

Coarse-to-Fine Combinatorial Matching for Dense Isometric Shape Correspondence

Y. Sahilliođlu and Y. Yemez

Department of Computer Engineering, Koç University, Istanbul, Turkey

Abstract

We present a dense correspondence method for isometric shapes, which is accurate yet computationally efficient. We minimize the isometric distortion directly in the 3D Euclidean space, i.e., in the domain where isometry is originally defined, by using a coarse-to-fine sampling and combinatorial matching algorithm. Our method does not require any initialization and aims to find an accurate solution in the minimum-distortion sense for perfectly isometric shapes. We demonstrate the performance of our method on various isometric (or nearly isometric) pairs of shapes.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: 3D Shape Correspondence—Coarse-to-fine sampling, combinatorial matching, isometric dense shape correspondence

1. Introduction

Finding correspondences between shapes is a fundamental problem in both computer vision and graphics with numerous applications such as mesh morphing, mesh parametrization, deformation transfer, shape registration, shape matching, analysis of sequential meshes and statistical shape modeling [vKZHC011]. In this paper we address the problem of establishing dense correspondence between isometric (or nearly isometric) shapes. Isometric shape correspondence is an important problem; not only since most real world deformations are isometric, but also because semantically similar shapes have similar metric structures, such as different poses of an articulated object, models of a mesh sequence representing the motion of a human actor, or two shapes representing different but semantically similar objects (e.g., two different humans or animals).

If two shapes are perfectly isometric, then there exists an isometry, i.e., a distance-preserving mapping, between these shapes such that the geodesic distance between any two points on one shape is exactly the same as the geodesic distance between their correspondences on the other. However, two digital shapes are hardly ever perfectly isometric, even for different poses of a rigid object, due to imperfections of the modeling process and/or geometry discretization errors. Hence the goal of isometric correspondence methods existing in the literature is rather to find a mapping that minimizes the amount of deviation from isometry.

When the shapes to be matched are given in the form of triangle meshes, the correspondence problem can be formulated as combinatorial optimization over all possible mappings from one vertex set to the other. There exist various ways of solving this optimization problem in the literature. One way is to minimize the deviation from isometry, indirectly in some embedding space, where Euclidean distances approximate geodesic distances. Euclidean embedding, in the context of shape analysis, can be achieved by using different techniques such as classical MDS (Multi-Dimensional Scaling) [JZ06], least-squares MDS [EK03], and spectral analysis of the graph Laplacian [MHK*08] or of the Laplace-Beltrami operator [OSG07]. Some recent methods propose to use embeddings such as the heat kernel [OMMG10] and the canonical diffusion embedding [SH10], which are invariant under isometric deformations, but which are not Euclidean in the multidimensional scaling sense. Other two recent examples are based on the Möbius transformation which is used for conformal embedding of the given shapes into a canonical coordinate frame on the complex plane where deviations from isometry are approximated based on mutually closest points [LF09], [ZWW*10]. A problem common to these embedding-based techniques is that they all produce approximate and/or ambiguous solutions since they can measure deviations from isometry only approximately in the embedding space. In order to decrease the approximation error, Bronstein et al. [BBK06b] propose

to embed one of the shapes to be matched into the surface of the other via the generalized MDS, which however requires minimization of a non-convex stress function that is difficult and expensive to optimize. Similarly, in [SY10], the isometric distortion is minimized via greedy optimization, which hence needs a good initialization and may otherwise get stuck with local maxima, yielding suboptimal solutions. Bronstein et al. propose a coarse-to-fine method in [BBK06a], to remedy the local convergence problem involved in optimization of the stress function, whereas Mé-moli and Sapiro [MS04] present a technique to compute a discrete approximation to Gromov-Hausdorff distance for comparing isometric point clouds, both methods coupled with a farthest point sampling procedure.

Isometric shape correspondence methods may also exploit local shape similarity in addition to isometric clues, such as curvature information as in [OMMG10], [ZWW*10]. Some methods may even solely rely on local shape similarity information, using descriptors such as spherical harmonics as in [FS06], histogram of oriented gradients as in [ZBVH09], and mean curvature as in [TBW*09]. Local shape similarity is an important clue for shape correspondence, especially in the case of non-isometric deformations, but otherwise it is considered as less reliable than global shape information such as isometry.

An important distinction between shape correspondence methods is whether they target sparse or dense correspondence. Most embedding-based methods naturally support dense correspondence but the computational load is usually a limiting factor. There also exist methods which primarily aim to find a small number of feature correspondences [ZSCO*08], [HAWG08], [SH10], [ACOT*10], [LF09], [KLF11]. These methods regard the sparse correspondence problem as the main challenge since, based on a sufficient number of reliable landmarks, cross-parametrization or some other form of interpolation technique can always be applied to obtain a dense correspondence [Ale02]. However, to decide on the degree of sparsity, that would lead to a robust dense correspondence, is always a problematic issue. A particular example to sparse correspondence methods is the deformation-driven approach of [ZSCO*08]. In this method, an optimal correspondence is sought between shape extremities via combinatorial tree traversal by pruning the search space according to some criteria based on local shape similarity and geodesic consistency. For each candidate correspondence set, the source shape is deformed to the target based on these small number of landmarks (anchor points), and the correspondence with the smallest distortion gives the best matching. Another similar deformation-driven method is presented also in [HAWG08]. Both methods can handle large deformations but their computational cost is extremely high due to the repeated deformation process that they involve.

There are two main contributions of this paper:

- We propose a dense shape correspondence method which is computationally efficient. We minimize the isometric distortion directly in the 3D Euclidean space, i.e., in the domain where isometry is originally defined, by using a coarse-to-fine combinatorial search algorithm. Our method does not require any initialization and aims to find an accurate solution in the minimum-distortion sense for perfectly isometric shapes.
- Our shape correspondence method is based upon a coarse-to-fine joint sampling technique that incrementally samples evenly-distributed salient vertices from a given mesh at increasing levels of detail.

We note that the source code and the executables for the method that we present in this paper are publicly available in <http://home.ku.edu.tr/~yyemez/c2f>.

2. Problem Statement and Overview

Our goal is to establish a dense correspondence between two given isometric (or nearly isometric) shapes. We assume that each shape is represented by a manifold surface mesh of sufficiently high resolution, on which geodesic distances can easily be computed. We designate one of the shapes as *source* and the other as *target*. Let S and T denote the vertex sets of the source and the target meshes, respectively. Let also that a mapping $\xi : S \rightarrow T$ (or a relation in the most general setting) is given. We then measure the isometric distortion D_{iso} as follows:

$$D_{\text{iso}}(\xi) = \frac{1}{|\xi|} \sum_{(s_i, t_j) \in \xi} d_{\text{iso}}(s_i, t_j) \quad (1)$$

where $d_{\text{iso}}(s_i, t_j)$ is the contribution of the individual correspondence (s_i, t_j) to the overall isometric distortion:

$$d_{\text{iso}}(s_i, t_j) = \frac{1}{|\xi'|} \sum_{(s_l, t_m) \in \xi'} |g(s_i, s_l) - g(t_j, t_m)| \quad (2)$$

where $g(\cdot, \cdot)$ is the geodesic distance between two vertices on a given surface and $\xi' = \xi - \{(s_i, t_j)\}$ in the most general setting. Both d_{iso} and D_{iso} take values in the interval $[0, 1]$ since the function g is normalized with respect to the maximum geodesic distance over the surface. Note that (1) can be seen as a variant of the generalized stress function defined in [BBK06b]. The problem can then be formulated as a combinatorial search over all possible mappings so as to minimize the isometric distortion function given in (1):

$$\xi^* = \arg \min_{\xi} D_{\text{iso}}(\xi) \quad (3)$$

Since S and T consist of vertices sampled on isometric (or nearly isometric) shapes, we require the optimal mapping ξ^* to establish a full correspondence between S and T so that every $s_i \in S$ is related to some $t_j \in T$, and likewise every $t_j \in T$ is related to some $s_i \in S$. If we further assume that the shapes are perfectly isometric and represented by uniform triangle meshes, then ξ^* is constrained to be a

one-to-one correspondence with $|S| = |T|$, and can be found by evaluating (1) for $N!$ different possible mappings, where $N = |S| = |T|$. This combinatorial solution however has factorial complexity and hence computationally intractable for typical values of N .

Motivated by the simplicity of this optimal but expensive solution, we provide a coarse-to-fine shape matching algorithm based on combinatorial search. The idea is to reduce the search space by exploiting the fact that the optimal mapping ξ^* maps nearby vertices on the source shape to nearby vertices on the target. This suggests that shape matching can be performed on a patch-by-patch basis in a coarse-to-fine fashion. To this effect, we incrementally sample evenly-distributed vertices from the surfaces of both shapes at increasing levels of detail (Section 3). In parallel to this sampling process, at each level of detail, we match the sampled vertices by combinatorial search on a patch-by-patch basis (Section 4), using Equations 1 and 3. Our algorithm is built upon the basic assumption that the shapes to be matched are perfectly isometric. However two digital shapes (or their mesh representations) are hardly ever so in practice. Moreover $|S|$ is not usually equal to $|T|$, hence the correspondence ξ^* , which is optimal in the minimum-distortion sense, is a many-to-many mapping. We note that the final output of our correspondence algorithm is also a many-to-many mapping.

3. Coarse-to-fine Sampling

In this section, we describe our coarse-to-fine algorithm that we use to sample *base vertices* from the mesh representations of the given shapes. We sample S and T separately, at increasing levels of detail, by using a sequence of decreasing *sampling radii* $\{r^{(k)}\}$ such that $r^{(k)} > r^{(k+1)}$ for $k = 0, 1, \dots, K$. Let $S^{(k)}$ denote the set of base vertices sampled from S at level k . We require the sets $\{S^{(k)}\}$: i) to be incremental such that $S^{(k)} \subset S^{(k+1)} \subseteq S$, and ii) to consist of samples as evenly distributed as possible on the shape surface. Note that the same notation applies also to T .

In order to obtain a consistent joint sampling between two shapes, we use an importance sampling strategy which prioritizes salient vertices (shape extremities and/or high curvature points) during selection of the samples. We initially sort the vertices of S into a list in descending order with respect to their saliency values and mark them all as base vertex candidates. The sampling algorithm starts, at the coarsest level $k = 0$, by selecting the top vertex of the list as the first base vertex. We launch the Dijkstra's shortest paths algorithm from this vertex and unmark all the vertices lying within a distance $r^{(0)}$. Then, the next base vertex is picked as the first marked vertex of the list. When this is repeated until no marked vertex is left, we obtain a sampling of the surface, where the base vertices are at least $r^{(0)}$ apart from each other. The same sampling procedure is applied to each level k by initializing the sampling with $S^{(k)} = S^{(k-1)}$, where

$S^{(-1)} = \emptyset$. At the beginning of each level k , we mark all the vertices but the ones which are within a distance $r^{(k)}$ from the base vertices inherited from the previous level. We then pick the first marked vertex in the list and unmark all the vertices within a distance $r^{(k)}$. When this is repeated until no marked vertex is left, we obtain the base vertex set $S^{(k)}$ such that the samples are at least $r^{(k)}$ apart from each other. The maximum level of sampling is obtained when all the vertices in S are picked as base vertices.

Each base vertex $s_i \in S^{(k)}$, sampled as described above, defines a surface patch around itself at level k . This patch, that we denote by $S_i^{(k)}$, includes all the vertices within a distance $r^{(k)}$ from the base s_i , as illustrated in Fig.1. The algorithm also ensures that each vertex is covered by the patch of at least one base vertex. Hence, the resulting patches partition the shape surface at each level into overlapping regions of approximately equal size such that $S^{(k)} = \bigcup_i S_i^{(k)}$. The sampling algorithm is given in pseudocode below.

Sampling algorithm:

Input: Vertex set S
Input: Sampling radii $\{r^{(k)}\}$ s.t. $r^{(0)} > r^{(1)} > \dots > r^{(K)}$
Output: Base vertex sets $\{S^{(k)}\}$ s.t. $S^{(0)} \subset S^{(1)} \subset \dots \subset S^{(K)}$
 $S^{(-1)} = \emptyset$; $k = -1$;
Sort S in descending order w.r.t. saliency;
Iterate on level k
 Mark all vertices in S as base vertex candidates;
 $S^{(k)} = S^{(k-1)}$;
 For each base $s_i \in S^{(k-1)}$
 Unmark all the vertices within patch $S_i^{(k)}$;
 Repeat
 Let s_j be the first marked vertex in S ;
 $S^{(k)} = S^{(k)} \cup \{s_j\}$;
 Unmark all the vertices within patch $S_j^{(k)}$;
 Until no marked vertex is left
Until all vertices in S are sampled (or maximum level K is achieved)

Our sampling algorithm is built on a sampling technique used in [HSKK01]. As described above, we extend this technique, which is uniresolution and which selects the base vertices evenly but in an arbitrary manner, so as to have a multiresolution sampling algorithm which prioritizes the selection of salient vertices. We measure saliency by the integral of geodesic distance function of [HSKK01] at the initial coarsest level, and by the Gauss curvature at all other levels. Hence the base vertices are placed on local maxima of Gauss curvature at every level except that they are selected from the shape extremities at the initial level.

4. Correspondence Algorithm

We establish the vertex correspondence between the source shape S and the target T , at increasing levels of detail, in

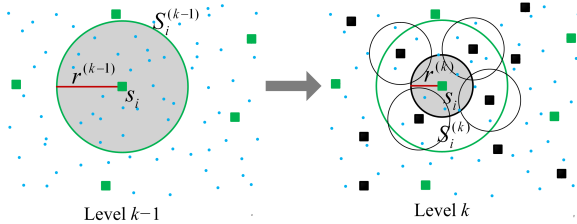


Figure 1: Coarse-to-fine sampling. Small blue dots represent the original vertex set S . (Left) Green points are the base vertices sampled at level $k-1$, where the base s_i defines the patch $S_i^{(k-1)}$ with radius $r^{(k-1)}$. (Right) The newly sampled base vertices at level k (in black), along with the ones inherited from the previous level (in green), constitute the base vertex set $S^{(k)}$ at level k , where each s_j defines a patch $S_j^{(k)}$ around itself with radius $r^{(k)}$. The patch $S_i^{(k-1)}$ is the parent of the base vertices that it covers at level k .

parallel to our coarse-to-fine sampling algorithm described in the previous section. Hence our shape correspondence algorithm produces a sequence of correspondences, $\{\xi^{(k)}\}$, which is refined as the level k increases, and at the maximum level possible, we expect to get the optimal dense correspondence: $\xi^* = \xi^{(K)}$. In the sequel, we explain how we obtain each $\xi^{(k)}$, i.e., the correspondence at each level k .

4.1. Patch-based Combinatorial Matching

Recall from Section 3 that each base vertex set $S^{(k-1)}$, or equivalently $T^{(k-1)}$, partitions the surface of the shape S (or T) into overlapping patches of approximately equal size $\{S_i^{(k-1)}\}$ (or $\{T_i^{(k-1)}\}$). Suppose that each such patch at level $k-1$, say $S_i^{(k-1)}$, contains a fixed number M of base vertices from the next level k , i.e., from $S^{(k)}$. If we designate $S_i^{(k-1)}$ as the parent of these k^{th} level base vertices, then our coarse-to-fine sampling procedure can be thought of as a process in which the surface of the shape S is recursively subdivided into M smaller patches as levels proceed. Note that, while a newly sampled base at level k (or its patch) may have multiple parents, a base vertex inherited from level $k-1$ has only one parent patch.

Given $\xi^{(k-1)}$, the correspondence $\xi^{(k)}$ can be obtained on a patch-by-patch basis, i.e., for every $(s_i, t_j) \in \xi^{(k-1)}$, by matching the k^{th} level base vertices inside the patches $S_i^{(k-1)}$ and $T_j^{(k-1)}$, respectively, as illustrated in Fig. 2. The M base vertices of two corresponding patches, say $S_i^{(k-1)}$ and $T_j^{(k-1)}$, can be matched by combinatorial search provided that M is sufficiently small. To this effect, we evaluate each time the isometric distortions of $M!$ possible one-to-one mappings ξ from the base vertices in $S_i^{(k-1)}$ to the base vertices in $T_j^{(k-1)}$ via Eq. 1 and pick the one with the least distortion. We denote this minimum-distortion patch-to-patch

matching by $\xi_m^{(k)}$. Recall that, to be able to compute $D_{\text{iso}}(\xi)$ in Eq. 1, we need to set a correspondence list ξ' to be used in Eq. 2. The list ξ' , which includes ξ by default, should be global enough to avoid mismatches due to local symmetries, yet local enough to perform well on details, hence we augment it with the matchings of the parent patches, whenever available: $\xi' = \xi \cup \cup_n \xi_n^{(k-1)}$, where each $\xi_n^{(k-1)}$ denotes the correspondence at level $k-1$ between the parent patches of s_i and t_j (note that a base vertex may have multiple parents).

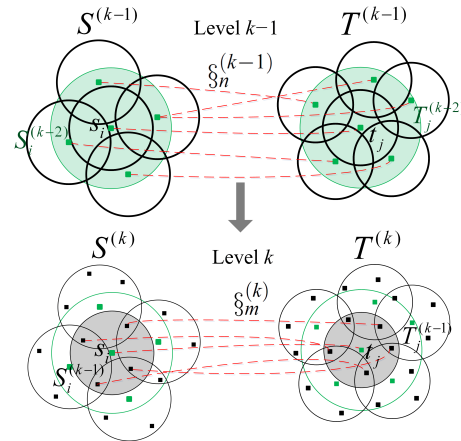


Figure 2: Coarse-to-fine patch-based combinatorial matching. At level k , the base vertices inside the (grey) patches $S_i^{(k-1)}$ and $T_j^{(k-1)}$ are matched by combinatorial search. At level $k-1$, the matching $\xi_n^{(k-1)}$ is one-to-many since the (green) patches $S_i^{(k-2)}$ and $T_j^{(k-2)}$ have different number of bases. Note that $\xi_n^{(k-1)}$ is included in ξ' while matching the patches $S_i^{(k-1)}$ and $T_j^{(k-1)}$ at level k .

At each level k , we set the value of the sampling radius $r^{(k)}$ based on the area of the largest patch at level $k-1$, denoted by $A^{(k-1)}$. We use the ad-hoc formula, $r^{(k)} = 0.6\sqrt{A^{(k-1)}/\pi}$, which ensures that the number M of bases sampled within a parent patch is about 5 (or more generally, $M \leq 6$), and that in turn allows us to match the resulting patches via combinatorial search.

Since two shapes are never perfectly isometric, two patches to be matched may indeed have different number of base vertices sampled, say M and M' and $M' > M$, as it is also the case in in Fig. 2. In such cases we have $\binom{M'}{M}M!$ different possible one-to-one mappings to evaluate for combinatorial matching. Once the minimum-distortion one-to-one mapping is found via combinatorial search, there remain unmatched vertices in one of the patches, to which we assign residual matches in order to assure that the whole surface is covered by the correspondence algorithm as depicted in Fig. 2. To achieve this, each unmatched vertex in one patch is paired with all the vertices in the other one by one, and the pair that minimizes the isometric distortion is picked as

the residual match. Hence each patch-to-patch matching at level k results in a mapping $\xi_m^{(k)}$, which can be one-to-one, many-to-one or one-to-many, depending on the number of base vertices sampled in each patch.

4.2. Correspondence Merging

At the end of each level k , we merge the patch-to-patch correspondences $\{\xi_m^{(k)}\}$, obtained via combinatorial search as explained in Section 4.1, into one global correspondence $\xi^{(k)}$ that covers the whole surface. Since patches partition the shape surface into overlapping regions, the union of these matchings, $\xi^{(k,0)} = \bigcup_m \xi_m^{(k)}$, gives an initial correspondence that needs to be simplified. The merging process at level k is carried out in the following three steps, that involve elimination of redundant multiple matches and trimming of the outliers (as illustrated in Fig. 3):

- **Step 1:** For every base vertex $s_i \in S^{(k)}$, we keep only one correspondence pair, the one with the minimum distortion among all $(s_i, t_j) \in \xi^{(k,0)}$, which gives us $\xi^{(k,1)}$, the output of the first step of the merging algorithm. We compute each isometric distortion $d_{\text{iso}}(s_i, t_j)$ via Eq.2 with the setting $\xi' = \xi^{(k,0)}$. Note that $\xi^{(k,1)}$ is many-to-one.
- **Step 2:** To every isolated base vertex $t_j \in T^{(k)}$ with no correspondence $s_i \in S^{(k)}$ s.t. $(s_i, t_j) \in \xi^{(k,1)}$, we assign the base $s_n \in S^{(k)}$ that yields the minimum distortion among all $(s_n, t_j) \in \xi^{(k,0)}$, which gives us $\xi^{(k,2)}$, the output of the second step. We compute each isometric distortion with the setting $\xi' = \xi^{(k,1)}$.
- **Step 3:** We replace every outlier $(s_i, t_j) \in \xi^{(k,2)}$ for which $d_{\text{iso}}(s_i, t_j) > 2 \cdot D_{\text{iso}}(\xi^{(k,2)})$, with (s_i, t_n) that yields the minimum distortion among all $t_n \in T^{(k)}$. We compute each isometric distortion with the setting $\xi' = \xi^{(k,2)}$. If the removal of an outlier creates an isolated target base, then step (2) is repeated. The output of this last step gives us the final correspondence $\xi^{(k)}$, which is a many-to-many mapping.

4.3. Overall Algorithm

The overall correspondence algorithm is composed of three basic tasks at each level of detail: sampling, combinatorial matching and merging. A critical parameter of the algorithm is the sampling radius $r^{(k)}$, which determines at each level k the number M of bases sampled within each parent patch. Recall that we set $r^{(k)}$ based on the area of the largest patch at level $k-1$, $A^{(k-1)}$, by using the ad-hoc formula given in Section 4.1, which ensures a sufficiently small value of M for combinatorial matching, $M \leq 6$. Hence in practice, we have two sampling radii at each level, $r_S^{(k)}$ and $r_T^{(k)}$, one for the source shape S and one for the target T , which have close but different values (since S and T are assumed to be nearly isometric). We initialize

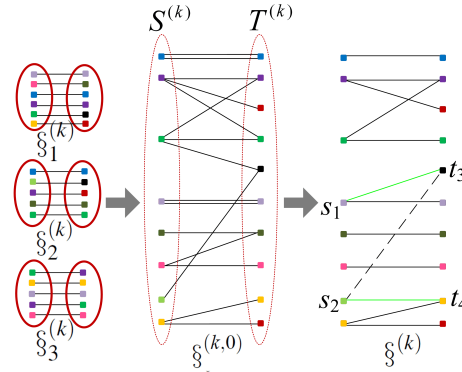


Figure 3: Merging of patch-based correspondences, $\xi_1^{(k)}$, $\xi_2^{(k)}$ and $\xi_3^{(k)}$, into one global mapping $\xi^{(k)}$ at level k . The illustration considers only three patches for demonstration purposes. Points from different patches with the same color correspond to the same base vertex. Note that some redundant matches are eliminated and the outlier (s_2, t_3) is replaced with (s_1, t_3) and (s_2, t_4) . The resulting $\xi^{(k)}$ covers every base vertex on the source shape as well as on the target.

the area parameters, as $A_S^{(-1)} = A_S$ and $A_T^{(-1)} = A_T$, where A_S and A_T denote the surface areas of the source and the target, respectively. The overall correspondence algorithm can then be written in pseudocode as below:

Correspondence Algorithm:

Input: Vertex sets S and T

Output: Dense Correspondence $\xi^{(k)} : S \rightarrow T$

Iterate on level k (initially $k = 0$)

Set the sampling radii $r_S^{(k)}$ and $r_T^{(k)}$ based on the largest patch areas at the previous level, $A_S^{(k-1)}$ and $A_T^{(k-1)}$;

Compute the base sets $S^{(k)}$ and $T^{(k)}$ by the sampling algorithm (Section 3);

For each $(s_i, t_i) \in \xi^{(k-1)}$

Match k^{th} level base vertices in patch $S_i^{(k-1)}$ and patch $T_i^{(k-1)}$

via combinatorial search, that gives $\xi_m^{(k)}$ (Section 4.1);

Merge patch-to-patch correspondences $\{\xi_m^{(k)}\}$

into $\xi^{(k)}$ (Section 4.2);

Until all vertices in S and T are matched;

5. An Insight to Why the Algorithm Works

We now show that, under certain conditions, a coarse-to-fine matching algorithm, such as ours, can be used to gradually localize an accurate dense correspondence as the level of detail increases, based on the following "inclusion assertion":

Inclusion Assertion: Suppose S and T are sampled at sufficiently high resolution from two perfectly isometric surfaces. Let $(s_i, t_i) \in \xi^*$, where ξ^* is the optimal correspondence that minimizes $D_{\text{iso}}(\xi)$, i.e., $D_{\text{iso}}(\xi^*) = 0$. Suppose

also that $\xi^{(k)}$ is the optimal correspondence at some level k , and that the optimal solution is unique, which exempts us from ambiguities due to symmetries. Then,

$$\forall i, j \quad (s_i, t_j) \in \xi^{(k)} \rightarrow t_i \in T_j^{(k)}$$

where $T_j^{(k)}$ is the patch with radius $r^{(k)}$ centered around t_j at level k .

We will not formally prove this assertion, but rather give an outline of its justification, that will provide us with an insight into why and under what conditions our coarse-to-fine correspondence algorithm works. First note that $D_{\text{iso}}(\xi)$ is locally a slowly changing convex function around its optimal point. Hence the mapping $\xi^{(k)}$, which is optimal at level k , will assign each base vertex s_i to some base t_j as nearest to the optimal t_i as possible. Moreover $S^{(k)}$ and $T^{(k)}$ consist of evenly-spaced base vertices sampled from perfectly isometric shapes, such that each vertex in S and T is included by the patch of at least one base vertex with radius $r^{(k)}$. Based on these, we can state the following: $\forall i, j \quad (s_i, t_j) \in \xi^{(k)} \rightarrow g(t_i, t_j) \leq r^{(k)}$, where $g(\cdot, \cdot)$ is the geodesic distance function, which directly implies the assertion. The inclusion assertion is illustrated in Fig. 4 and demonstrated on a pair of isometric shapes in Fig. 5.

In the case of perfect isometry, the inclusion assertion basically suggests that, based on the optimal correspondence $\xi^{(k-1)}$ at level $k-1$, a patch-based matching algorithm, such as ours, can be used to find the optimal correspondence $\xi^{(k)}$ at level k . Starting from the coarsest level $k=0$, our algorithm aims to maintain the optimality at each level k by patched-based combinatorial matching and merging, assuming the optimality of the correspondence found at the previous level. The algorithm is expected to eventually converge to the optimal correspondence ξ^* as levels proceed, since at the last level there will be only one vertex left in each patch.

The inclusion assertion, hence our algorithm, discards the symmetry problem which is however inherent to all isometry-based correspondence methods. We note that, since our algorithm is coarse-to-fine, even though the shapes to be matched are not symmetric as a whole, symmetric flips may occasionally arise at the initial level due to coarse sampling, which can then effect further levels of matching. Moreover, since real shapes are usually only nearly isometric, the initial sampling based on extremities or Gaussian curvature with particular choices of the sampling radius can lead to failure cases. That means, the optimal correspondence ξ^* ideally given by Eq. 3 may not always match the dense correspondence $\xi^{(K)}$ that our algorithm produces.

6. Computational Complexity

Sorting the vertices of the original mesh w.r.t. saliency prior to the sampling process takes $O(N \log N)$ time where $N =$

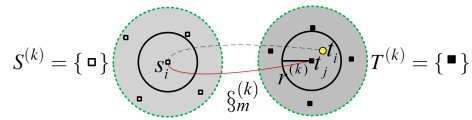


Figure 4: Illustration of the inclusion assertion at level k . The optimal correspondence t_i of s_i is included in the patch defined by t_j , whenever $(s_i, t_j) \in \xi_m^{(k)}$.

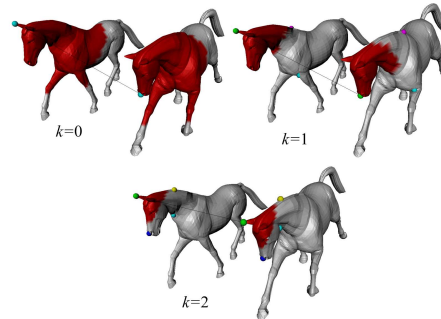


Figure 5: Demonstration of the inclusion assertion. Note how the correspondence between the right ears of two horses is refined as levels proceed. At levels $k=0$ and $k=1$, the correspondence is far from the optimal, yet the optimal target vertex is included at each level by the patch of the matched base, as required by the inclusion assertion. At level $k=2$, the left ears are correctly matched. The corresponding patches are painted in red at each level.

$\max(|S|, |T|)$. Our correspondence algorithm then proceeds with the following computations:

- *Coarse-to-fine sampling.* We sample about $M=5$ base vertices within each parent patch. Restricted to the patch to be sampled, Dijkstra's shortest paths algorithm, when applied to all active patches, takes $O(\sum_{k=1}^K M^k \cdot \frac{N}{M^k} \log_M \frac{N}{M^k})$ time, where M^k and $\frac{N}{M^k}$ can be interpreted as the number and the size of current patches, respectively, and $K = \log_M N$ is the maximum level of detail that can be achieved. The summation can be expanded as $N \log_M 1 + N \log_M M + \dots + N \log_M N$, yielding a total of $O(N \log N)$ time complexity.
- *Patch-based combinatorial matching.* Each patch pair is matched in constant time $O(M!)$ since there are $M!$ mappings to be evaluated via D_{iso} that can be computed also in constant time. Hence all patch pairs at all levels are matched in $O(\sum_{k=1}^K M^k \cdot M!)$ time, which unfolds to a total of $O(N \log N)$ complexity since $M^K = N$.
- *Merging.* Given a global initial mapping $\xi^{(k,0)}$ of size M^k at the end of each level k , its evenly-spaced subset that contains E matches ($E \ll N$) is used as the list ξ_j' to be traversed in the computation of d_{iso} (see step 1 in Section 4.2). Hence, the total time complexity can first be written as $O(\sum_{k=1}^K 3 \cdot M^k E)$, where the term $M^k E$ is due to d_{iso} computations in the three-step merging algorithm.

As before, the total complexity then reduces to $O(N \log N)$ with a proper choice of E such as 100.

The complexity analysis above reveals $O(N \log N)$ overall time complexity which scales well to large meshes, especially when compared to other isometric shape correspondence algorithms in the literature, such as $O(N^2 \log N)$ complexity of [JZ06] and [BBK06b], and $O(N^4 \log N)$ complexity of [LF09].

7. Experimental Results

We have tested our algorithm on several shape datasets: 1) on various mesh sequences and 2) on Nonrigid World database [BBK06a]. The mesh sequences are namely *Ballerina* [CGF09], *Jumping Man* [SMP03] and *Horse Gallop* [SP04], each containing different poses of an articulated object. While *Jumping Man* and *Horse Gallop* are given as fixed-connectivity mesh sequences sampled uniformly at high-resolution with 16K and 9K vertices, respectively, the *Ballerina* sequence exhibits severe non-uniformity as well as different number of vertices and connectivity among its 5 available different poses with around 6K vertices. On the other hand, the part of the Nonrigid World database that we use contains mesh models of 9 cats, 11 dogs, 3 wolves, 17 horses, 24 female figures, and two different male figures, containing 15 and 20 poses, where each object has approximately 3K vertices with arbitrary connectivity.

We measure the performance of our shape correspondence algorithm in terms of deviation from the ground-truth as well as isometric distortion. To quantify isometric distortion, we use the average distortion measure D_{iso} defined in (1), and also a maximum distortion measure that we denote by d_{iso}^\dagger :

$$d_{\text{iso}}^\dagger = \max_{(s_i, t_j) \in \xi} d_{\text{iso}}(s_i, t_j), \quad (4)$$

where d_{iso} is the isometric distortion function given in (2). Similarly we compute average and maximum ground-truth correspondence errors respectively by (whenever the ground-truth correspondence is available),

$$D_{\text{grd}}(\xi) = \frac{1}{|\xi|} \sum_{(s_i, t_j) \in \xi} g(t_i, t_j), \quad (5)$$

and

$$d_{\text{grd}}^\dagger = \max_{(s_i, t_j) \in \xi} g(t_i, t_j), \quad (6)$$

where each (s_i, t_i) stands for a ground-truth correspondence pair. Note that in computation of the performance measures, we use normalized geodesic distances so that the maximum geodesic distance on a shape surface is taken as 1.0.

In the figures where shape correspondences are displayed, the worst matchings, with respect to isometric distortion d_{iso}^\dagger and the ground-truth correspondence error d_{grd}^\dagger (whenever available), are highlighted with bold red and green lines, respectively. In Fig. 6, we display the dense correspondence

obtained on a shape pair from the *Ballerina* sequence, and zoom on the fingers to show the accuracy and smoothness of the mapping obtained.

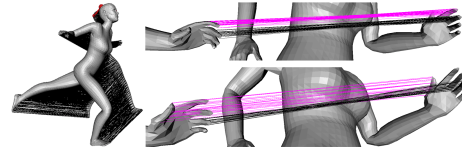


Figure 6: Dense correspondence for a *Ballerina* pair ($\xi^{(K)}$ with $K = 11$), zooming on the fingers of the left hands. The worst match w.r.t isometric distortion, as measured by d_{iso}^\dagger , is indicated with a bold red line.

In Fig. 7, we display the correspondences obtained at increasing levels of detail for a *Ballerina* pair and a *Dog-Wolf* pair from the Nonrigid World. We observe that the correspondences are improved and refined as levels proceed, and eventually we obtain a very accurate correspondence, even on the *Dog-Wolf* pair which contains severe non-isometries as compared to the articulated *Ballerina* pair. As a general rule, the mapping becomes denser and hence the isometric distortion is expected to decrease as levels proceed. This is verified by the plot given in Fig. 8, except that there is a consistent small increase in distortion from level $k = 0$ to $k = 1$, which is due to accurate joint sampling of shape extremities at the initial level. In Fig. 8, we plot the isometric distortion for varying k , computed by averaging the results over 6 sample pairs (whenever available).

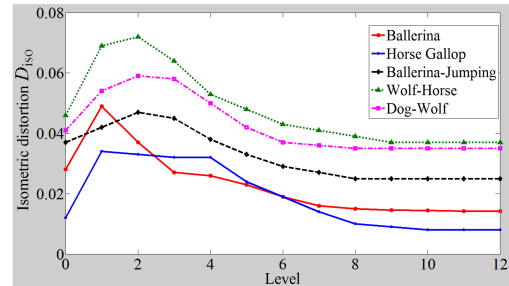


Figure 8: The isometric distortion (D_{iso}) of the correspondence obtained at each level k for different shape datasets.

The isometric quality of the input is an important factor for the performance of our algorithm. To measure isometric quality, we assign each dataset an isometric distortion value, denoted by ξ , which is computed as the value of D_{iso} for the manual one-to-one matching of 10 shape extremities: (from the most isometric to the least) 0.034 for *Ballerina*, 0.039 for *Horse Gallop*, 0.047 for the *Jumping Man-Ballerina* pair, 0.060 for the *Dog-Wolf* pair, and 0.064 for the *Wolf-Horse* pair. Although any pair can be regarded as only nearly isometric from a rigorous point of view, the *Ballerina* and *Horse*

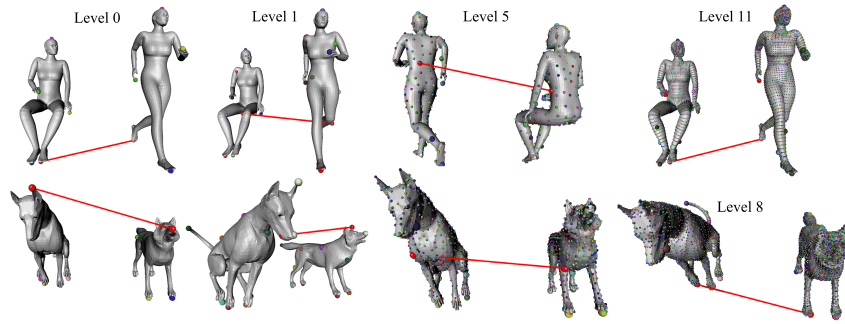


Figure 7: Correspondences obtained at increasing levels of detail for a *Ballerina* pair (top) and for the *Dog-Wolf* pair (bottom).

Gallop sequences, having higher input quality, provide better outputs when compared to the other hybrid pairs, as observed in Table 1 as well as in Fig. 8. Nevertheless, the hybrid pairs still yield high-quality results despite significant non-isometries that they contain, e.g., due to the fingers of the *Ballerina* matched with the hand of the *Jumping Man* (with no fingers) as shown in Fig. 9, or due to the scale differences between the corresponding parts of the *Wolf* and *Dog* shapes such as tails, legs, and faces (Fig. 7).

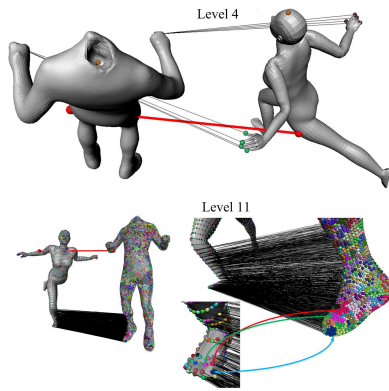


Figure 9: Correspondence results on a *Jumping Man - Ballerina* hybrid pair. At level $k = 4$, the fingers of the *Ballerina* are mapped to a single vertex on the hand of the *Jumping Man* as desired (top). Clustered matchings are observed on the *Jumping Man* at the last level, which is as expected since its mesh representation is significantly denser than the *Ballerina* mesh (bottom).

We compare our isometric shape correspondence method with two state-of-the-art techniques: the spectral method of [JZ06] and the generalized multidimensional scaling (GMDS) method of [BBK06a]. We have obtained the results of these two methods by using the Matlab codes made publicly available by the authors. Since neither the spectral method nor the GMDS scales well to large meshes, in the comparison experiments we have stopped our coarse-to-fine

algorithm at an intermediate level ($k = 4$, ~ 200 base vertices for the spectral method and $k = 5$, ~ 300 base vertices for GMDS). For comparisons with the spectral method, we have run their code on the same set of base vertices that we have used for our algorithm. The GMDS method however has its own sampling procedure, and the publicly available code can be set so as to generate the same number of vertices that our method produces. Hence in the latter case, the sampled vertices are different but the same in number.

All the performance measures provided in Table 1 for comparison with the spectral method are each computed and averaged over 10 different runs of the algorithm on 10 different pairs, except one pair for the *Dog-Wolf* and 5 pairs for *Ballerina*, where each run matches two spatially-apart poses of the articulated object in the corresponding sequence. In Table 2, we provide the results of the comparison of our method with GMDS on the Nonrigid World database. In this case, we evaluate the average performance in three different categories: 1) within human models, 2) within animal models, and 3) across human models. In the first and second categories, all possible shape pairs are tested such that a human or animal model is paired up only with models from the same class, e.g., a dog to another dog, whereas in the third category, a human object is matched to a model if and only if it represents a different human. We note that all three methods can result in symmetric flips since they are all isometry-based. Hence when comparing two methods, we have included only the pairs that can be matched as free of symmetry problems by both methods in consideration. Note also that, in our case, if the value of M_0 , i.e., the number of extremities at the coarsest level, is increased in the range $5 \leq M_0 \leq 9$, the number of symmetric flips tend to decrease.

The spectral method, as well as the GMDS method, generates as output a many-to-one mapping from the source shape to the target while our method produces a many-to-many correspondence. Hence to make our result compatible, for comparison we have used the many-to-one mapping, $\xi^{(k,1)}$, that the first step of our merging algorithm generates (see Section 4.2). In Table 1, we observe that our

many-to-one mapping, $\xi^{(k,1)}$, outperforms the results of the spectral method for all datasets, and that the improvement is more significant on more isometric Ballerina and Horse Gallop pairs. In the table, we also provide the performance results for the dense mapping, $\xi^{(K)}$, obtained with our method, which are significantly better than the results obtained at intermediate levels. Note that, on all datasets, the isometric distortion of our final many-to-many dense mapping is lower than the corresponding input distortion ξ . We have the ground-truth correspondences only for the Horse Gallop sequence, hence we compare the two methods in this case also w.r.t their ground-truth error performances. Our method outperforms the spectral method with $D_{\text{grd}} = 0.039$ and $d_{\text{grd}}^{\dagger} = 0.102$ against $D_{\text{grd}} = 0.062$ and $d_{\text{grd}}^{\dagger} = 0.195$, when the results are averaged over 10 different runs of the algorithm on 10 different pairs. In Fig. 10, we visually compare the performance of our algorithm with the spectral method on a Horse Gallop pair. We observe that our method outperforms the spectral method significantly in this case, the worst matching being significantly better with our method as well as the correspondences being generally more accurate. While our method correctly matches all the shape extremities, the tips of the ears on the Horse are not for example correctly matched with the spectral method.

When compared with the GMDS method, we observe in Table 2 that our method is performance-wise on a par in terms of isometric distortion, even slightly better in some cases (recall that the complexity of GMDS is $O(N^2 \log N)$ against our $O(N \log N)$ complexity). This is visually demonstrated on two examples in Fig. 11. In the figure, we see that the matched vertices with GMDS are not as evenly distributed as they are in our case. This is mainly due to the fact that the GMDS algorithm is an iterative optimization process that produces sub-vertex matchings which do not necessarily coincide with the initial sampling. Hence while computing the resulting isometric distortion value for a GMDS output, we round sub-vertex coordinates to their nearest vertices on the surface mesh. The effect of this rounding process to the computed distortion values is however negligible since the mesh models in the Nonrigid World database are almost uniform and at relatively high resolution. We also observe from the visual comparison that our method can match the salient points of a shape more successfully than GMDS (e.g., ear tips of the cats), thanks to our saliency-based sampling.

Finally we note that, on a 6GB 2.53GHz 64-bit workstation, the overall execution time of our implementation, needed to match all the vertices so as to obtain a dense correspondence, is 38, 103, 110, 281 and 337 seconds for Dog-Wolf, Ballerina, Wolf - Horse Gallop, Horse Gallop and Jumping Man-Ballerina pairs, respectively.

	Many-to-one [JZ06]	Many-to-one (our method)	Many-to-many (our method)
Pair	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$
Ballerina	.023, .054	.016, .038	.014, .024
Horse Gallop	.021, .040	.013, .026	.008, .015
Jumping-Baller.	.041, .069	.035, .058	.028, .051
Dog-Wolf	.050, .105	.044, .075	.036, .055
Wolf-Horse Gal.	.055, .087	.048, .079	.037, .066

Table 1: Quantitative performance of our method in comparison with the spectral method of [JZ06].

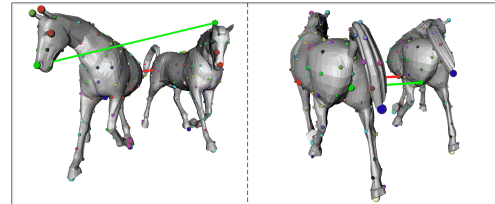


Figure 10: Many-to-one mappings obtained with the spectral method [JZ06] (left) and with our method at level $k = 4$ (right) for a Horse Gallop pair. Green and red lines indicate the worst matches w.r.t. ground-truth and isometric distortion, respectively.

8. Conclusion

We have proposed a dense isometric shape correspondence method based on coarse-to-fine sampling and combinatorial matching. Our findings can be summarized as follows:

- Our method is computationally very efficient with $O(N \log N)$ complexity, and hence scales well to large meshes.
- The dense correspondences that we obtain are very accurate since the isometric distortion is minimized in the original 3D Euclidean space, as free of approximation errors that embedding-based methods usually suffer from.
- Our method performs well not only on isometric shapes, but also on pairs of shapes which are nearly isometric, such as mesh representations of two different humans or animals, i.e., for shapes of the objects which are different but semantically and/or structurally similar.
- Our method produces accurate correspondences at differ-

	GMDS	Our method
Pair	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$	$D_{\text{iso}}, d_{\text{iso}}^{\dagger}$
Within animals	.022, .087	.021, .096
Within humans	.018, .076	.020, .079
Across humans	.026, .068	.038, .082

Table 2: Quantitative performance of our method in comparison with the GMDS method of [BKK06a] on Nonrigid World database.

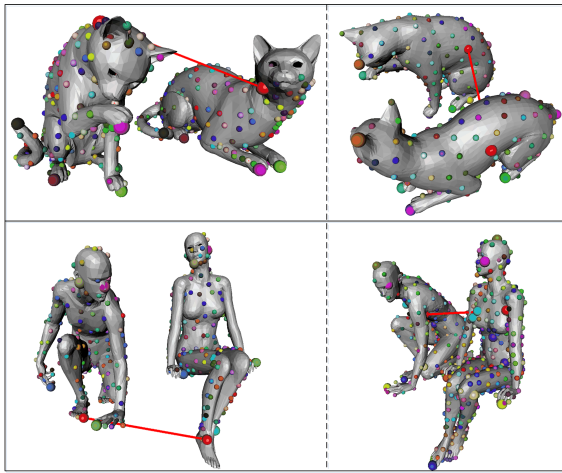


Figure 11: Many-to-one mappings obtained with the GMDS method [BBK06a] (left) and with our method at level $k = 5$ (right) for a within-animal pair (top) and for a across-human pair (bottom).

ent levels of detail thanks to our coarse-to-fine joint sampling algorithm.

A shortcoming of our method is due to the symmetry problem, which is inherent to all isometry-based correspondence algorithms. Moreover, since our algorithm is coarse-to-fine, even though the shapes to be matched are not symmetric as a whole, symmetric flips may occasionally arise at the initial level due to coarse sampling. This also relates to the difficult problem of finding a reliable initial correspondence as discussed in Section 1, which we plan to address as future work. We also plan to evaluate the performance of our method with larger-scale comparisons to more recent methods, using benchmark datasets such as in [KLF11].

Acknowledgement

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

References

- [ACOT*10] AU O. K.-C., COHEN-OR D., TAI C.-L., FU H., ZHENG Y.: Electors voting for fast automatic shape correspondence. *Computer Graphics Forum* 29, 2 (2010), 645–654.
- [Ale02] ALEXA M.: Recent advances in mesh morphing. *Computer Graphics Forum* 21 (2002), 173–196.
- [BBK06a] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing* 28, 5 (2006).
- [BBK06b] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Generalized multidimensional scaling: A framework for isometry invariant partial surface matching. *Proceedings of the National Academy of Science* 103, 5 (2006), 1168–1172.
- [CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3D mesh segmentation. *Proc. SIGGRAPH* (2009).
- [EK03] ELAD A., KIMMEL R.: On bending invariant signatures for surfaces. *IEEE Trans. PAMI* 25 (2003), 1285–1295.
- [FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3D shapes with priority-driven search. *Symposium on Geometry Processing (SGP)* (2006).
- [HAWG08] HUANG Q., ADAMS B., WICKE M., GUIBAS L.: Non-rigid registration under isometric deformations. *Symposium on Geometry Processing (SGP)* (2008), 1149–1458.
- [HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T.: Topology matching for fully automatic similarity estimation of 3D shapes. *Proc. SIGGRAPH* (2001), 203–212.
- [JZ06] JAIN V., ZHANG H.: Robust 3D shape correspondence in the spectral domain. *IEEE Int. Conf. on Shape Modeling and Applications (SMI)* (2006), 118–129.
- [KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *Proc. SIGGRAPH* (2011).
- [LF09] LIPMAN Y., FUNKHOUSER T.: Mobius voting for surface correspondence. *Proc. SIGGRAPH* 28, 3 (2009).
- [MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.: Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. *Proc. Computer Vision and Pattern Recognition (CVPR)* (2008).
- [MS04] MÉMOLI F., SAPIRO G.: Comparing point clouds. *Symposium on Geometry Processing (SGP)* (2004), 32–40.
- [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.
- [OSG07] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. *Comp. Graph. Forum* 27, 5 (2007).
- [SH10] SHARMA A., HORAUD R. P.: Shape matching based on diffusion embedding and on mutual isometric consistency. *Proc. Workshop on NORDIA* (2010).
- [SMP03] SAND P., MCMILLAN L., POPOVIC J.: Continuous capture of skin deformation. *Int. Conf. on Computer Graphics and Interactive Techniques* (2003).
- [SP04] SUMNER R. W., POPOVIC J.: Deformation transfer for triangle meshes. *Proc. SIGGRAPH* 23, 3 (2004).
- [SY10] SAHILLIOĞLU Y., YEMEZ Y.: 3D shape correspondence by isometry-driven greedy optimization. *Proc. Computer Vision and Pattern Recognition (CVPR)* (2010), 453–458.
- [TBW*09] TEVS A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: Isometric registration of ambiguous and partial data. *Proc. Comp. Vision and Pattern Recog. (CVPR)* (2009).
- [vKZHC011] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum* (2011).
- [ZBVH09] ZAHARESCU A., BOYER E., VARANASI K., HORAUD R.: Surface feature detection and description with applications to mesh matching. *Proc. Comp. Vision and Pattern Recog. (CVPR)* (2009).
- [ZSCO*08] ZHANG H., SHEFFER A., COHEN-OR D., ZHOU Q., VAN KAICK O., TAGLIASACCHIL A.: Deformation-driven shape correspondence. *Computer Graphics Forum* (2008).
- [ZWW*10] ZENG Y., WANG C., WANG Y., GU X., SAMARAS D., PARAGIOS N.: Dense non-rigid surface registration using high-order graph matching. *Proc. Comp. Vision and Pattern Recog. (CVPR)* (2010).