspondence algorithm is hence dominated by the construction time of G , which is $O(M\hat{N}W \log W + MR\hat{N}^2 \log \hat{N})$, yielding fast execution times on a standard computer under the assumption $N \leq 10$ and $\hat{N} \leq 300$.

7. Experimental results

We have tested the performance of our multiple shape correspondence algorithm on high-resolution mesh sequences with non-uniform triangulation and noise. The first dataset is a reconstructed pose sequence of a human actor from the SCAPE benchmark [ASK*05], which contains 71 non-uniformly sampled models, whereas the second one consists of *Noise* and *Shotnoise* classes of the SHREC'11 shape benchmark [BBB*11] with 5 meshes in each. The third set is based on the TOSCA shape benchmark [BBK08] with uniformly-sampled models from various classes *Cat*, *Centaur*, *David*, *Gorilla*, *Horse*, and *Victoria*, each representing the motion of an articulated object with 11, 6, 7, 4, 8, and 12 meshes, respectively. We also create a hybrid class, that we refer to as *Across Humans*, which is a mixture of all 7 males from *David* and the first 7 females from *Victoria*. We add a combination of noise (random amount of displacement in normal direction) and shotnoise (random direction of displacement) to all TOSCA meshes. Specifically, we set the amount of displacement to a random number in $[0, 10\varepsilon]$, where ε is the average edge length in the original mesh. For compactness, we pack the TOSCA results concerning humans and animal into *Within Humans* and *Within Animals* classes, respectively. While the ground-truth correspondences for SCAPE and TOSCA are due to fixed-connectivity meshing, we manually define the ground-truths for the corresponding SHREC'11 classes.

7.1. Evaluation measures

We measure the overall distortion of a multiple correspondence Ψ by the average of all pairwise isometric distortions, hence by $\mathcal{D}(\Psi)$ in Eq. 1. Similarly we use $\mathcal{D}_{\text{grd}}(\Psi)$ to quantify the deviation of Ψ from the ground-truth correspondence:

$$\mathcal{D}_{\text{grd}}(\Psi) = \frac{1}{|\Psi|} \sum_{k < l} \mathcal{D}_{\text{grd}}(\Psi_{k,l}), \quad (7)$$

where

$$\mathcal{D}_{\text{grd}}(\Psi_{k,l}) = \frac{1}{|\hat{\Psi}_{k,l}|} \sum_{(s_k^a, s_l^b) \in \hat{\Psi}_{k,l}} g(\vartheta(s_k^a, s_l^b)) \quad (8)$$

Here $\vartheta(s_k^a)$ stands for the ground-truth correspondence of s_k^a on the other shape, $g(\cdot, \cdot)$ is the geodesic distance function as defined before in Eq. 2, and $\hat{\Psi}_{k,l}$ represents the dense version of the coarse map $\Psi_{k,l}$.

We also define another measure, denoted by $\tilde{\mathcal{D}}(\pi)$, that computes the distortion of a given path π only over its constituent maps:

$$\tilde{\mathcal{D}}(\pi) = \frac{1}{|\pi|} \sum_{\phi_i^j \in \pi} \mathcal{D}(\phi_i^j), \quad (9)$$

and its deviation from ground truth is given by:

$$\tilde{\mathcal{D}}_{\text{grd}}(\pi) = \sum_{\phi_i^j \in \pi} \mathcal{D}_{\text{grd}}(\phi_i^j) \quad (10)$$

The symmetric flip problem that may arise especially due

to sparse sampling is expected to be alleviated with our method due to its dense matching framework. We measure this improvement by the ratio of symmetrically flipped mappings on a path π by $\tilde{\mathcal{F}}$ and over all induced pairwise maps by \mathcal{F} . Note that we use the notation π to represent a map path that may be given as output by the exponential algorithm, the sequential algorithm, or our algorithm.

7.2. Comparison to Baseline Methods

We compare our method with two baseline algorithms that we refer to as *sequential* and *exponential*. Recall that the exponential method computes the theoretical optimal correspondence by a brute-search algorithm, whereas the sequential algorithm finds a sequence of minimum-distortion maps between individual pairs of consecutive shapes, which can then be used to generate a consistent set of maps through map compositions. In addition to the visual results given in Figures 4, 7, and 8, we quantitatively evaluate the correspondence performance via isometric and ground-truth distortion measures and resulting symmetric flip counts (Table 1).

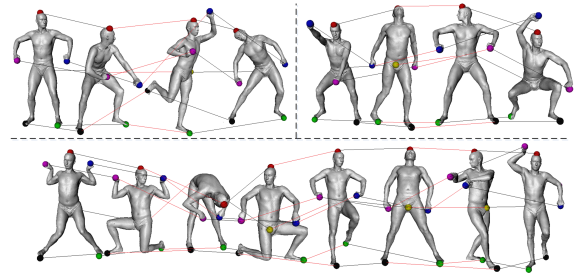


Figure 4: Multiple correspondence obtained by our method on SCAPE dataset (three different subsequences are shown).

In all experiments, we set the parameters of our algorithm to $N = 6$ (number of shape extremities) or $N = 9$ (for *Centaur* only), $\hat{N} = 50$ (number of dense samples), and $R = 16$ (number of enumerated maps), based on the assumptions that the number of representative extremities on a given shape does not usually exceed 9, 50 samples are enough to uniformly cover the shape, and a pairwise mapping outside the top-16 minimum-distortion maps can safely be discarded throughout the whole process. Increased values of these parameters have not led to any performance improvement. We however note that the choice of R is closely related to the number of extremities, N , as well as intrinsic symmetries of the given shape. In principle, the value of R should be larger than the number of possible symmetrically flipped correspondences, and the setting $R = 16$ was sufficient for all the shapes in our database.

We plot in Fig. 5, for varying values of M (number of shapes), the behavior of the isometric and ground-truth distortion measures in three different cases: our method, the sequential method and the exponential algorithm. We compare the distortions obtained over all pairwise mappings as well as over only the maps that constitute the resulting paths. We observe that the path-based isometric distortion $\tilde{\mathcal{D}}$ for the sequential algorithm bounds the other $\tilde{\mathcal{D}}$ plots from below

for all M as expected since it optimizes the path distance, whereas the overall distortion \mathcal{D} of the sequential method is an upper bound for the others with its highest distortion. Similarly, \mathcal{D} for the exponential algorithm bounds the other \mathcal{D} plots from below since it explicitly optimizes the overall distortion. Note that our result closely approximates the exponential result in this case as well as for \mathcal{D}_{grd} and $\tilde{\mathcal{D}}_{\text{grd}}$ measures. When the collection is populated with $M = 8$ shapes, as far as \mathcal{D}_{grd} is concerned, the methods are ordered, from the worst to the best, as the sequential algorithm, our algorithm, and the exponential algorithm, yet our method can work with arbitrary number of shapes whereas the exponential one can handle at most $M = 8$ due to its heavy computational load.

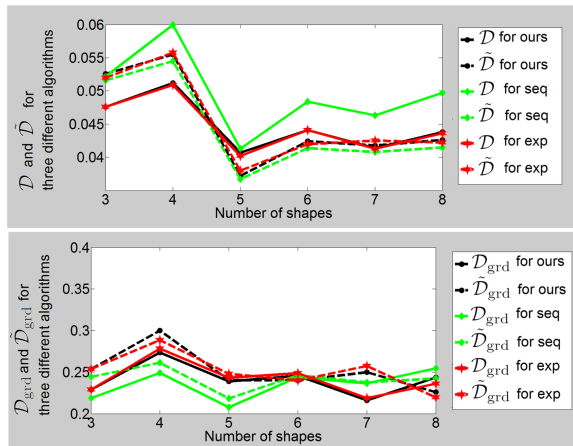


Figure 5: Overall (\mathcal{D} , solid lines) and path-based ($\tilde{\mathcal{D}}$, dashed lines) isometric (top) and ground-truth (bottom) distortions as a function of number of shapes from the SCAPE dataset.

We have also tested the exact optimality of our method, and its superior performance over the sequential method, for collections of size $M = 3$ over 100 random triplets of SCAPE meshes. Our method results in an overall isometric distortion of .05 on average, which is better than the resulting .0526 of the sequential algorithm. The resulting ground-truth distortion and the symmetric flip ratio are also less in the case of our method, being .225 vs. .291, and 102/300 vs. 153/300, respectively, whereas the exponential algorithm replicates our results in this experiment. We observe that symmetric flips are most often inevitable in the case of coarse matching. We however note that our dense evaluation framework significantly alleviates this problem; we have run the same experiments without using dense evaluation, that is, by setting $D(\phi_i^j) = D_{\text{iso}}(\phi_i^j)$ in Eq. 3, and we have obtained the symmetric flip ratios as 142/300 vs. 182/300.

For the quantitative evaluation given in Table 1, we average the results of our algorithm and the sequential algorithm over 5 random orderings of all models of a given class from the corresponding dataset. We provide results for all available datasets (see the beginning of Section 7), including TOSCA classes with noise (w/n). We observe that the

overall isometric distortion \mathcal{D} of our method is always better than that of the sequential method whose path-based distortion $\tilde{\mathcal{D}}$ is always smaller as expected. Consequently, \mathcal{D}_{grd} values always favor our method, and more interestingly, in the case of SCAPE we obtain a slightly better $\tilde{\mathcal{D}}_{\text{grd}}$ value since close isometric distortion values may indeed prefer a flipped mapping over a correct one, which can however be resolved in some cases incorporating additional information from the collection as our dynamic programming algorithm does. We also provide the ratios of the symmetrically flipped maps, \mathcal{F} and $\tilde{\mathcal{F}}$, which basically follow the same pattern as in overall distortions. We see that the symmetric flip ratio \mathcal{F} of our method is considerably smaller especially in the case of SCAPE dataset. For the rest of the cases, our method results in \mathcal{F} and $\tilde{\mathcal{F}}$ values which are still better than the values resulting from the sequential method but not as effective as the SCAPE case, mainly because of the large number of shapes in SCAPE that enables a better integration of context information. As for the performance under noise, comparing the results over TOSCA classes with and without noise, we observe that the performance values deteriorate only very slightly with noise, mainly thanks to the coarse-to-dense framework that we employ.

We examine the sensitivity of our algorithm to the order of the shapes in a given collection in Fig. 6, where the overall distortion \mathcal{D} is plotted as a function of number of shapes, separately for 5 different orderings. We observe that our algorithm exhibits consistent performance in terms of overall distortion, and the variations remain insignificant under different orderings. We see that the overall distortion curve exhibits a similar behavior under different orderings with relatively rapid increase at the beginning due to accumulation of approximation errors (recall that the solution is exactly optimal only for $M \leq 3$), and then tendency to saturate at the same value as the shape collection is sufficiently populated. This is as expected since the SCAPE sequence is an isometric shape collection.

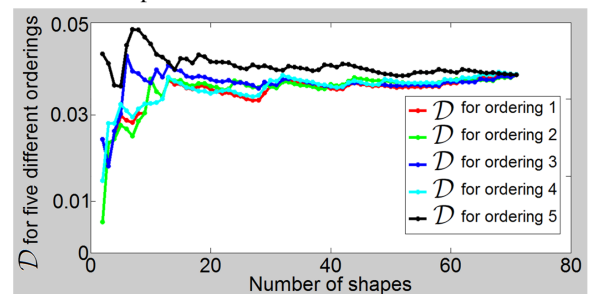
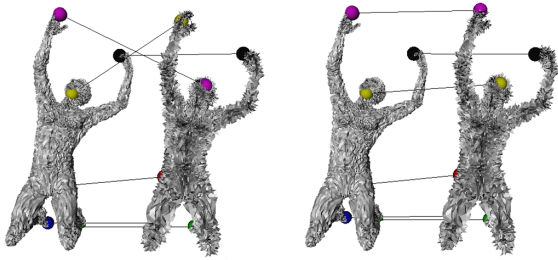


Figure 6: Isometric distortions for five different orderings of the SCAPE dataset as a function of number of shapes.

The execution times of our method on a 2.53GHz PC for SCAPE dataset of $M = 71$ shapes (each with 12.5K vertices) is 96 seconds of graph construction followed by 25 seconds of dynamic programming. *Noise*, *Shotnoise*, and *David* (from TOSCA) classes deal with only $M = 5$ or 7 shapes and consequently take about 40 seconds of graph creation

Table 1: Quantitative evaluation of our multiple shape correspondence method in comparison with sequential method.

Dataset	Our method			Sequential method		
	$\mathcal{D}, \bar{\mathcal{D}}$	$\mathcal{D}_{\text{grd}}, \bar{\mathcal{D}}_{\text{grd}}$	$\mathcal{F}, \bar{\mathcal{F}}$	$\mathcal{D}, \bar{\mathcal{D}}$	$\mathcal{D}_{\text{grd}}, \bar{\mathcal{D}}_{\text{grd}}$	$\mathcal{F}, \bar{\mathcal{F}}$
SCAPE	.0321 , .0404	.228 , .224	1192.9/2556 , 37.7/71	.0756, .0401	.303, .239	1825.1/2556, 40.5/71
Noise (SHREC)	.0202 , .0259	.153 , .166	2.4/10 , 1.2/4	.0200, .0251	.166, .160	3.5/10, 1.1/4
Shotnoise (SHREC)	.0211 , .0350	.162 , .165	2.7/10 , 1.2/4	.0207, .0291	.176, .164	3.5/10, 1.2/4
Within Humans (w/n)	.0430 , .0473	.133 , .147	33.9/87 , 5.9/17	.0496, .0466	.179, .146	47.1/87, 7.3/17
Across Humans (w/n)	.0447 , .0515	.213 , .292	40.9/91 , 6.2/13	.0519, .0498	.299, .232	63.1/91, 8.3/13
Within Animals (w/n)	.0475 , .0467	.195 , .188	26.5/55 , 5.9/10	.0601, .0451	.250, .185	42.1/55, 6.3/10
Within Humans	.0422 , .0469	.131 , .157	33.1/87 , 6.2/17	.0449, .0460	.161, .144	45.4/87, 7.1/17
Across Humans	.0431 , .0501	.197 , .182	37.2/91 , 6.1/13	.0489, .0469	.233, .170	50.8/91, 5.4/13
Within Animals	.0471 , .0460	.185 , .161	30.5/55 , 6.4/10	.0508, .0447	.205, .160	40.1/55, 6.2/10

**Figure 7:** Resulting pairwise map of HAS method [HZG*12] (left) and our method (right) on Noise dataset.

followed by 0.4 seconds of dynamic programming, despite their large number of vertices that is around 52K. The *Across Humans* class doubles this last execution time with $M = 14$ shapes in the collection. The sequential method yields very similar execution times.

7.3. Comparison to HAS Method

Visual comparisons of our method and the HAS (hub-and-spoke) method of [HZG*12] are available through Figures 7 and 8. In Table 2, we also provide quantitative evaluation of our matching algorithm, in comparison to the HAS method (averaged over 5 different orderings as in Section 7.2). For comparison, we have run the HAS code made publicly available by the authors of [HZG*12].

We first note that the HAS method produces many-to-one maps, hence boosting the distortion performance but yielding clustered correspondences. Our method, on the other hand, generates bijections between shapes, which is more intuitive in the case of complete isometric shape correspondence. For fair comparisons, HAS method is evaluated based on the samples that are closest to the 50 dense samples used by our algorithm, hence performance of both methods are based on the whole shape, not just the extremities. Since it is hard to infer the symmetric flip situation of the HAS results, e.g., uncategorized head \leftrightarrow arm or hand \leftrightarrow head matches in the middle row of Fig. 8, we avoid symmetric flip count

comparisons in this case. Similarly, the isometric distortion measure that we explicitly minimize does not carry important information in this setting.

We first observe that, in the case of perfect mesh models without any non-uniformities and noise, such as in TOSCA classes, the HAS method significantly outperforms our method, while for the rest of the cases the results clearly favor our method over HAS. The success of HAS method depends highly on the number and quality of the initial full dense maps provided to the system. Their default initialization choice is the BIM method [KLF11], which works well for genus zero meshes with uniform triangulation and without noise. In the case of SCAPE dataset, HAS method suffers both from the number and quality requirements of the initial maps as we initialize it with the BIM maps computed over only 70 pairs of consecutive shapes due to the relatively high population of this dataset. Quality of these maps is also not so perfect due to the non-uniform triangulations. Since the other collections in Table 2 contain considerably less number of shapes, in these cases the HAS method can be initialized with the BIM maps computed over all possible pairs. Some of these maps, however, are of low-quality as can be seen by the top row of Fig. 8.

The multiple correspondences that we generate are always perfectly consistent unlike the HAS outputs containing cycle-consistency distortions. We have to also note that the tolerance of the HAS method is higher to possible non-isometries in the collection while our method is designed rather for nearly isometric shape sequences.

An important advantage of our method, when compared to state-of-the-art methods in the literature, is that it does not require any "good" initialization of pairwise matchings. For example, computing a multiple correspondence between 50 points with the HAS method takes slightly more than an hour on the SCAPE sequence, assuming that the method is initialized with 70 dense maps, all computed using the BIM method between consecutive shape pairs. Our algorithm, on the other hand, performs the same task in about 2 minutes. We have to however note that with almost the same computation time, the HAS method can readily deliver full dense

maps much beyond $\hat{N} = 50$ as well while the resolution of our results is limited by $O(\hat{N}^2 \log \hat{N})$ complexity due to the perfect matching step.

We finally note that there is a fine line between coarse and dense analysis in the success of a matching algorithm, especially in the noisy environments. BIM and HAS methods, tailored to match shapes at high resolution, hence considering all vertices, are likely to result in poor performance under noise and non-uniformities. On the other hand, seeking for correspondences at a coarse scale, as our extremity matching case suggests, helps improve resiliency against noise but may in turn increase occurrence rate of symmetrically flipped matches. We address this symmetrical flip issue by considering $\hat{N} = 50$ samples spread evenly over the shape surface for evaluation of coarse maps and thereby generation of denser correspondences (Section 4.2). In Fig. 9, we display an example of the denser correspondence that our algorithm generates as by product, where full dense maps between models are interpolated based on the available $\hat{N} = 50$ pairs as described in [KLF11].

Table 2: Quantitative evaluation of our multiple shape correspondence method in comparison with HAS [HZG* 12].

	Our method	HAS
Dataset	$\mathcal{D}_{\text{grd}}, \mathcal{D}_{\text{grd}}$	$\mathcal{D}_{\text{grd}}, \mathcal{D}_{\text{grd}}$
SCAPE	.109, .105	.153, .169
Noise (SHREC)	.033, .042	.120, .131
Shotnoise (SHREC)	.029, .040	.124, .113
Within Humans (w/n)	.039, .051	.103, .123
Across Humans (w/n)	.113, .176	.242, .287
Within Animals (w/n)	.077, .082	.159, .167
Within Humans	.041, .059	.012, .032
Across Humans	.063, .077	.020, .033
Within Animals	.057, .072	.010, .025

8. Conclusion

We have presented a multiple shape correspondence method that computes consistent mappings between all shape pairs of a given collection without using any initial correspondence information. The experiments conducted show that our dynamic programming based algorithm closely approximates the theoretical optimal solution that minimizes the overall isometric distortion, and performs better than the intuitive alternative that composes maps based on a sequence of minimum-distortion maps computed between pairs of consecutive shapes without leveraging any context information from the collection. Another finding of this work is that, by considering the shape collection as a whole and by preserving consistency, it is possible to obtain distortion-wise better correspondences than one would obtain by pairwise matching, not only over the totality of a given collection of (nearly) isometric shapes but in some cases even over individual shape pairs, hence a proof of concept for the multiple shape correspondence idea.

Our method should be preferable to its competitors available in the literature when the problem is to find consistent bijections over all pairs of an isometric shape collection in

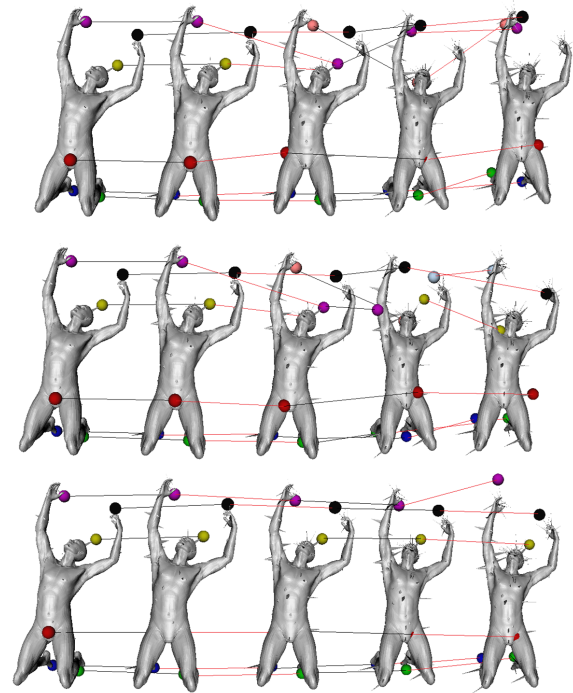


Figure 8: Extremity matches over the sequence of meshes in the Shotnoise dataset, generated by BIM method of [KLF11] (top), HAS method of [HZG* 12], and our method (bottom).



Figure 9: Multiple correspondence generated by our method on a Horse subsequence for $\hat{N} = 50$ samples (spheres), interpolated and visualized as full dense maps.

a computationally very efficient way. We have also demonstrated the benefit of our method over non-uniformly sampled and/or noisy shape collections. One key component of our method is the coarse-to-dense matching framework that improves its resiliency against noise and alleviates the symmetrical flip problem inherent to coarse matching. Our experiments show that, for isometric shape collections with non-uniform triangulation and noise, our method can compute relatively dense correspondences reasonably fast and outperform state of the art in terms of accuracy.

Nevertheless, symmetric flips are still observed in the results, which are often inevitable with any method in the literature. We note that symmetric flips are not actually much of a problem in some applications such as statistical shape analysis and shape recognition but may cause complications in other cases such as shape morphing and registration. One

possible remedy to resolve the symmetric flip problem here would be to incorporate explicit symmetry information by resorting to methods that can detect global intrinsic symmetries such as [KLCF10, OSG07]. Another limitation of our method is that it works only for collections of (nearly) isometric shapes with distinct extremities, and fails in the case of arbitrarily deforming objects, such as cloths and faces, exhibiting severe non-isometries and/or changing extremities.

The key idea in our multiple shape correspondence scheme is explicit minimization of the overall isometric distortion function, rather than improving a given set of initial pairwise maps. Our future work will involve further exploitation of this idea to deal with collections that contain shapes which are partially isometric and/or which exhibit non-isometries.

Acknowledgement

This work has been supported by TUBITAK under the project EEEAG-109E274.

References

- [ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: Shape completion and animation of people. *ACM Trans. Graph.* 24, 3 (2005), 408–416. 6
- [ATR*08] AHMED N., THEOBALT C., ROESSL C., THRUN S., SEIDEL H. P.: Dense correspondence finding for parametrization-free animation reconstruction from video. *Proc. Computer Vision and Pattern Recognition (CVPR)* (2008). 1, 2
- [BA03] B. ALLEN B. CURLESS Z. P.: The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.* 22, 3 (2003), 587–594. 1
- [BBB*11] BOYER E., BRONSTEIN A., BRONSTEIN M., BUSTOS B., ET AL.: SHREC 2011: Robust feature detection and description benchmark. *Eurographics Workshop on 3D Object Retrieval* (2011). 6
- [BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scient. Comput.* 28, 5 (2006), 1812–1836. 2, 3
- [BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008. 1, 6
- [BHKH13] BUDD C., HUANG P., KLAUDINY M., HILTON A.: Global non-rigid alignment of surface sequences. *Int. Journal of Computer Vision* 102, 1-3 (2013), 256–270. 2
- [Cha09] CHARPIAT G.: Learning shape metrics based on deformations and transport. In *Proc. NORDIA* (2009). 2
- [DMW*07] DALAL P., MUNSELL B. C., WANG S., TANG J., OLIVER K.: A fast 3d correspondence method for statistical shape modeling. *Proc. Computer Vision and Pattern Recognition (CVPR)* (2007). 2
- [ELPZ97] ELДАР Y., LINDENBAUM M., PORAT M., ZEEVI Y.: The farthest point strategy for progressive image sampling. *IEEE Trans. Image Processing* 6 (1997), 1305–1315. 3
- [HG13] HUANG Q., GUIBAS L.: Consistent shape maps via semidefinite programming. *Computer Graphics Forum (Proc. SGP)* 32, 5 (2013), 177–186. 2
- [HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint shape segmentation with linear programming. *SIGGRAPHAsia* (2011). 2
- [HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T.: Topology matching for fully automatic similarity estimation of 3d shapes. *Proc. SIGGRAPH* (2001). 3
- [HZG*12] HUANG Q., ZHANG G., GAO L., HU S., BUSTCHER A., GUIBAS L.: An optimization approach for extracting and encoding consistent maps in a shape collection. *Proc. SIGGRAPH Asia* 31, 6 (2012). 2, 8, 9
- [KCKK12] KALOGERAKIS E., CHAUDHURI S., KOLLER D., KOLTUN V.: A probabilistic model of component-based shape synthesis. *Proc. SIGGRAPH Asia* 31, 4 (2012). 2
- [KLCF10] KIM V. G., LIPMAN Y., CHEN X., FUNKHOUSER T.: Möbius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum (Symposium on Geometry Processing)* 29, 5 (2010). 10
- [KLF11] KIM V., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *Proc. SIGGRAPH* 30, 4 (2011). 2, 8, 9
- [KLM*12] KIM V., LI W., MITRA N., DIVERDI S., FUNKHOUSER T.: Exploring collections of 3d models using fuzzy correspondences. *Proc. SIGGRAPH* (2012). 2
- [Kol09] KOLMOGOROV V.: Blossom V: A new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation (MPC)* (2009), 43–67. 3
- [LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM Trans. Graph.* 28, 3 (2009). 2
- [MTSW12] MUNSELL B. C., TEMLYAKOV A., STYNER M., WANG S.: Pre-organizing shape instances for landmark-based shape correspondence. *Int. Journal of Computer Vision (IJCV)* 97, 2 (2012), 210–228. 2
- [NBCW*11] NGUYEN A., BEN-CHEN M., WELNICKA K., YE Y., GUIBAS L.: An optimization approach to improving collections of shape maps. *Computer Graphics Forum (Proc. SGP)* 30, 5 (2011), 1481–1491. 2
- [OLGM12] OVSJANIKOV M., LI W., GUIBAS L., MITRA N.: Exploration of continuous variability in collections of 3d shapes. *Proc. SIGGRAPH* (2012). 2
- [OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. *Computer Graphics Forum* 29, 5 (2010), 1555–1564. 2
- [OSG07] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. *Computer Graphics Forum* 27, 5 (2007), 1341–1348. 10
- [SY12] SAHILLIOĞLU Y., YEMEZ Y.: Minimum-distortion isometric shape correspondence using EM algorithm. *IEEE Trans. PAMI* 34, 11 (2012), 2203–2215. 2, 3
- [SY13] SAHILLIOĞLU Y., YEMEZ Y.: Coarse-to-fine isometric shape correspondence by tracking symmetric flips. *Computer Graphics Forum* 32, 1 (2013), 177–189. 3
- [TM10] TUNG T., MATSUYAMA T.: Dynamic surface matching by geodesic mapping for 3D animation transfer. *Proc. Computer Vision and Pattern Recognition (CVPR)* (2010), 1402–1409. 2
- [vKZHC011] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707. 2
- [WavK*12] WANG Y., ASAFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B.: Active co-analysis of a set of shapes. *Proc. SIGGRAPH Asia* 31, 6 (2012). 2
- [WH10] WARD A., HAMARNEH B.: The groupwise medial axis transform for fuzzy skeletonization and pruning. *IEEE Trans. PAMI* 32, 6 (2010), 1084–1096. 1
- [XZCOC12] XU K., ZHANG H., COHEN-OR D., CHEN B.: Fit and diverse: Set evolution for inspiring 3d shape galleries. *Proc. SIGGRAPH* (2012). 2